Scanned Datasheet
Notice: You cannot copy or search the text in this PDF file, because this PDF file is converted from a scanned image of printed materials.

Resources
Datasheet Archive - Datasheet & Application Note Search Engine
SupplyFrame - Datasheet & Application Note Search Engine

LH0080 Z80 CPU Central Processing Unit

■ Description

The LH0080 Z80 CPU (Z80 CPU for short below) is a general-purpose 8-bit microprocessor fabricated using an N-channel silicon-gate process.

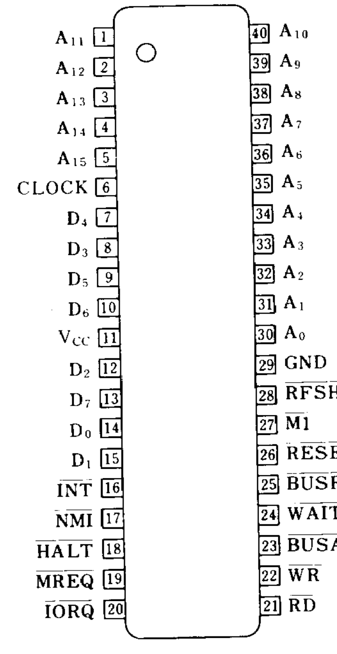
The LH0080A Z80A, LH0080B Z80B, LH0080E Z80E CPU are the high speed version which can operate at the 4MHz, 6MHz and 8MHz system clock, respectively.

■ Features

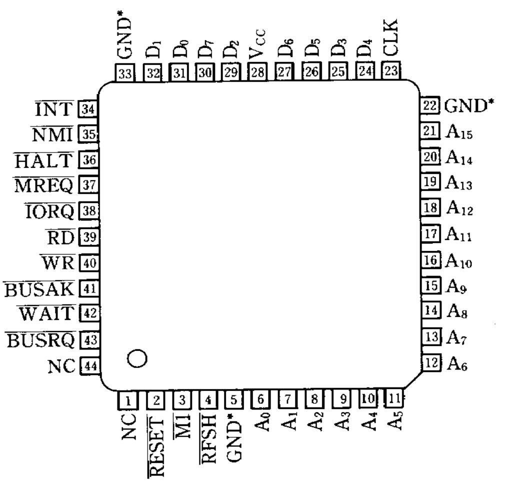
1. 8-bit parallel processing microprocessor
2. N-channel silicon-gate process
3. 158 instructions (The instruction of the 8080A are included as a subset ; 8080A software compatibility is maintained)
4. 22 registers
5. The capability of 3 modes maskable interrupt and non-maskable interrupt
6. On-chip dynamic memory refresh counter
7. Instruction fetch cycle : 1.6μs(Z80), 1.0μs (Z80A), 0.67μs (Z80B), 0.5μs (Z80E)
8. Single +5V power supply and single phase clock
9. All inputs and outputs fully TTL compatible
10. 40-pin DIP (DIP40-P-600)
    44-pin QFP (QFP44-P-1010A)
    44-pin QFJ (QFJ44-P-S650)

■ Pin Connections

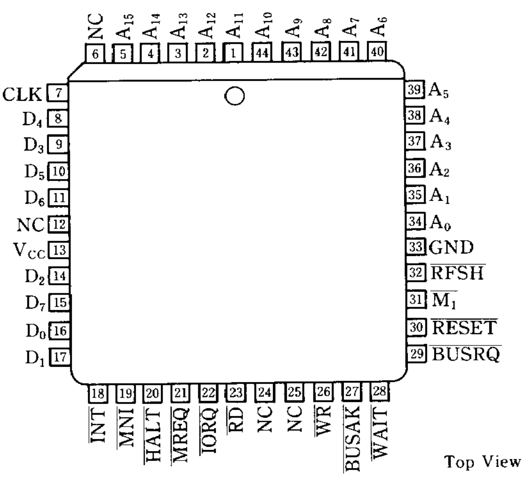
LH0080/LH0080A/LH0080B/LH0080E
LH0080H/LH0080AH



LH0080M/LH0080AM



LH0080U/LH0080AU/LH0080BU



* The GND pins must be connected to the GND level.

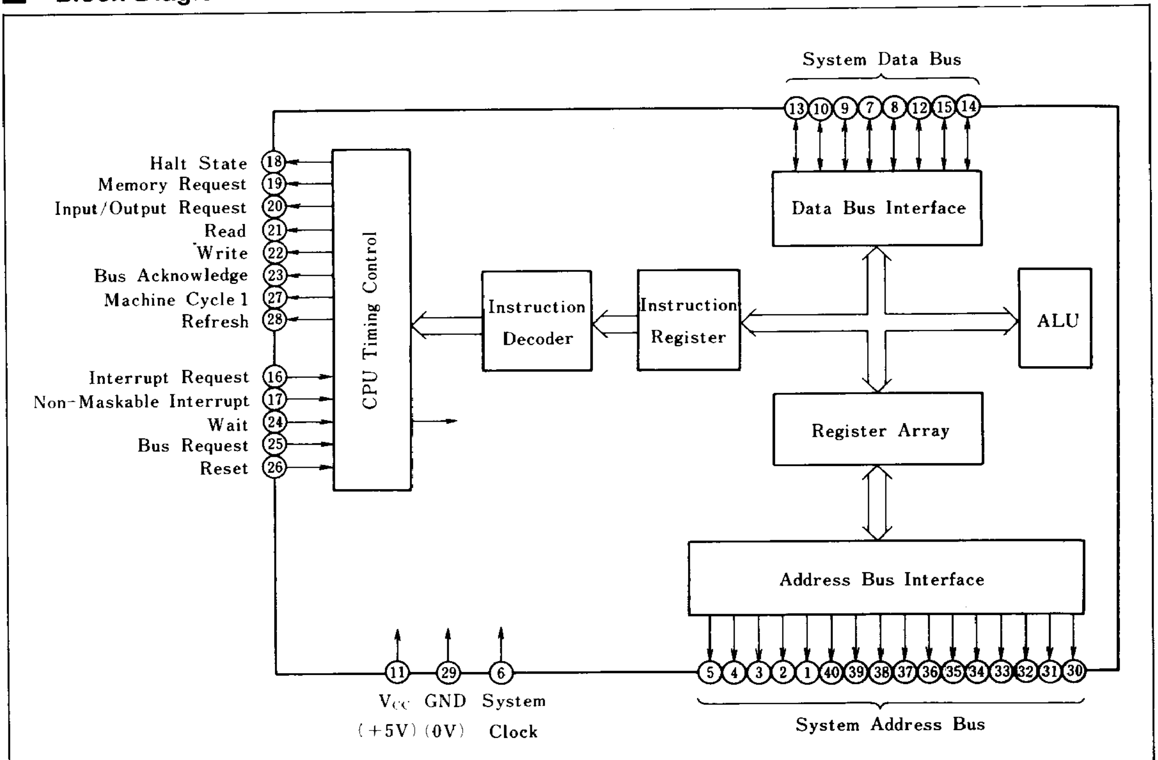
Powered by ICminer.com Electronic-Library Service CopyRight 2003

Ordering Information

Product	Z80 CPU	Z80A CPU	Z80B CPU	Z80E CPU	Package	Operating temperature
Clock frequency	2.5MHz	4MHz	6MHz	8MHz		
Model No.	LH0080	LH0080A	LH0080B	LH0080E	40-pin DIP	0℃ to +70℃
	LH0080H*	LH0080AH*				−20℃ to +85℃
	LH0080M	LH0080AM			44-pin QFP	0℃ to +60℃
	LH0080U	LH0080AU	LH0080BU		44-pin QFJ	0℃ to +70℃

* H suffix is a wide temperature spec, packaged in 40-pin DIP.

Block Diagram



Pin numbers apply to 40-pin DIP.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Pin Description

Signal	Pin name	I/O	Function
A_0-A_{15}	Address bus	3-state O	System address bus
D_0-D_7	Data bus	Bidirectional 3-state	System data bus
$\overline{M1}$	Machine cycle one	O	Active "Low". Indicates that the current machine cycle is the OP code fetch cycle of an instruction execution.
$\overline{MREQ}$	Memory request	3-state O	Active "Low". Indicates that the address bus holds a valid address for a memory read or memory write operation.
$\overline{IORQ}$	I/O request	3-state O	Active "Low". Indicates that the lower 8 bits of the address bus holds a valid I/O address for an I/O read or write operation. Also generated concurrently with $\overline{M1}$ during an interrupt acknowledge cycle to indicate an interrupt response.
$\overline{RD}$	Memory read	3-state O	Active "Low". Indicates that the CPU wants to read data from memory or an I/O device.
$\overline{WR}$	Memory write	3-state O	Active "Low". Indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.
$\overline{RFSH}$	Refresh	O	Active "Low". Indicates that the lower 7 bits of the system address bus can be used as a refresh address to the system's dynamic memories. Together with $\overline{MREQ}$ at "Low".
$\overline{HALT}$	Halt state	O	Active "Low". Indicates that a Halt instruction is being executed. While halted, the CPU executes NOPs to maintain memory refresh. The Halt state is cleared with $\overline{RESET}$, $\overline{NMI}$, or $\overline{INT}$ (when allowed).
$\overline{WAIT}$	Wait	I	Active "Low". Indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a wait state as long as this signal is active.
$\overline{INT}$	Maskable interrupt request	I	Active "Low". Generated by I/O devices. The CPU honors a request at the end of the current instruction if the interrupt enable flip-flop is enabled.
$\overline{NMI}$	Non-maskable interrupt request	I	Active "Low". Has a higher priority than $\overline{INT}$. Always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. Automatically forces the Z80 CPU to restart at location 0066H.
$\overline{RESET}$	Reset	I	Active "Low". Resets the interrupt enable flip-flop, the program counter interrupt vector register and the memory refresh register, and sets the interrupt status to Mode O, in order to initialize the CPU.
$\overline{BUSRQ}$	Bus request	I	Active "Low". Has a higher priority than NMI. Always recognized at the end of the current machine cycle. Activated to allow a bus master other than the CPU to control the system bus.
$\overline{BUSAK}$	Bus acknowledge	O	Active "Low". Indicates to the requesting device that the external circuitry can control the system bus.
CLOCK	System clock	I	Inputs+5V single-phase clock.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Note
Input voltage	V_{IN}	−0.3 to +7.0	V	
Output voltage	V_{OUT}	−0.3 to +7.0	V	
Operating temperature	Topr	0 to +70	℃	1
		0 to +60		2
		−20 to +85		3
Storage temperature	Tstg	−65 to +150	℃	

Note 1: 40-pin DIP and 44-pin QFJ
Note 2: 44-pin QFP
Note 3: 40-pin DIP with wide temperature spec.

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

All ac parameters assume a load capacitance of 100 pF. Add 10 ns delay for each 50 pF increase in load up to a maximum of 200 pF for the data bus and 100 pF for address and control lines.

DC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃ Note 1)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
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}	I_{OL}=1.8mA				0.4	V
Output high voltage	V_{OH}	I_{OH}=−250 μA		2.4			V
Current consumption	I_{CC}		LH0080			150	mA
			LH0080A			200	mA
			LH0080B			200	mA
			LH0080E			200	mA
Input leakage current	$\|I_{LI}\|$	$0 \leq V_{IN} \leq V_{CC}$				10	μA
3-state output leakage current in float	$\|I_{LEAK}\|$	V_{OUT}=0.4V to V_{CC}				10	μA

Note 1: Ta=0 to +60℃ for 44-pin QFP
Ta=−20 to +85℃ for 40-pin DIP with wide temperature spec.

Capacitance

(f=1MHz, Ta=25℃)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Clock capacitance	C_{CLOCK}	Unmeasured pins returned to ground			35	pF
Input capacitance	C_{IN}				5	pF
Output capacitance	C_{OUT}				10	pF

Powered by ICminer.com Electronic-Library Service CopyRight 2003

■ AC Characteristics

(V_{CC}=5V±5%, Ta=0 to +70℃ Note 1)

No.	Parameter	Symbol	LH0080		LH0080A		LH0080B		LH0080E*		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
1	Clock cycle time	TcC	400*		250*		165*		125*		ns
2	Clock pulse width (High)	TwCh	180*		110*		65*		55*		ns
3	Clock pulse width (Low)	TwCl	180	2000	110	2000	65	2000	55	2000	ns
4	Clock fall time	TfC		30		30		20		10	ns
5	Clock rise time	TrC		30		30		20		10	ns
6	Clock ↑ to address valid delay	TdCr (A)		145		110		90		80	ns
7	Addreess valid to MREQ ↓ delay	TdA (MREQf)	125*		65*		35*		20*		ns
8	Clock ↓ MREQ ↓ delay	TdCf (MREQf)		100		85		70		60	ns
9	Clock ↑ to MREQ ↑ delay	TdCr (MREQr)		100		85		70		60	ns
10	MREQ pulse width (High)	TwMREQh	170*		110*		65*		45*		ns
11	MREQ pulse width (Low)	TwMREQl	360*		220*		135*		100*		ns
12	Clock ↓ to MREQ ↑ delay	TdCf (MREQr)		100		85		70		60	ns
13	Clock ↓ to RD ↓ delay	TdCf (RDf)		130		95		80		70	ns
14	Clock ↑ to RD ↑ delay	TdCr (RDr)		100		85		70		60	ns
15	Data setup time to clock ↑	TsD (Cr)	50		35		30		30		ns
16	Data hold time from RD ↑	ThD (RDr)	0		0		0		0		ns
17	WAIT setup time to clock ↓	TsWAIT (Cf)	70		70		60		50		ns
18	WAIT hold time after clock ↓	ThWAIT (Cf)	0		0		0		0		ns
19	Clock ↑ to M1 ↓ delay	TdCr (M1f)		130		100		80		70	ns
20	Clock ↑ to M1 ↑ delay	TdCr (M1r)		130		100		80		70	ns
21	Clock ↑ to RFSH ↓ delay	TdCr (RFSHf)		180		130		110		95	ns
22	Clock ↑ to RFSH ↑ delay	TdCr (RFSHr)		150		120		100		85	ns
23	Clock ↓ to RD ↑ delay	TdCf (RDr)		110		85		70		60	ns
24	Clock ↑ to RD ↓ delay	TdCr (RDf)		100		85		70		60	ns
25	Data Setup to clock ↑ during M_2, M_3, M_4 or M_5 cycles	TsD (Cf)	60		50		40		30		ns
26	Address stable prior to IORQ ↓	TdA (IORQf)	320*		180*		110*		75*		ns
27	Clock ↑ IORQ ↓ delay	TdCr (IORQf)		90		75		65		55	ns
28	Clock ↓ to IORQ ↑ delay	TdCf (IORQr)		110		85		70		60	ns
29	Data stable prior to WR ↓	TdDm (WRf)	190*		80*		25*		5*		ns
30	Clock ↓ WR ↓ delay	TdCf (WRf)		90		80		70		60	ns
31	WR pulse width	TwWR	360*		220*		135*		100*		ns
32	Clock ↓ to WR ↑ delay	TdCr (WRr)		100		80		70		60	ns
33	Data stable prior to WR ↓	TdDi (WRf)	20*		−10*		−55*		−55*		ns
34	Clock ↑ to WR ↓ delay	TdCr (WRf)		80		65		60		55	ns
35	Data stable from WR ↑	TdWRr (D)	120*		60*		30*		15*		ns
36	Clock ↓ to HALT ↑ or ↓	TdCf (HALT)		300		300		260		225	ns
37	NMI pulse width	TwNMI	80		80		70		80		ns
38	BUSREQ setup time to clock ↑	TsBUSRQ (Cr)	80		50		50		40		ns
39	BUSREQ hold time after clock ↑	ThBUSRQ (Cr)	0		0		0		0		ns
40	Clock ↑ to BUSACK ↓ delay	TdCr (BUSAKf)		120		100		90		80	ns
41	Clock ↓ to BUSACK ↑ delay	TdCf (BUSAKr)		110		100		90		80	ns
42	Clock ↑ to data float delay	TdCr (Dz)		90		90		80		70	ns
43	Clock ↑ to control output float delay (MREQ, IORQ, RD, and WR)	TdCr (CTz)		110		80		70		60	ns
44	Clock ↑ to address float delay	TdCr (Az)		110		90		80		70	ns
45	MREQ ↑, IORQ ↑, RD and WR ↑ to address hold time	TdCTr (A)	160*		80*		35*		20*		ns

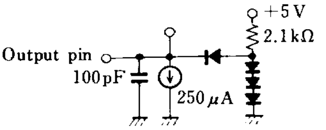
↑ Rising edge, ↓ Falling edge

Note 1: Ta=0 to +60℃ for 44-pin QFP.
Ta=−20 to +85℃ for 40-pin DIP with wide temperature spec.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

No.	Parameter	Symbol	LH0080		LH0080A		LH0080B		LH0080E*		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
46	RESET ↓ to clock ↑ setup time	TsRESET (Cr)	90		60		60		45		ns
47	RESET from clock! ↑ hold time	ThRESET (Cr)	0		0		0		0		ns
48	INT to clock ↑ setup time	TsINTf (Cr)	80		80		70		55		ns
49	INT from clock ↑ hold time	ThINTr (Cr)	0		0		0		0		ns
50	M1 ↓ to IORQ ↓ delay	TdM1f (IORQf)	920*		565*		365*		270*		ns
51	Clockk ↓ to IORQ ↓ delay	TdCf (IORQf)		110		85		70		60	ns
52	Clock ↑ to IORQ ↑ delay	TdCf (IORQr)		100		85		70		60	ns
53	Clock ↓ to data valid delay	TdCf (D)		230		150		130		115	ns

All ac parameters assume a load capacitance of 100 pF. Add 10 μs delay for each 50 pF increase in load up to a maximum of 200 pF for the data bus and 100 pF for address and control lines.
*For clock periods other than the minimums shown in the table, calculate parameters using the following expressions.



■ Footnotes to AC Characteristics

No.	Symbol	LH0080	LH0080A	LH0080B	LH0080E
1	TcC	TwCh+TwCl+TrC+TfC	TwCh+TwCl+TrC+TfC	TwCh+TwCl+TrC+TfC	TwCh+TwCl+TrC+TfC
2	TwCh	MAX. 200 μs	MAX. 200 μs	MAX. 200 μs	MAX. 200 μs
7	TdA (MREQf)	TwCh+TfC−75	TwCh+TfC−65	TwCh+TfC−50	TwCh+TfC−45
10	TwMREQh	TwCh+TfC−30	TwCh+TfC−20	TwCh+TfC−20	TwCh+TfC−20
11	TwMREQ1	TcC−40	TcC−30	TcC−30	TcC−25
26	TdA (IORQf)	TcC−80	TcC−70	TcC−55	TcC−50
29	TdD (WRf)	TcC−210	TcC−170	TcC−140	TcC−120
31	TwWR	TcC−40	TcC−30	TcC−30	TcC−25
33	TdD (WRf)	TwCl+TrC−180	TwCl+TrC−140	TwCl+TrC−140	TwCl+TrC−120
35	TdWRr (D)	TwCl+TrC−80	TwCl+TrC−70	TwCl+TrC−55	TwCl+TrC−50
45	TdCTr (A)	TwCl+TrC−40	TwCl+TrC−50	TwCl+TrC−50	TwCl+TrC−45
50	TdMlf (IORQf)	2Tch+TwCh+TfC−80	2TcC+TwCh+TfC−65	2TcC+TwCh+TfC−50	2TcC+TwCh+TfC−45

5

AC Test Conditions :

$V_{IH}=2.0V$ $V_{IHC}=V_{CC}-0.6V$ $V_{OH}=2.0V$ FLOAT $=\pm 0.5$
$V_{IL}=0.8V$ $V_{ILC}=0.45V$ $V_{OL}=0.8V$

Powered by ICminer.com Electronic-Library Service CopyRight 2003

CPU Timing

The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

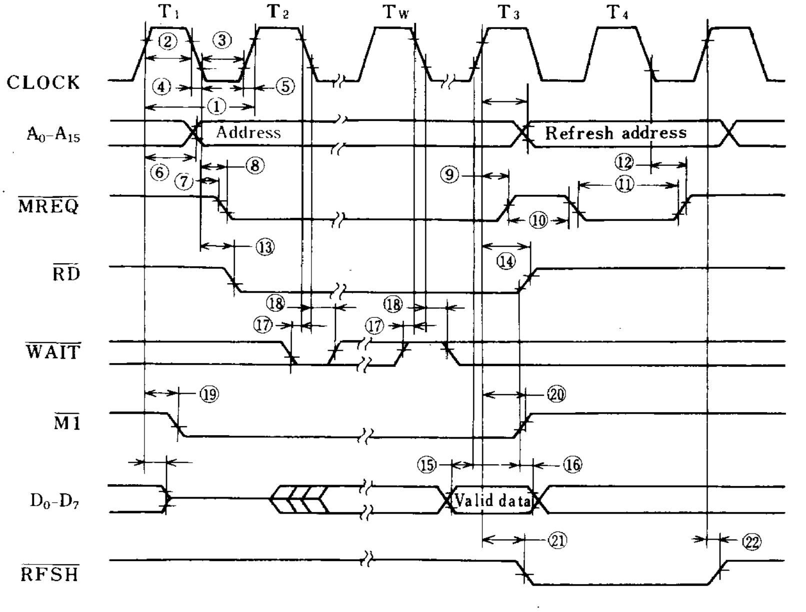
- Memory read or write
- I/O device read or write
- Interrupt acknowledge

The basic clock period is referred to as a T time or cycle, and three or more T cycles make up a machine cycle (M1, M2 or M3 for instance). Machine cycles can be extended either by the CPU automatically inserting one or more Wait states or by the insertion of one or more Wait states by the user.

(1) Instruction Opcode Fetch

The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle (Fig. 1). Approximately one-half clock cycle later, MREQ goes active. When active, RD indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the WAIT input with the falling edge of clock state T_2. During clock states T_3 and T_4 of an M1 cycle dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.



Note: T_w-Wait cycle added when necessary for slow ancilliary devices.

Fig. 1 Instruction opcode fetch

(2) Memory Read or Write Cycles

Fig. 2 shows the timing of memory read or write cycles other than an opcode fetch (M1) cycle.
The MREQ and RD signals function exactly as in the fetch cycle. In a memory write cycle, MREQ also becomes active when the address bus is stable. The WR line is active when the data bus is stable, so that it can be used directly as an R/W pulse to most semiconductor memories.

(3) Input or Output Cycles

Fig. 3 shows the timing for an I/O read or I/O write operation.
During I/O operations, the CPU automatically inserts a single wait state (T_w). This extra wait state allows sufficient time for an I/O port to decode the address from the port address lines.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

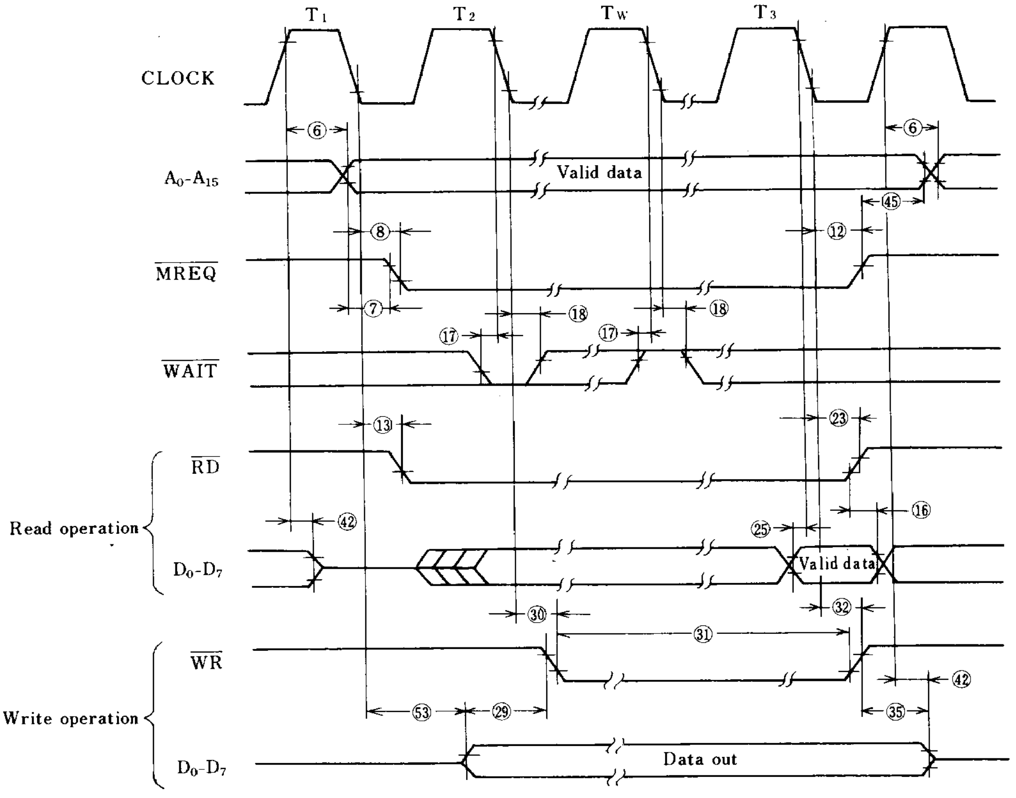
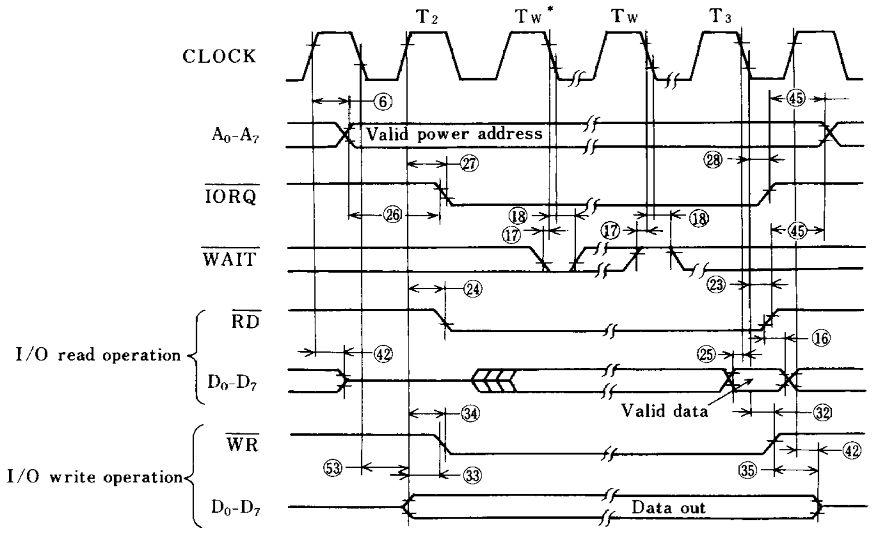


Fig. 2 Memory read or write cycles

Powered by ICminer.com Electronic-Library Service CopyRight 2003

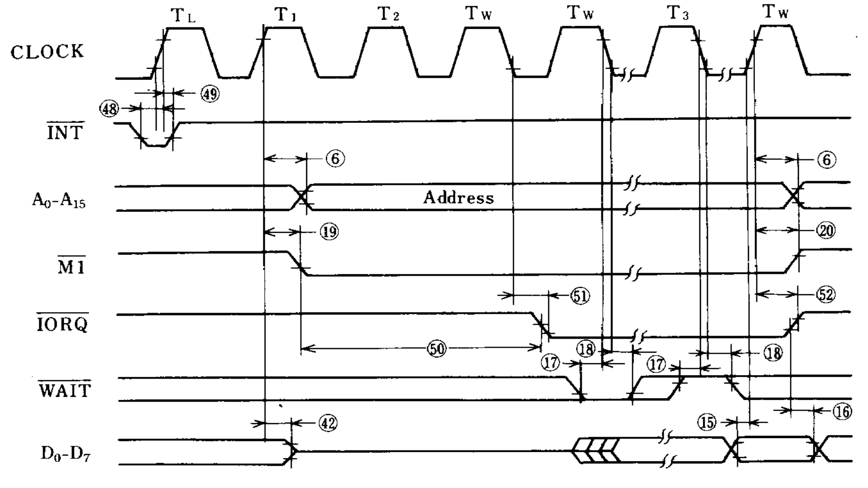


Note : Tw = One wait cycle automatically inserted by CPU.

Fig. 3 Input or output

(4) Interrupt request/acknowledge cycle

The CPU samples the interrupt signal with the rising edge of the last clock at the end of any instruction (Fig. 4). When an interrupt is accepted, a special M1 cycle is generated. During this M1 cycle, IORQ becomes active (instead of MREQ) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two wait states to this cycle.



Note 1 : TL = Last state of previous instruction.
Note 2 : Two wait cycles automatically inserted by CPU (*).

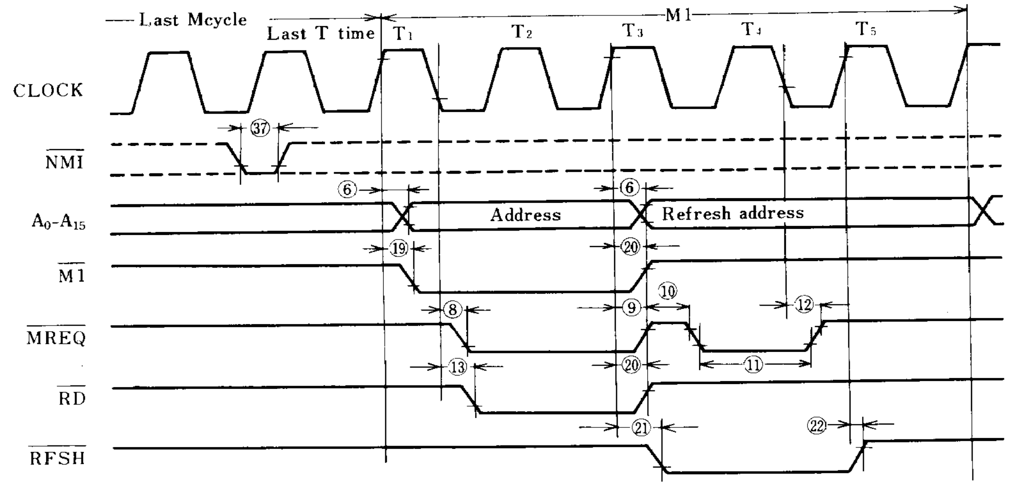
Fig. 4 Interrupt request/acknowledge cycle

Powered by ICminer.com Electronic-Library Service CopyRight 2003

(5) Non-maskable interrupt request cycle

NMI is sampled at the same time as the maskable interrupt INT but has higher priority and cannot be disabled under software control.

The subsequent timing is similar to that of a normal instruction fetch except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the NMI service routine located at address 0066H (Fig. 5).



*Although NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, NMI's falling edge must occur no later than rising edge of the clock cycle preceding T_{LAST}.

Fig. 5 Non-maskable interrupt request operation

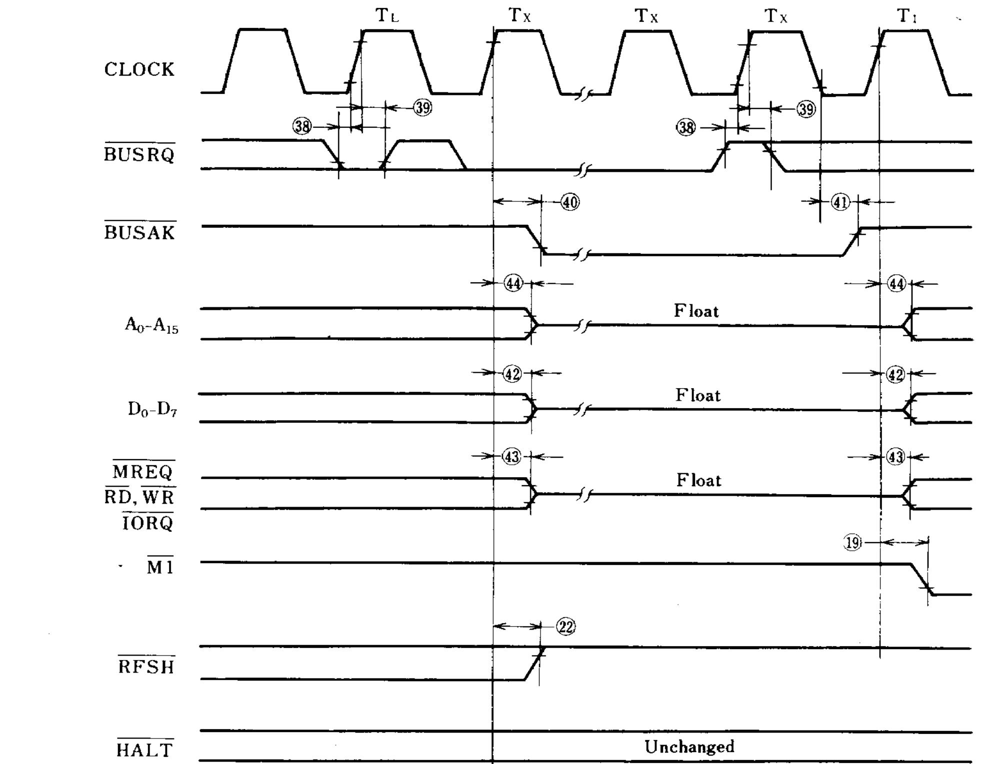
(6) Bus request/acknowledge cycle

The CPU samples BUSREQ with the rising edge of the last clock period of any machine cycle (Fig. 6). If BUSREQ is active, the CPU sets its address, data, and MREQ, IORQ, RD, and WR lines to a high-impedance state with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.

(7) Reset cycle

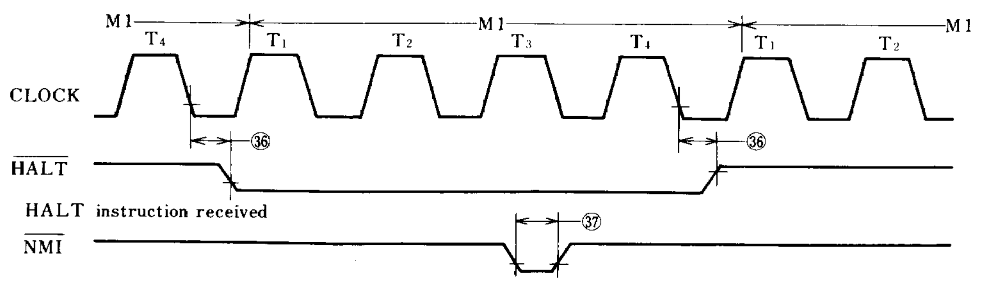
RESET must be active for at least three clock cycles for the CPU to properly accept it. As long as RESET remains active, the address and data buses float, and the control outputs are inactive. Once RESET goes inactive, three internal T cycles are consumed before the CPU resumes normal processing operation. RESET clears the PC register, so the first opcode fetch will be location 0000 (Fig. 8).

Powered by ICminer.com Electronic-Library Service CopyRight 2003



Note: TL=Last state of any M cycle.
TX=An arbitrary clock cycle used by requesting device.

Fig. 6 Z-bus request/acknowledge cycle



Note: INT will also force a Halt exit.

Fig. 7 Halt acknowledge cycle

Powered by ICminer.com Electronic-Library Service CopyRight 2003

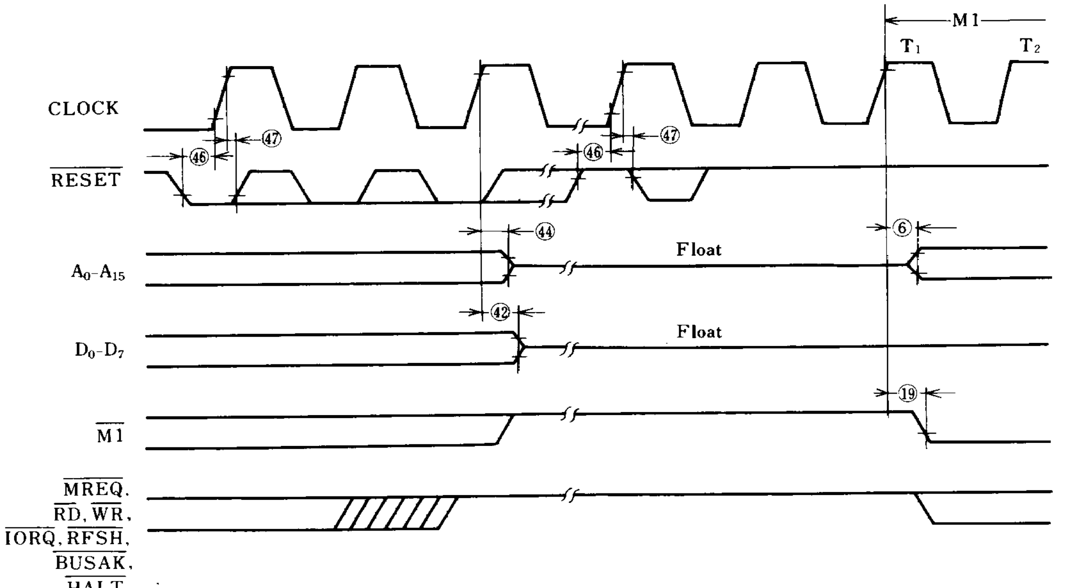


Fig. 8 Reset cycle

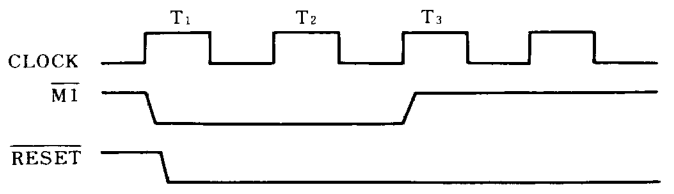


Fig. 9 Timing diagram when M1 cycle has no wait state

Powered by ICminer.com Electronic-Library Service CopyRight 2003

⟨Reference⟩

The RAM contents may be adversely affected by resetting the CPU while it is in operation.

To prevent this, a $\overline{RESET}$ signal should be input in the following timings.

(1) No walt state in the M1 cycle

Input a $\overline{RESET}$ signal to start sampling this signal at the clock rising in the M1 cycle's T_2 state. (See Fig. 9.)

(2) A walt state in the M1 cycle

Input a RESET signal to start sampling this signal at the clock rising in the M1 cycle's T_3 state. (See Fig. 10.)

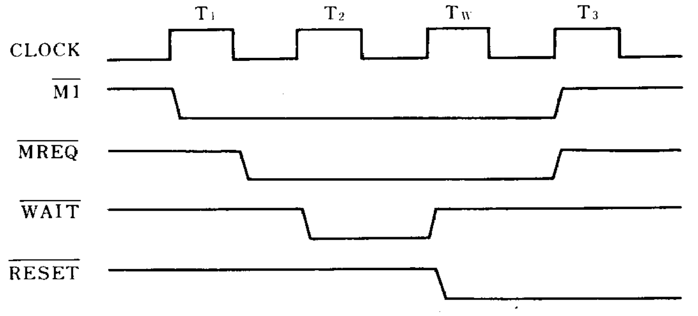


Fig. 10 Reset circuit and timing diagram when M1 cycle has a wait state

Powered by ICminer.com Electronic-Library Service CopyRight 2003

CPU Registers

A Accumulator	F Flag Register	A′ Accumulator	F′ Flag Register
B General Purpose	C General Purpose	B′ General Purpose	C′ General Purpose
D General Purpose	E General Purpose	D′ General Purpose	E′ General Purpose
H General Purpose	L General Purpose	H′ General Purpose	L′ General Purpose

← 8 bits →

I Interrupt Vector		R Memory Refresh
IX	Index Register	
IY	Index Register	
SP	Stack pointer	
PC	Program Counter	

← 16 bits →

Architecture

(1) CPU Registers

(i) **Program Counter (PC)** The program counter holds the 16 bits memory address of a current instruction. The CPU fetches the contents from memory address specified by the PC.

The PC feeds the data to the address line, automatically setting the PC value to +1. When a program jump takes place, a new value is directly set to the PC.

(ii) **Stack Pointer (SP)** The stack pointer holds the top 16-bit address of the stack with an external RAM. An external file is based on LIFO (Last-In, First-Out).

The data are transferred between a CPU-specified register and the stack by a PUSH or POP instruction. The last-pushed data are first popped from the stack.

(iii) **Index Register (IX & IY)** For index mode addressing, there are independent index registers IX and IY, each of which holds 16-bit reference address.

In the index mode, the index registers are used to designate the memory area for data input/output.

With an INDEX ADDRESSING instruction, an effective address comes by adding a one-byte displacement to the register content. This displacement is an integral signed two's complement number

(iv) **Interrupt Register (I)** The Z80 CPU has indirect subroutine call mode for any memory area according to an interrupt. For this purpose, this register stores the upper 8 bits of memory address for vectored interrupt processing and the lower 8 bits for the interrupting device.

(v) **Refresh Register (R)** The built-in refresh register provides user-transparent dynamic memory refresh. Its lower 7 bits are automatically incremented during each instruction fetch cycle.

While the CPU records a fetched instruction and executes the instruction, the refresh register data are placed on the address bus by a REFRESH control signal.

(vi) **Accumulator and Flag Register (A & F)** The CPU has also two independent 8-bit accumulators in combination with two 8-bit flag registers.

The accumulators store an operand or the results of an 8-bit operation. The flag registers, on the other hand, deal with the results of an 8-bit or 16-bit operation; for example, seeing if the result is equal to 0 or not.

(vii) **General-Purpose Registers** There are several pairs of general-purpose registers. In each pair, they can be used separately or as a 16-bit paired register. The paired registers are BC, DE, HL, as well as BC′ DE′ HL′. Either of these sets can work by an "Exchange" instruction at any time on a program.

(2) Arithmetic/Logical Unit (ALU)

An 8-bit arithmetic/logical operation instruction is executed by the ALU inside the CPU. The ALU connects to each register through the internal bus for data transfer between them.

(3) Instruction Register, CPU Control

Each instruction is read out of the memory, held in the instruction register, and decoded. The con-

5

Powered by ICminer.com Electronic-Library Service CopyRight 2003

trol unit controls this action and gives control signals necessary to read and write data from and to the registers.

The control unit also makes ALU control signal and other external control signals.

<Interrupts : General Operation> The Z80 CPU accepts two interrupt input signals: $\overline{NMI}$ and $\overline{INT}$. The $\overline{NMI}$ is a non-maskable interrupt and has the highest priority. $\overline{INT}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate.

(1) Non-Maskable Interrupt ($\overline{NMI}$)

The non-maskable interrupt will be accepted at all times by the CPU.

After recognition of the $\overline{NMI}$ signal, the CPU jumps to restart location 0066H.

(2) Maskable Interrupt ($\overline{INT}$)

The maskable interrupt, $\overline{INT}$, has three programmable response modes available.

(i) **Mode 0 Interrupt Operation.** This mode is similar to the 8080A microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is a Restart instruction or a Call instruction.

(ii) **Mode 1 Interrupt Operation.** Mode 1 operation is very similar to that for the $\overline{NMI}$. The principal difference is that the Mode 1 interrupt has a restart location of 0038H only.

(iii) **Mode 2 Interrupt Operation.** This interrupt mode has been designed to utilize most effectively the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address (16 bits) of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8-bits and the contents of the I register as the upper 8-bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that address.

All the Z80 peripheral devices have the interrupt priority circuit with a daisy-chain configuration. During an interrupt acknowledge cycle, vectors are automatically fed. For more details, refer to the Z80 PIO description.

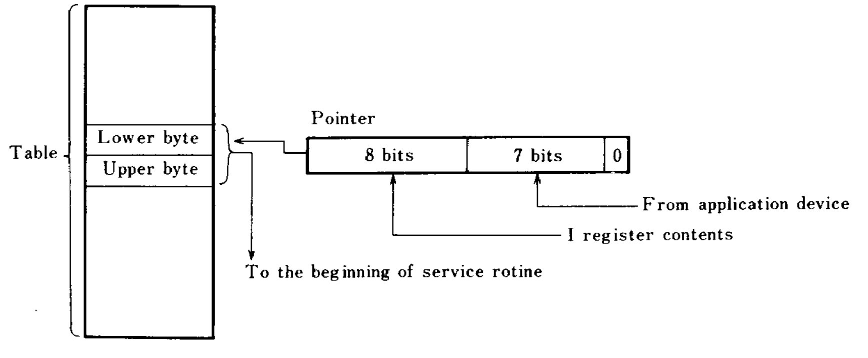


Fig. 1 Mode 2 interrupt diagram

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Instruction Set

Table 1 8-bit load group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD r, r′	r←r′	01 r r′	40+	●	●	●	●	●	●	1	1	4	
LD r, n	r←n	00 r 110 ← n →	06+	●	●	●	●	●	●	2	2	7	
LD r, (HL)	r← (HL)	01 r 110	46+	●	●	●	●	●	●	1	2	7	
LD r, (IX+d)	r← (IX+d)	11 011 101 01 r 110 ← d →	DD 46+	●	●	●	●	●	●	3	5	19	
LD r, (IY+d)	r← (IY+d)	11 111 101 01 r 110 ← d →	FD 46	●	●	●	●	●	●	3	5	19	
LD (HL), r	(HL)←r	01 110 r	70+	●	●	●	●	●	●	1	2	7	
LD (IX+d), r	(IX+d)←r	11 011 101 01 110 r ← d →	DD 70+	●	●	●	●	●	●	3	5	19	
LD (IY+d), r	(IY+d)←r	11 111 101 01 110 r ← d →	FD 70+	●	●	●	●	●	●	3	5	19	
LD (HL), n	(HL)←n	00 110 110 ← n →	36	●	●	●	●	●	●	2	3	10	
LD (IX+d), n	(IX+d)←n	11 011 101 00 110 110 ← d → ← n →	DD 36	●	●	●	●	●	●	4	5	19	
LD (IY+d), n	(IY+d)←n	11 111 101 00 110 110 ← d → ← n →	FD 36	●	●	●	●	●	●	4	5	19	
LD A, (BC)	A ← (BC)	00 001 010	0A	●	●	●	●	●	●	1	2	7	
LD A, (DE)	A ← (DE)	00 011 010	1A	●	●	●	●	●	●	1	2	7	
LD A, (nn)	A ← (nn)	00 111 010 ← n → ← n →	3A	●	●	●	●	●	●	3	4	13	
LD (BC), A	(BC) ← A	00 000 010	02	●	●	●	●	●	●	1	2	7	
LD (DE), A	(DE) ← A	00 010 010	12	●	●	●	●	●	●	1	2	7	
LD (nn), A	(nn) ← A	00 110 010 ← n → ← n →	32	●	●	●	●	●	●	3	4	13	
LD A, I	A ← I	11 101 101 01 010 111	ED 57	●	↕	IFF	↕	0	0	2	2	9	
LD A, R	A ← R	11 101 101 01 011 111	ED 5F	●	↕	IFF	↕	0	0	2	2	9	
LD I, A	I ← A	11 101 101 01 000 111	ED 47	●	●	●	●	●	●	2	2	9	
LD R, A	R ← A	11 101 101 01 001 111	ED 4F	●	●	●	●	●	●	2	2	9	

Comments:

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

Notes : r, r′ means any of the registers A, B, C, D, E, H, L, IFF the content of the interrupt enable flip-flop, (IFF) is copied into the P/V flag.
Flags : C (carry), Z (zero), S (sign), P/V (parity/overflow), H (half carry), N (add/substract).
: ● =unchanged, 0=reset, 1=set, X=undefined.
: ↕ set or reset according to the result of the operation.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 2 16-bit load group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	00 dd0 001 ← n → ← n →	01+	●	●	●	●	●	●	3	3	10	dd Reg.
LD IX, nn	IX ← nn	11 011 101 00 100 001 ← n → ← n →	DD 21	●	●	●	●	●	●	4	4	14	00 BC 01 DE 10 HL
LD IY, nn	IY ← nn	11 111 101 00 100 001 ← n → ← n →	FD 21	●	●	●	●	●	●	4	4	14	11 SP
LD HL, (nn)	H ← (nn+1) L ← (nn)	00 101 010 ← n → ← n →	2A	●	●	●	●	●	●	3	5	16	nn : 2-byte number. Lower byte just after opcode.
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	11 101 101 01 dd1 011 ← n → ← n →	ED 4B+	●	●	●	●	●	●	4	6	20	Upper byte comes next.
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	11 011 101 00 101 010 ← n → ← n →	DD 2A	●	●	●	●	●	●	4	6	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	11 111 101 00 101 010 ← n →	FD 2A	●	●	●	●	●	●	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	00 100 010 ← n → ← n →	22	●	●	●	●	●	●	3	5	16	
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	11 101 101 01 dd0 011 ← n → ← n →	ED 43+	●	●	●	●	●	●	4	6	20	
LD (nn), IX	(nn+1)←IX_H (nn) ← IX_L	11 011 101 00 100 010 ← n → ← n →	DD 22	●	●	●	●	●	●	4	6	20	
LD (nn), IY	(nn+1)←IY_H (nn) ← IY_L	11 111 101 00 100 010 ← n → ← n →	FD 22	●	●	●	●	●	●	4	6	20	
LD SP, HL	SP ← HL	11 111 001	F9	●	●	●	●	●	●	1	1	6	
LD SP, IX	SP ← IX	11 011 101 11 111 001	DD F9	●	●	●	●	●	●	2	2	10	
LD SP, IY	SP ← IY	11 111 101 11 111 001	FD F9	●	●	●	●	●	●	2	2	10	
PUSH qq	(SP−2)←qq_L (SP−1)←qq_H	11 qq0 101	C5+	●	●	●	●	●	●	1	3	11	qq Reg.
PUSH IX	(SP−2)←IX_L (SP−1)←IX_H	11 011 101 11 100 101	DD E5	●	●	●	●	●	●	2	4	15	00 BC 01 DE
PUSH IY	(SP−2)←IY_L (SP−1)←IY_H	11 111 101 11 100 101	FD E5	●	●	●	●	●	●	2	4	15	10 HL
POP qq	qq_H ← (SP+1) qq_L ← (SP)	11 qq0 001	C1+	●	●	●	●	●	●	1	3	10	11 AF
POP IX	IX_H←(SP+1) IX_L ← (SP)	11 011 101 11 100 001	DD E1	●	●	●	●	●	●	2	4	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP)	11 111 101 11 100 001	FD E1	●	●	●	●	●	●	2	4	14	

Notes : dd is any of the register pairs BC, DE, HL, SP.
qq is any of the register pairs AF, BC, DE, HL.
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively, e.g., BC_L=C, AF_H=A.
Flags : ● =unchanged, 0=reset, 1=set, X=undefined, ↕ =set or reset according to the result of the operation

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 3 Exchange, block transfer, block search groups

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE, HL	DE ↔ HL	11 101 011	EB	●	●	●	●	●	●	1	1	4	
EX AF, AF′	AF ↔ AF′	00 001 000	08	●	●	●	●	●	●	1	1	4	
EXX	(BC, DE, HL) ↔ (BC′, DE′, HL′)	11 011 001	D9	●	●	●	●	●	●	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	11 100 011	E3	●	●	●	●	●	●	1	5	19	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	11 011 101 11 100 011	DD E3	●	●	●	●	●	●	2	6	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	11 111 101 11 100 011	FD E3	●	●	●	●	●	●	2	6	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC−1	11 101 101 10 100 000	ED A0	●	●	↕ ①	●	0	0	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC−1	11 101 101 10 110 000	ED B0	●	●	0	●	0	0	2	5	21	If BC≠0
	If BC=0 end									2	4	16	If BC=0
LDD	(DE) ← (HL) DE ← DE−1 HL ← HL−1 BC ← BC−1	11 101 101 10 101 000	ED A8	●	●	↕ ①	●	0	0	2	4	16	
LDDR	(DE) ← (HL) DE ← DE−1 HL ← HL−1 BC ← BC−1	11 101 101 10 111 000	ED B8	●	●	0	●	0	0	2	5	21	If BC≠0
	If BC=0 end									2	4	16	If BC=0
CPI	A− (HL) HL ← HL+1 BC ← BC−1	11 101 101 10 100 001	ED A1	●	↕ ②	↕ ①	↕	1	↕	2	4	16	
CPIR	A− (HL) HL ← HL+1 BC ← BC−1	11 101 101 10 110 001	ED B1	●	↕ ②	↕ ①	↕	1	↕	2	5	21	If BC≠0 and A≠(HL)
	If A= (HL) or BC=0 end									2	4	16	If BC=0 or A= (HL)
CPD	A− (HL) HL ← HL−1 BC ← BC−1	11 101 101 10 101 001	ED A9	●	↕ ②	↕ ①	↕	1	↕	2	4	16	
CPDR	A− (HL) HL ← HL−1 BC ← BC−1	11 101 101 10 111 001	ED B9	●	↕ ②	↕ ①	↕	1	↕	2	5	21	If BC≠0 and A≠(HL)
	If A= (HL) or BC=0 end									2	4	16	If BC=0 or A= (HL)

Note: ①P/V flag is 0 if the result of BC=0, otherwise P/V=1
②Z flag is 1 if A= (HL), otherwise Z=0

Flags : ● = unchanged
0 = set, 1 = reset
↕ = set or reset according to the result of the operation

5

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 4 8-bit arithmetic and logical group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States
ADD A, r	A ← A+r	10 k r	80+	↕	↕	V	↕	0	↕	1	1	4
ADD A, n	A ← A+n	11 k 110 ← n →	C6+	↕	↕	V	↕	0	↕	2	2	7
ADD A, (HL)	A ← A+(HL)	10 k 110	86+	↕	↕	V	↕	0	↕	1	2	7
ADD A, (IX+d)	A ← A+(IX+d)	11 011 101 10 k 110 ← d →	DD 86+	↕	↕	V	↕	0	↕	3	5	19
ADD A, (IY+d)	A ← A+(IY+d)	11 111 101 10 k 110 ← d →	FD 86+	↕	↕	V	↕	0	↕	3	5	19
ADC A, s	A ← A+s+C	4 types available based on the above ADD instruction (see Comments)		↕	↕	V	↕	0	↕	1※1 2 1 3	1※1 2 2 5	4※1 7 7 19
SUB s	A ← A−s			↕	↕	V	↕	1	↕			
SBC A, s	A ← A−s−C			↕	↕	V	↕	1	↕			
AND s	A ← A∧s			0	↕	P	↕	0	1			
OR s	A ← A∨s			0	↕	P	↕	0	0			
XOR s	A ← A⊕s			0	↕	P	↕	0	0			
CP s	A−s			↕	↕	V	↕	1	↕			
INC r	r ← r+1	00 r ℓ	00+	●	↕	V	↕	0	↕	1	1	4
INC (HL)	(HL) ← (HL)+1	00 110 ℓ	30+	●	↕	V	↕	0	↕	1	3	11
INC (IX+d)	(IX+d) ← (IX+d)+1	11 011 101 00 110 ℓ ← d →	DD 30+	●	↕	V	↕	0	↕	3	6	23
INC (IY+d)	(IY+d) ← (IY+d)+1	11 111 101 00 110 ℓ ← d →	FD 30+	●	↕	V	↕	0	↕	3	6	23
DEC m	m ← m−1	4 types available based on the above INC instruction		●	↕	V	↕	1	↕	1※2 1 3 3	1※2 3 6 6	4※2 11 23 23

Comments:

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

Mnemonic	k
ADD	000
ADC	001
SUB	010
SBC	011
AND	100
OR	110
XOR	101
CP	111

S=r, n, (HL), (IX+d), (IY+d)

Mnemonic	ℓ
INC	100
DEC	101

m=r, (HL), (IX+d), (IY+d)

Note : V and P mean overflow and parity, respectively.
Flags : ● =unchanged
0=reset
1=set
X=undefined
↕ =set or reset according to the result of the operation

※1: depends on s.
※2: depends on m.

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 5 General purpose arithmetic and CPU control groups

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	Decimal adjustment (add/subtract)	00 100 111	27	↕	↕	P	↕	●	↕	1	1	4	Decimal adjust accumulator.
CPL	A ← Ā	00 101 111	2F	●	●	●	●	1	1	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0−A	11 101 101 01 000 100	ED 44	↕	↕	V	↕	1	↕	2	2	8	Negate acc. (two's complement).
CCF	C ← C̄	00 111 111	3F	↕	●	●	●	0	X	1	1	4	Complement carry flag.
SCF	C ← 1	00 110 111	37	1	●	●	●	0	0	1	1	4	Set carry flag.
NOP	No operation	00 000 000	00	●	●	●	●	●	●	1	1	4	
HALT	CPU halted	01 110 110	76	●	●	●	●	●	●	1	1	4	
DI	IFF ← 0	11 110 011	F3	●	●	●	●	●	●	1	1	4	Interrupt not enable
EI	IFF ← 1	11 111 011	FB	●	●	●	●	●	●	1	1	4	Interrupt enable
IM 0	Set interrupt mode 0	11 101 101 01 000 110	ED 46	●	●	●	●	●	●	2	2	8	Set interrupt mode.
IM 1	Set interrupt mode 1	11 101 101 01 010 110	ED 56	●	●	●	●	●	●	2	2	8	
IM 2	Set interrupt mode 2	11 101 101 01 011 110	ED 5E	●	●	●	●	●	●	2	2	8	

Note : IFF indicates the interrupt enable flip-flop, CY indicates the carry flip-flop.
Flags : ● =unchanged, 0=reset, 1=set, X=undefined, ↕ =set or reset according to the result of the operation

Table 6 16-bit arithmetic group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL, ss	HL ← HL +ss	00 ss1 001	09+	↕	●	●	●	0	X	1	3	11	
ADC HL, ss	HL ← HL +ss+C	11 101 101 01 ss1 010	ED 4A+	↕	↕	V	↕	0	X	2	4	15	
SBC HL, ss	HL ← HL −ss−C	11 101 101 01 ss0 010	ED 42+	↕	↕	V	↕	1	X	2	4	15	
ADD IX, pp	IX ← IX+pp	11 011 101 00 pp1 001	DD 09+	↕	●	●	●	0	X	2	4	15	
ADD IY, rr	IY ← IY+rr	11 111 101 00 rr1 001	FD 09+	↕	●	●	●	0	X	2	4	15	
INC ss	ss ← ss+1	00 ss0 011	03+	●	●	●	●	●	●	1	1	6	
INC IX	IX ← IX+1	11 011 101 00 100 011	DD 23	●	●	●	●	●	●	2	2	10	
INC IY	IY ← IY+1	11 111 101 00 100 011	FD 23	●	●	●	●	●	●	2	2	10	
DEC ss	ss ← ss−1	00 ss1 011	0B+	●	●	●	●	●	●	1	1	6	
DEC IX	IX ← IX−1	11 011 101 00 101 011	DD 2B	●	●	●	●	●	●	2	2	10	
DEC IY	IY ← IY−1	11 111 101 00 101 011	FD 2B	●	●	●	●	●	●	2	2	10	

ss	Reg.
00	BC
01	DE
10	HL
11	SP

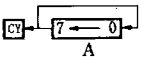
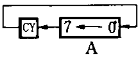
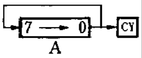
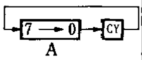
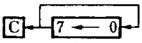
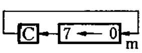
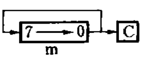
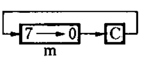
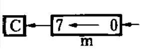
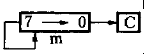
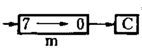
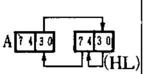
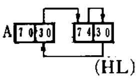
pp	Reg.
00	BC
01	DE
10	IX
11	SP

rr	Reg.
00	BC
01	DE
10	IY
11	SP

Note : ss is any of the register pairs BC, DE, HL, SP.
pp is any of the register pairs BC, DE, IX, SP.
rr is any of the register pairs BC, DE, IY, SP.
Flags : ● =unchanged, 0=reset, 1=set, X=undefinede, ↕ =set or reset according to the result of the operation

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 7 Rotate and shift groups

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		00 000 111	07	↕	●	●	●	0	0	1	1	4	Rotate left circular accumulator.
RLA		00 010 111	17	↕	●	●	●	0	0	1	1	4	Rotate left accumulator.
RRCA		00 001 111	0F	↕	●	●	●	0	0	1	1	4	Rotate right circular accumulator.
RRA		00 011 111	1F	↕	●	●	●	0	0	1	1	4	Rotate right accumulator.
RLCr		11 001 011 00 k r	CB 00+	↕	↕	P	↕	0	0	2	2	8	Rotate left circular register r.
RLC (HL)		11 001 011 00 k 110	CB 06+	↕	↕	P	↕	0	0	2	4	15	r / Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	 r, (HL), (IX+d), (IY+d)	11 011 101 11 001 011 ← d → 00 k 110	DD CB 06+	↕	↕	P	↕	0	0	4	6	23	
RLC (IY+d)		11 111 101 11 001 011 ← d → 00 k 110	FD CB 06+	↕	↕	P	↕	0	0	4	6	23	
RL m				↕	↕	P	↕	0	0				Mnemonic / k RLC 000 RRC 001 RL 010 RR 011 SLA 100 SRA 101 SRL 111
RRC m				↕	↕	P	↕	0	0				
RR m				↕	↕	P	↕	0	0	2* 2 4 4	2* 4 6 6	8* 15 23 23	
SLA m				↕	↕	P	↕	0	0				
SRA m				↕	↕	P	↕	0	0				m=r, (HL), (IX+d), (IY+d)
SRL m				↕	↕	P	↕	0	0				*depends on m.
RLD		11 101 101 01 101 111	ED 6F	●	↕	P	↕	0	0	2	5	18	Rotate digit left and right between the accumulator and location (HL).
RRD		11 101 101 01 100 111	ED 67	●	↕	P	↕	0	0	2	5	18	The content of the upper half of the accumulator is unaffected.

Flags : ● = unchanged
0 = reset
1 = set
× = undefined
↕ = set or reset according to the result of the operation

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 8 Bit set, reset and test group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b, r	$Z \leftarrow \overline{r_b}$	11 001 011 01 b r	CB 40+	●	↕	X	X	0	1	2	2	8	
BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	11 001 011 01 b 110	CB 46+	●	↕	X	X	0	1	2	3	12	
BIT b, (IX+d)	$Z \leftarrow \overline{(IX+d)_b}$	11 011 101 11 001 011 ← d → 01 b 110	DD CB 46+	●	↕	X	X	0	1	4	5	20	
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)_b}$	11 111 101 11 001 011 ← d → 01 b 110	FD CB 46+	●	↕	X	X	0	1	4	5	20	
SET b, r	$r_b \leftarrow 1$	11 001 011 a b r	CB	●	●	●	●	●	●	2	2	8	
SET b, (HL)	$(HL)_b \leftarrow 1$	11 001 011 a b 110	CB 06+	●	●	●	●	●	●	2	4	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	11 011 101 11 001 011 ← d → a b 110	DD CB 06+	●	●	●	●	●	●	4	6	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	11 111 101 11 001 011 ← d → a b 110	FD CB 06+	●	●	●	●	●	●	4	6	23	
RES b, m	$m_b \leftarrow 0$									2* 2 4 4	2* 4 6 6	8* 15 23 23	m=r, (HL), (IX+d), (IY+d) *depends on m

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

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

Mnemonic	a
SET	11
RES	10

Note : The notation m_b indicates bit b (0 to 7) or location m.
Flags : ● = unchanged
0 = reset
1 = set
X = undefined
↕ = set or reset according to the result of the operation

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 9 Jump group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States
JP nn	PC ← nn	11 000 011 ← n → ← n →	C3	●	●	●	●	●	●	3	3	10
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	11 cc 010 ← n → ← n →	C2+	●	●	●	●	●	●	3	3	10
										3	3	10
JR e	PC ← PC+e	00 011 000 ← e-2 →	18	●	●	●	●	●	●	2	3	12
JR C, e	If C=1 PC ← PC+e	00 111 000 ← e-2 →	38	●	●	●	●	●	●	2	3	12
	If C=0 continue									2	2	7
JR NC, e	If C=0 PC ← PC+e	00 110 000 ← e-2 →	30	●	●	●	●	●	●	2	3	12
	If C=1 continue									2	2	7
JR Z, e	If Z=1 PC ← PC+e	00 101 000 ← e-2 →	28	●	●	●	●	●	●	2	3	12
	If Z=0 continue									2	2	7
JR NZ, e	If Z=0 PC ← PC+e	00 100 000 ← e-2 →	20	●	●	●	●	●	●	2	3	12
	If Z=1 continue									2	2	7
JP (HL)	PC ← HL	11 101 001	E9	●	●	●	●	●	●	1	1	4
JP (IX)	PC ← IX	11 011 101 11 101 001	DD E9	●	●	●	●	●	●	2	2	8
JP (IY)	PC ← IY	11 111 101 11 101 001	FD E9	●	●	●	●	●	●	2	2	8
DJNZ, e	If B ← B−1 B≠0 PC ← PC+1	00 010 000 ← e-2 →	10	●	●	●	●	●	●	2	3	13
	If B=0 continue									2	2	8

Comments

cc	Condition
000	NZ
001	Z
010	NC
011	C
100	PO
101	PE
110	P
111	M

NZ : non-zero
Z : zero
C : carry
PO : parity odd
PE : parity even
P : sign positive
M : sign negative

Note : e represents the extension in the relative addressing mode.
e is a signed two's complement number in the range <−126, 129>
e −2 in the opcode provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.
e itself is obtained from opcode position.

Flags : ● =unchanged
0=reset
1=set
X=undefined
‡ =set or reset according to the result of the operation

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 10 Call and return group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States
CALL nn	(SP−1) ← PCH	11 001 101	CD	●	●	●	●	●	●	3	5	17
	(SP−2) ← PCL	← n →										
	PC ← nn	← n →										
CALL cc, nn	If condition cc is false continue,	11 cc 100	C4+	●	●	●	●	●	●	3	5	17
		← n →										
	otherwise same as CALL nn	← n →								3	3	10
RET	PCL ← (SP)	11 001 001	C9	●	●	●	●	●	●	1	3	10
	PCH ← (SP+1)											
RET cc	If condition cc is false continue,	11 cc 000	C0+	●	●	●	●	●	●	1	3	11
	otherwise same as RET									1	1	5
RETI	Return from interrupt	11 101 101	ED	●	●	●	●	●	●	2	4	14
		01 001 101	4D									
RETN	Return from non-maskable interrupt	11 101 101	ED	●	●	●	●	●	●	2	4	14
		01 000 101	45									
RST p	(SP−1) ← PCH	11 t 111	C7+	●	●	●	●	●	●	1	3	11
	(SP−2) ← PCL											
	PCH ← 0											
	PCL ← p											

Comments

cc	Condition
000	NZ
001	Z
010	NC
011	C
100	PO
101	PE
110	P
111	M

r	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

Flags : ● = unchanged
0 = reset
1 = set
X = undefined
‡ = set or reset according to the result of the operation

5

Powered by ICminer.com Electronic-Library Service CopyRight 2003

Table 11 Input and output group

Mnemonic	Symbolic operation	OP code 76 543 210	HEX code (Basic)	Flags C	Z	P/V	S	N	H	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A, (n)	A ← (n)	11 011 011 ← n →	DB	●	●	●	●	●	●	2	3	11	n → A_0-A_7 Acc → A_8-A_{15}
IN r, (C)	r ← (C)	11 101 101 01 r 000	ED 40+	●	↕	P	↕	0	↕	2	3	12	
INI	(HL) ← (C) B ← B−1 HL ← HL+1	11 101 101 10 100 010	ED A2	X	↕ ①	X	X	1	X	2	4	16	C → A_0-A_7 B → A_8-A_{15}
INIR	(HL) ← (C) B ← B−1 HL ← HL+1 Repeat until B=0	11 101 101 10 110 010	ED B2	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	
IND	(HL) ← (C) B ← B−1 HL ← HL−1	11 101 101 10 101 010	ED AA	X	↕ ①	X	X	1	X	2	4	16	
INDR	(HL) ← (C) B ← B−1 HL ← HL−1 Repeat until B=0	11 101 101 10 111 010	ED BA	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	
OUT (n), A	(n) ← A	11 010 011 ← n →	D3	●	●	●	●	●	●	2	3	11	n → $(A\text{-}BUS)_{0\text{-}7}$ Acc → $(A\text{-}BUS)_{8\text{-}15}$
OUT (C), r	(C) ← r	11 101 101 01 r 001	ED 41+	●	●	●	●	●	●	2	3	12	
OUTI	(C) ← (HL) B ← B−1 HL ← HL+1	11 101 101 10 100 011	ED A3	X	↕ ①	X	X	1	X	2	4	16	C → A_0-A_7
OTIR	(C) ← (HL) B ← B−1 HL ← HL+1 Repeat until B=0	11 101 101 10 110 011	ED B3	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	B → A_8-A_{15}
OUTD	(C) ← (HL) B ← B−1 HL ← HL−1	11 101 101 10 101 011	ED AB	X	↕ ①	X	X	1	X	2	4	16	
OTDR	(C) ← (HL) B ← B−1 HL ← HL−1 Repeat until B=0	11 101 101 10 111 011	ED BB	X	1 ②	X	X	1	X	2 2	5 (If B≠0) 4 (If B=0)	21 16	

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

Note : ①If the result of B−1 is zero the Z flag is set, otherwise it is reset.
②Z flag is set upon instruction completion only.

Flags : ● =unchanged
0=reset
1=set
X=undefined
↕ =set or reset according to the result of the operation

Powered by ICminer.com Electronic-Library Service CopyRight 2003