INSTRUCTION	C	Z	P/V	S	N	H	COMMENTS
ADD s; ADC s	↕	↕	V	↕	0	↕	8-bit add or add with carry
SUB s; SBC s, CP s, NEG	↕	↕	V	↕	1	↕	8-bit subtract, subtract with carry, compare and negate accumulator
AND s	0	↕	P	↕	0	1	Logical operations
OR s; XOR x	0	↕	P	↕	0	0	And sets different flags
INC s	•	↕	V	↕	0	↕	8-bit increment
DEC s	•	↕	V	↕	1	↕	8-bit decrement
ADD DD, SS	↕	•	•	•	0	X	16-bit add
ADC HL, SS	↕	↕	V	↕	0	X	16-bit add with carry
SBC HL, SS	↕	↕	V	↕	1	X	16-bit subtract with carry
RLA; RLCA, RRA, RRCA	↕	•	•	•	0	0	Rotate accumulator
RL s; RLC s; RR s; RRC s SLA s; SRA s; SRL s	↕	↕	P	↕	0	0	Rotate and shift location s
RLD, RRD	•	↕	P	↕	0	0	Rotate digit left and right
DAA	↕	↕	P	↕	•	↕	Decimal adjust accumulator
CPL	•	•	•	•	1	1	Complement accumulator
SCF	1	•	•	•	0	0	Set carry
CCF	↕	•	•	•	0	X	Complement carry
IN r, (C)	•	↕	P	↕	0	0	Input register indirect
INI; IND; OUTI; OUTD	•	↕	X	X	1	X	Block input and output
INIR; INDR; OTIR; OTDR	•	1	X	X	1	X	Z = 0 if B ≠ 0 otherwise Z = 1
LDI, LDD	•	X	↕	X	0	0	Block transfer instructions
LDIR, LDDR	•	X	0	X	0	0	P/V = 1 if BC ≠ 0, otherwise P/V = 0
CPI, CPIR, CPD, CPDR	•	↕	↕	X	1	X	Block search instructions Z = 1 if A = (HL), otherwise Z = 0 P/V = 1 if BC ≠ 0, otherwise P/V = 0
LD A, I; LD A, R	•	↕	IFF	↕	0	0	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag
BIT b, s	•	↕	X	X	0	1	The state of bit b of location s is copied into the Z flag
NEG	↕	↕	V	↕	1	↕	Negate accumulator.

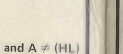
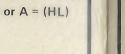
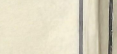
The following notation is used in this table:

SYMBOL	OPERATION
C	Carry/link flag. C=1 if the operation produced a carry from the MSB of the operand or result.
Z	Zero flag. Z=1 if the result of the operation is zero.
S	Sign flag. S=1 if the MSB of the result is one.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V=1 if the result of the operation is even, P/V=0 if result is odd. If P/V holds overflow, P/V=1 if the result of the operation produced an overflow.
H	Half-carry flag. H=1 if the add or subtract operation produced a carry into or borrow from into bit 4 of the accumulator.
N	Add/subtract flag. N=1 if the previous operation was a subtract.
	H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.
↕	The flag is affected according to the result of the operation.
•	The flag is unchanged by the operation.
0	The flag is reset by the operation.
1	The flag is set by the operation.
X	The flag is a "don't care."
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one of the CPU registers A, B, C, D, E, H, L.
s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16-bit location for all the addressing modes allowed for that instruction.
ii	Any one of the two index registers IX or IY.
R	Refresh counter.
n	8-bit value in range <0, 255>.
nn	16-bit value in range <0, 65535>.

SUMMARY OF FLAG OPERATION

MNEMONIC	SYMBOLIC OPERATION	FLAGS C	Z	P/V	S	N	H	OP-CODE 76 543 210	NO. OF T CYCLES	COMMENTS
LD r, r'	r ← r'	•	•	•	•	•	•	01 r r'	4	r, r' Reg: 000 B; 001 C; 010 D; 011 E; 100 H; 101 L; 111 A
LD r, n	r ← n	•	•	•	•	•	•	00 r 110 ← n →	7	
LD r, (HL)	r ← (HL)	•	•	•	•	•	•	01 r 110	7	
LD r, (IX+d)	r ← (IX+d)	•	•	•	•	•	•	11 011 101 01 r 110 ← d →	19	
LD r, (IY+d)	r ← (IY+d)	•	•	•	•	•	•	11 111 101 01 r 110 ← d →	19	
LD (HL), r	(HL) ← r	•	•	•	•	•	•	01 110 r	7	
LD (IX+d), r	(IX+d) ← r	•	•	•	•	•	•	11 011 101 01 110 r ← d →	19	
LD (IY+d), r	(IY+d) ← r	•	•	•	•	•	•	11 111 101 01 110 r ← d →	19	
LD (HL), n	(HL) ← n	•	•	•	•	•	•	00 110 110 ← n →	10	
LD (IX+d), n	(IX+d) ← n	•	•	•	•	•	•	11 011 101 00 110 110 ← d → ← n →	19	
LD (IY+d), n	(IY+d) ← n	•	•	•	•	•	•	11 111 101 00 110 110 ← d → ← n →	19	
LD A, (BC)	A ← (BC)	•	•	•	•	•	•	00 001 010	7	
LD A, (DE)	A ← (DE)	•	•	•	•	•	•	00 011 010	7	
LD A, (nn)	A ← (nn)	•	•	•	•	•	•	00 111 010 ← n → ← n →	13	
LD (BC), A	(BC) ← A	•	•	•	•	•	•	00 000 010	7	
LD (DE), A	(DE) ← A	•	•	•	•	•	•	00 010 010	7	
LD (nn), A	(nn) ← A	•	•	•	•	•	•	00 110 010 ← n → ← n →	13	
LD A, I	A ← I	•	↕	IFF	↕	0	0	11 101 101 01 010 111	9	
LD A, R	A ← R	•	↕	IFF	↕	0	0	11 101 101 01 011 111	9	
LD I, A	I ← A	•	•	•	•	•	•	11 101 101 01 000 111	9	
LD R, A	R ← A	•	•	•	•	•	•	11 101 101 01 001 111	9	
LD dd, nn	dd ← nn	•	•	•	•	•	•	00 dd0 001 ← n → ← n →	10	dd: 00 BC; 01 DE; 10 HL; 11 SP
LD IX, nn	IX ← nn	•	•	•	•	•	•	11 011 101 00 100 001 ← n → ← n →	14	
LD IY, nn	IY ← nn	•	•	•	•	•	•	11 111 101 00 100 001 ← n → ← n →	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	•	•	•	•	00 101 010 ← n → ← n →	16	
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	•	•	•	•	•	•	11 101 101 01 dd1 011 ← n → ← n →	20	
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	•	•	•	•	•	•	11 011 101 00 101 010 ← n → ← n →	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	•	•	•	•	•	•	00 101 010 00 101 010 ← n → ← n →	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	•	•	•	•	00 100 010 ← n → ← n →	16	

MNEMONIC	SYMBOLIC OPERATION	FLAGS C	Z	P/V	S	N	H	OP-CODE 76 543 210	NO. OF T CYCLES	COMMENTS
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	•	•	•	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	20	
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	•	•	•	•	•	•	11 011 101 00 100 010 ← n → ← n →	20	
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	•	•	•	•	•	•	11 111 101 00 100 010 ← n → ← n →	20	
LD SP, HL	SP ← HL	•	•	•	•	•	•	11 111 001	6	
LD SP, IX	SP ← IX	•	•	•	•	•	•	11 011 101 11 111 001	10	
LD SP, IY	SP ← IY	•	•	•	•	•	•	11 111 101 11 111 001	10	
PUSH qq	(SP-2) ← qq_L (SP-1) ← qq_H	•	•	•	•	•	•	11 qq0 101	11	qq: 00 BC; 01 DE; 10 HL; 11 AF
PUSH IX	(SP-2) ← IX_L (SP-1) ← IX_H	•	•	•	•	•	•	11 011 101 11 100 101	15	
PUSH IY	(SP-2) ← IY_L (SP-1) ← IY_H	•	•	•	•	•	•	11 111 101 11 100 101	15	
POP qq	qq_H ← (SP+1) qq_L ← (SP)	•	•	•	•	•	•	11 qq0 001	10	
POP IX	IX_H ← (SP+1) IX_L ← (SP)	•	•	•	•	•	•	11 011 101 11 100 001	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP)	•	•	•	•	•	•	11 111 101 11 100 001	14	
EX DE, HL	DE ↔ HL	•	•	•	•	•	•	11 101 011	4	
EX AF, AF'	AF ↔ AF'	•	•	•	•	•	•	00 001 000	4	
EXX	BC BC' DE ↔ DE' HL HL'	•	•	•	•	•	•	11 011 001	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	•	•	•	•	•	•	11 100 011	19	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	•	•	•	•	•	•	11 011 101 11 100 011	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	•	•	•	•	•	•	11 111 101 11 100 011	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	•	•	① ↕	•	0	0	11 101 101 10 100 000	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 Repeat until BC = 0	•	•	0	•	0	0	11 101 101 10 110 000	21 16	If BC ≠ 0 If BC = 0
LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	•	•	① ↕	•	0	0	11 101 101 10 101 000	16	
LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Repeat until BC = 0	•	•	0	•	0	0	11 101 101 10 111 000	21 16	If BC ≠ 0 If BC = 0
CPI	A - (HL) HL ← HL+1 BC ← BC-1	•	② ↕	① ↕	↕	1	↕	11 101 101 10 100 001	16	
CPIR	A - (HL) HL ← HL+1 BC ← BC-1 Repeat until A = (HL) or BC = 0	•	② ↕	① ↕	↕	1	↕	11 101 101 10 110 001	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A - (HL) HL ← HL-1 BC ← BC-1	•	② ↕	① ↕	↕	1	↕	11 101 101 10 101 001	16	

MNEMONIC	SYMBOLIC OPERATION	C
CPDR	A - (HL) HL ← HL-1 BC ← BC-1 Repeat until A = (HL) or BC = 0	•
ADD r	A ← A + r	↕
ADD n	A ← A + n	↕
ADD (HL)	A ← A + (HL)	↕
ADD(IX+d)	A ← A + (IX+d)	↕
ADD (IY+d)	A ← A(IY+d)	↕
ADC s	A ← A + s + CY	↕
SUB s	A ← A - s	↕
SBC s	A ← A - s - CY	↕
AND s	A ← A ∧ s	0
OR s	A ← A ∨ s	0
XOR s	A ← A ⊻ s	0
CP s	A - s	↕
INC r	r ← r + 1	•
INC (HL)	(HL) ← (HL)+1	•
INC(IX+d)	(IX+d) ← (IX+d)+1	•
INC (IY+d)	(IY+d) ← (IY+d)+1	•
DEC d	d ← d -1	•
ADD HL, ss	HL ← HL+ss	↕
ADC HL, ss	HL ← HL+ss+CY	↕
SBC HL, ss	HL ← HL-ss-CY	↕
ADD IX, pp	IX ← IX + pp	↕
ADD IY, rr	IY ← IY+rr	↕
INC ss	ss ← ss + 1	•
INC IX	IX ← IX + 1	•
INC IY	IY ← IY + 1	•
DEC ss	ss ← ss -1	•
DEC IX	IX ← IX -1	•
DEC IY	IY ← IY -1	•
RLCA		↕
RLA		↕
RRCA		↕
RRA		↕