Instruction set classifications

Instruction set determines, what functions the microprocessor can perform with a single instruction.

Instructions can be classified into five categories.

(1) Data transfer [copy] instructions.
(2) Arithmetic operations.
(3) Logical operations.
(4) Branching operations.
(5) Machine controlled operations:

Data can be available at:

1. Immediately available (Im)
2. Available at some memory (M)
3. at some register (R).
4. at some IO (IO).

Data can be required at:

(1) some memory
2. in some register.
3. at some IO.

Table representation of possible data transfer.

Data available	Data to transfer.
Im	R, M
R	R, M
M	R
IO	A - Accumulator
A	IO

Arithmetic operation

1. Addition.
2. Subtraction. - 8085 do not perform 16 bit subtraction
3. Increment / Decrement [Accumulator [illegible]]
   └ These operations can be performed directly in the source itself.

Data Transfer Instruction

① Move (R_d R_s). Register to Register. Register Addressing. One Byte. 4T. total

opcode fetch- 4T.

Move A, B.

Move A, A
B
C
D
E
H
L
(7)

MOV B, A
MOV B H
MOV B, L
(7)

, MOV C, A
(7)
MOV C, L
MOV D, A
(7)
MOV D L

, MOV E, A
MOV E L
(7).
MOV H, A
MOV H, L
(7)

MOV L, A
L, B
L, C
L, L
(7)

② Memory to Register. [MOV R_s, M] $A_d \leftarrow M$, [$R_d \leftarrow HL$]
Register Indirect.
Content of memory addressed by HL pair is moved to A register

One byte, Register indirect.
op-code fetch- 4T.
Memory read - 3T
7T
Total instructions = 7,

MOV A, M
B, M
C, M
D, M
E, M.

MOV H M.
MOV L, M

③ Move Register to memory:- Register Indirect

One Byte, 7 in number,
MOV (M, R_s).

opcode fetch- 4 T.
Memory write = 3T
7T

MOV A, M,
B, M
C, M
H, M
L, M

~~MVI (Rd, d) Rd ← 8b.~~

(4). ~~Move M, Rs~~
LDAX rp. $A \leftarrow M$ or $A \leftarrow rp$

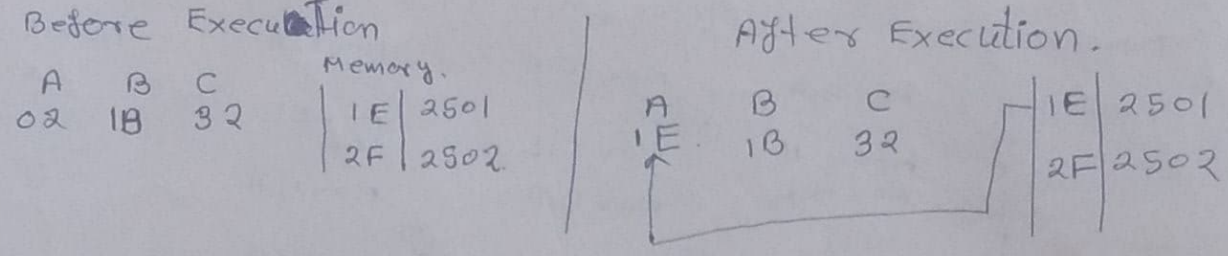
Content of memory addressed by the register pair (
is moved to accumulator. [The content of register p
is in memory address). register pair can be BC

One Byte, Register indirect.
Total No of instruction 2
Two machine cycle: ① Op code fetch - 4T
Memory Read - 3T
7T.

Example:

Move the content of memory location addressed by BC pair is moved to accumulator, ie Register A.

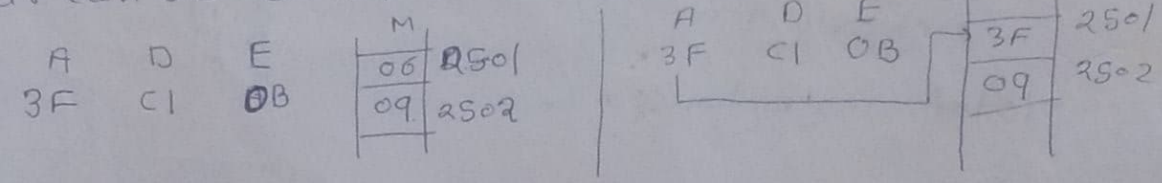
LDAX → B , A ← BC. or A ← M.



(6) STAX rp STAX D M ← A or DE ← A

One byte, Register indirect.

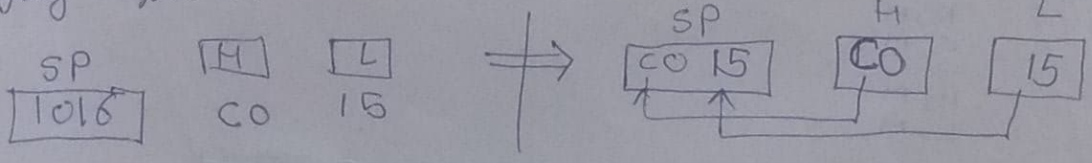
Content of the accumulator is moved to memory addressed by the register pair (rp). [The content of register pair is in the memory address). The register pair can be either BC or DE.



(7) SPHL (SP) ← (HL).

one byte Implied addressing

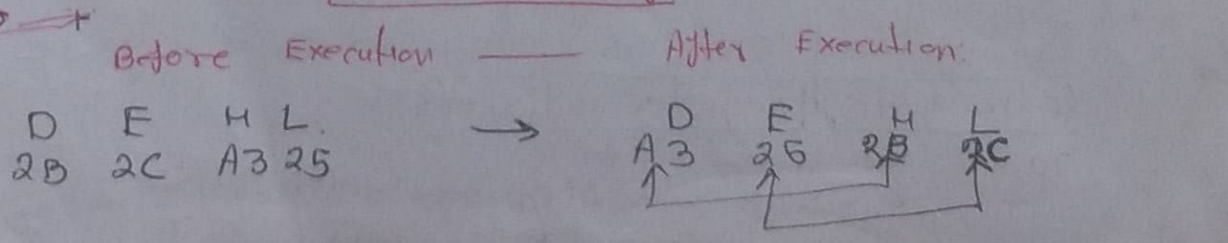
Content of HL pair is moved to stack pointer. NO flag affected.



(8) Exchange: XCHG E ↔ L, D ↔ H

contenteut of HL pair is exchanged with DE pair.

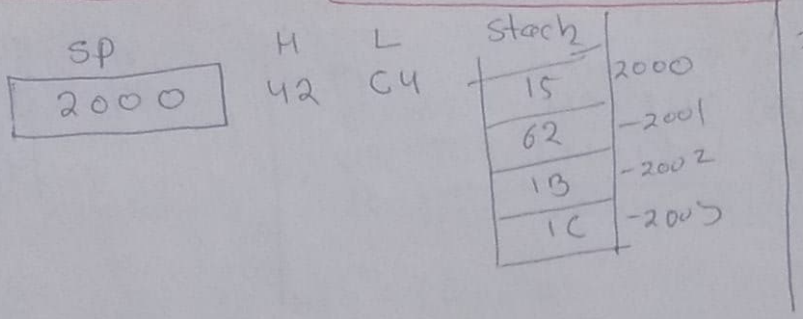
One byte, Implied addressing

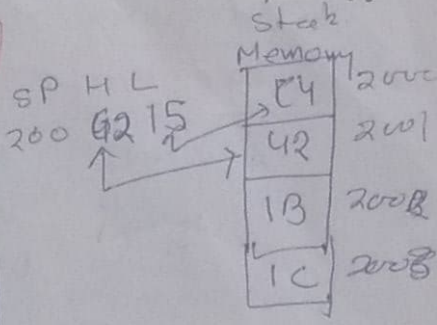


(9) XTHL (HL) ⟷ (M) or HL ⟷ SP

Content of stack Pointer is exchanged with HL pair.

One byte | Implied addressing





(10) Push rp :- (SP) ← SP-1 : (SP) ← (rp)H
(SP) ← SP-1 (SP) ← (rp)L

one byte | Register indirect

The content of register pair (rp) is pushed to stack. After execution of this instruction, the content of stackpoint will be less than the earlier value by a factor of 2.

(11) POP rp rp ← (SP) , SP ← SP+1.
(rp)H ← SP , SP ← SP+1.

Exp- ADD A, Add H→
ADD E Add L→

: Arithmatic Instruction:

1. Add reg A ← (A + reg) A ← (A + B)

One byte, ~~Im~~ Register addressing, Flag affected

Content of register B is added to A-register.

Before execution		Addition	After execution.	
A	B	C2 = 1100 0010	A	B
C2	B8	B8 = 1011 1000	7A	A
		carry 1 0111 1010	CF = 1	
			SF = 0	
			PF = 0	
			AF = 0	
			ZF = 0	

(2) Add M $A \leftarrow (A+M)$. or $A \leftarrow A + (HL)$

The content of memory addressed by HL pair is added to the content of A register. After addition, the result is stored in register A, All flags affected.

One byte | Register indirected

A	HL
44	C00A

↑ address of HL pair

(44) 0100 0100
(73) 0111 0011
(B7)= 1011 0111

C00A – 73
C00B – 2B
C00C – 3E

A	(HL)	
B7	73	C00A
	2B	C00B
	3E	C00C

(3) ADC reg. $A \leftarrow (A) + d8 + CF$

One byte, register addressing

Content of register and carry flag are added to the content of A-register. After ~~the result~~ operation result is stored in register. All flag affected.

A	H
43	7A

CF = 1
PF = 0
AF = 0
ZF = 0
SF = 1

Add
43 = 01000011
7A = 01111010
CF = 1
10111110
S = BE
Carry = 0

After execution

A	H
BE	7A

CF = 0
PF = 1
AF = 0
ZF = 0
SF = 1.

(4) ADC M. $A \leftarrow A+M+CF$. or $A \leftarrow A+(HL)+CF$.

One byte
Register indirect.

(5) Sub reg $A \leftarrow A-reg$ $A \leftarrow A - C$

content of register C is substracted from register CA

One byte, register addressing. Total No = 7.
Sub A, ... B ... E
H, Sub L

(6) Sub M. $A \leftarrow A-M$, or $A \leftarrow A-(HL)$.

One byte, register indrected.

(7) SBB M → $A \leftarrow A-M-CF$. , $A \leftarrow A-HL-CF$.

content of memory location addressed by HL and the value of the carry C before executing the instruction) are subtracted from accumulator. After subtracting result is stored in register A. All flag affected.

(8) DAA – Decimal adjust Accumulator.

After BCD addition, the DAA is executed to get result in BCD. When DAA instruction is executed, the content of the accumulator is altered or adjusted.

One byte, implied addressing.

i) If the sum of lower nibble exceeds 09_H or auxiliary carry is set, then a correction of $(06)_H$ (0110) is added to lower nibble.

(ii) If the sum of the upper nibbles exceeds $(09)_H$ or carry is set, then a correction of $(06)_H$ (0110) is added to sum of upper nibble.

(14) DAD rp. – Doubble Addition: The content of register pair is added to the content of the HL-pair. After addition the result is stored in the H-L pair. Only carry flag is affected. Register pair can be BC, DE, HL, or SP.

One byte instruction.

Register addressing.

DAD-B, DAD-D, DAD H.
DAD SP.

(15) INR reg :→ reg ← (reg) + 01.

Content of register is incremented by register 1. Except carry flag all other flags are affected. The register can be any one of the general purpose register A, B, C, D, H, E or L.

One byte, register addressing.

(16) INR M:– M ← M + 01

06	2501
AF	2502
FF	2503

2501 = 00000110
+1 00000001
00000111

07	2501
AF	2502
EF	2503

One byte register indirect addressing.

(17) DEC M:– M ← (M) – 01 or HL ← (HL – 01)

Contented of ~~memory~~ addressed by HL pair is decremented by one. Except carry all other flags are affected.

One Byte.

Register indirect.

HL
2010

FA	2010
0B	2011

⟹

HL	Memory	
2010	F9	2010
		2011

(18) :- Dec reg. reg ← (reg) - 01.

content of register is decremented by one.

E ← (E) - 1

example = E = 60, ⟹ E = 5F.

(01)$_H$ = 00000001

11111110

= (11111111) FF.

60 = 01100000

10011111

5 F.

one byte, register addressing.

(19) INR rp rp ← rp + 01.

INR H (H) ← HL + 01.

content of HL pair is incremented by one.

HL ⟹ HL

00FF 0100

one byte, Register addressing.

20 : DCX rp rp ← rp - 01.

content of register pair is decremented by one.

one byte, register addressing.