Assembly Programming

Instruction Set

I. Data Transfer

Instruction with example	Working
(1) MOV B C	Moves data from one location to another.
(2) MVI B 05h → Data.	Move Immediate. Moves data directly into a location.
(3) LXI B 2000h	Load Pair Immediate. Loads address immediately in register pair.
(4) LDA 3000h	~~a~~ Load Accumulator. Go to the mentioned location in RAM and load the contents of that location in accumulator.
(5) STA 3000h	Store Accumulator. Save the contents of accumulator in the mentioned location in RAM.
(6) LDAX B	Load Accumulator from pair Load Accumulator with data in the address saved in the register pair.
(7) STAX B	Store Accumulator into pair. Store the contents of Accumulator in the memory address saved in the register pair.

II. Arithmetic

1.	ADD	B	AC ← AC + B (Direct)
2.	ADI	05h	AC ← AC + 05h (Immediate)
3.	ADD	M	AC ← AC + (HL) M denotes (HL) pair i.e, Memory
4.	ADC	B	Add with carry. AC ← AC + B + Cy ← previous Carry
5.	SUB	B	AC ← AC − B
6.	SUI	05h	AC ← AC − 05h
7.	SBB	B	Subtract with borrow. AC ← AC − B − Borrow.
8.	INR	B	B ← B + 1. Increase value of data by 1.
9.	INX	B	BC ← BC + 1 Increase address stored in register pair by 1.
10.	DCR	B	B ← B − 1
11.	DCX	B	BC ← BC − 1
12.	DAD	B	HL ← HL + BC Add 2 memory addresses, stored in register ~~pair~~ pairs. Destination is always stored in HL pair.

III. Logical

1. ANA B

AND with Accumulator.

AC ← AC & B

2. ANI 05h

AND with Accumulator Immediate.

AC ← AC & 05h

3. ORA B

OR with AC

AC ← AC | B

4. ORI 07h

OR with AC Immediate.

AC ← AC | 07h

5. XRA B

XOR with AC.

6. XRI 05h

XOR with AC Immediate.

7. CMP B

Compare with AC.

If,

A > B	C=0	Z=0	S=0
A < B	C=1	Z=0	S=1
A = B	C=0	Z=1	S=0

Here, C → Carry flag
Z → Zero flag
S → Sign flag.

8. CMP M

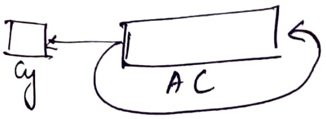
Compare AC with data saved in HL pair.

9. CPI 05h

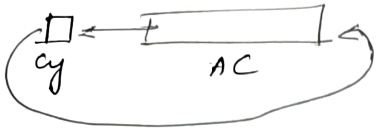
Compare AC immediate

10. RLC

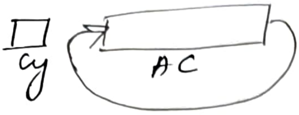
Rotate left without Carry



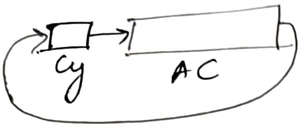
11. RAL — Rotate left <u>with</u> Carry



12. RRC — Rotate right <u>without</u> carry.



13. RAR — Rotate righ <u>with</u> carry.



14. CMA — Complement AC. $AC \leftarrow \overline{AC}$

15. STC — Set Carry. $C \leftarrow 1$.

16. CMC — Complement Carry. $C \leftarrow \overline{C}$

IV <u>Branching</u>

1. JC		Jump if Carry is set.
2. JNC		Jump if Carry is not set.
3. JZ		Jump if Zero is set.
4. JNZ		Jump if Zero is not set.
5. JMP		Jump
6. JPO	(Jump if odd)	Jump if Parity is set. (Odd)
7. JRE	(Jump if even)	Jump if Parity is not set. (Even)
8. JP	(Jump if positive)	Jump if Sign is not set. (+ve)
9. JM	(Jump if minus)	Jump if Sign is set. (−ve)