8085 Microprocessor and Its Programming

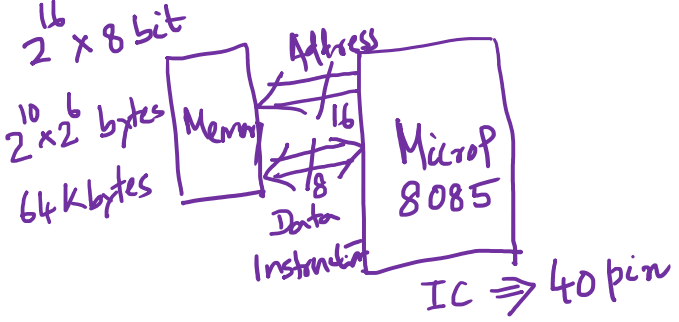
Intel → 1976

8-bit → Processing Capability
ADD, SUB, MUL, etc. 8-bit numbers
Internal Datalbus size = 8 bit

8085 → Digital Processor → 240+ instructions (Assembly)

Programming → Writing Programs using 8085 instructions

Programming Language → High level Language → C, Java (Handled by humanbeings)

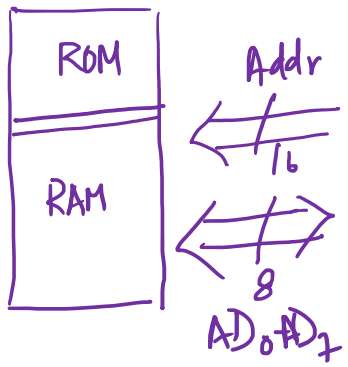


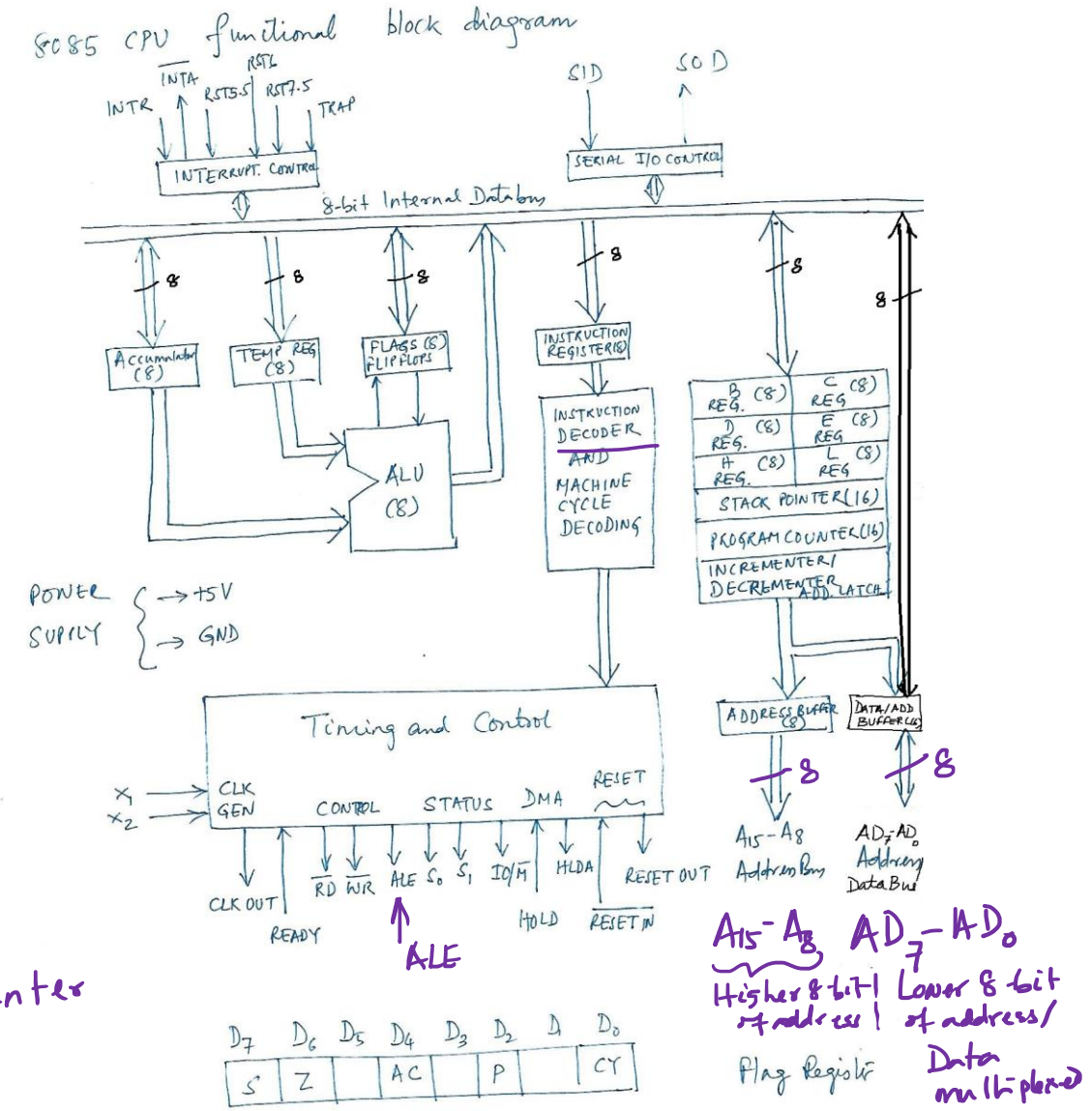
⇒ Assembly Language (Low level Language)

↓ Assembler (Software) / Manually by handcoding

Machine Code ← fed to the memory of the microprocessor.

8085 Programming Model





ALE

A_{15}-A_8 Higher 8 bit of address

AD_7-AD_0 Lower 8 bit of address / Data multiplexed

SP ← Stack Pointer (16)

PC ← Program Counter (16)

→ 6 General purpose registers to store 8-bit data
B, C, D, E, H, L
Can be combined as register pair to perform 16-bit operation
BC, DE, HL

ALU → Arithmetic Logic Unit

```
  1011
  1001
------
 ←0100
CY 1
```

ADD A, B A ← A+B

↓

Machine Code

↓

Instruction Register

↓

Instruction Decoder

Registers → Storage blocks

B, C, D, E, H, L ← 8-bit 8 bit Registers

BC
DE
HL
} 16-bit Registers

HL — memory pointer (M)

HL is a special register used as a memory pointer

Accumulator (Usually called A reg.)

8-bit reg → used in ALU
Used to store the results of arithmetic and logic operations.

Flags
- Have critical importance in the decision making process of the MP.

Z - Zero flag, set to 1 when the result is zero.

CY - Carry flag. Set when the result of an arithmetic operation results in a carry.

S - Sign flag, Set if the bit D_7 of the result is 1.

P - Parity flag, Set if the result has even number of 1.

AC - Auxiliary carry. Set when a carry is generated by digit D_3 and is passed to digit D_4.

Program Counter (PC) - 16 bit register

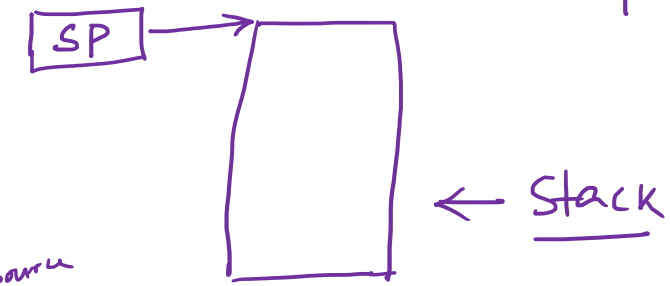
- PC is used to sequence the execution of instructions.
- The function of the PC is to point to the memory address from which the next byte is to be fetched

- When a byte is being fetched, the PC is incremented by one to point to the next memory location.

Stack Pointer (SP)

- A 16-bit register used as a memory pointer, pointing to the R/W memory location, called Stack.

Part of the memory (Temporary storage)



8085 Instruction Set:

five functional categories

- Data transfer (copy) — MOV A, B (Dest, Source) — A ← B
- Arithmetic operations — ADD B — A ← A + B
- Logical operation
- Branching operation — JMP <addr16>
- Machine control operations — HLT

246 Instructions in total

A total of 246 instructions.

Data Transfer Operation

- Between registers — A ← B — MOV A, B
- Specific data to a register or memory location — B ← 32H — MVI B, 32H; M ← 32H — MVI M, 32H
- Between memory location and a register — B ← (Mem) — MOV B, M
- Between I/O device and the accumulator — A ← (I/O add.) — IN PORT

MVI B, 32H — B ← 32H

MOV B, M — B ← (HL) (HL is the memory pointer)

IN PORT — A ← (PORT) (External Dev)

Compare

A byte, reg or (mem) can be compared with the contents of A.

Equal
GT
LT

Complement

$A \leftarrow \overline{A}$

Branching Operations

Jump → Conditionally based on flags
Jump → or Unconditionally

Call, Return, Restart

Machine Control Operation:

Control machine operations

Halt — HLT
Interrupt — RST
Do nothing — NOP

Arithmetic Instructions

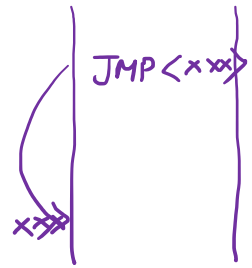
ADD B — $A \leftarrow A + B$

Logical Instructions

AND, OR, Ex-OR

ANA B — $A \leftarrow A$. AND. B

JMP <address 16>



Addressing Modes in 8085 Instructions

How does 8085 access the Data?

Direct Addressing: Address of the data is directly specified in the instruction.

LDA <Address 16>
(Load Accumulator)

A ← (Address16)

LDA 8020H A ← (8020H)

Indirect Addressing

The memory/data address is indirectly specified, i.e., the data address is available in a register pair.

MOV A, M

A ← (HL)

Immediate Addressing

Data is immediately available in the instruction

MVI A, 32H A ← 32H

32H ↑ immediate data

Register Addressing

Data is available in the register specified

ADD B A ← A + B

Instruction Format

Each instruction has two parts.

Operation code (Op-code) = Task to be performed.

Operand = Data to be operated on
Reg., 8-bit or 16-bit data, mem. location
8/16 bit address.
. Sometimes operand is implicit.

Instruction word size

1-byte instructions
2-byte instructions
3-byte instructions

1 byte instructions (Mnemonic followed by a letter or two letters representing register)

Opcode	Operand	Binary code	Hex code
MOV	C, A	0100 1111	4FH
ADD	B	1000 0000	80H
CMA		0010 1111	2FH

2-byte instructions (A mnemonic followed by 8-bit data)

Opcode	Operand	Binary code	Hex code
MVI	A, 32H	0011 1110	3E
		0011 0010	32

3-byte instructions (A mnemonic followed by a 16-bit address)

Opcode	Operand	Binary code	Hex code
LDA	2050H	0011 1010	3A
		0101 0000	50
		0010 0000	20

Higher byte, Lower byte