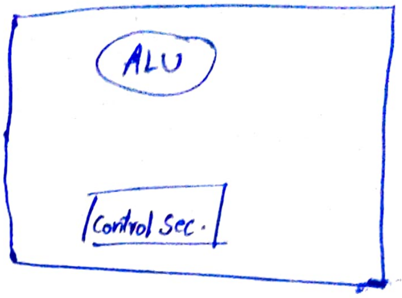
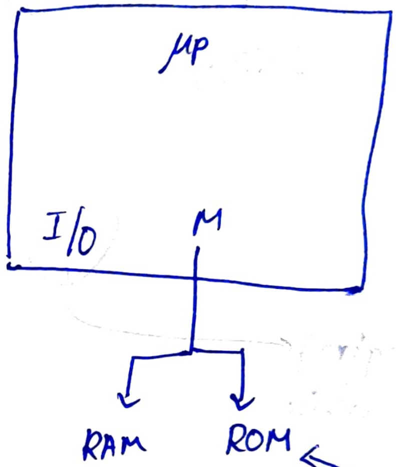
Microproccesor => Can run multiple applications

-> It is an integrated chip in which multiple things are embeded.



Micro Controller => made to run single applications

-> Integrated circuit with μP & I/O & Memory.



BIOS in Rom

Bootloader - helps in programable Rom

stored inside Rom

↳ helps when to use external programer.

=> 8 Core means 8 μP Can run parallely.

Intel major contributor of μP & μC.

↳ 8080 (voltage issues)

=> 8085 (1st comm. Successfull)

=> 8086 (To ↑ speed they made this) (Adv. μP)

=> INTEL + IBM

↳ 80 286
80 386
80 486
80586 (around 1995)

→ Basics 8085, 8086.

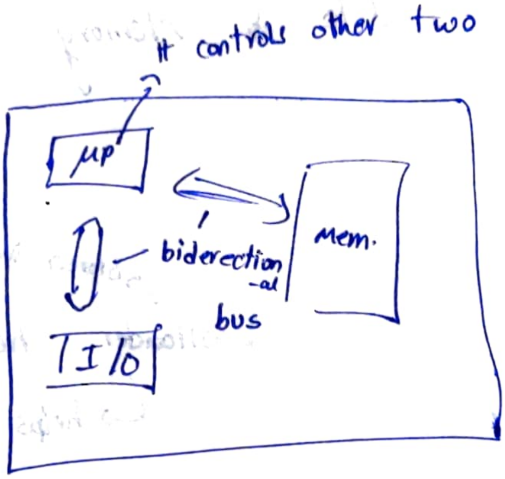
8085 ↓ 8-bit μp

8086 ↳ 16-bit μp

→

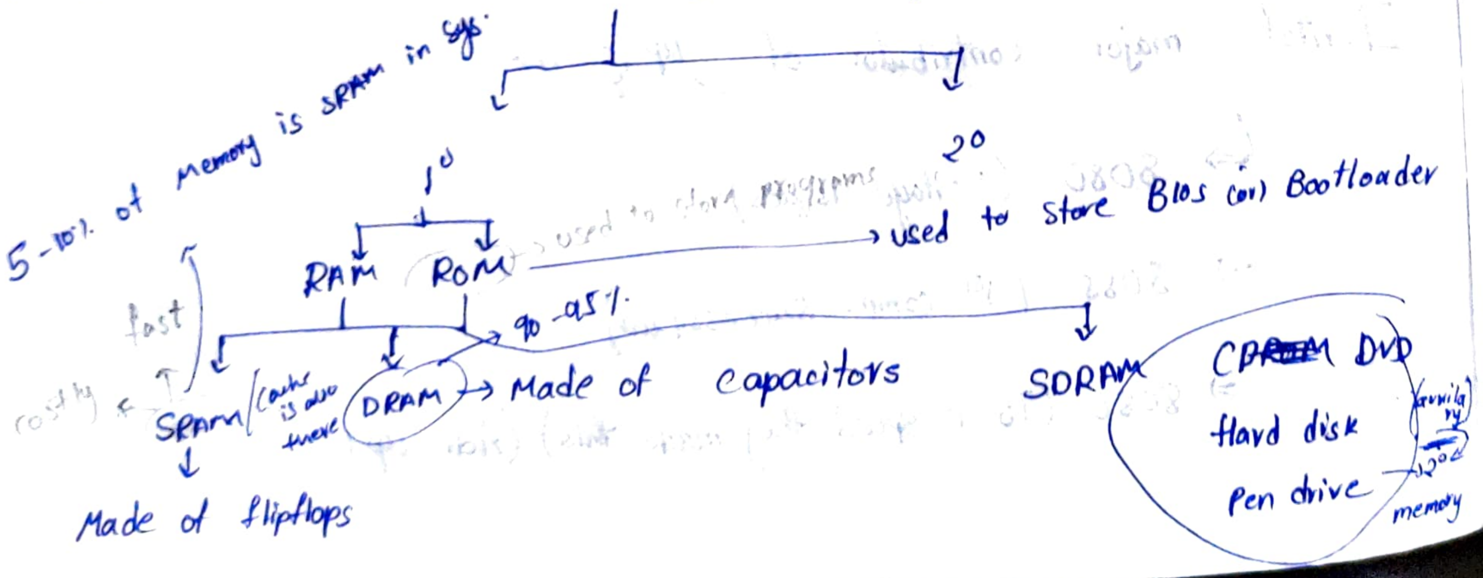
Input devices

- → Mouse
- → keyboard
- → Camera
- → Micro phone
- → Sensors
- → Scanner



Computer Sys.

=> Memory = Program + data



→ Stored in 0, 1's

0 → 0V 1 → High logic

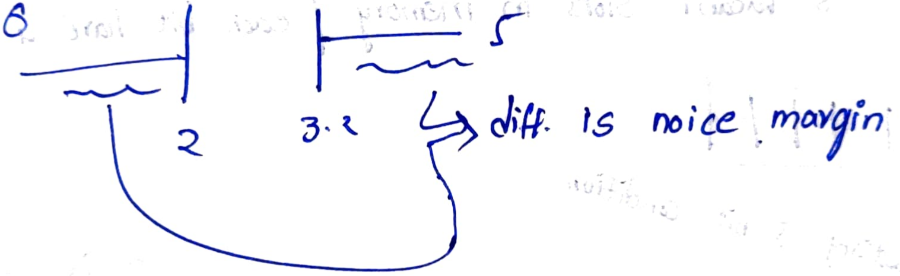
> 3.2 V ⇒ logic high.

If voltage in between 0 – 2.2 V ⇒ Logic 0

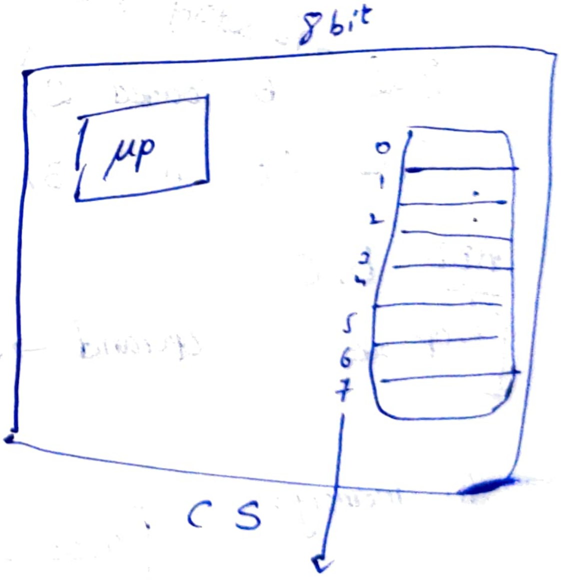
" " " " 2.2 – 3.2V → Garbage

" " " " ≥ 3.2 v → Logic High (1)

– ≤5



↳ diff. is noice margin



→ 2+3 = 5 → High level language

```
int x = 2;
int y = 5;

z = x+y;
```

Compiler

HLL MLL

If no bug then compiler turns into MLL.

Numbering always in Hexadecimal

8 bit → byte

8+8 bit → word

µp won't have variables they have registers (made up of flip flops)

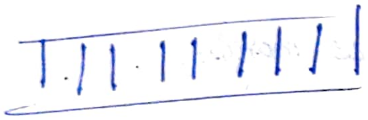
General purpose

A → Accumilator
B
C
D } → general purpose
E
F

Spl purpose
↓
Can't be used by user

→ Registers are semi conductor based used to store data temporarily in µp

8-bit means 8 vacant slots in memory / each bit have a flipflop.



B,02 ↗ To satisfy 8-bit condition (B loaded 2)
C,03 (C " 3)

2 + 3
↓
0000 0010

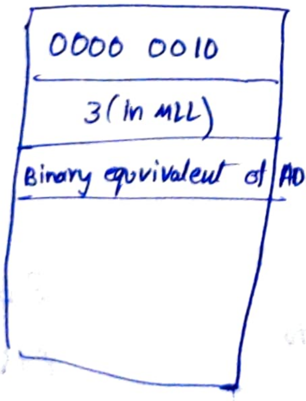
2 is 4-bit only

but we need to use 8-bit value

So 2 = 0000 0010

ADD B, C
[↳ op code operand → data]

In memory:


0000 0010
3 (in MLL)
Binary equivalent of ADD is stored here

Now µp task is to add these numbers.

µp ⟺ it interacts with bus

Fetching : Generates the address and reads the data.

Control
↳ tells to read / write

* μp decodes meaning of Binary equi. of ADD and gets to know it Should add and then executes.

=> Fetch
Decode
Execute -> Instruction Cycle (IC) / Machine cycles

* 8085 needs 4 cycles to give 1 IC