Modern Digital Signal processors : (EEG303E)

* Importance... Why ?

↳ Easiest electronics device !! - Easier than transistor.
↳ project.... Robotics... Automation...
↳ In the Job market, survival of the fittest.
↳ Overall development.

Basics :-

• Computer Languages

↳ to instructing the computer to act as we want.
↳ Coding, program.

"key words" → already defined
"Syntax" → (like grammer)
Set of instructions → Called program.

① High level language — C, Basic, Java, Python
↳ 'Easy'
AI ML

② Assembly language → Mnemonics
↓ (Registers)

```
MOV  A, B
ADD  X, Y, Z
SUB  R1, R2
```

③ M/c Language → consists only combinations of 0 and 1.

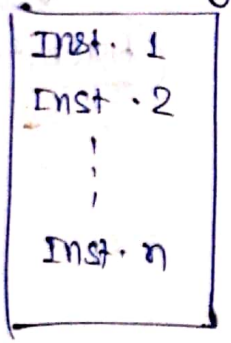
Adv. and Disadv. of each languages.

Computer language

↳ instructing the computer to act as we want.
Coding.... program...

- Key words ⟷ Words
- Set of Rules — syntax ⟷ Grammer.

Programs → set of instruction written to do a function.



① HLL :- C, Java, Basis, python (AI/ML)
↳ libraries are there which is used in

* High level language:- (portability) ⟷ compiler, Interpreter

↳ English like language — with proper grammer, called, 'Syntax'.

↳ Far more simpler ~~than~~ to understand for the humans, than the assembly level language or machine level language.

↳ There are determined statements for writing, each and every instruction.

↳ However, whatever language you learn, you need to have a fine conception of the basic of that computer language.

↳

Architecture :-

(2) Assembly language :-

↳ Building block ⟺ Mnemonics → Assembler → OP-code

ADD A, B, C
MOV R_1, R_2
SROT X, 4

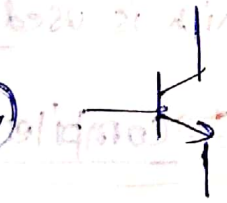
OP-code ← set of 0's & 1's
OP-code → Represented by hexadecimal code (BJT)

all the info. is given in Data sheets → Architecture

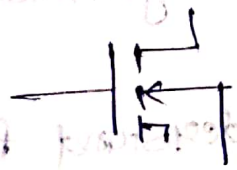
Adv. of digital over analog ?

Switch → Mech.
→ electromechanical (Relay)
→ purely electrical/electronics/semiconductor switch.

BJT



MOS



~~Sica, Gaanti~~
Diode
↓
transistor
↓
SCR
↓
BJT's

Char:

0 1 1 0 1 1 0 1 1 0 1 0 1 0

2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

* Number systems —

Binary (Base 2: Digits are 0, 1, --? $)_{10}$

Octal (Base 8: 0, 1, 2, 3, - - - 7 - ? $)_{10}$

Decimal (Base 10: 0, 1, 2, - - - 9 ? $)_{10}$

Hexadecimal (Base 16: 0, 1, 2, - - 9, A, B, C, D, E, F ? $)_{10}$

2) 57 (1
2) 28 (0
2) 14 (0
2) 7 (1
2) 3 (1
1

1011 0101 → Binary

B 5 → Hexadecimal

H.W

* Convert from any base to any other Base

- $(876)_{10}$ to binary, Hex, octal
- $(24A)_H$ to " decimal octal
- same → $(1011101)_2$ to decimal Hex, octal
- same → $(1011101)_2$ to ——

ASCII CODE ? ($0 = 30_H$)

BCD code ?

* **ASCII Code :-** (American Standard code for Information interchange.) ⇒ Character encoding standard for electronic communication.

↓

↳ (7 Bits)

↳ (8 bit for extended Ascii)

↳ it's range is from 0 to 127.

- first 31 (0–31) is for control character.
- codes (32–127) are for printable character, it represent letters, digits and some symbols.

eg:- (48 – 57) → (0 to 9)

(65 – 90) → (A to Z)

(97 – 122) → (a to z)

"alpha neumeric character."

* **BCD Code :-** Binary Coded decimal.

↓

- each decimal digit is Represented by a 4-bit binary number.
- Positional weights are 8-4-2-1

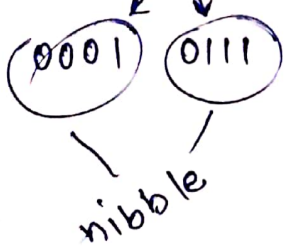
(0 → 9) ⇒ decimal digits.

there are six BCD Representations which are invalid → (1010, 1011, 1100, 1101, 1110, 1111)

10 15

- ~~dec~~ decimal → BCD

eg: $(17)_{10}$ → 0001 0111



BCD → Decimal

0001 0100 → $(14)_{10}$

1 4

↳ BCD is taking more bits so it is less efficient than binary.

- it is sometimes desirable to manipulate numbers in decimal instead of converting them to Binary.
  b/c → decimal to binary & binary to decimal conversion process is complex.

(The binary ckts are easy and more efficients.)

So, we use BCD to directly convert decimal to binary by just seeing it.

5	7
N_1	N_2

→ Packed BCD

(1B)

0	7
N_1	N_2

↳ unpacked BCD.

83 → Packed BCD

(08 03) → unpacked BCD.

processor .... The past .....

First microprocessor.

Intel → 4004

↓

full form ?

INtegrated Electronics

intel – 4004

↳ 4-Bit microprocessor
(Process 4 bit of data at a time)

Word size (n-bit means process n bit at a time.)

- 4-bit microprocessor (ALU can perform 4 bit data at a time.)
- 8-bit "
- 16-bit "
- 32-bit "
- 64-bit "

↳ MOOR'S Law (it governs the speed of the processor)

↓

every two year, the data which is getting processed should get doubbled. (i.e; the bits should get double)

- First computer mad by Charls Babbage – Analytical Engine / machines

↓ using → Gears → (elec. equivalent → transformer)

↓

Static computer

↓

Speed changes

• History of Microprocessor :-

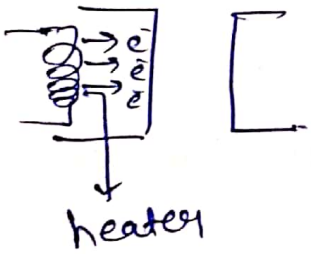
Z3 → first electro mech., general purpose Relays program-controlled computer built by Germans during world war-2.

COLOSSUS

ENIGMA

ENIAC.

Vaccum tubes



Transistors

IC → to integrate many transistors.

↳ 2500 vowels and 8KW power required.
↳ Reliable.

* ENIGMA :- (working based on electric ckts.)
Used to encrypt and decrypt msg.
Used by German military in 1930s and in world war 2.
There is a keyboard of 26 letters.

3 rotors ⟹ plugboard

Battery → key switch → plugboard → Rotors → plugboard → keyswitch → Light bulb → Battery.

↳ letters changes 7 to 9 times in between.

* ENIAC :-(Electrical numerical integrator & computer)
↳ first computer having the facilities of reprograming.

area - 1800 sq.ft
Height - 8 ft.
Length - 80 ft
Depth - 3 ft.
weight - 30 tons.

↳ No storage.
↳ Calculation speed - 5000/sec.

~~Ballistic~~

In feb. 1944

- World first digital electronics computer.
- Made by switches & wiring cables.

→ use IBM punch card reader for input and IBM Card Punch m/c for o/p.
→ 40 panels and 80K vaccum tubes, 70K resistors, 10K capaci-

* EDVAC (Electronic Discrete variable Automatic computer.)
↳ it was binary., designed to be a stored-program computer.
↳ delivered to the Ballistics Research laboratory in 1949.

- **Z_3 (Computer)** → German electromechanical computer. designed by Konrad Zuse (1938 - 1941)
  - ↳ World 1st programable, fully automated, digital Computer. (Build with 2600 relays)
  - ↳ operated as a Stack machine with a stack of two registers, R_1 and R_2.

Specifications

- Avg calculation speed ⇒ Addition – 0.8 sec. multiplication – 3 sec.
- Arithmetic unit → Binary - floating point, 22-bit, add, sub. multiply, divide, sq. root.
- data memory → 64 22-bit words.
- I/P – O/P → decimal floating point numbers.
- freq^n → 5 - 10 Hz.
- Power consumption : around 4000 W
- weight → Arround 1 tonne.

- **COLOSSUS** (Developed by british codebreakers in (1943-45)
  - ↳ The World first electronic computer → (Degined by Tommy Flowers)
    programmed by switches and plugs.
    Programmed to help in cryptanalysis. (To encrypt the msg. from German high command.)

it recieve Radio msg. onto paper tape and load it onto Colossus. as an endless loop.

- ↳ then it read optically by colossus @ 5000 characters a sec.
- ↳ COLOSSUS has no memory.
- ↳ programming done on switch panels.

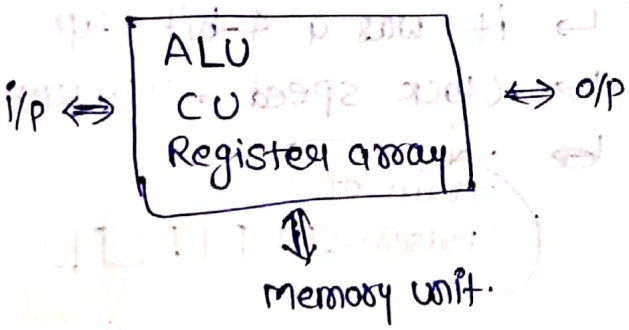
* History of Microprocessor :-

- First generation comp. (1939 - 1954) ⇒ Made up with the help of vaccum tubes.
- Second generation (1954 - 1959) ⇒ Transistor
- Third generation (1959 - 1971) ⇒ IC
- fourth generation (1971 - present) ⇒ Microprocessor.

↳ µp is identified with word size of data.

- 4-bit processors : INTEL 4004, INTEL 4040
- 8-bit processors : 8008, 8080 (2MHZ), 8085 (5MHZ), MOTOROLA 6800 (M6800)
- 16-bit processors : 8086 (5MHZ), 8088 (5MZ), Zilog Z800, 80186, 80286 (6 MHZ)
- 32-bit processors : INTEL 80386 (16MHZ), 80387, 80486 (25 MHZ), INTEL PENTIUM (60MHZ), PENTIUM PRO, PENTIUM II (233MHZ), PENTIUM III (450 MHZ), PENTIUM IV (1.5GHZ), DUAL CORE
- 64-bit processors :- INTEL CORE 2, CORE I7, CORE I5, CORE I3

µp



SMD
BGA

IC-packages

TERAFLOP

↓

Used to handle No., decimal po

Moore's law

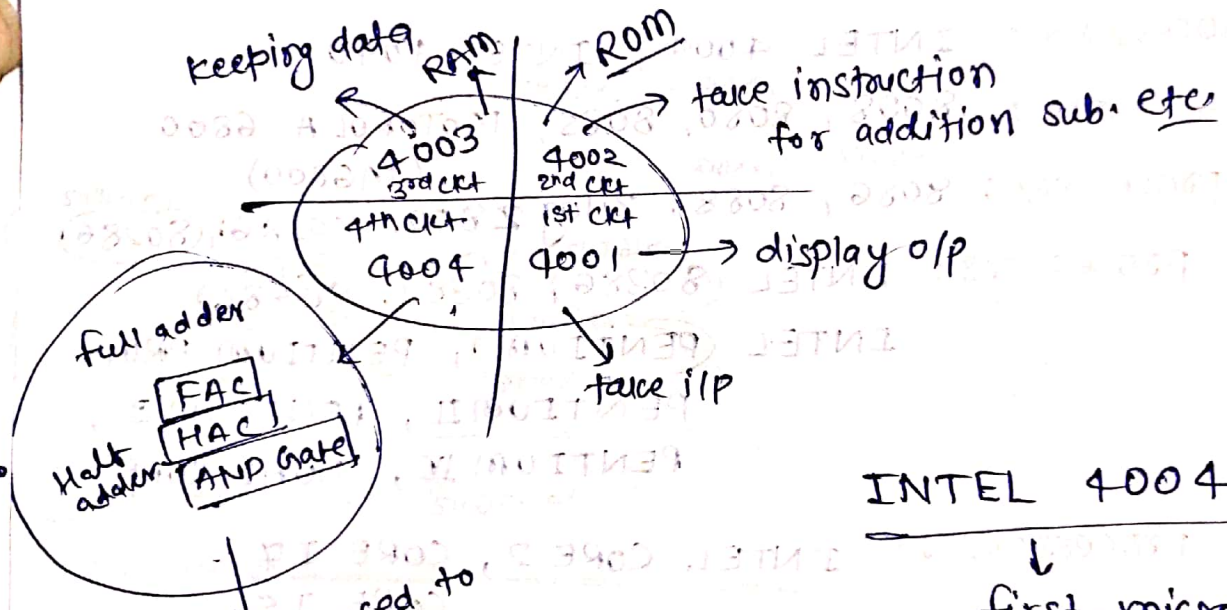
affordable, ~~storing~~ do memory

* The first Computer

↳ Fairchild semiconductors (founded in 1957) invented the first IC in 1959.

↳ In 1968, Robert Noyce, Gordan moore, Andrew Grove resigned from fairchild semiconductors.

Ted Hoff → designed the chip (IC) for Japanies based company for making computer.



INTEL 4004

↓

first microprocessor.

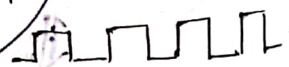
↳ 2300 transistors.

↳ Execute 60000 inst./sec.

↳ it was a 4-bit µp

↳ Clock speed - 740 KHz

↳ train of pulses.



data
↳

*

. INTEL - 8008 :- first 8-bit µp.
- 8080
- 8085

↳ Intel 8085 :- introduced in 1976
8-bit µp
clock speed 3MHz
. having 6500 transistors
. data bus is 8 and address bus is 16.

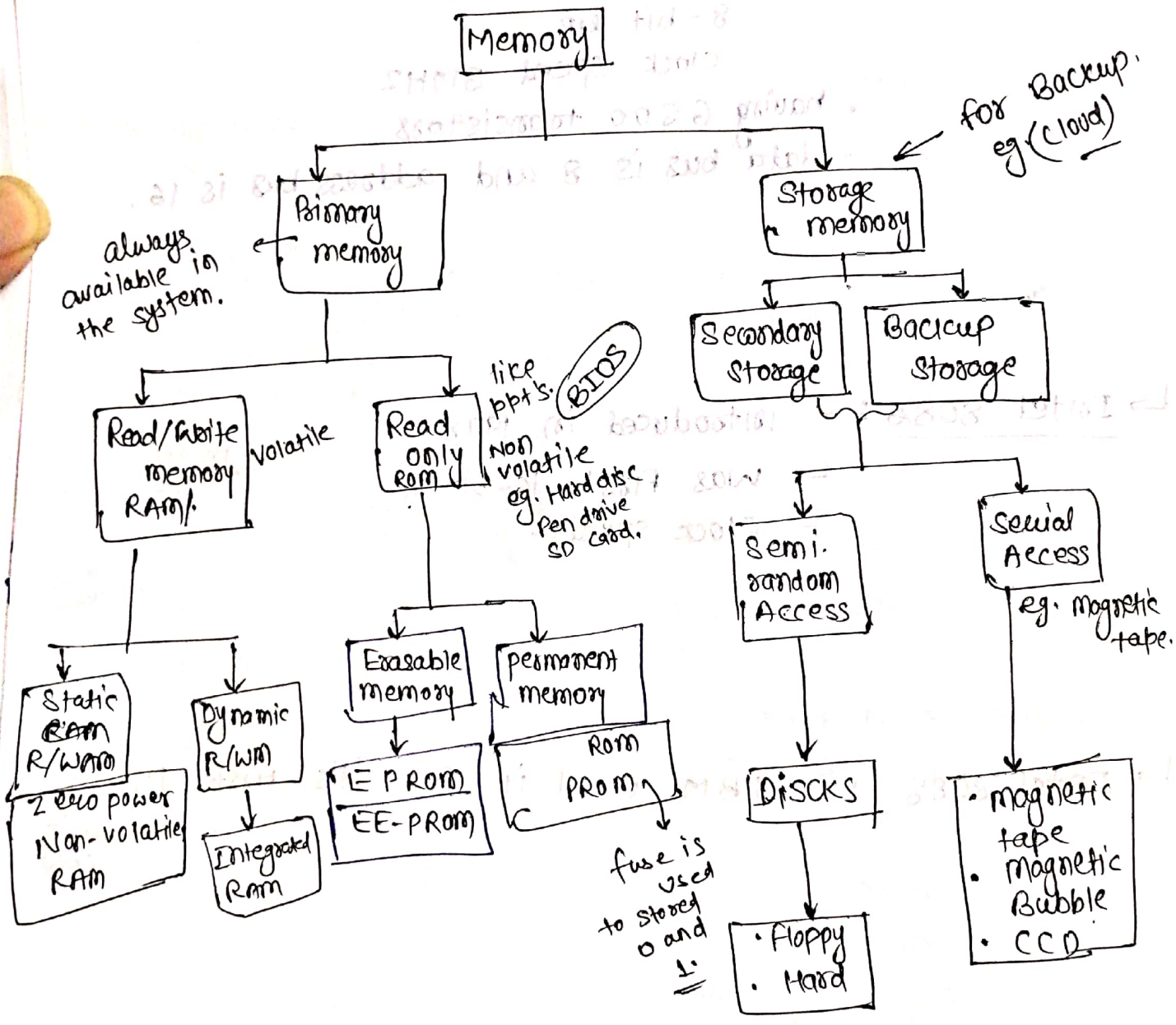
↳ Intel 8086 :- introduced in 1978
↳ was first 16-bit µp.
↳ Clock speed -

High in cost

→ Intel 8088 :- IBM used it for its first PC.

Low in cost

* Memory → Store digital information.
  - → instruction → applicn. software → O.S.
  - → data
  - → Configuration information. / control info.



- Hierarchy of Speed and size of memory.

- **Intel 4004:**
  - Introduced in 1971.
  - it was first μp by intel.
  - it was a 4-bit μp.
  - it's clock speed was 740 KHz.
  - it had 2300 transistors
  - it could execute around 60,000 instruction per second.

- **Intel 8008:**
  - it was First 8-bit μp.
  - introduced in 1972.
  - it has 14-bit address bus that could address 16 KB of memory.
  - it had 3500 transistors.
  - clock speed 0.5 MHz — 0.8 MHz.

- **Intel 8080:** 2nd 8-bit μp.
  - introduced in 1974.
  - it had 4500 to 6000 transistors.
  - it had clock speed of 2 to 3.125 MHz.

- **Intel 8085:**
  - Introduced in 1976
  - it was also a 8-bit μp.
  - it's clock speed was 3 MHz.
  - data bus - 8 bit & address bus - 16 bit
  - it had 6500 transistors.
  - could execute 7,69,230 instruction per second.
  - it could access 64 KB of memory.
  - it had 246 instructions.
  - over 100 million copies were sold.

7 Registers

7 byte.

* Intel 8086 :
- introduced in 1978.
- it was first 16-bit µp.
- Clock-speed is 4.77 MHz, 8 MHz and 10 MHz, depending on the version.
- data-bus - 16-bit, address bus 20-bit.
- it had 29,000 transistors.
- Could execute 2.5 million instruction per second.
- it could access 1MB of memory.
- it had 22,000 instructions.
- it had Multiply and Divide instructions.

* Intel 8088 :-
- Introduced in 1979.
- It was also a 16-bit µp.
- cheaper version of Intel's 8086.
- 16-bit processor with an 8-bit external bus.
- could execute 2.5 million instructions per second.
- IBM used it for it's first PC.

* Intel 80186 & 80188 :-
- Introduced in 1982.
- They were 16-bit µps.
- Clock speed was 6 MHz.
- 80188 was a cheaper version of 80186. With an 8-bit external data bus.
- They have additional component like — Interrupt Controller
  — Clock generator
  — Local bus controller
  — Counters

* Intel 80286 :-
- Introduced in 1982.
- It was also a 16-bit µp.
- it uses nearly ~~125~~ 134,000 transistors.
- clock-speed 5 to 25 MHz.

* Intel 80386 :
- introduced in 1986.
- It was first 32-bit µp.
- data bus - 32 bit and address bus - 32 bit
- it could address 4 GB of memory.
- It had 2,75,000 transistors.
- clock-speed varies from 16 MHz to 33 MHz.

Different versions:
- 80386 DX
- 80386 SX
- 80386 SL

↳ Intel 80386 becomes the best selling microprocessor in history.

* Intel 80486 :
- Introduced in 1989.
- It was 32-bit µp.
- It had 1.2 million transistors.
- Its clock speed varied from 16 MHz to 100 MHz.

↳ it had 5 different versions
- 80486 DX
- 80486 SX
- 80486 DX2
- 80486 SL
- 80486 DX4

- 8 KB of cache memory was introduced.

* Intel ~~per~~ PENTIUM :
- Introduced in 1993.
- 32-bit µp.
- originally named 80586.
- Clock-speed 66 MHz.
- data bus - 32 bit, address bus - 32 bit
- could address 4GB of memory.
- could execute 110 million instructions per second.
- cache memory —
  8 KB for instruction.
  8 KB For data.

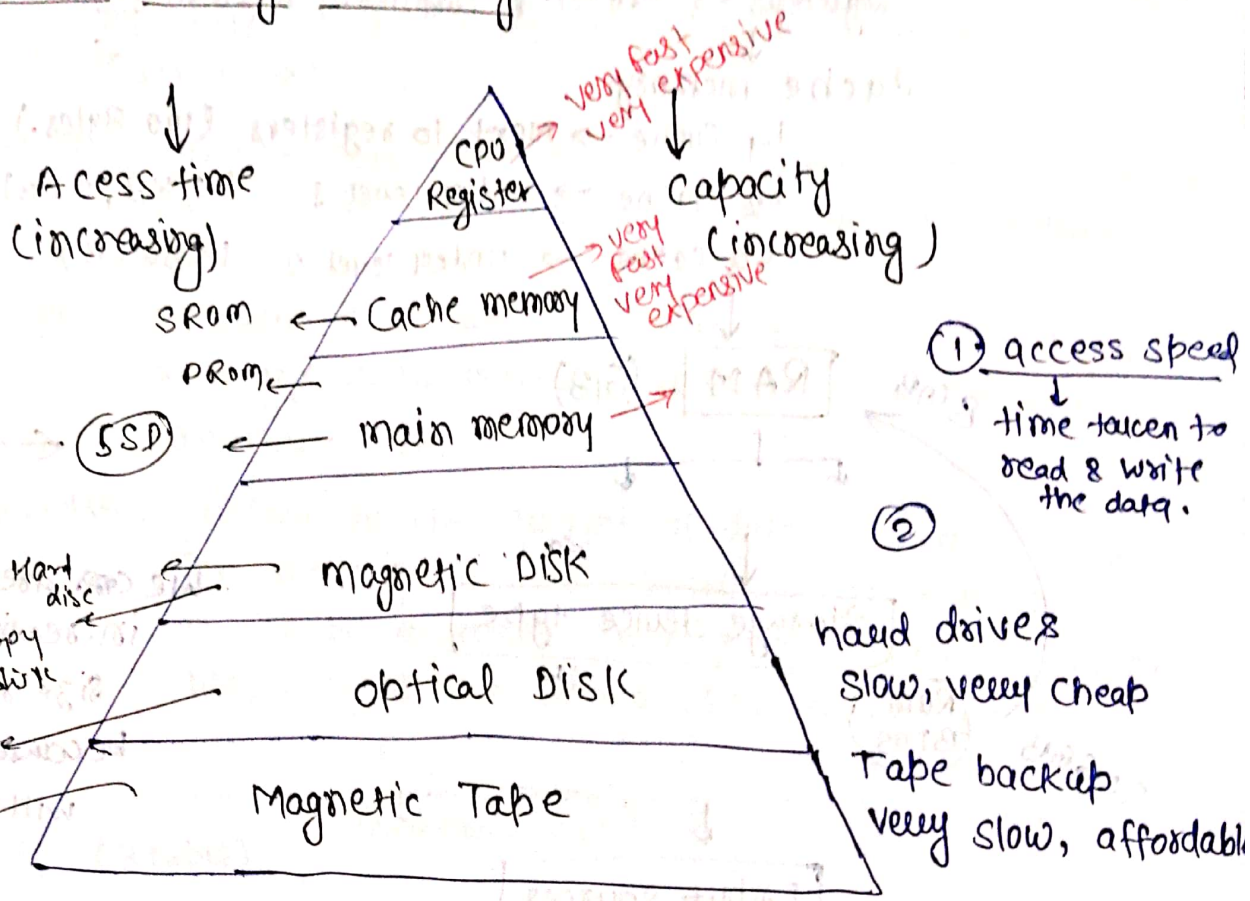
* Intel PENTIUM PRO :-
- Introduced in 1995.
- Clock-speed 150-200 MHz.
- Transistors used 5.5 million.
- 32 bit µp.

* Intel PENTIUM II :-
- Introduced in 1997
- clock speed 233 to 450 MHz
- 7.5 million transistors used.

* Intel PENTIUM III :-
- Introduced in 1999.
- clock speed 450 MHz to 1.4 GHz.
- 9.5 million transistors used.

* Intel PENTIUM IV :-
- introduced in 2000
- clock speed 1.3 to 3.8 GHz.
- transistors used 42 million.

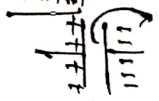
* Computer Memory Hierarchy :-



RAM · Rom → Because it is only Read by Laptop / Desktop.
↳ for program we need another system from taking out from our own system.

RAM → fastest → used for data
Rom → slowest → used for program.

↳ in dynamic cell all the bit is stored using capacitor. (multiple cap.)



size ↓ cost ↓

Refresh cycles. (Power req ↑) time req ↑

data is lost due to self discharge. So we have to periodically give charge to it. that's why it is called dynamics.

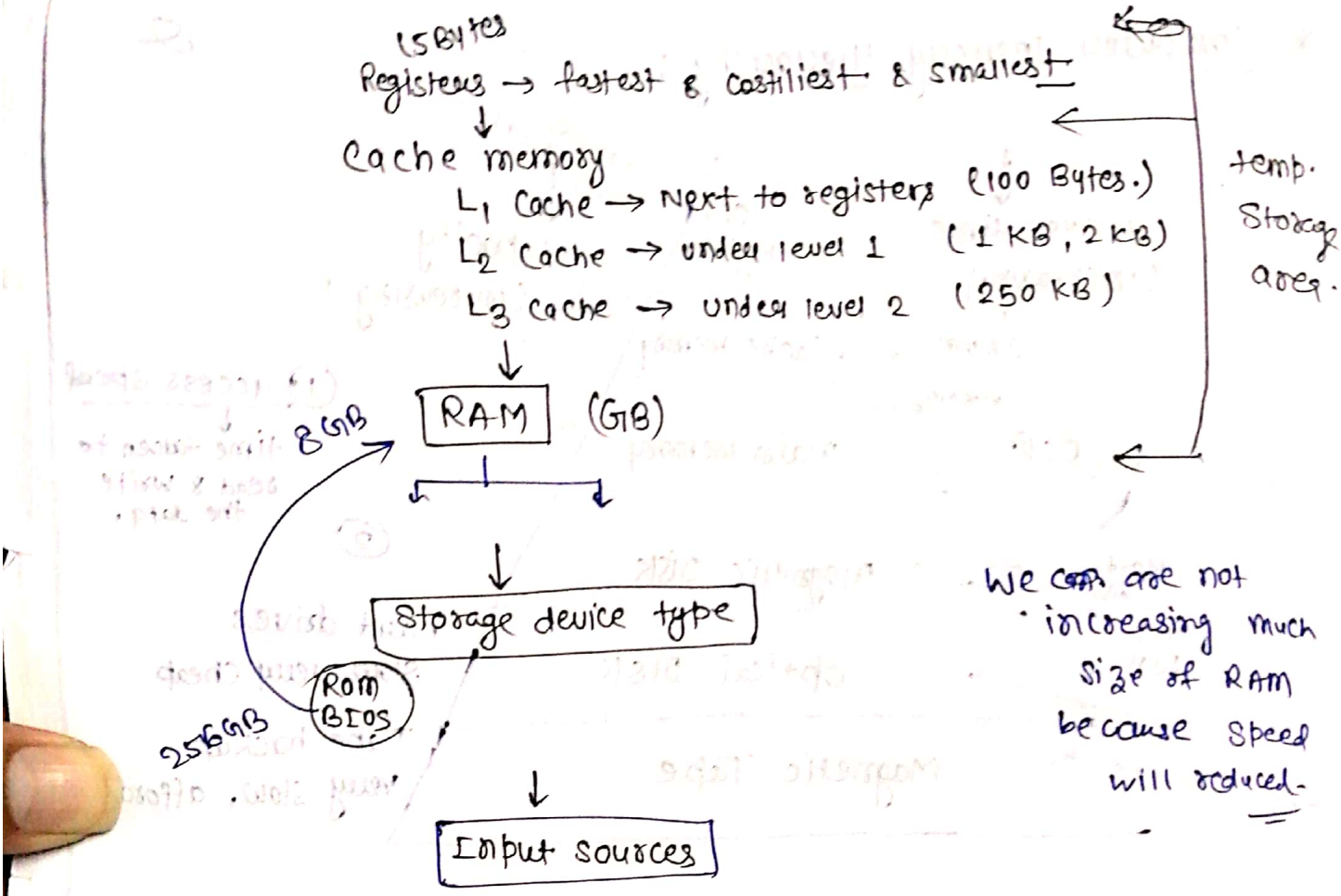
↳ Static RAM ⇒ Transistors are used (2-P type, & n-type)
6 Transistors are used to store one bit of data. (6T cell.)

Linear
cross connection ↓ Bi-stable multivibrator.

so here data remains stored always else we change it.

No refresh required
Will take less time

Cost ↑

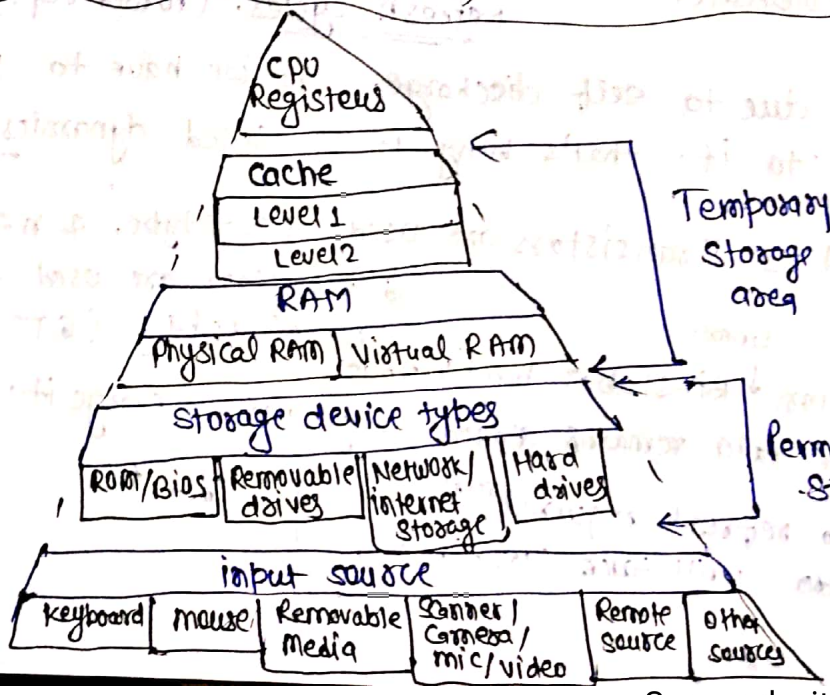


We can are not increasing much size of RAM because speed will reduced.

→ Bus types? & their need.

① Access speed :- How fast the data can be transfered

② post d.speed :- Maxm data transfer rate b/w your device & the local Network to which it is connected.
(How quickly the data can be send or received within the Network.)



- **Data rate** :- it expressed in heatz (Hz) and represents the number of bits which can be transferred within the chip each second.
- **Access time**:- The elapsed time from the moment the CPU calls for data to the moment it receives the request back.
- **cycle time**:- Represent the time necessary to complete a single read ~~and~~ (or) write process and to reset the chip for another cycle.
- **Bandwidth** :- Refers to the amount of data a memory chip can move (or) process within a set time frame. it can be expressed in bits per second (bps (or) b/s), bytes per second (BPS (or) B/s) or hertz. cyc/sec.

Collection of wires.

* **Bus types** (3 types)

① Address bus:- Transport memory addresses from the processor to additional parts, including primary storage and i/p/o/p devices. (unidirectional)

② Data bus: Transfer the data from the processor to other parts. (Bidirectional)

③ Control Bus:- transporting control signals along with the clk pulses across the control bus to other peripherals from the processor. (unidirectional)

↳ They are needed because of the fast internal connection between the components and is also used to transfer data and control signal b/w the processor and other parts.

- Z_2 (Computer) → German electromechanical Comp...

* Dynamic RAM (DRAM) memory technology:-

Type

FPM
EDO
SDRAM
DDR SDRAM
DDR2 SDRAM
DDR3 SDRAM
DDR4 SDRAM

Graphics memory. → Part of DRAM

* FPM

↳ Fast page mode RAM → Dynamic RAM
allows faster access to data in the same row or page.

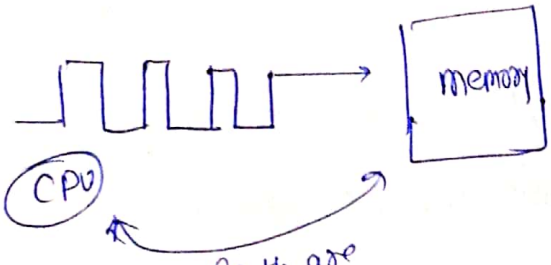
↳ work by

* EDO DRAM (Extended data out DRAM)

Burst EDO DRAM (Can process four memory address
↓
a sudden flow

* SD RAM

- synchronous dynamic RAM
- synchronizes itself with the CPU's bus and is capable of running at 133 MHz.
- 3 times faster than conventional FPM RAM
- twice as fast EDO DRAM and BEDO DRAM.



Both are using same CLK pulse in synch. Dynamic RAM.

∴ it will become faster.

Why CLK is needed

in order to continuously produce transition.

↓

B/c on transition only the data will latches (flip flop).

* RD RAM

* DDR SDRAM (double data Rate)

↳ in both +ve & -ve edge the reading & writing is happen.

addition register required.

PCI slots → peripheral Component interconnect full form?

SATA slots → Serial Advanced technology Attachment

SMPS → Why needed.

↳ B/C multiple power supply is required in our desktop.

We cannot add different supply.

So SMPS used.

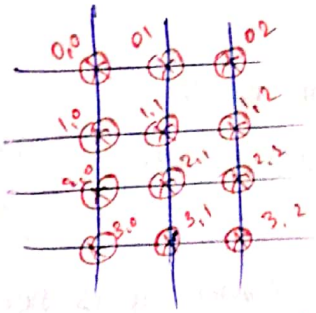
Problem with SMPS?

Z_3 (Computer) → German electromechanical Computer.

another processor → Graphics processor. (GPU). NVIDIA

↑

AI/ML



about 60 crores data

Bright

R, G, B

---

* BUS :-

1.

ALE
$\overline{RD}$
$\overline{WR}$
$\overline{CE}$ / $\overline{CS}$
} example of control signals.

— (bar)

↓

active low signal

(to write you make that signal to 0.)

↳ Why all these signals are active low in nature ??
- To reduce the noise interference error
- Bic of n-channel mosfet

↳ Having more no. of wires (pins) in the address bus ~~if no.~~ then more no. of memory can be attached i.e., more no. of data can be addressed.

n bit address bus

↳ $(2)^n$ memory location (or)

2^n bytes can be stored.

* Data Bus :- Size ↑ then system speed ↑

* Control Bus :- (Processor by default have control bus.)
  ↳ No need of large size of control bus.

(HOLD) → DMA controller uses HOLD s/g to hold the

HOLDA

* Why size of bit is directly related to speed.
  ↓
  more data can be transfered at a time.

Byte addressed memory → only 8 bits is stored
(or) word addressed memory → 16 bits stored in one memory.

(1 - word) ⇒ size ?

* Why most of them are word address. ?
* one is linear addressing
  What is the other type of addressing ?
  ↳ segmented addressing :-

• Z- (computer) → Germanium electromechanical computer

* Type of processors
① Microprocessor (CPU)
② GPU (Graphical processing unit)
③ Microcontrollers.
④ Digital signal processors.
⑤ Fuield programable gate array (FPGA)

• Micro-computer :

set of pneumonics ↓ instruction set

Tensor flow – ?
↓
it is one of the library available in Python. (used in A.I., M.L)

* Types of PUS

① MCUs/MPU (Microprocessors / Microcontrollers)
② ~~ASIC~~ (Application specific instruction set processor) – specific application like graphics.
③ ~~ASIP~~ :
③ ASIC ⇒ (Application specific integrated CKt)
④ FPGA (field programable Gate array)
⑤ DSP processors
⑥ GPU – Graphics processing unit / VPU – "Visual processing unit"
⑦ Tensor processing unit (TPU)
⑧ Neural processing unit (NPU)
⑨ physics processing unit (PPU)
⑩ Image signal processor (ISP)
⑪ synergistic processing element (or unit (SPE (or) SPU)

① • sensor
② • Analog to digital converter
③ i/p & o/p should be getting connected to processor using a port.
④ cost should be less.
⑤ speed of the processor. ↑
⑥ Memory.
⑦ word size should ~~match with~~ be considered towards speed of system.
⑧ power consumption. should be minimum. (for portable, battery usable device.)

How to start a processor ?

various types of IC available.

* Choosing a DSP:

→ Many DSP's ↔ Many manufacturers – Good in their own area
TI, Motorola, Freescale, Philips, Atmel, Zilog, microchip.

↳ each of them have a unique instruction set & Registers in them.
↳ 8 bit, 16 bit, 32 bit DSPs.
↳ Not compatible with each other.
↳ program written (Assymbly/machine) for one will not run on the others.
↳

meeting the computational needs —

- speed
- cost
- memory
- pheripherals available
- power requirement

→ Efficient soln for the problem
→ Word size needed
→ Amount of RAM & Rom needed – Available
→ Number of I/O pins.
→ Other peripherals available/support
→ Ease to upgrade
→ Speed
→ Package
→ power consumption
→ Less cost

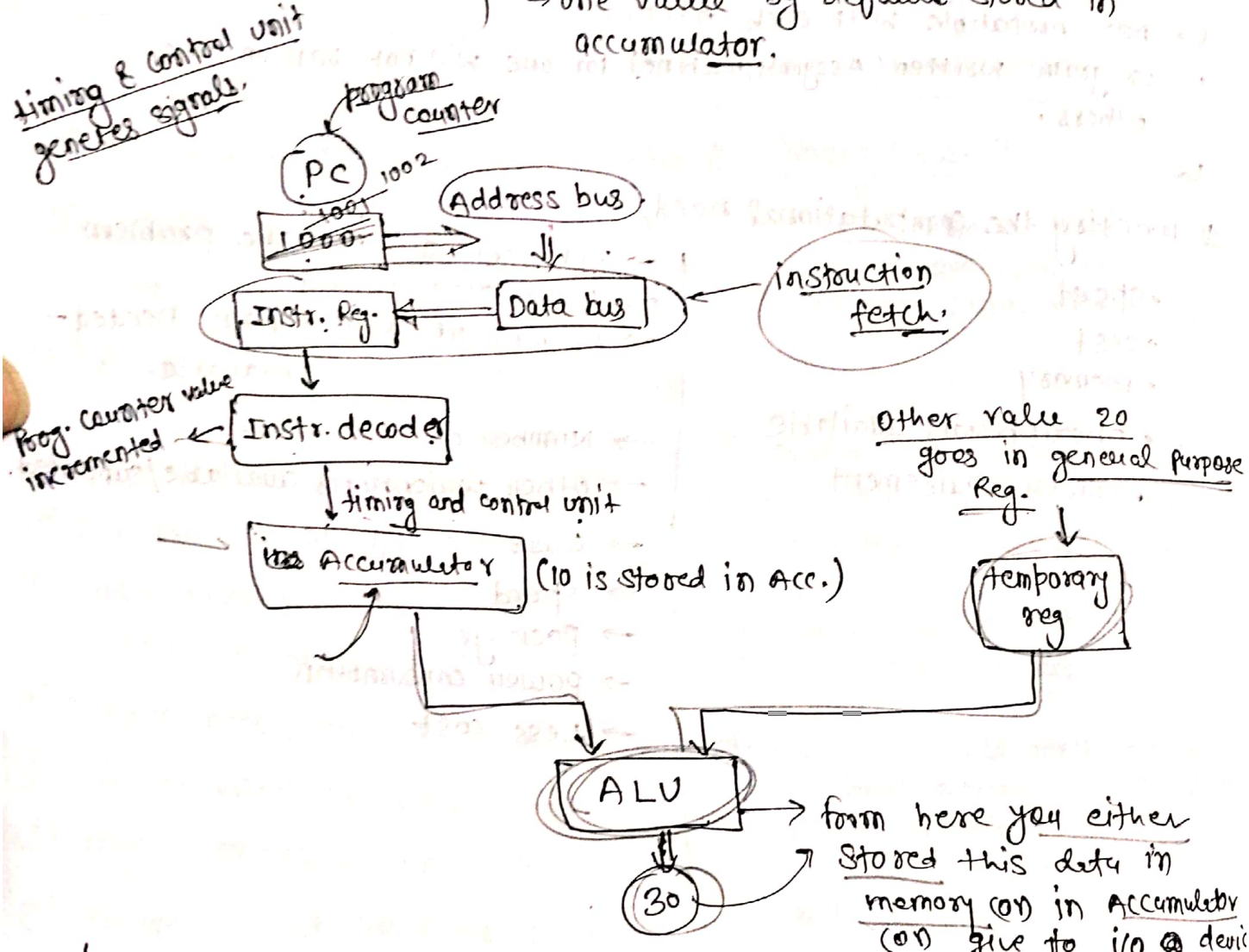
eg.

```
int a, b, c
a = 10;
b = 20;
c = a + b
```

eg.

I_1	MOV A, 0AH	1000
I_2	MOV B, 14H	1001
I_3	ADD B.	1002

↳ one value by default stored in accumulator.



↳ Program Counter loaded with initial address
↳ it is placed on add. Bus
↳ Address reaches to program memory location
↳ CPU generates necessary signal to read data.
↳ data is placed on data bus. Reached to control unit.
↳ opcode goes to instr. register and then to instr. decoder.

↳ Necessary operands are fetched.
↳ operands reaches ALU.
↳ operands executed.
↳ Results are stored.

Dual inline package (DIP) → goes upto 40 pins.

SMD → Surface mound device (for more bit processors where millions of transistors use)

BGA (600 – 800) pins.

PLCC

LCC

* What is microprocessor :- it is a digital ckt. (Transistor connected to form digital ckts)

↳ it is a digital ckts with a set of general purpose Registers and their control logic —
Tens of thousands of semiconductor switches (diodes / transistors / mos devices...) are properly interconnected. (GPU)

↳ Data can be moved into or out of the registers and can be manupulated using instructions.
A sequence of instruction is a program.

↳ the program and data are stored in a memory external to the µp.

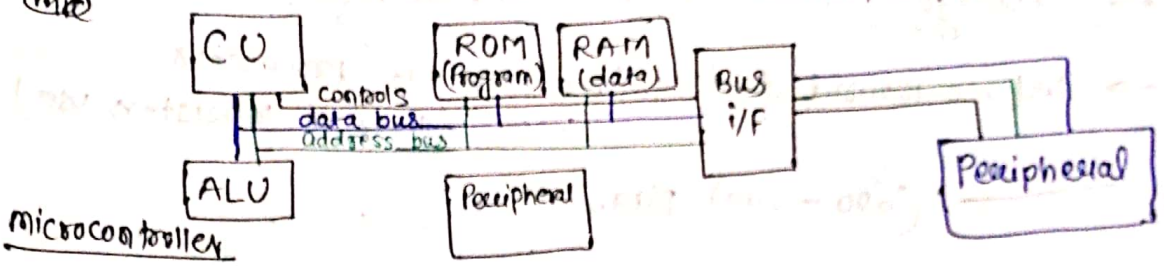
↳ it interacts with outside world through I/O port.

↳ monitors input : process the data ; changes the output if necessary !!

↳ min^m 30 pins (wires) should be connected to ROM & also RAM.

* (Embaded Processor/sys.)
Microcontroller :- (microcomputer on a single chip.)

↳ first microcontroller — 8051 → (made by intel)
↳ self sufficient up.
• Contains a up and more. — RAM included, ROM included, some peripheral may be included.



Microcontroller

Adv.
↳ Few component
↳ lower size
↳ lower cost

Disadv.
↳ fixed sets of peripherals.
↳ many types to keep stock (costs more)
↳ Hardware solutions can be reverse engineered.

Microcontrollers generally have —
• No external address or data bus.
• No provision to access external memory.
• Relatively small amount of memory.
• Relatively small instruction set.
• No ability to do intensive math operations

instruction set. → individual Mnumonics.

• Classification
↳ Based on word size — 8 bit, 16 bit, 32 bit etc.
↳ Based on architecture :- Von Newmann, ~~Hard~~ Haarward etc.
↳ Based on memory :- OTP, EPROM, FLASH etc.

* inside a processor?
• ALU
• Registers : (General purpose, Special purpose)
↓
(Same like scratch pad)
↓
(Programmers / users disposal)

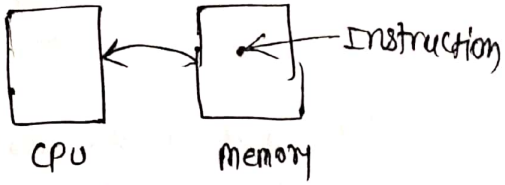
↳ used for configuring & controlling.
(it is designed & used by manufactured.)

• program counter :- pointer register (Can only carry addresses → Pointer Reg.)

↳ Contains the address of next instruction.

to be fetched.

transferring instr. from memory to CPU → Fetching



• Instruction Registor → Contains next instruction to be executed.

• Instruction decoding unit →

opcode → Comb^{n}. of zero and 1's.

↓

operating codes

decodes opcodes

decoder required to decode mnemonics to opcodes.

the 0's & 1's goes from instruction Registor to instruction decoder.

↓

80% transistors are used by decoder.

if we are using BCD addition in order to do that AC is needed.

• Status flags (...

↓

Represents the status of results of the last executed instructions.

Carry flag (Bit) set 1 → when data overflow is there. (exceeding maxm no. of value contains in Register.)

Parity flags :- (even & no. of 1's → P→1) ← in the result of last instruction.

Von-Neumann :- consists single, shared memory for program & data.

↳ Single bus for memory access.

↳ Architecture unit

↳ program control unit

↳ it operates fetching & execution cycles seriously.

Stores data and instruction in main memory → and moving them b/w memory & processor.

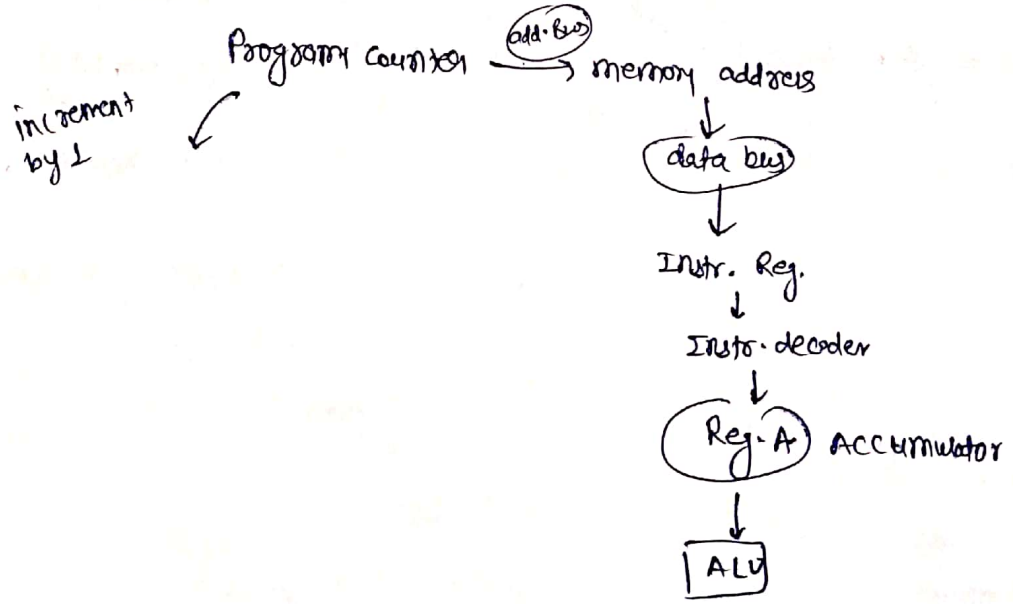
↳ Von-Neumann arch. is an ancient type of computer architecture that follows the concept of stored-program computer.

↳ Harvard arch. follows relay based model. (modern type)
It contains seperate bus for instruction & data such that CPU can access instr. & read/write data at the same time.

↳ When we do signed calculation. (OV) flag is used. "over flow"
in 8 bit only 7 bit's are used.
one bit is used for representing sign.

(1) 0 0 1 1 0 1 1 1
↑ ↑ x

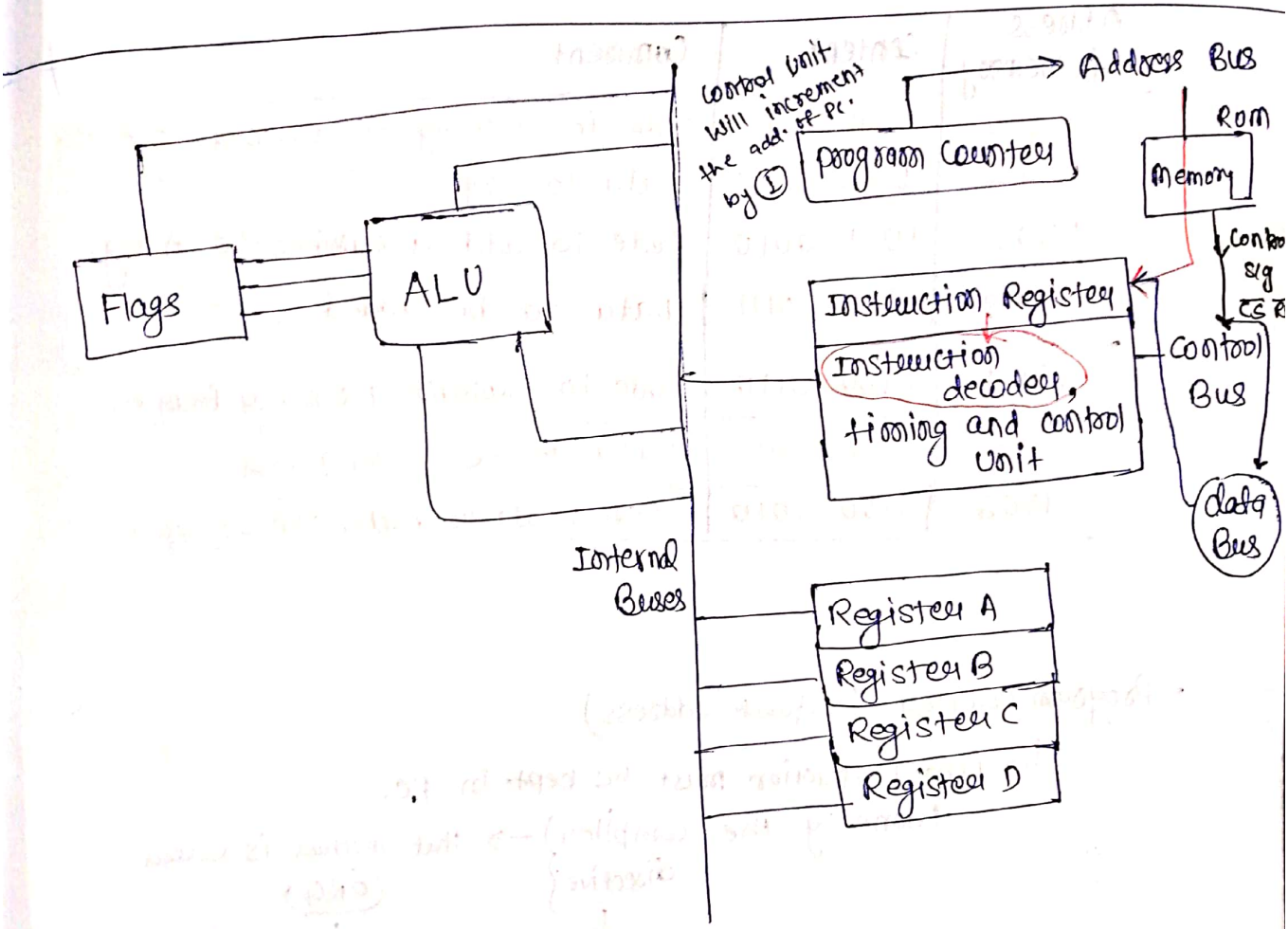
When 7th to 8 Carry is not there but 8th to output Carry is there.
(OR)



Importance of default value of program counter :-

(directive to compiler)
include <stdio.h>
↳ Compiler instruction.

ORG 50
→ ADD A, B



$\overline{CS} = 0$
$\overline{RD} = 0$

* Turning on a processor

Action	Code	Data
Load the number 34 in Register A	BAH	34H
Add the number 57 to the content of register A and keep the result in register A itself.	B2H	857H
Subtract 21 from the content of register A.	46H	21H

Address of memory	Content	Comment
1500	1000 1010	Code for moving the number to A reg.
1501	0011 0100	Data to load.
1502	1011 0010	Code to add a number to A reg.
1503	0101 0111	Data to be added
1504	0100 0110	Code to subtract a number from A.
1505	0010 0001	Data to be subtracted
1506	1100 1010	CA : Code to halll the processor

- Program counter (default address)
  - ↳ first instruction must be kept in P.C.
    - (done by the compiler) directive → that method is called ORG
      - ↳ Compiler directive.
    - ↓
    - #include
- Compiler is a software