

- Computer Architecture
- Von-Neumann Architecture
- Architecture
- Idea of stored program.

Formation/
constitution
of microprocessor

Key terms:

- **Definition**
- **Registers**
 - Special purpose
 - General purpose

- **FDEC**
- System Buses (IAS)
- Immediate Access Store

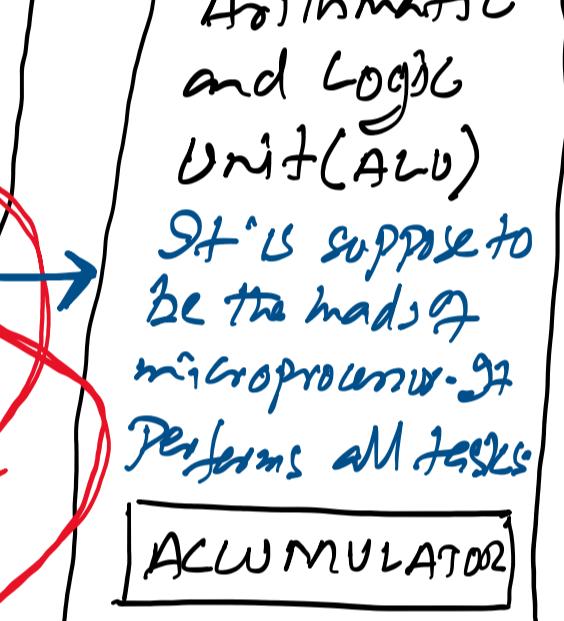
This was an idea for the general purpose computer that when given any program can execute it effortlessly.

You don't have to change a computer to make it work differently.

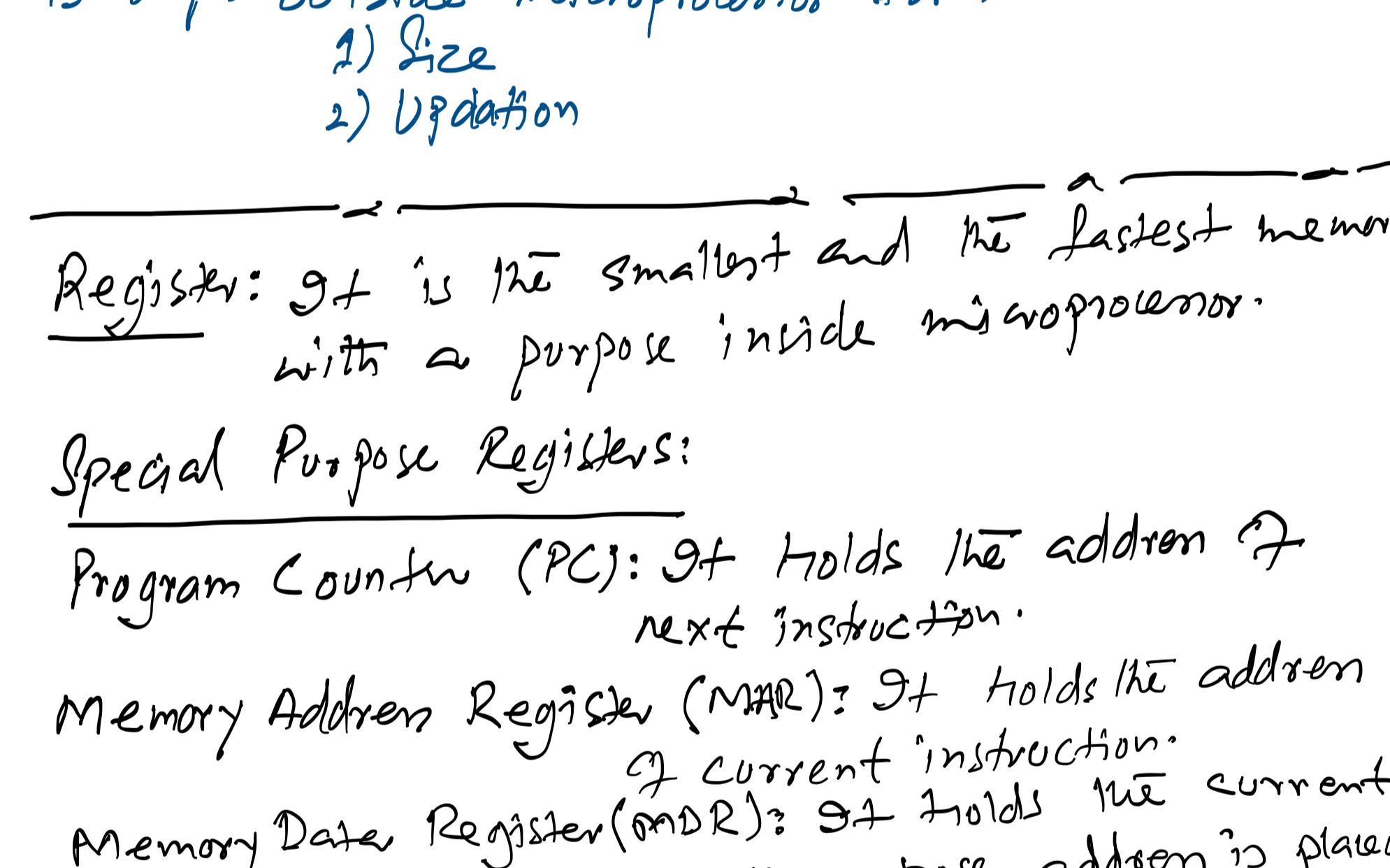
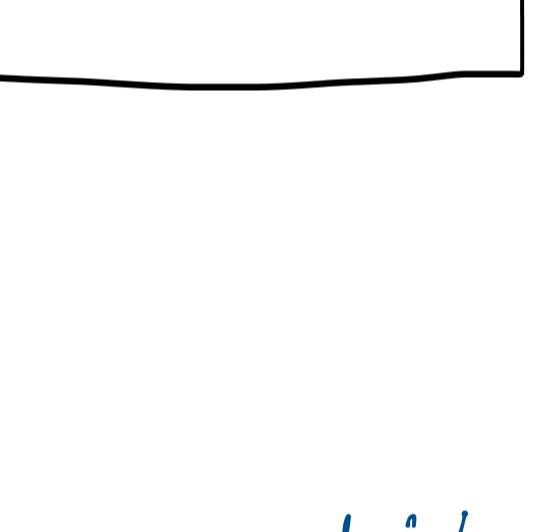
Definition: What/State/Describe/Explain

- It is a single processor made up of Control unit (CU), Arithmetic Logic unit (ALU) and Memory unit (MU).
- Both programs (instructions) and

* Data are indistinguishable in binary form and will be kept in the same main memory.



- Use of input, output, and storage.
- It's a sequential/serial machine



- Main memory is part of the microprocessor and it is kept outside microprocessor for two reasons:

- 1) Size
- 2) Updation

Registers: It is the smallest and the fastest memory with a purpose inside microprocessor.

Special Purpose Registers:

Program Counter (PC): It holds the address of next instruction.

Memory Address Register (MAR): It holds the address of current instruction.

Memory Data Register (MDR): It holds the current instruction whose address is placed in MAR.

Current Instruction Register (CIR): It decodes & executes current instruction.

General Purpose Registers:

Accumulator: All the data produced during the execution of current program is kept in accumulator.

FETCH DECODE EXECUTE CYCLE (FDEC):

- Address of the next instruction from PC goes to MAR.

- Instruction in memory whose address is in MAR arrives in MDR.

- From MDR, current instruction goes to CIR.

- PC increments itself by 1.

- CIR decodes and executes current instruction.

Fetch

Fetch. $\xrightarrow{\text{PC} \rightarrow \text{MAR}} \xrightarrow{\text{MAR} \rightarrow \text{MDR}}$ $\xrightarrow{\text{MDR} \rightarrow \text{CIR}}$

$\xrightarrow{\text{30}} \xrightarrow{\text{31 Add. Bus}} \xrightarrow{\text{LDD}}$

$\xrightarrow{\text{LDD}}$

\x