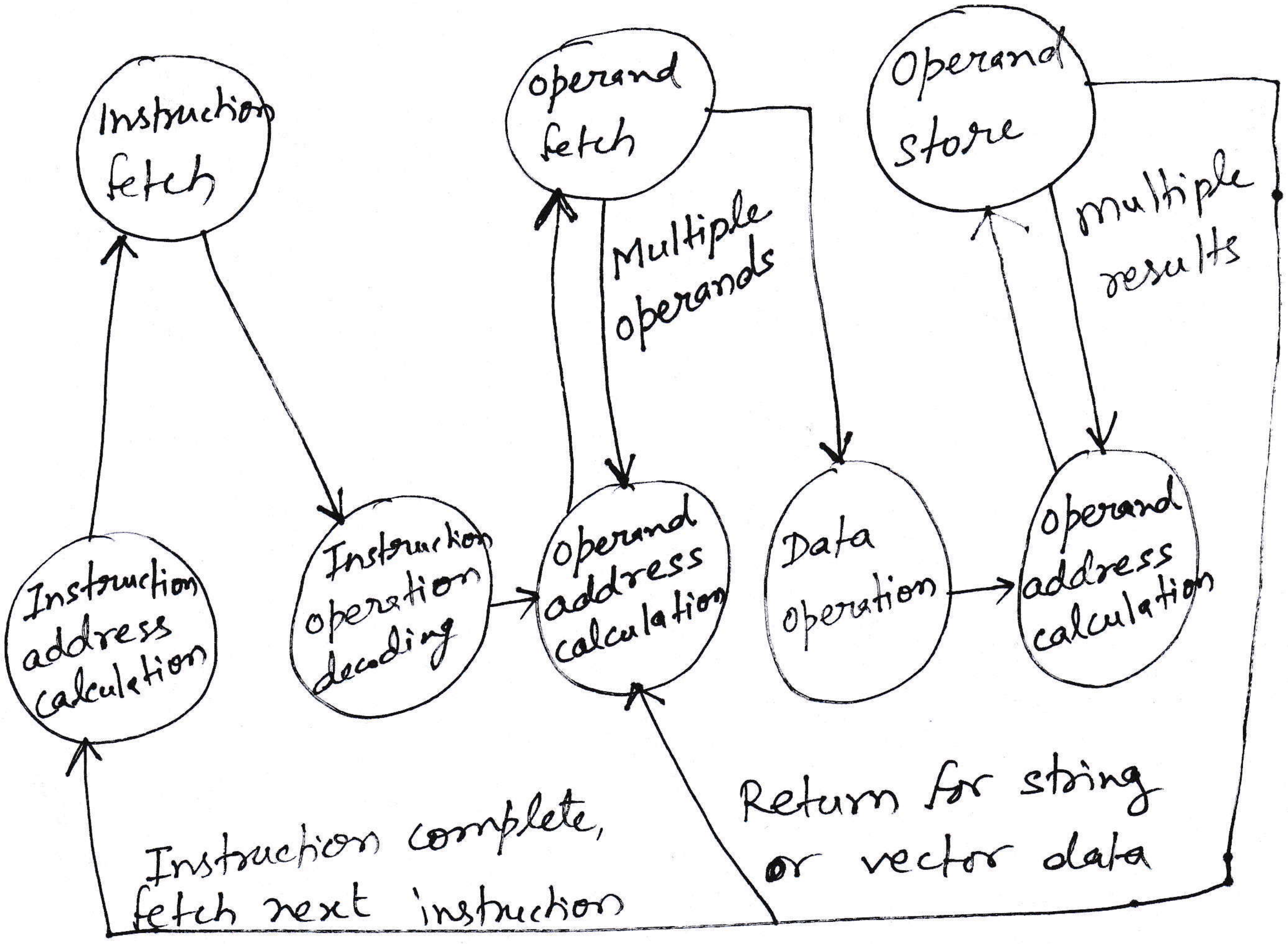
Control Unit

Instruction Cycle:

An instruction cycle is the complete process of fetching, decoding and executing the instruction.


Instruction fetch
operand fetch
Operand store
Multiple operands
multiple results
Instruction address calculation
Instruction operation decoding
Operand address calculation
Data operation
Operand address calculation
Return for string or vector data
Instruction complete, fetch next instruction

Control Unit

Steps (Instruction Cycle)

1) PC gives the address to fetch an instruction from the memory.

2.) Once fetched, the instruction opcode is decoded.

3.) This identifies, if there are any operands to be fetched from memory.

4.) The operand address is calculated.

5) Operands are fetched from the memory.

6.) Now the data operation is performed on the operands, and a result is generated

7.) If the result has to be stored in register, the instruction ends here.

8.) If the destination is memory, then first the destination address has to be calculated.

9.) The result is then stored in the memory.

10.) Now the current instruction has been executed.

11.) Side by side PC is incremented to calculate address of the next instruction.

12.) The above instruction cycle then repeats for further instructions.

Micro - operations & Control Signals

→ A program is a set of instructions

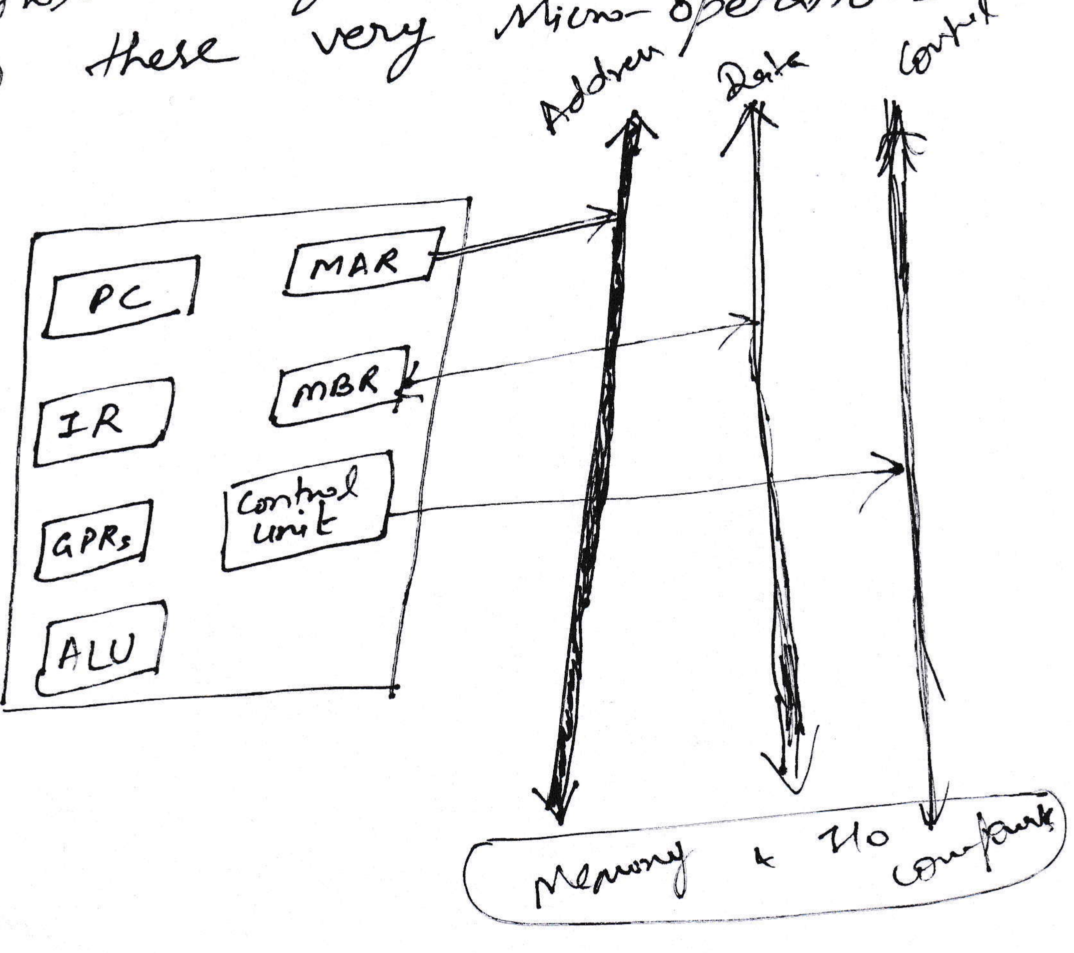
→ An instruction, requires a set of small operations called Micro- operations.

→ A <u>micro-operations</u> is a finite activity performed by the processor in one clock cycle. One clock-cycle is also called as one T-state (Transition state).

→ One Micro-Operation requires One T-state.

→ Several Independent Micro-operations in the same T-state.

→ Control unit generates control signals to perform these very Micro-operations



Micro-operations for Instruction Fetching

one-way	other-way
T1: MAR ← PC	T1: MAR ← PC
T2: MDR ← Memory (Instr)	T2: MDR ← Memory (Instr) PC ← PC+1
T3: IR ← MDR PC ← PC+1	T3: IR ← MDR

OR

- Examples → Immediate Addressing mode

1) MOV R1, 25H; { R1 register gets the immediate value 25H }

T1: MAR ← PC
T2: MDR ← memory (Instr)
T3: IR ← MDR
PC ← PC+1
T4: R1 ← 25H (IR) // R1 gets value 25H from IR.

2) Register Addressing Mode
MOV R1, R2; // R1 gets the data from R2

T1: MAR ← PC
T2: MDR ← memory (Instr)
T3: IR ← MDR
PC ← PC+1

T4: R1 ← R2

(3) Direct Addressing mode

MOV R1, [2000H]; R1 gets data from memory location 2000H

T1 : MAR ← PC
T2 : MDR ← memory (Instr)
T3 : IR ← MDR
PC ← PC+1

T4 : MAR ← IR (2000H)
T5 : MDR ← memory ([2000H])
T6 : R1 ← MDR

(4) Indirect Addressing mode :

MOV R1, [R2]; {R1 gets data from memory location pointed by R2

T1: MAR ← PC
T2 : MDR ← memory (Instr)
T3 : IR ← MDR
PC ← PC+1

T4 : MAR ← R2
T5: MDR ← memory ([R2])
T6 : R1 ← MDR