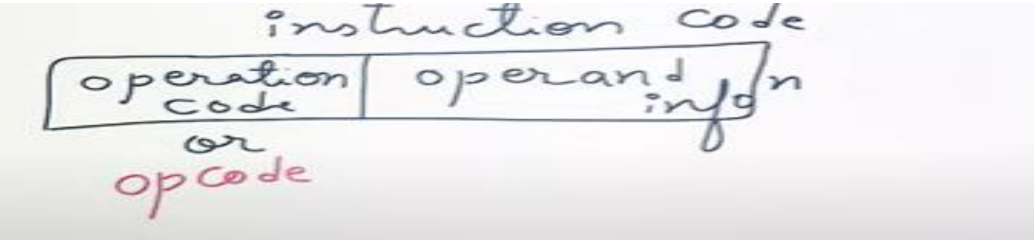
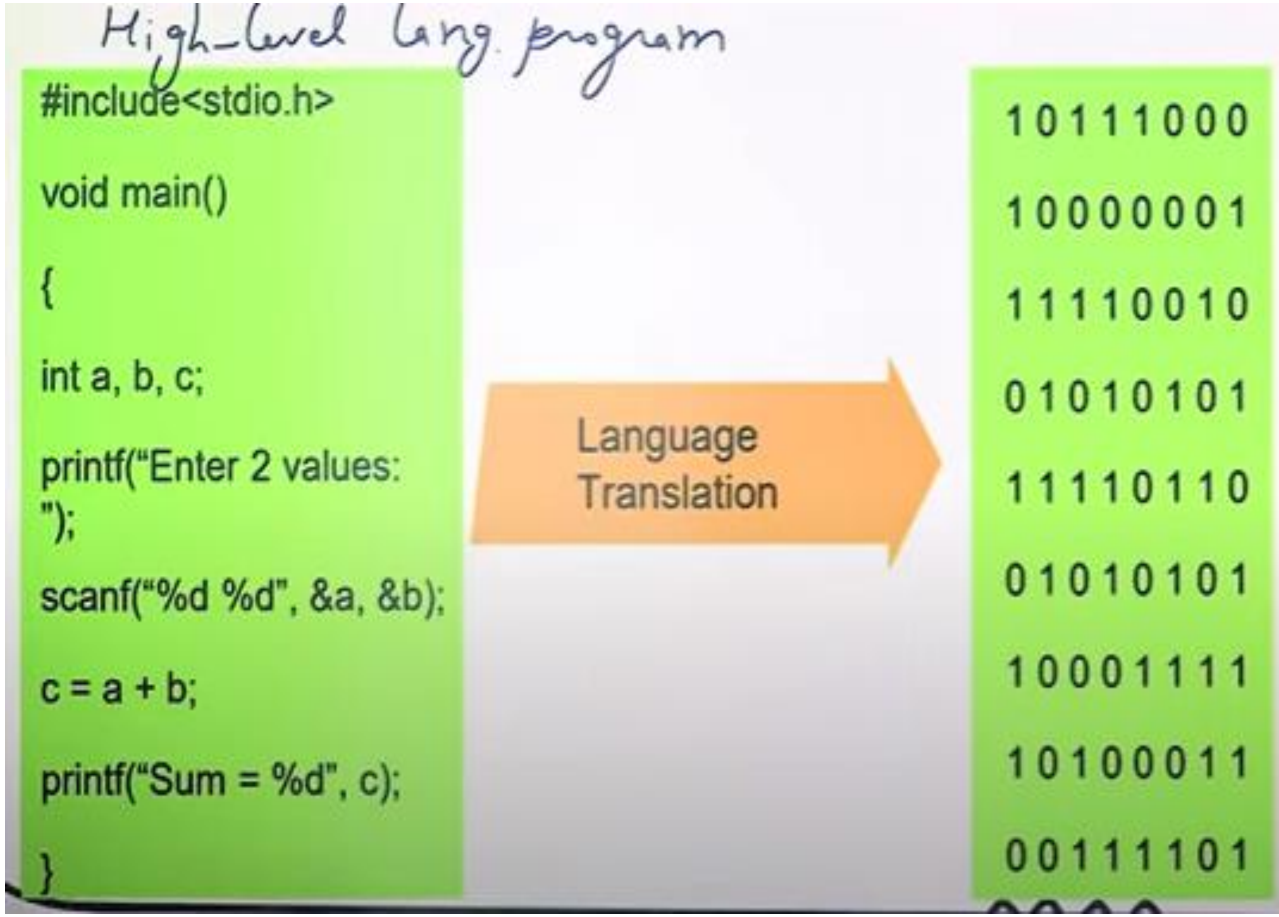
Unit 3

Instruction generation and Execution

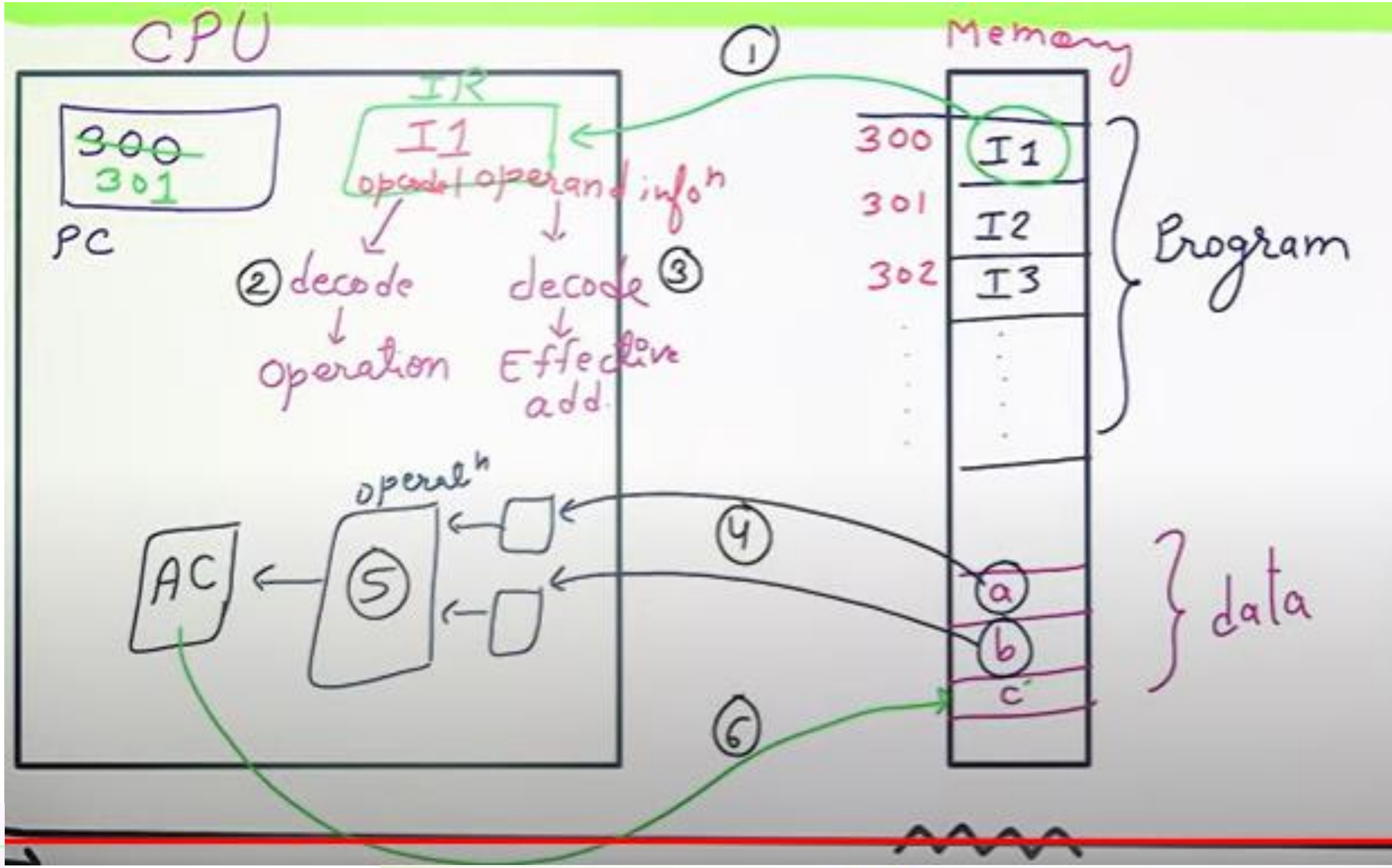
- Instruction are binary sequence that tells computer to do some operations.
- Instructions are generated via complier.
- Complier generates instruction that is supported by computer (CPU-based Architecture).
- For e.g.- register based architecture- instruction will collect data from registers.





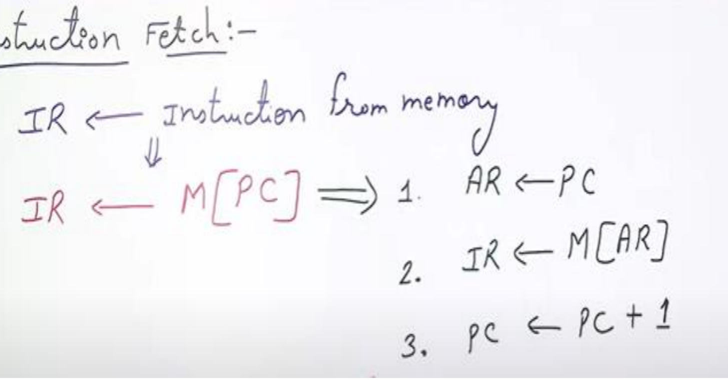
Instruction Execution

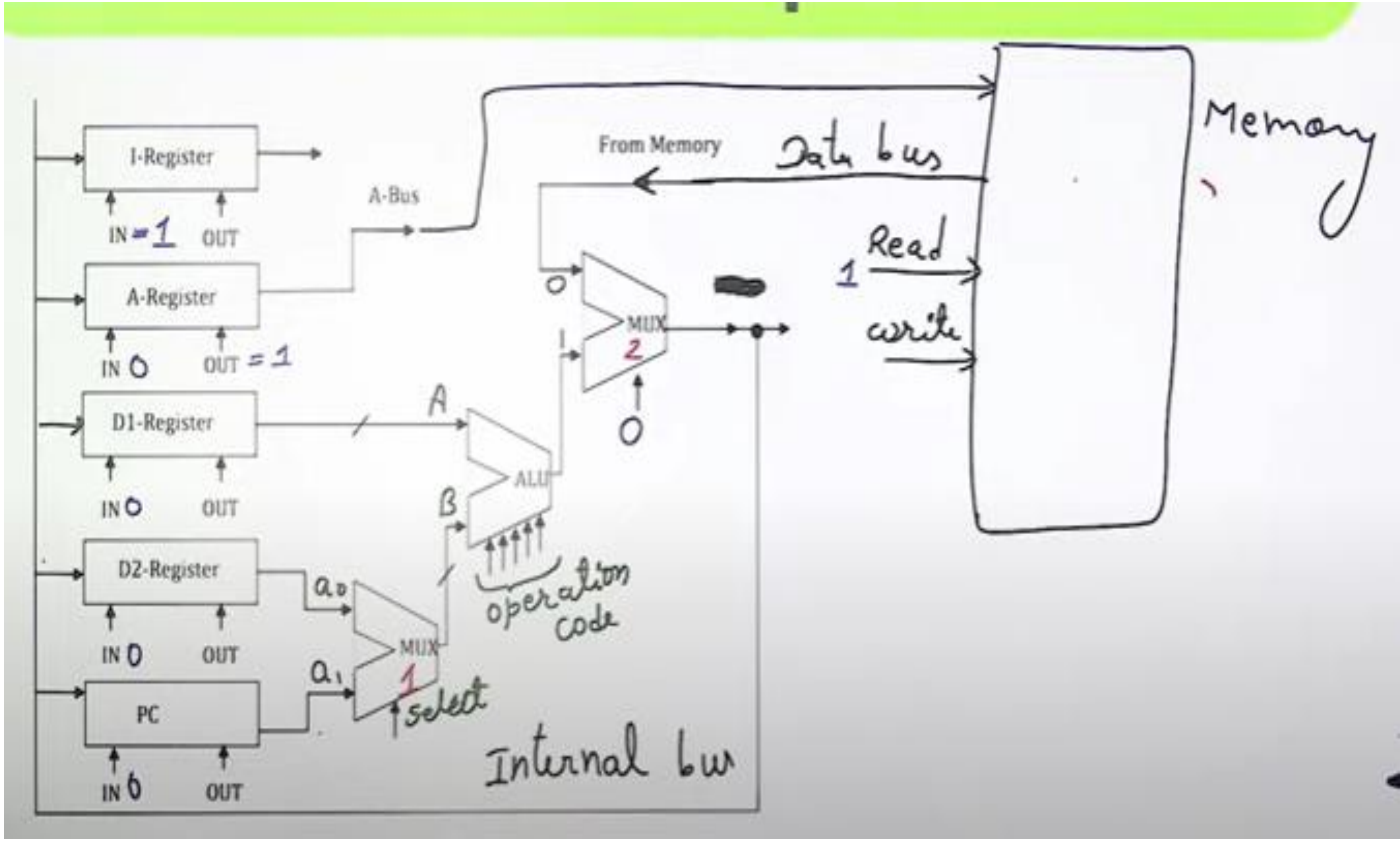
- Performed inside CPU. (Program instructions and data –main memory).
- Operand actual location- Effective address.
- Execution at ALU
- Result first stored at AC.
- Then it is write back to storage location
- Whole process is known as Instruction cycle



Datapath

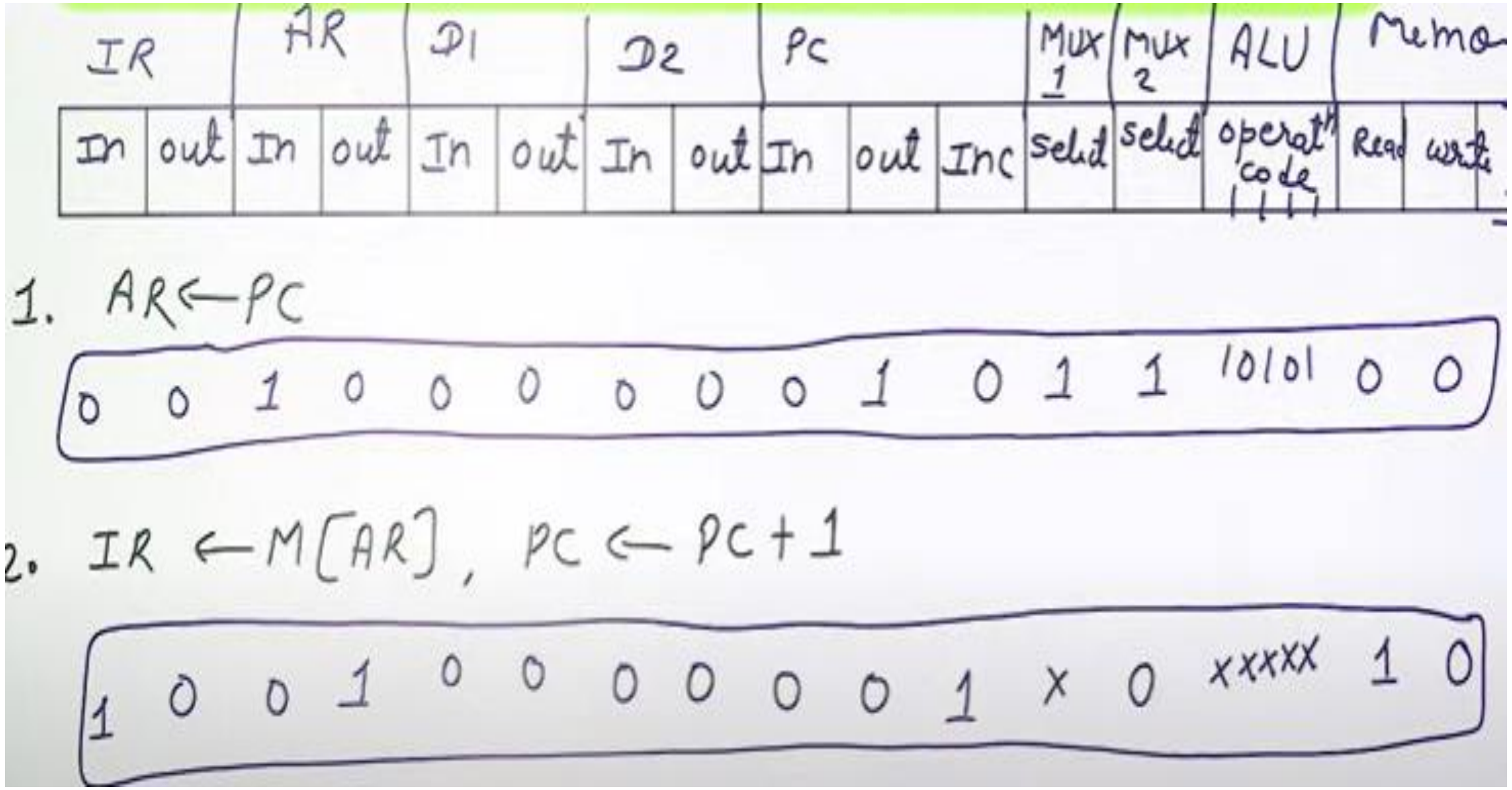
- Collection of functional units such as ALU, multiplexers, etc to perform any data operations.





Control Unit

- Generates control signals and send this to all components of computer(act accordingly).
- Control variable – name given to control signal (READ, WRITE, IN, OUT)
- Control Words: Collection of signals generated by control unit at once.
- Working:
- Control unit generates control words and the signal in control words is send to respective component to act accordingly(for microoperations).
- Whole process is known as CPU cycle.



Microoperations

- Smallest atomic operation performed by the CPU.
- Operations executed on values stored in registers.
- E.g. Gmail login.
- To denote microoperation a new type of symbolic language : Register transfer language.
- E.g. R1<- R2, PC=PC+1, Read from memory R1<- M[Address]
- Both microoperations can be done simultaneously , if both microoperations requires mutually exclusive set of components.
- Control unit generates control signal.
- Generation of control signal depends upon the design of control unit.
- Design means how control unit generates the control signal(control words).

Design of Control unit

- How CU generates control signal
  - Hardwired CU
  - Microprogrammed CU

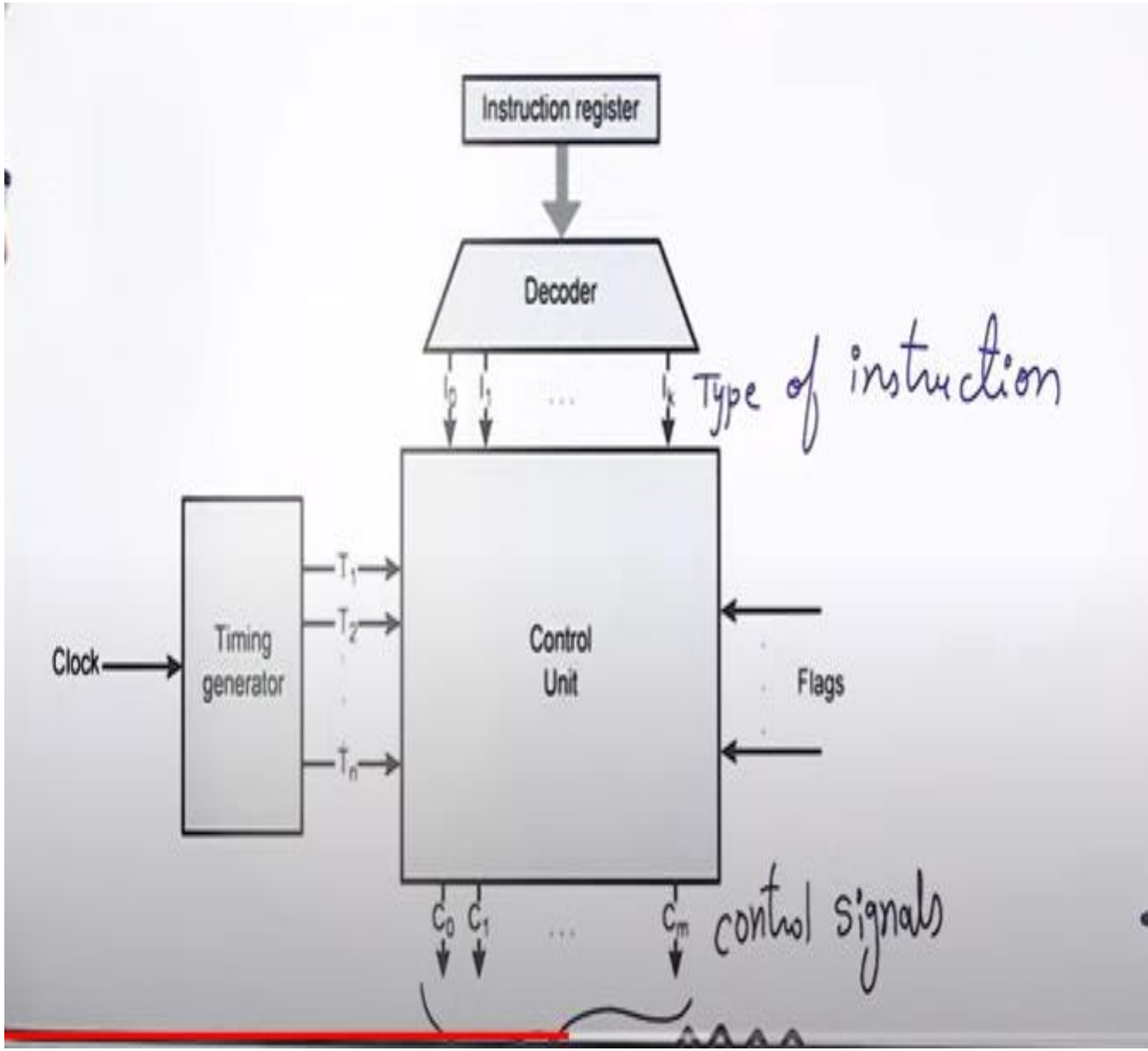
In Hardwired , control signal generates through hardware, (flipflop, logic gates, decoder, and other digital circuits.

CU generates control signal(control words) based on requirements.

Advantages: Very fast due to implementation on hardware

Disadvantages: Rearranging wires among circuit is difficult.

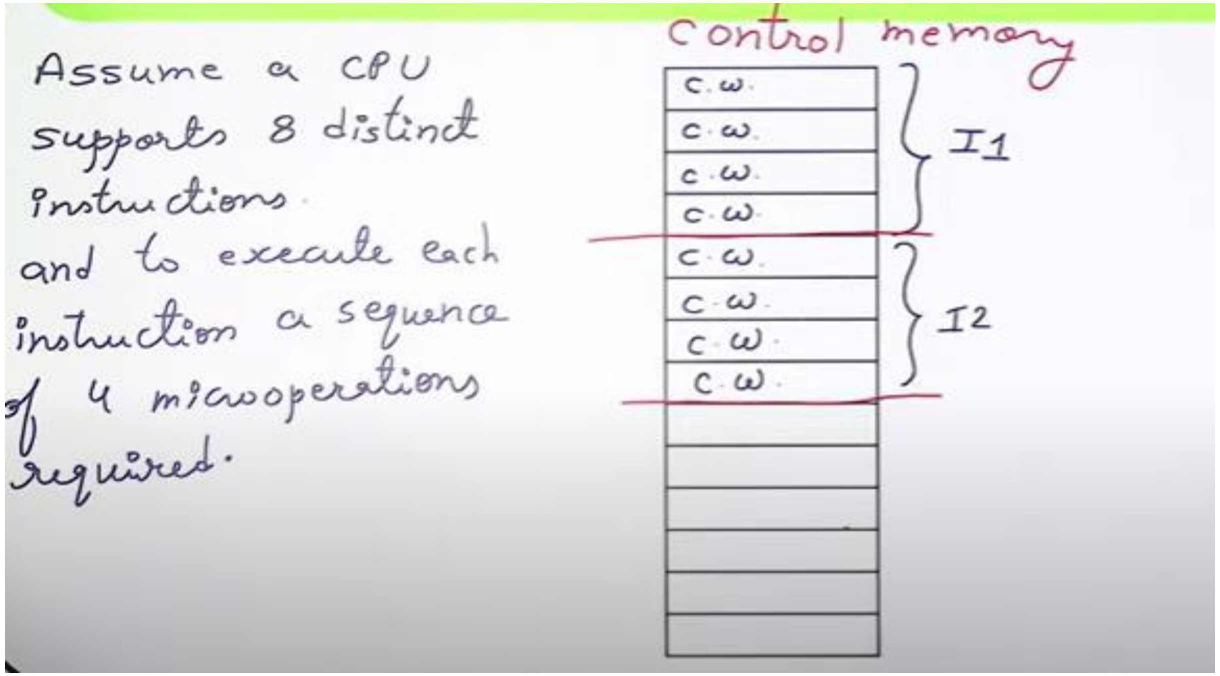
Architect have knowledge of which micro instruction have what control words.



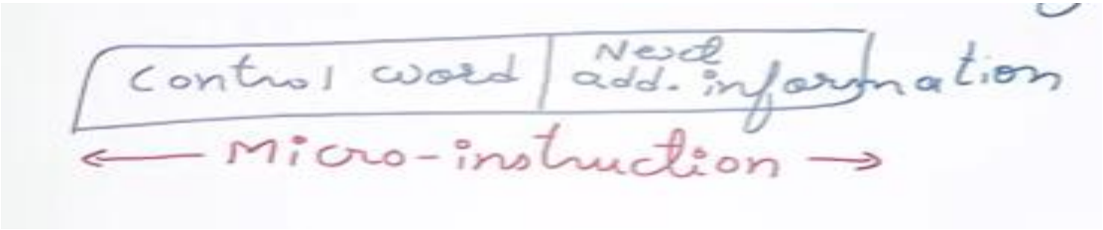
Microprogrammed

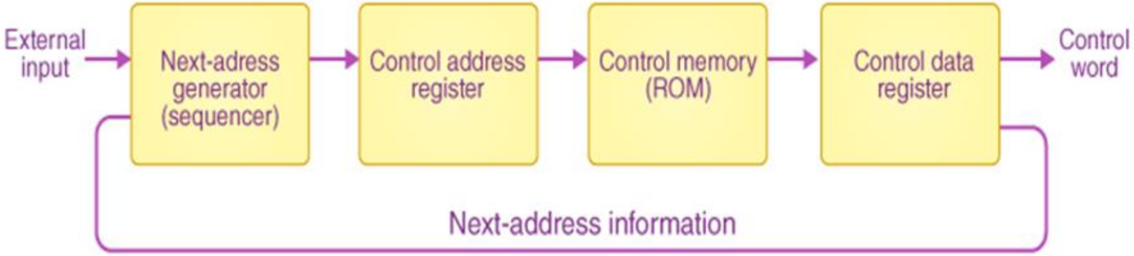
- Control words is already stored in control memory.
- E.g., - doing addition using hand or mind.
- For every microoperations there is specific control words (generated by hardware).
- In microprogrammed, we kept for all possible microoperations, control words for stored in control memory (inside control unit).
- So for each requirements, we look in control memory and generates control words.
- Advantage : Updating control logic is easy
- Disadvantage: Slower than hardwired.(because memory access is slower than hardware access)

- For execution of instruction , we require sequences of microinstructions
- For each microoperation, CU generates control words
- For each instruction, required control words of each microoperations are kept in sequences
- Control word Sequencing- Arranging control words of each instruction sequentially so that in between no external input is required.
- E.g.- Prayer in school diary



- Since control memory is also memory type, each control word is stored at some address.
- Next address generator generates address from where next control word is to be fetched.
- Control address register stores address of control word
- Control memory (ROM) is used to read that control word.
- Fetched control word is stored in control data register.
- Control words stores control signal along with next address info(to help next address generator to next address of control words).
- Control memory is located inside CPU control unit.
- Collection of microinstruction is called microprogram.





Microprogrammed control unit of a Basic Computer

Horizontal Microprogrammed	Vertical Microprogrammed
For each control signal one bit	• Signals are divided into multiple groups in such a way within each group at a time only one signal can be active. Each group info is stored in encoded form • E.g., AR<- PC , AR in signal only one while other register in signal is zero. • Group encoded size = log group size
Large control word size	Small control word
Decoder not required	Decoder is required for (generating actual signal from encoded form)
Faster	Slower(Due to decoder delay)

Instruction execution :

Instruction execution needs the following steps, which are

•PC (program counter) register gives the address of the instruction which needs to be fetched from the memory.

•Then the instruction opcode is decoded.

•On decoding, the processor identifies the number of operands.

•If there is any operand to be fetched from the memory, then that operand address (EA) is calculated.

•Operands are fetched from the memory. If there is more than one operand, repeat the above step.

•After this, the data operation is performed on the operands, and a result is generated.

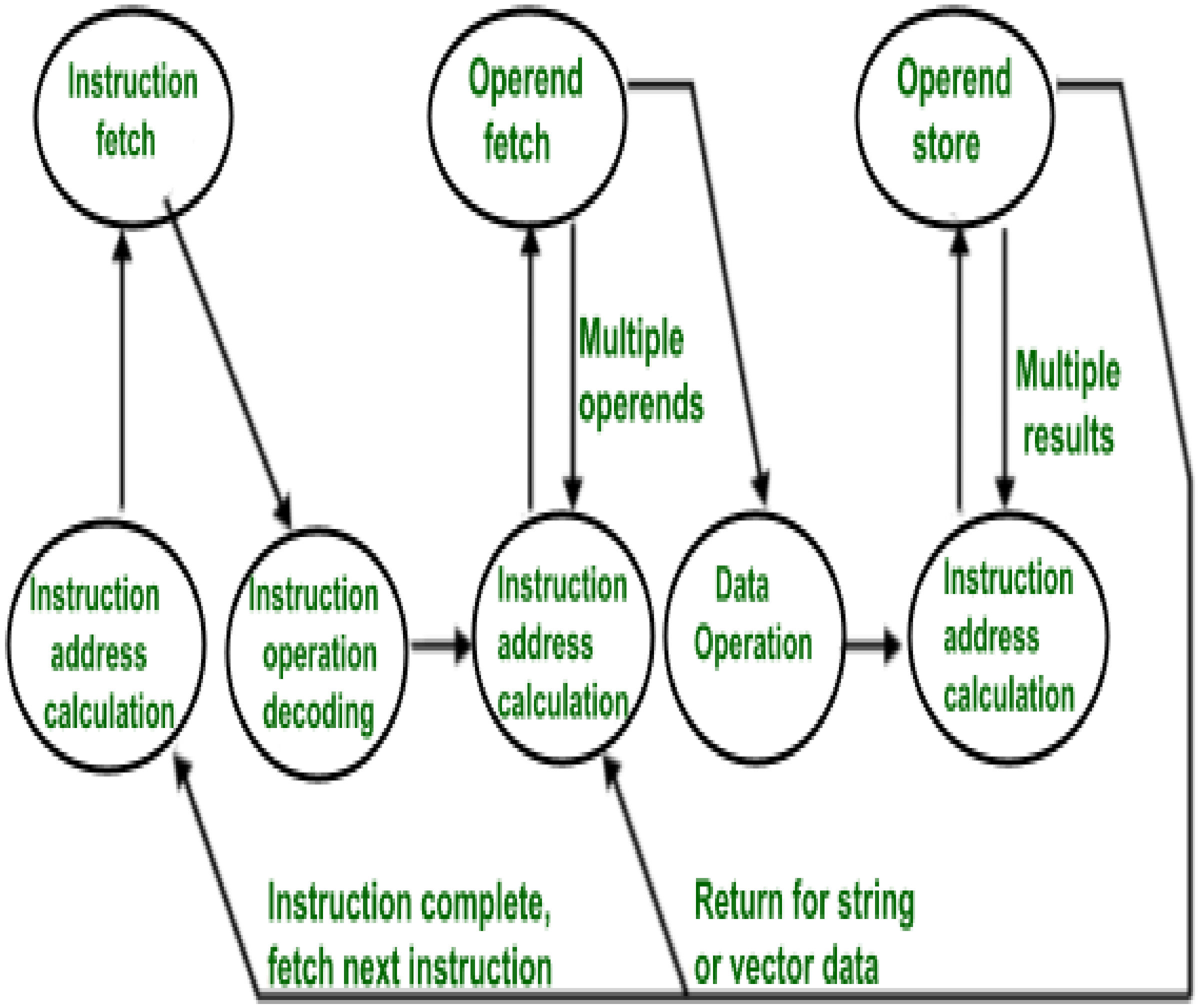
  •If the result has to be stored in a register, the instructions end here.

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

•Now the current instructions have been executed.

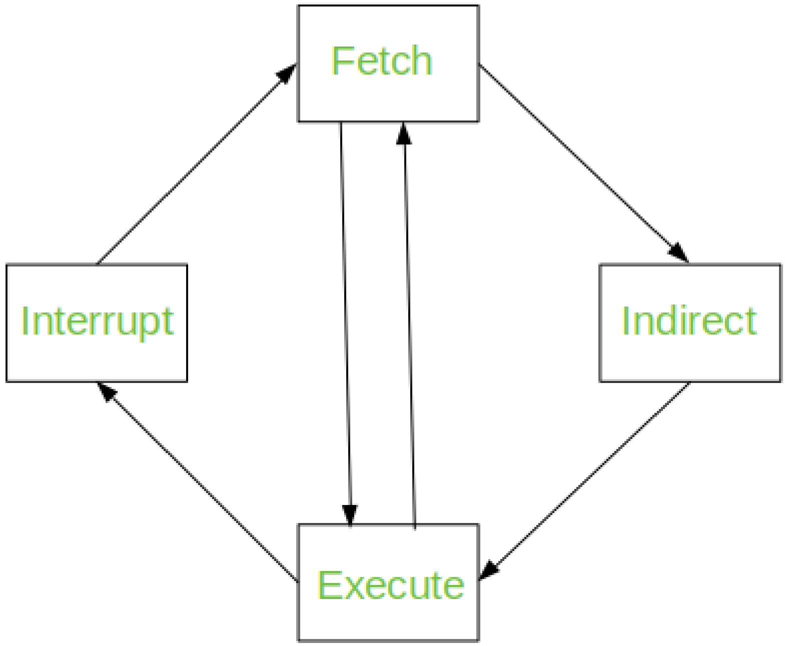
•Side by side, the PC is incremented to calculate the address of the next instruction.

•The above instruction cycle then repeats for further instructions.



1.Fetch: In the fetch cycle, the CPU retrieves the instruction from memory. The PC is then incremented to point to the next instruction in memory.
2.Decode. (Opcode and Operand identification)
3.Execute: In the execute cycle, the CPU performs the operation specified by the instruction. (reading or writing data from or to memory, performing arithmetic or logic operations on data,)

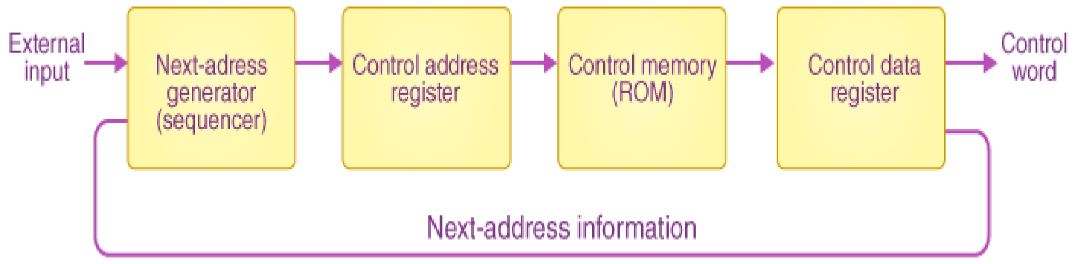
- Fetch operands: In some CPUs, the operands needed for an instruction are fetched during a separate cycle before the execute cycle. This is called the fetch operands cycle.
- Store results: In some CPUs, the results of an instruction are stored during a separate cycle after the execute cycle. This is called the store results cycle.
- Interrupt handling: In some CPUs, interrupt handling may occur during any cycle of the instruction cycle.
- Indirect Cycle: Once an instruction is fetched, the next step is to fetch source operands. *Source Operand* is being fetched by indirect addressing(it can be fetched by any addressing mode, here its done by indirect addressing).



The Instruction Cycle

Control Memory

- Control Unit- defines sequence of microoperations.
- Control signal generated using hardware – hardwired
- Microprogrammed- control unit is designed using microinstructions.
- Microinstruction is stored in permanent storage unit – ROM.
- Microinstruction is sequences of microoperation , represented using binary variables (control variables).
- Control variables at given time can be represented by a string of 1's and 0's known as control word.
- Since microinstruction is stored in ROM, it is not changeable during operation.
- A computer that employs micro programmed control unit has two memory – main memory and control memory.
- Main memory –program storage. Contol memory- fixed microprogram that can not be change by occasional users.
- Microinstruction- sequences of microoperations (fetch, calculate EA, DECODE, execute).



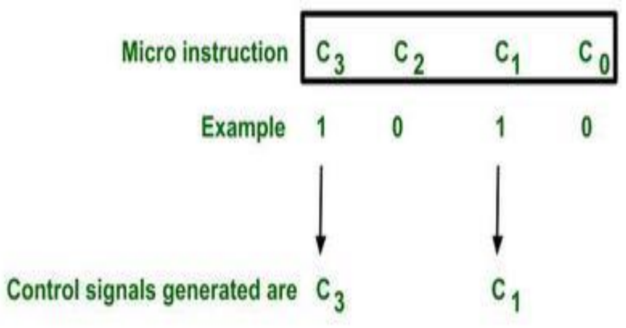
Microprogrammed control unit of a Basic Computer

- Control address register- address of microinstruction
- Control data register- microinstruction reads from control memory
- Next address generator- address of next microinstructions

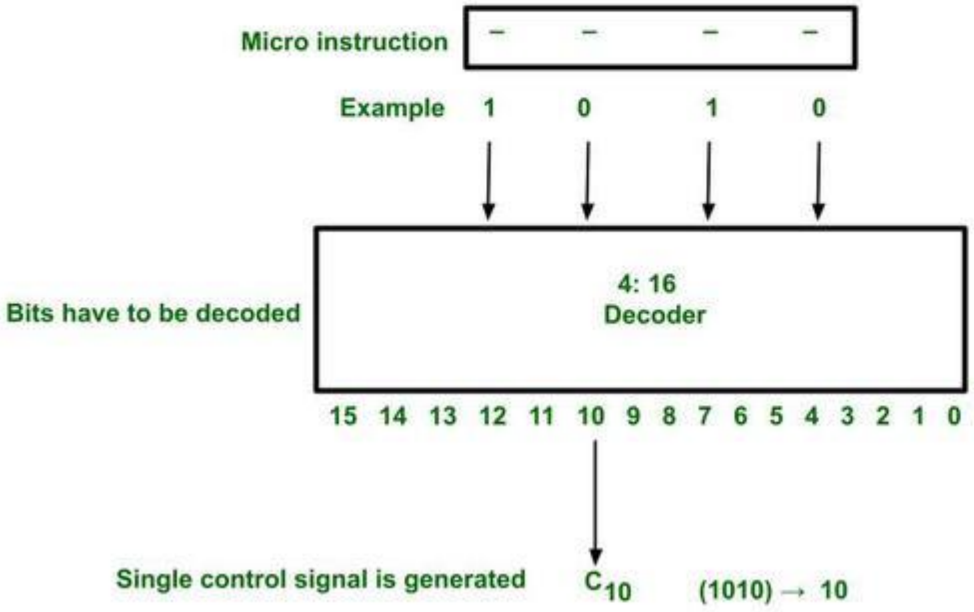
Hardwired Control Unit	Microprogrammed Control Unit
Hardwired control unit generates the control signals needed for the processor using logic circuits	Microprogrammed control unit generates the control signals with the help of micro instructions stored in control memory
Hardwired control unit is faster when compared to microprogrammed control unit as the required control signals are generated with the help of hardwares	This is slower than the other as micro instructions are used for generating signals here
Difficult to modify as the control signals that need to be generated are hard wired	Easy to modify as the modification need to be done only at the instruction level
More costlier as everything has to be realized in terms of logic gates	Less costlier than hardwired control as only micro instructions are used for generating control signals
It cannot handle complex instructions as the circuit design for it becomes complex	It can handle complex instructions
Only limited number of instructions are used due to the hardware implementation	Control signals for many instructions can be generated
Used in computer that makes use of Reduced Instruction Set Computers(RISC)	Used in computer that makes use of Complex Instruction Set Computers(CISC)

Nano Programming

- **Micro instruction format :**
  The control field of the micro-instruction decides the control signals to be produced. It is of two different formats Horizontal or Vertical. Let's discuss it one by one.
- **Type-1 :**
  Horizontal micro-instruction :
  Here, we will discuss the horizontal micro-instruction format as follows.
- Every bit of the micro-instruction corresponds to a control signal.
- The bit which is 1, that corresponding control signal will be produced by the micro-instruction. If there are N bits in micro-instruction then it can produce N control signals by that micro-instruction.
- As the control signals increase, the micro-instruction grows wider. Therefore, the control memory grows horizontally.
- Executes faster because no decoder is used.
- The control memory is large because the micro-instructions are wide.

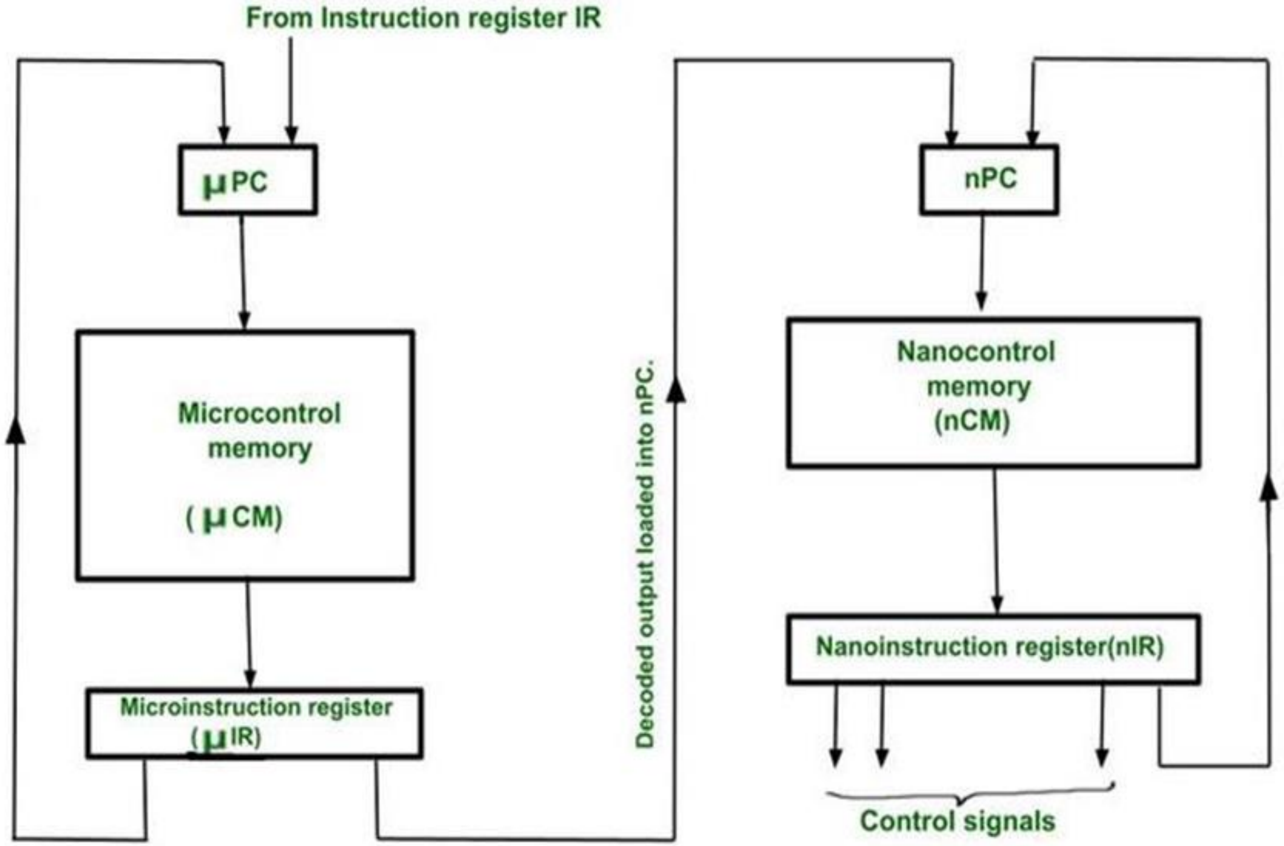


- **Type-2 :**
  Vertical micro-instruction :
- Here, we will discuss the vertical micro-instruction format as follows.
- The bits of the micro-instruction are decoded.
- The decoded output decides the control signal to be produced.
- N bits in the micro-instruction will totally generate 2^N control signals.
- But one control can be generated by one micro-instruction.
- More micro-instructions are needed.
- Also, the decoding makes the execution slower.
- The circuit is complex.



- **Need for nano programming :**
- Horizontal micro-instructions can produce multiple control signals simultaneously but are very wide. This makes the control memory very large.
- Vertical micro-instructions are narrow, but after decoding, only produce one control signal. This makes the control memory small, but the execution is slow.
- Both methods have their own pros and cons.
- Hence, a combination of both techniques is needed to be called nano programming.
- **Nano programming :**

1. Here we have a two-level control memory.
2. The instruction is fetched from the main memory into instruction register IR.
3. Using its opcode we load address of its first micro-instruction into μPC,
4. Using this address we fetch the micro-instruction from micro control memory (μCM) into micro instruction register μIR.
5. This is in vertical form and decoded by a decoder.
6. The decoded output loads a new address in a nano program counter (nPC).
7. By using this address, the nano-instruction is fetched from nano-control memory (nCM) into nano instruction register (nIR).
8. This is in horizontal form and can directly generate control signals which can be multiple at a time.
9. Such a combination gives advantage of both techniques.
10. The size of the control Memory is small as micro-instructions are vertical.



- Suppose processors have 32 instructions (8 for arithmetic operations 8 for data transfer operations, 8 for logical operations 8 for shift operation).
- If we implement it by horizontal micro-instruction, then the size of each micro-instruction should be 32 bits
- If we use nano programming, then micro-control memory has instructions for identifying the type of the instruction by decoding i.e using vertical micro-instruction. After identifying the type of instruction the nano control memory is responsible for generating control signals for that particular instruction type that is identified previously. S
- o the advantage of that is every nano instruction size reduces to 8 bits and every micro-instruction size reduces to 2 bits(for identifying 4 different types of instruction and for 2×4 decoder) i.e advantage of vertical micro-instruction also we can produce multiple control signal for a particular instruction type simultaneously which is an advantage of horizontal instruction.

Disadvantage of Nano programming
Increased memory access time:- The main disadvantage of the two level memory approaches is the loss of speed due to the extra memory access required for Nano control memory.

- **Unit III: Control Unit Design**
- Introduction, Instruction Interpretation & Execution, Control Transfer, Fetch Cycle, Micro programmed Control, Control Memory, Micro programmed vs. Hardwired Control Unit, Nano Programming, Superscalar processing.
- **Unit IV: Memory Organization** **4 lecture hours**
- Memory Locations & Addresses, Semiconductor Memory, Static and Dynamic Memory, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Secondary Memories: Optical Magnetic Tape, Magnetic Disk and Controllers.