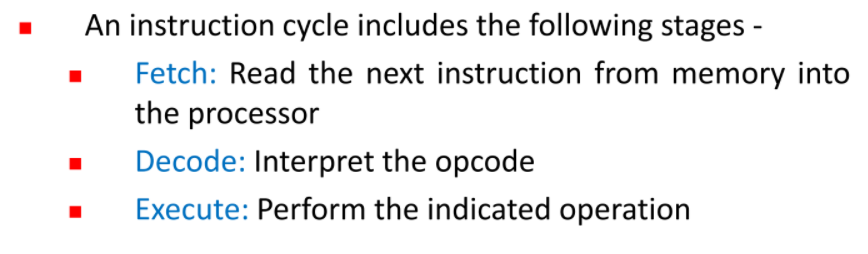
Q1. Demonstrate the Instruction Cycle highlighting data flow during fetch cycle, indirect cycle and

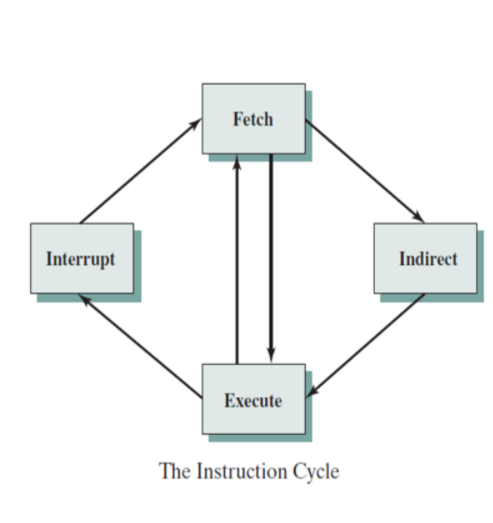
interrupt cycle (10)

a. Define Instruction Cycle (1)



b. Explain with neat diagram the Instruction Cycle State diagram that includes fetch cycle, indirect

cycle and interrupt cycle (3)



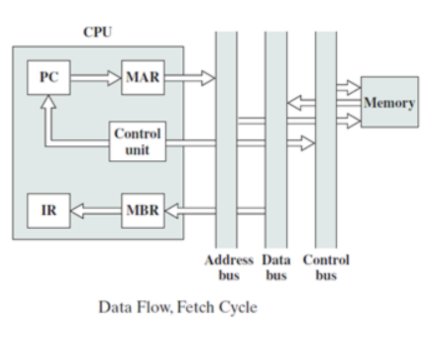
**The Fetch Cycle –**  
During the **fetch** execute **cycle**, the computer retrieves a program instruction from its memory. It then establishes and carries out the actions that are required for that instruction. The **cycle** of **fetching**, decoding, and executing an instruction is continually repeated by the CPU whilst the computer is turned on.

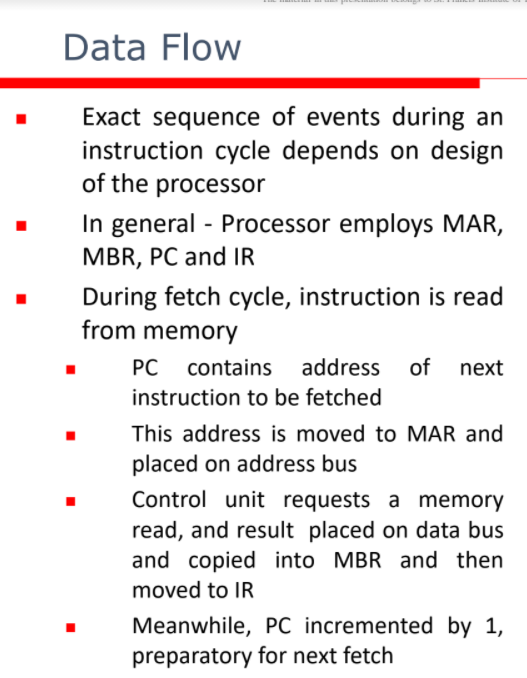
**The Indirect Cycles –**

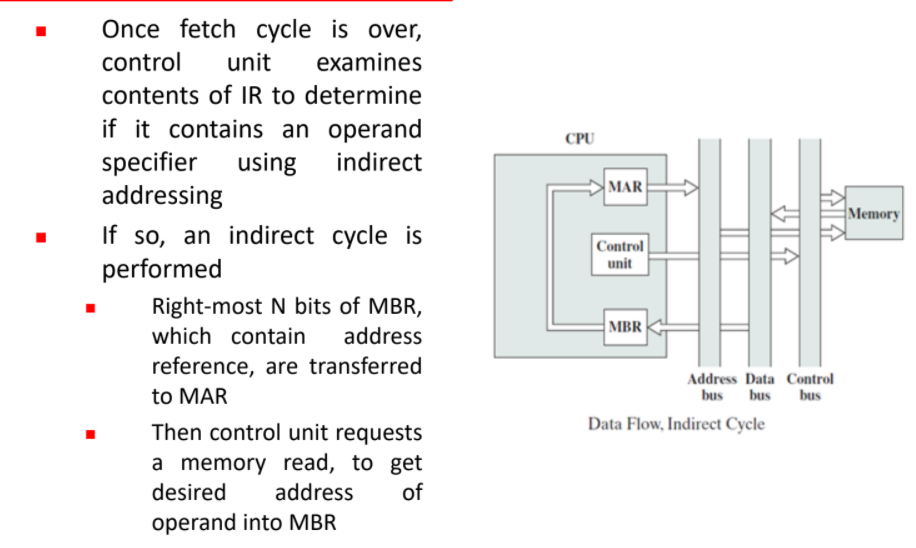
The execution of an instruction may involve one or more operands in memory, each of which requires a memory access. Further, if **indirect** addressing is used, then additional memory accesses are required.

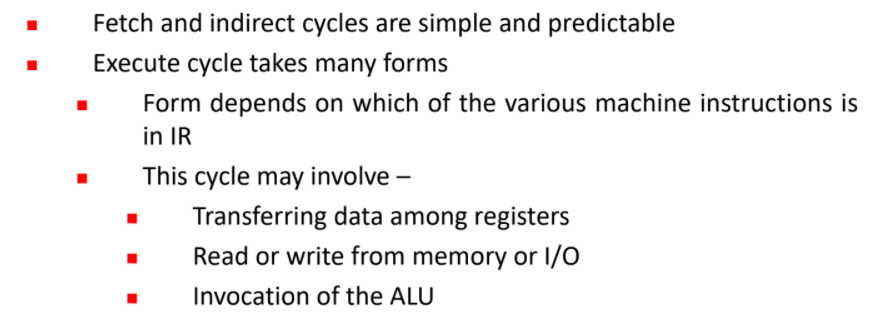
**The Interrupt Cycle**:  
At the completion of the Execute Cycle, a test is made to determine whether any enabled interrupt has occurred or not. If an enabled interrupt has occurred then Interrupt Cycle occurs. The nature of this cycle varies greatly from one machine to another.

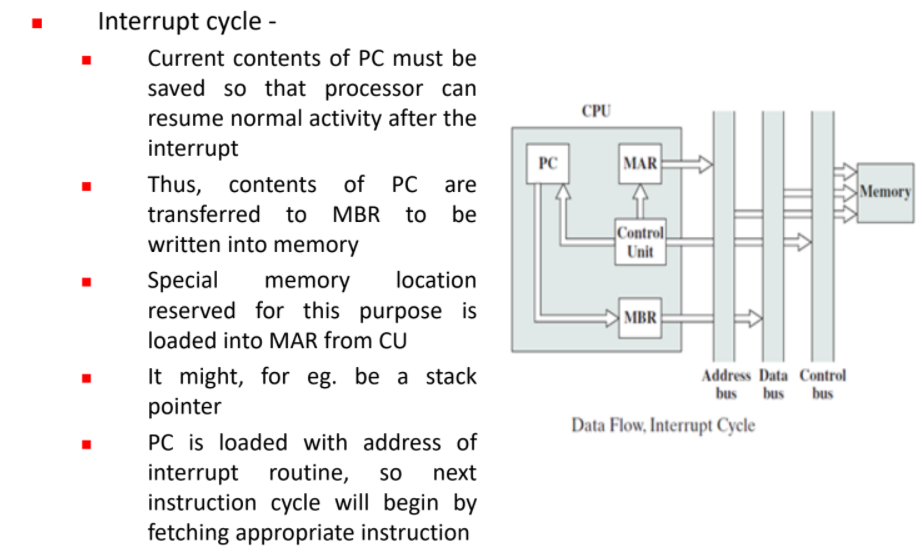
c. Elaborate on data flow during fetch cycle, indirect cycle and interrupt cycle with neat diagrams (6)











Q2. Explain microinstruction sequencing and execution. (10)

a. Define Microinstructions and classify them (2)

The two basic tasks performed by a microprogrammed control unit are as follows:

* Microinstruction sequencing: Get the next microinstruction from the control memory.
* Microinstruction execution: Generate the control signals needed to execute the microinstruction.

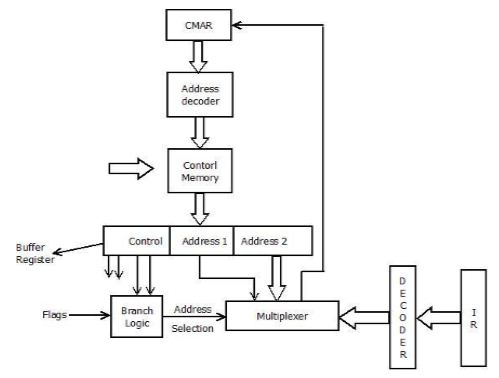
b. Discuss with neat diagram Two address, Single address and Variable format sequencing

techniques (6)

There are three general techniques based on the format of the address information in the microinstruction:

1. Two Address Field.
2. Single Address Field.
3. Variable Format

**Two Address Field:**



The simplest approach is to provide two address field in each microinstruction and multiplexer is provided to select:

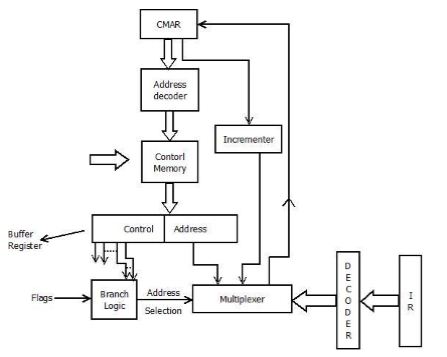
* Address from the second address field.
* Starting address based on the OPcode field in the current instruction.

The address selection signals are provided by a branch logic module whose input consists of control unit flags plus bits from the control partition of the micro instruction.

**Single Address Field:**

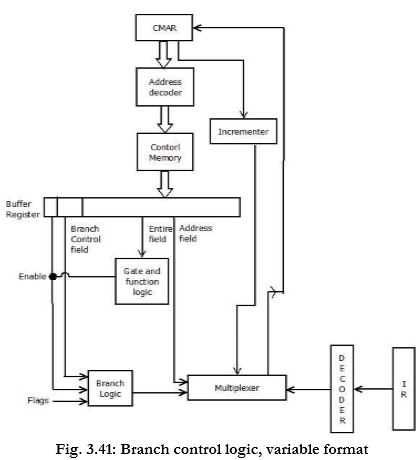
Two-address approach is simple but it requires more bits in the microinstruction. With a simpler approach, we can have a single address field in the micro instruction with the following options for the next address.

* Address Field.
* Based on OPcode in instruction register.
* Next Sequential Address.



The address selection signals determine which option is selected. This approach reduces the number of address field to one. In most cases (in case of sequential execution) the address field will not be used. Thus the microinstruction encoding does not efficiently utilize the entire microinstruction.

**Variable Format:**



In this approach, there are two entirely different microinstruction formats. One bit designates which format is being used. In this first format, the remaining bits are used to activate control signals. In the second format, some bits drive the branch logic module, and the remaining bits provide the address. With the first format, the next address is either the next sequential address or an address derived from the instruction register. With the second format, either a conditional or unconditional branch is specified.

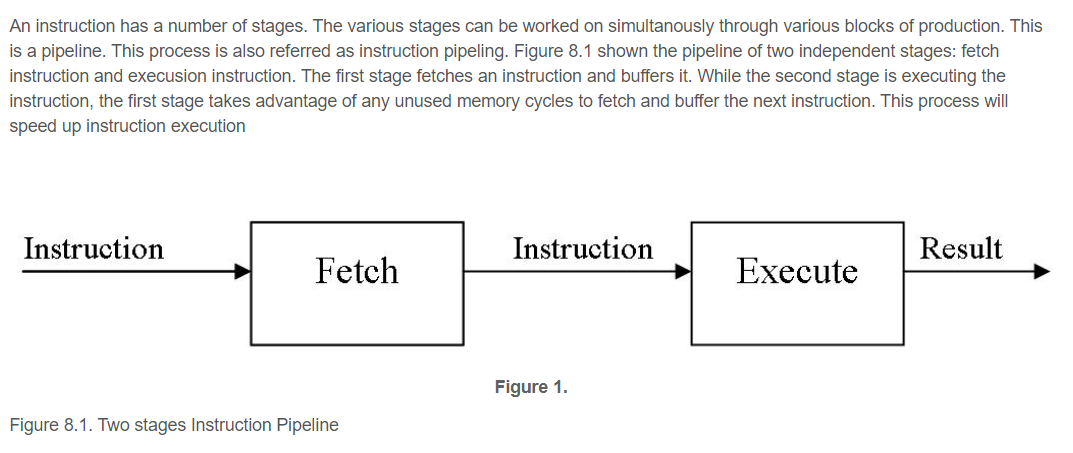
c. Elaborate on Microinstruction Cycle (2)

Q3. What is instruction pipelining? Illustrate the concept of instruction pipelining using a 6-stage

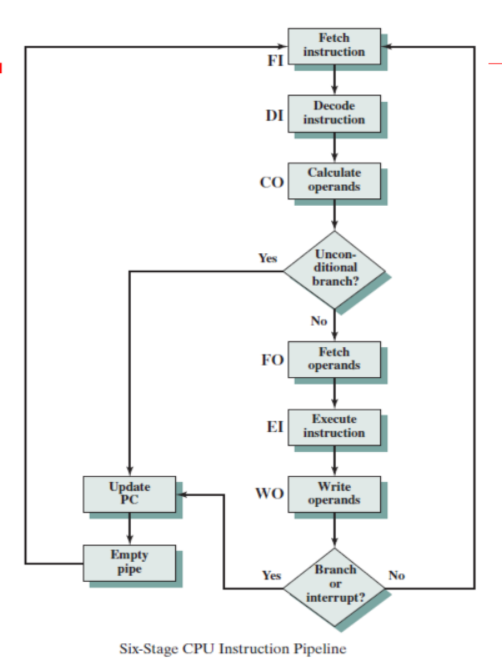
instruction pipeline. Also, highlight the logic needed for pipelining to account for branches using a

flow diagram. (10)

a. Define and elaborate the concept of Instruction Pipeline (two-stage) with neat diagram (3)



b. Discuss the six-stage Instruction Pipeline with neat diagrams (5)



1. A typical instruction cycle can be split into many sub cycles like Fetch instruction, Decode instruction, Execute and Store. The instruction cycle and the corresponding sub cycles are performed for each instruction. These sub cycles for different instructions can thus be interleaved or in other words these sub cycles of many instructions can be carried out simultaneously, resulting in reduced overall execution time. This is called instruction pipelining.
2. The more are the stages in the pipeline, the more the throughput is of the CPU.
3. If the instruction processing is split into six phases, the pipelined CPU will have six different stages for the execution of the sub phases.
4. The six stages are as follows:

* Fetch instruction (FI):
* Decode instruction ((DI):
* Calculate operand (CO):
* Fetch operands (FO):
* Execute Instruction (EI):
* Write operand (WO):

**Fetch instruction:** Instructions are fetched from the memory into a temporary buffer before it gets executed.

**Decode instruction:** The instruction is decoded by the CPU so that the necessary op codes and operands can be determined.

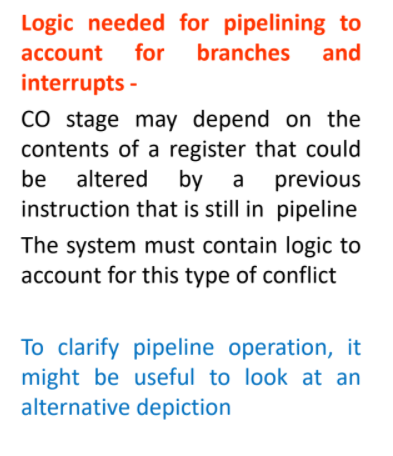
**Calculate operand:** Based on the addressing scheme used, either operands are directly provided in the instruction or the effective address has to be calculated.

**Fetch Operand:** Once the address is calculated, the operands need to be fetched from the address that was calculated. This is done in this phase.

**Execute Instruction:** The instruction can now be executed.

**Write operand:** Once the instruction is executed, the result from the execution needs to be stored or written back in the memory.

c. Highlight the logic needed for pipelining to account for branches using a flow diagram. (2)



Q4. What is instruction pipelining? What are the advantages and disadvantages of pipelining? (5)

**Instruction pipelining is a powerful technique for enhancing**

**performance but requires careful design to achieve optimum**

**results with reasonable complexity**

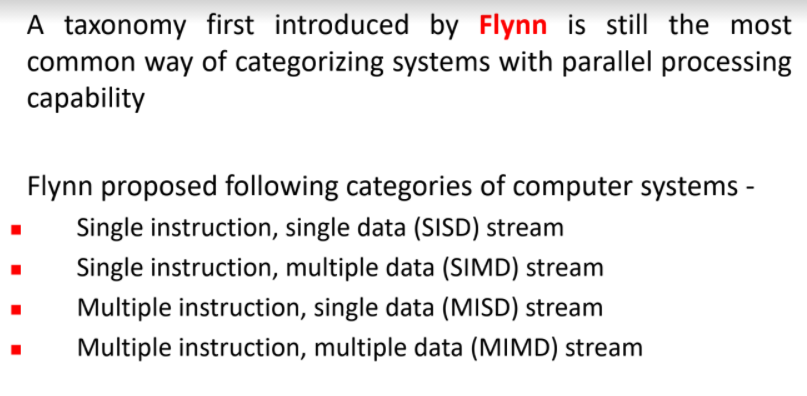
**Advantages of Pipelining**

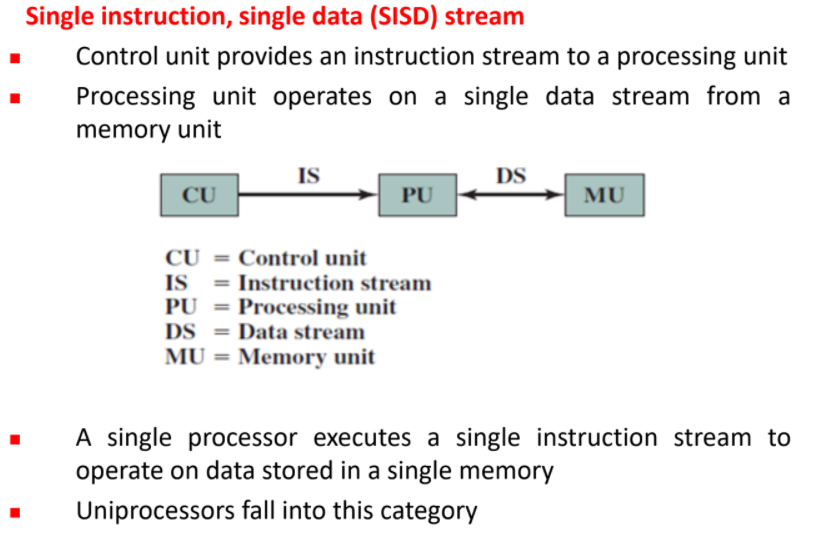
* Instruction throughput increases.
* Increase in the number of pipeline stages increases the number of instructions executed simultaneously.
* Faster ALU can be designed when pipelining is used.
* Pipelined CPU’s works at higher clock frequencies than the RAM.
* Pipelining increases the overall performance of the CPU.

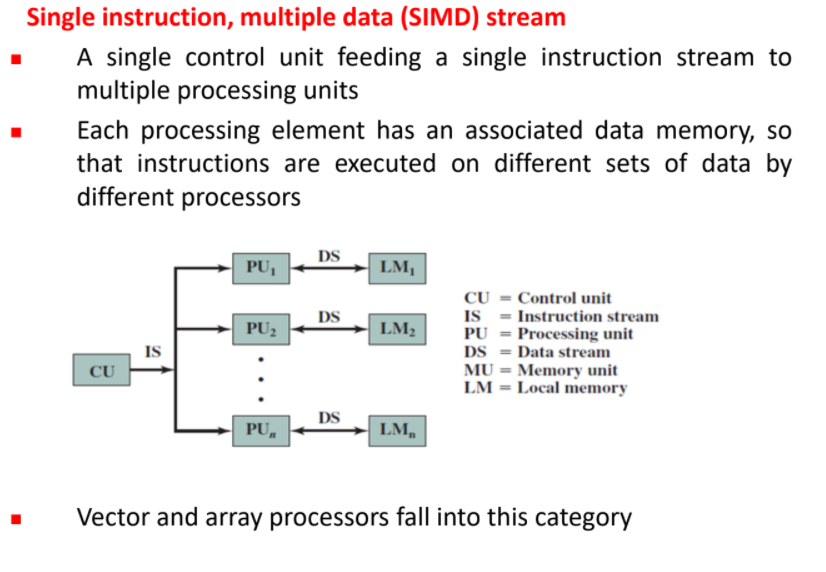
**Disadvantages of Pipelining**

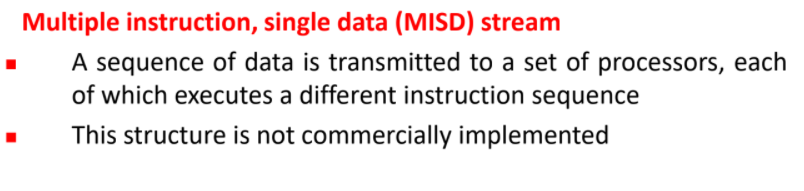
* Designing of the pipelined processor is complex.
* Instruction latency increases in pipelined processors.
* The throughput of a pipelined processor is difficult to predict.
* The longer the pipeline, worse the problem of hazard for branch instructions.

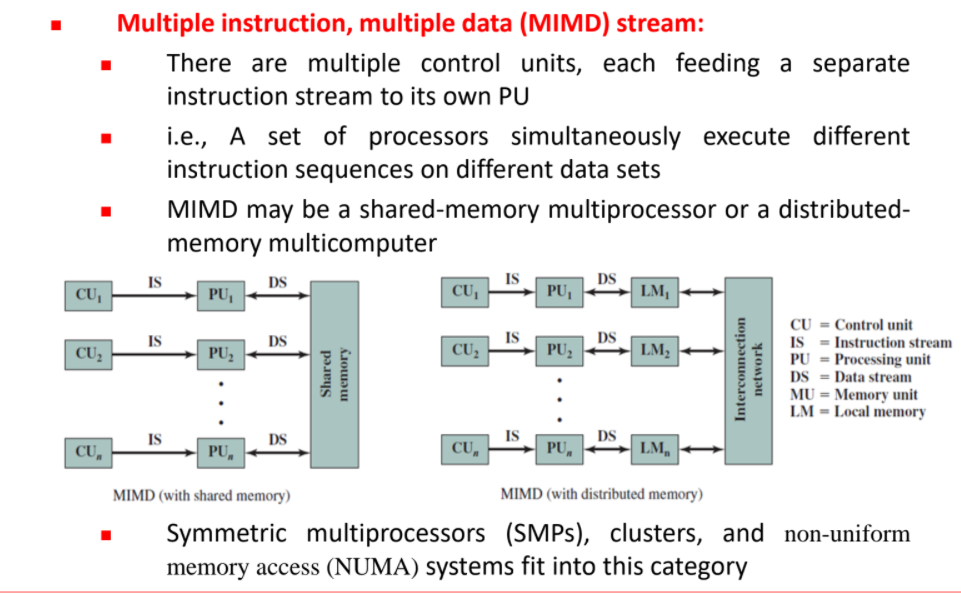
Q5. Discuss Flynn’s classification of parallel processing systems. (5)



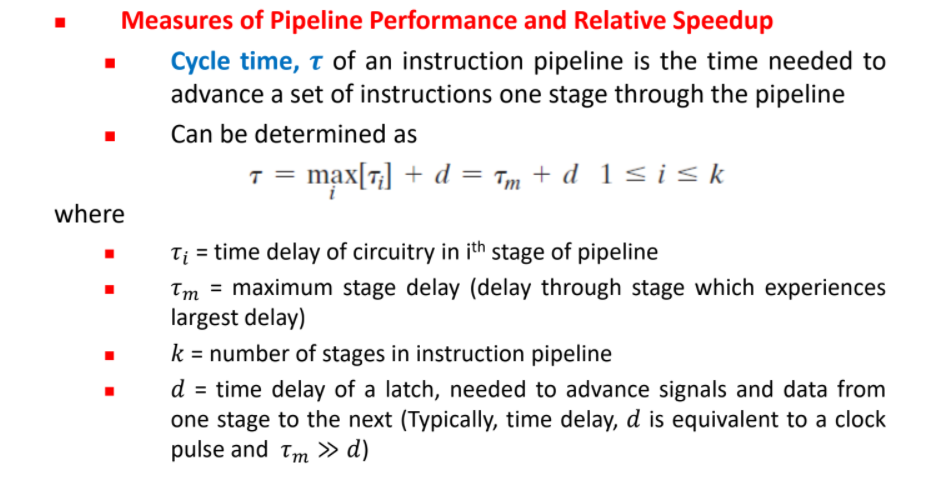


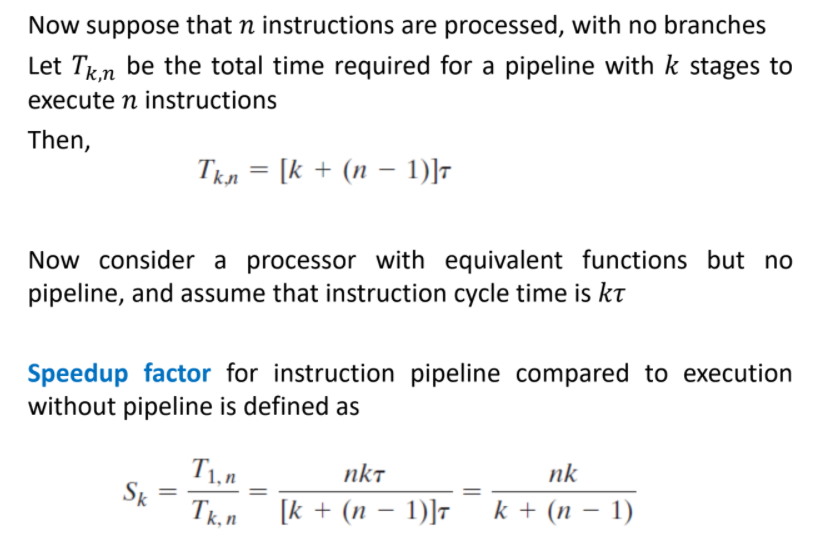




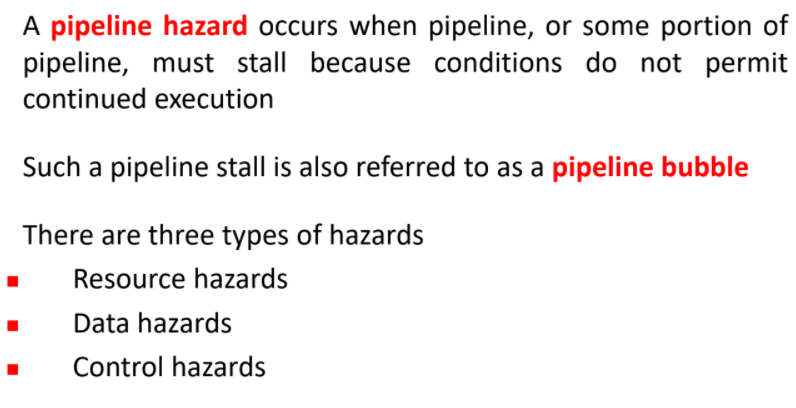


Q6. Enumerate the measures of pipeline performance and relative speedup. (5)





Q7. Give a detailed overview of types of pipeline hazards. (5)



**Data Hazards:**

A data hazard is any condition in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline. As a result of which some operation has to be delayed and the pipeline stalls. Whenever there are two instructions one of which depends on the data obtained from the other.

A=3+A

B=A\*4

For the above sequence, the second instruction needs the value of ‘A’ computed in the first instruction.

Thus the second instruction is said to depend on the first.

If the execution is done in a pipelined processor, it is highly likely that the interleaving of these two instructions can lead to incorrect results due to data dependency between the instructions. Thus the pipeline needs to be stalled as and when necessary to avoid errors.

ii. **Structural Hazards:**

This situation arises mainly when two instructions require a given hardware resource at the same time and hence for one of the instructions the pipeline needs to be stalled.

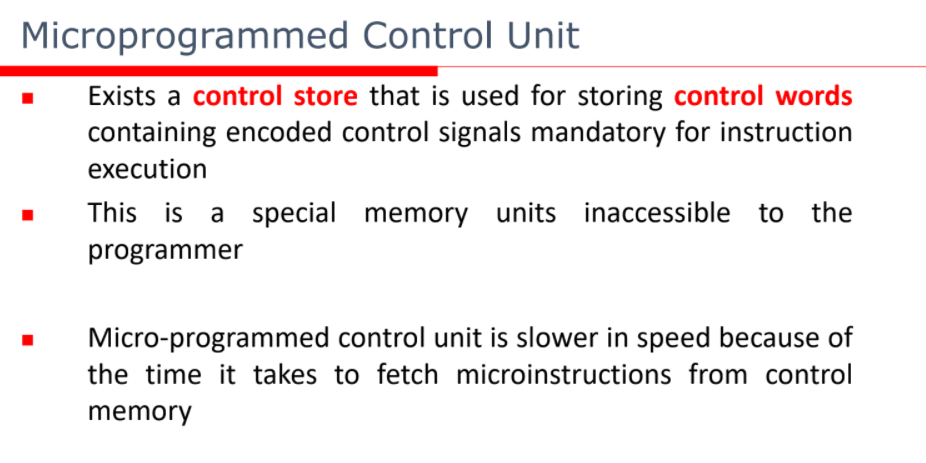
The most common case is when memory is accessed at the same time by two instructions. One instruction may need to access the memory as part of the Execute or Write back phase while other instruction is being fetched. In this case if both the instructions and data reside in the same memory. Both the instructions can’t proceed together and one of them needs to be stalled till the other is done with the memory access part. Thus in general sufficient hardware resources are needed for avoiding structural hazards.

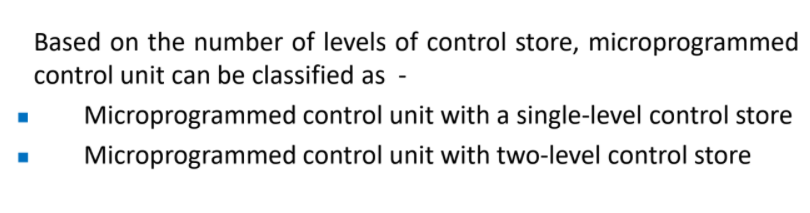
iii. **Control hazards:**

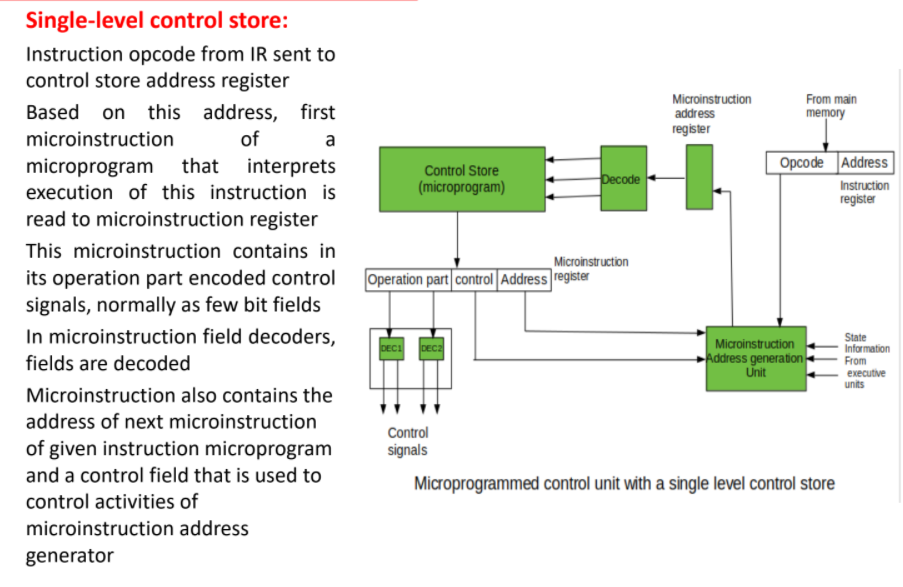
The instruction fetch unit of the CPU is responsible for providing a stream of instructions to the execution unit. The instructions fetched by the fetch unit are in consecutive memory locations and they are executed.

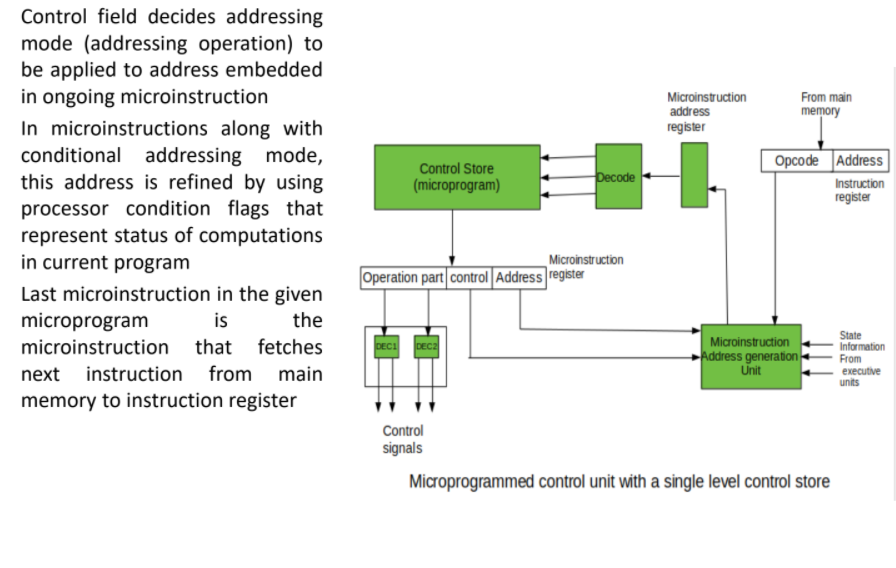
However the problem arises when one of the instructions is a branching instruction to some other memory location. Thus all the instruction fetched in the pipeline from consecutive memory locations are invalid now and need to removed(also called flushing of the pipeline).This induces a stall till new instructions are again fetched from the memory address specified in the branch instruction.

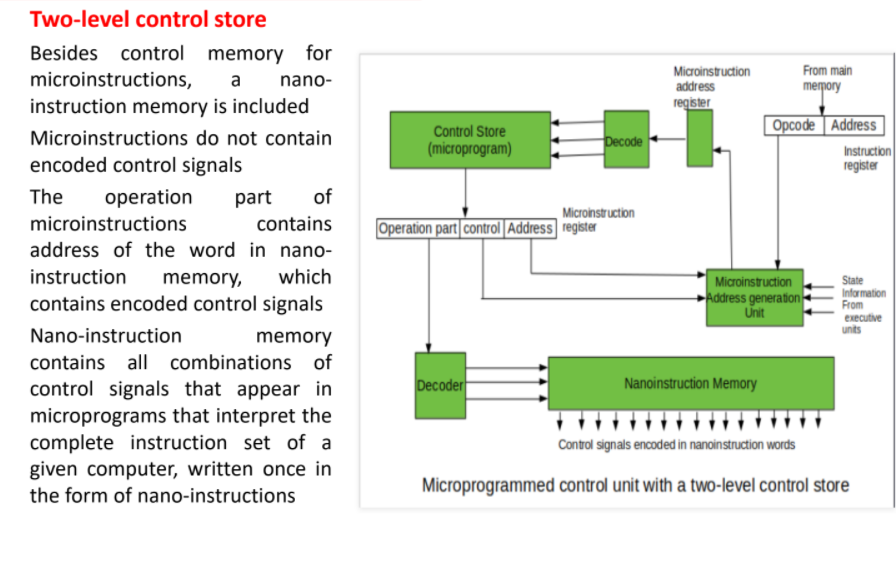
Q8. Describe microprogrammed control unit and specify its advantages and disadvantages. (5)

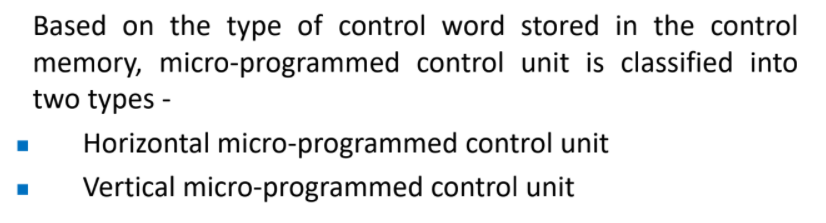


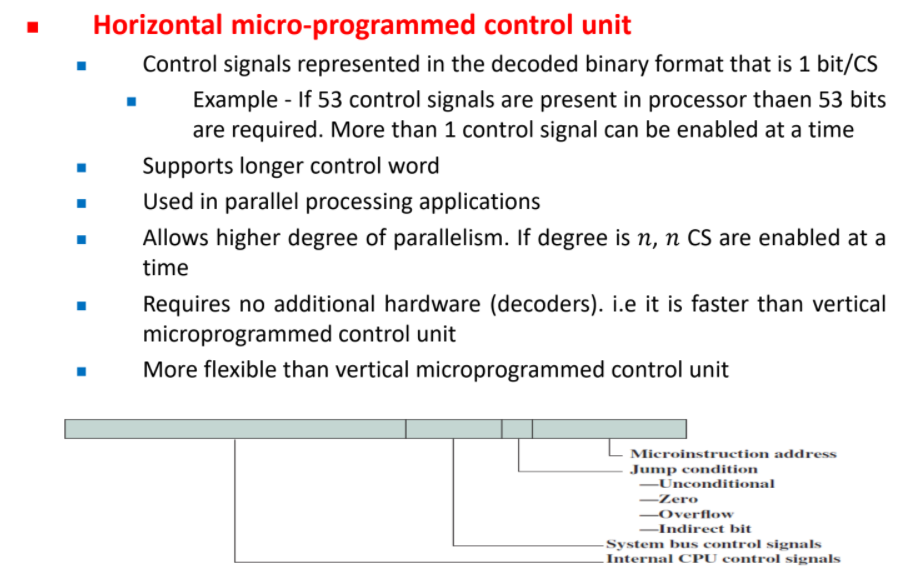


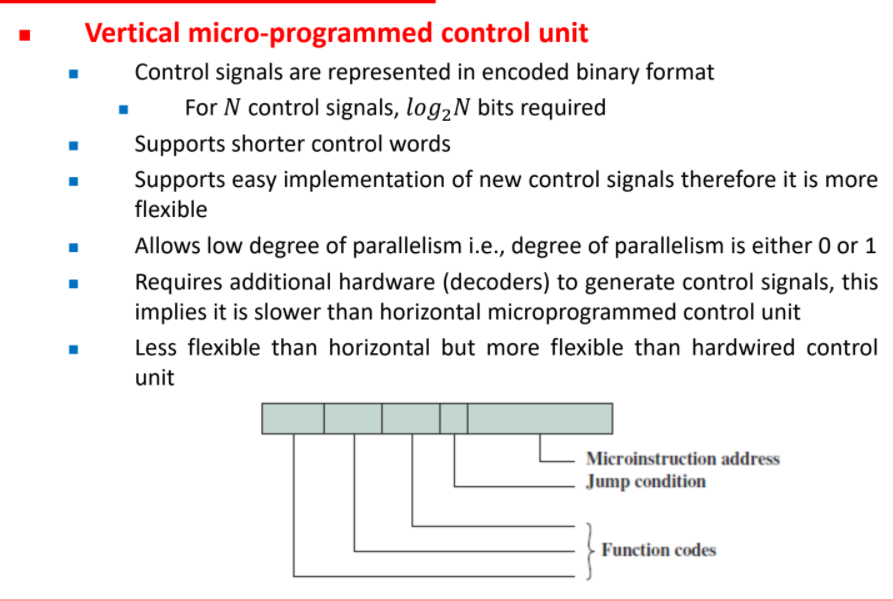




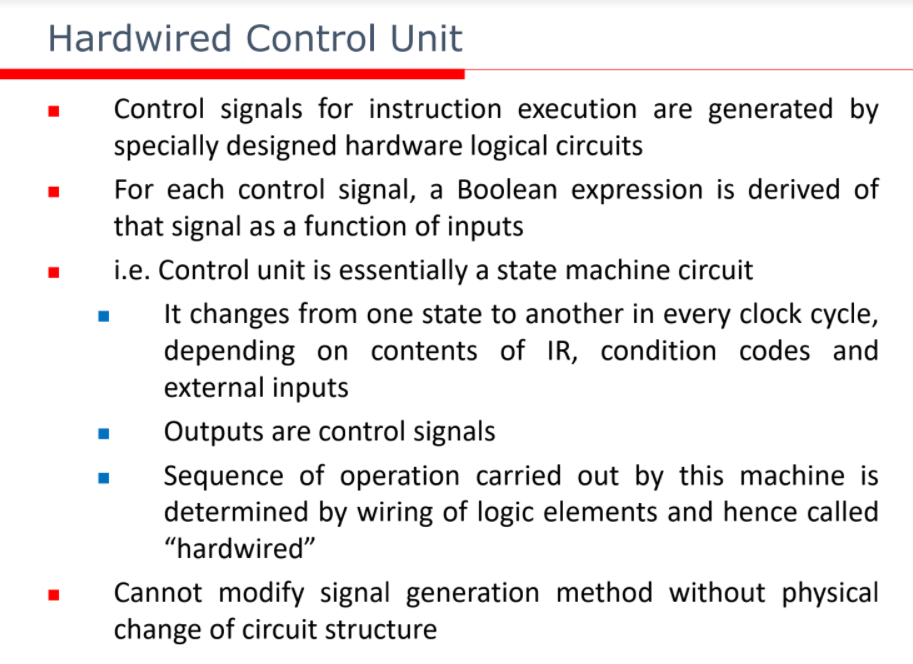


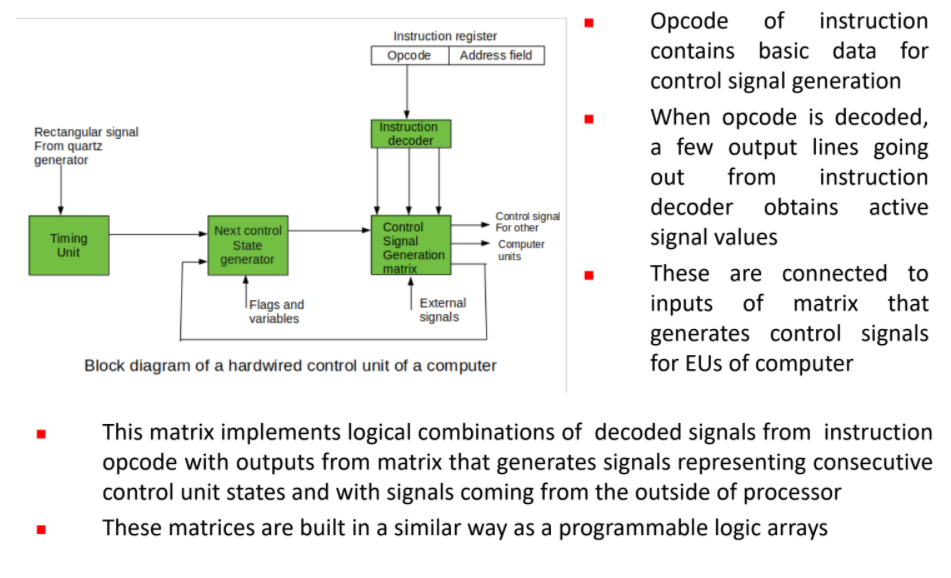






Q9. Describe hardwired control unit and specify its advantages and disadvantages. (5)





**Advantages of Hardwired Control unit:**

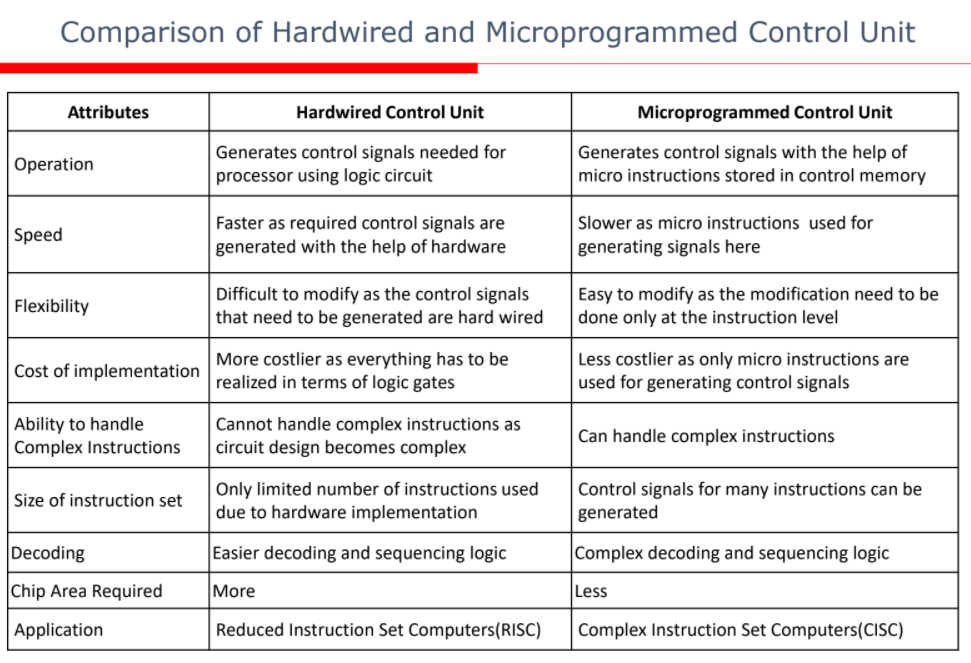
1. It is faster than the microprogrammed control unit.
2. It can be optimized to produce the fast mode of operation.

**Disadvantages of Hardwired control unit:**

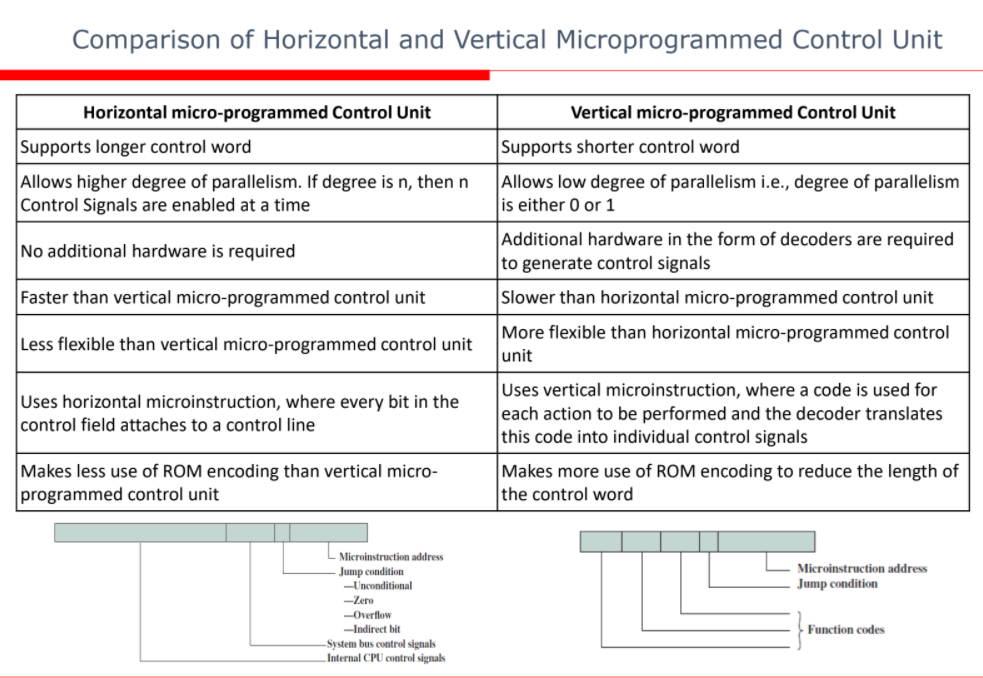
1. Instruction set, the control logic is directly implemented.
2. Requires change in wiring if the design has to be controlled.
3. An occurrence of an error is more.
4. Complex decoding and sequencing logic.
5. It requires a more chip area, therefore, it is a costlier control unit.

Q10. Compare – (5)

(a) Hardwired and microprogrammed control unit



(b) Horizontal and vertical microprogrammed control unit



Q11. Discuss the concept of nano-programming in control unit organization. (5)

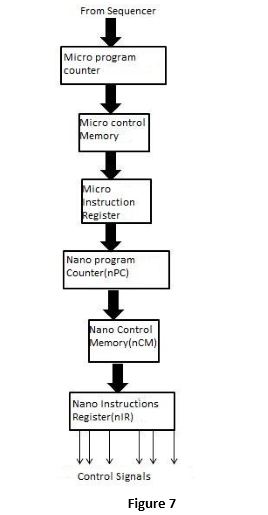
1. In microprogrammed processors, an instruction fetched from memory is interpreted by a micro program stored in a single control memory CM; whereas in other microprogrammed processors, the micro instructions are not directly used by the decoder to generate control signals.

This is achieved by the use of a second control memory called a Nano control memory (nCM).

1. So now there are two levels of control memories, a higher level control memory is known as micro control memory (µCM) and a lower level control memory is known as Nano control memory (nCM). This is shown in Figure 7.
2. Thus a microinstruction is in primary control-store memory, it then has the control signals generated for each microinstruction using a secondary control store memory The output word from the secondary memory is called Nano instruction.
3. The µCM stores micro instructions whereas nCM stores nano instructions.

The decoder uses Nano instructions from nCM to generate control signals.

Thus Nano programming gives an alternative strategy to generate control signals. The process of generation of control signals using nano instructions is shown in Figure 7.



1. Nano instruction addresses are generated by a nano program counter and nano instructions are placed in a register nIR. The next address of nIR is directly obtained. The next address is generated by either incrementing the nano program counter or loading it from external source(branch field or address from micro instruction opcode)

**Advantages of Nano programming**

**1. Reduces total size of required control memory**

In two level control design technique, the total control memory size S2can be calculated as

S2=HmxWm+HnxWnS2=HmxWm+HnxWn

Where H−mnH−mn represents the number of words in the high level memory

WmWm represents the size of word in the high level memory

HnHn represents the number of words in the low level memory

WnWn represents the size of word in the low level memory

Usually, the micro programs are vertically organized so HmHm is large and WmWm is small. In Nano programming, we have a highly parallel horizontal organization, which makes Wn large andHnis small. This gives the compatible size for single level control unit as

S1=HmxS1=Hmx Wn which is larger than S2S2. The reduced size of control memory reduces the total chip area.

**2. Greater design flexibility**

Because of two level memories organization more design flexibility exists between instructions and hardware.

**Disadvantage of Nano programming**

**1. Increased memory access time:**

The main disadvantage of the two level memory approaches is the loss of speed due to theextra memory access required for Nano control memory.

Q12. What are different types of Instruction formats in computer architecture? (5)