# University of Mumbai

## Examination 2020 under cluster __(Lead College: _____)

**Examinations Commencing from 10th April 2021 to 17th April 2021**

Program: Bachelor of Engineering in Computer Engineering

Curriculum Scheme: Rev2019

Examination: DSE SemesterIII

Course Code: **CSC304** and Course Name: **Digital Logic & Computer Architecture**

Time: 2 hour Max. Marks: 80

==========

<table>
<tr><td>Q1.</td><td>Choose the correct option for following questions. All the Questions are compulsory and carry equal marks.</td></tr>
<tr><td></td><td></td></tr>
<tr><td>1.</td><td>Which of the following options represents the correct matching?
<table>
<tr><th>Addressing Mode</th><th>Description</th></tr>
<tr><td>1. Immediate</td><td>A. the address field refers to the address of a word in the memory, which in-turn contains the address of the operand</td></tr>
<tr><td>2. Direct</td><td>B. the address field contains the address (in main memory) where the operand is stored</td></tr>
<tr><td>3. Indirect</td><td>C. operand value is present in the instruction itself (address field)</td></tr>
<tr><td>4. Register Direct</td><td>D. the address field of the operand is a register</td></tr>
</table></td></tr>
<tr><td>Option A:</td><td>1->A; 2->D; 3->C; 4->B;</td></tr>
<tr><td>Option B:</td><td>1->C; 2->B; 3->D; 4->A;</td></tr>
<tr><td>Option C:</td><td>1->C; 2->B; 3->A; 4->D;</td></tr>
<tr><td>Option D:</td><td>1->A; 2->D; 3->B; 4->C;</td></tr>
<tr><td></td><td></td></tr>
<tr><td>2.</td><td>Consider an example of memory organization as shown in the figure below. Which value will be loaded into the accumulator when the instruction “LOAD DIRECT 3” is executed?
<table>
<tr><td>Memory Location address</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>Content</td><td>10</td><td>23</td><td>25</td><td>20</td><td>12</td><td>3</td><td>1</td><td>2</td></tr>
</table></td></tr>
<tr><td>Option A:</td><td>3</td></tr>
<tr><td>Option B:</td><td>25</td></tr>
<tr><td>Option C:</td><td>12</td></tr>
<tr><td>Option D:</td><td>20</td></tr>
<tr><td></td><td></td></tr>
</table>

| | |
|---|---|
| 3. | For a 0-address instruction format, what would be the top element of the stack following sequences of instructions? PUSH 20; PUSH 5; PUSH 5; ADD; SUB; PUSH 20; MULT |
| Option A: | 100 |
| Option B: | 200 |
| Option C: | 10 |
| Option D: | 5 |
| | |
| 4. | What is the value of n in Booth's multiplication of 110* 1000? |
| Option A: | 2 |
| Option B: | 3 |
| Option C: | 4 |
| Option D: | 0 |
| | |
| 5. | In restoring division algorithm, after performing operations (1) left shift operation on A,Q and (2) A=A-M, if magnitude of A > 0 then ? |
| Option A: | Q0=0, A=A+M |
| Option B: | A=A+M |
| Option C: | Q0=1 |
| Option D: | A=A-M |
| | |
| 6. | In non-restoring division algorithm, after performing left shift operation on A, Qregisters,if magnitude of A < 0 then? |
| Option A: | Q0=0, A=A+M |
| Option B: | A=A+M |
| Option C: | Q0=1 |
| Option D: | A=A-M |
| | |
| 7. | In single precision, IEEE754 floating point standard exponent represent by bits and mantissa represent by bits. |
| Option A: | 8, 23 |
| Option B: | 7, 24 |
| Option C: | 7, 23 |
| Option D: | 8, 24 |
| | |
| 8. | How many bits of opcode is required to implement a CPU with 10 arithmetic and logical instructions, 2 control instructions, and 5 data transfer instructions? |
| Option A: | 2 |
| Option B: | 3 |
| Option C: | 4 |
| Option D: | 5 |
| | |
| 9. | In a J-K flip-flop, if J=K the resulting flip-flop is referred to as |
| Option A: | D flip-flop |

| | |
|---|---|
| Option B: | S-R flip-flop |
| Option C: | T flip-flop |
| Option D: | S-K flip-flop |
| | |
| 10. | The instruction read from memory is then placed in the_____ and contents of program counter is_____ so that it contains the address of _____ instruction in the program. |
| Option A: | Program counter, incremented and next |
| Option B: | Instruction register, incremented and previous |
| Option C: | Instruction register, incremented and next |
| Option D: | Address register, decremented and next |
| | |
| 11. | Which is the simplest method of implementing hardwired control unit? |
| Option A: | State Table Method |
| Option B: | Delay Element Method |
| Option C: | Sequence Counter Method |
| Option D: | Using combinational Circuits |
| | |
| 12. | Which instruction does the following set of micro-operations refer to:<br>Steps Action<br>1 PCout, MARin, Read, Select4, Add, Zin<br>2 Zout, PCin, Yin, WMFC<br>3 MDRout, IRin<br>4 R1out, Yin<br>5 R2out, SelectY, Add, Zin<br>6 Zout, R1in, End |
| Option A: | ADD R2, R1 |
| Option B: | ADD R1, R2 |
| Option C: | MOVE R1, R2 |
| Option D: | MOVE R2, R1 |
| | |
| 13. | Which of the following statements is false? |
| Option A: | Diagonal micro-instructions encoding requires multiple decoders. |
| Option B: | In vertical micro-instructions encoding, more than one control signals cannot be activated at a time. |
| Option C: | Horizontal micro-instructions encoding has a lower cost of implementation. |
| Option D: | On one end of a spectrum, a *vertical* microinstruction is highly encoded and may look like a simple macroinstruction containing a single opcode field and one or two operand specifiers. |
| | |
| 14. | In _____ mapping, the data can be mapped anywhere in the Cache Memory. |
| Option A: | Associative |
| Option B: | Direct |
| Option C: | Set Associative |
| Option D: | Indirect |

| | |
|---|---|
| | |
| 15. | A second factor in locality of reference is the presence of loops in programs. Instructions in a loop, even when they are far apart in spatial terms, are executed repeatedly, resulting in a high frequency of reference to their addresses. This characteristic is referred to as _____. |
| Option A: | Spatial locality. |
| Option B: | temporal locality |
| Option C: | branch locality. |
| Option D: | Equidistant locality |
| | |
| 16. | _____ consists essentially of internal flip-flops that store the binary information. |
| Option A: | Static RAM |
| Option B: | Dynamic RAM |
| Option C: | PROM |
| Option D: | EEPROM |
| | |
| 17. | SIMD represents an organization that _____. |
| Option A: | refers to a computer system capable of processing several programs at the same time. |
| Option B: | represents organization of single computer containing a control unit, processor unit and a memory unit. |
| Option C: | includes many processing units under the supervision of a common control unit. |
| Option D: | similar to Von Neumann architecture. |
| | |
| 18. | In parallelization, if P is the proportion of a system or program that can be made parallel, and 1-P is the proportion that remains serial, then the maximum speedup that can be achieved using N number of processors is 1/((1P)+(P/N). This law is called _____ |
| Option A: | Newton's law |
| Option B: | Ohms law |
| Option C: | Amdahl's law |
| Option D: | Flynn's law |
| | |
| 19. | To resolve the clash over the access of the system BUS we use _____ |
| Option A: | Multiple BUS |
| Option B: | BUS arbitrator |
| Option C: | Priority access |
| Option D: | DMA controller |
| | |
| 20. | Select true statement from the following. |
| Option A: | USB is a parallel mode of transmission of data and this enables for the fast speeds of data transfers. |
| Option B: | In USB the devices can communicate with each other. |

| | |
|---|---|
| Option C: | The type/s of packets sent by the USB is/are Data. |
| Option D: | When the USB is connected to a system, its root hub is connected to the Processor BUS. |

**Q.2 Solve any Four out of Six.**

**a)** Briefly describe the Von Neumann Model computer architecture. **5**

**b)** Write a short note on Interleaved and Associative Memory. **5**

**c)** Differentiate between hardwired control unit and Microprogrammed Control unit. **5**

**d)** What is meaning of delayed branch and branch prediction? Write a difference between them. **5**

**e)** Draw and explain instruction cycle state diagram. **5**

**f)** Multiply (-10) and (-8) using Booth's algorithm. **5**

**Q.3 Solve any Two out of Three.**

**a)** Draw the flowchart of Restoring Division Algorithm & perform 10 /3 using this Algorithm. **10**

**b)** Explain with suitable diagrams - Flynn's Classification of Computer Architecture. **10**

**c)** Consider a Cache memory of 16 words. Each block consists of 4 words. Size of the main memory is 128 bytes. Draw the Associative Mapping and Calculate the TAG and WORD size. **10**

# University of Mumbai

## Examination 2020 under cluster __ (Lead College: _____)

**Examinations Commencing from 10th April 2021 to 17th April 2021**

Program: Bachelor of Engineering in Computer Engineering

Curriculum Scheme: Rev2019

Examination: DSE Semester III

Course Code: **CSC304** and Course Name: **Digital Logic & Computer Architecture**

Time: 2 hour Max. Marks: 80

==========

| Question Number | Correct Option (Enter either 'A' or 'B' or 'C' or 'D') |
|---|---|
| Q1. | C |
| Q2. | D |
| Q3. | B |
| Q4 | C |
| Q5 | C |
| Q6 | B |
| Q7 | A |
| Q8. | D |
| Q9. | C |
| Q10. | C |
| Q11. | A |
| Q12. | B |
| Q13. | C |
| Q14. | A |
| Q15. | B |
| Q16. | A |
| Q17. | C |
| Q18. | C |
| Q19. | B |
| Q20. | D |

**Q.2 Solve any Four out of Six.**

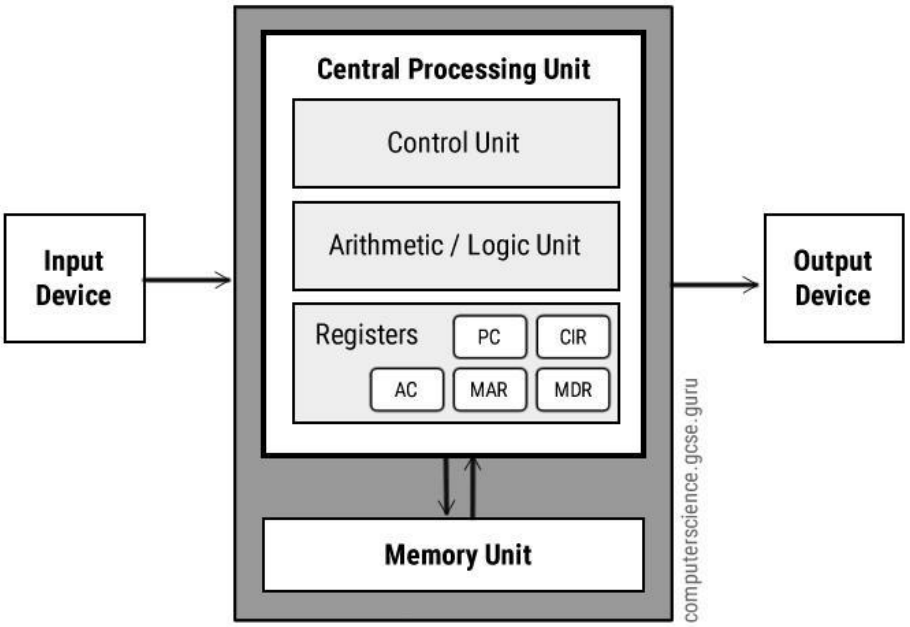
**a) Briefly describe the Von Neumann Model computer architecture. 5 Marks**

**Solution:**

***Diagram 2 marks and 1 marks for each component briefly description.***

## Von-Neumann Model

- Von Neumann architecture was first published by John von Neumann in 1945.
- His computer architecture design consists of a Control Unit, Arithmetic and Logic Unit (ALU), Memory Unit, Registers and Inputs/Outputs.
- Von Neumann architecture is based on the stored-program computer concept, where instruction data and program data are stored in the same memory. This design is still used in most computers produced today.




**Central Processing Unit (CPU)**

- The Central Processing Unit (CPU) is the electronic circuit responsible for executing the instructions of a computer program.
- It is sometimes referred to as the microprocessor or processor.
- The CPU contains the ALU, CU and a variety of registers.

**Registers**

Registers are high speed storage areas in the CPU. All data must be stored in a register before it can be processed.

| | | |
|---|---|---|
| **MAR** | **Memory Address Register** | **Holds the memory location of data that needs to be accessed** |
| **MDR** | Memory Data Register | Holds data that is being transferred to or from memory |
| **AC** | Accumulator | Where intermediate arithmetic and logic results are stored |
| **PC** | Program Counter | Contains the address of the next instruction to be executed |
| **CIR** | Current Instruction Register | Contains the current instruction during processing |

**Arithmetic and Logic Unit (ALU):** The ALU allows arithmetic (add, subtract etc) and logic (AND, OR, NOT etc) operations to be carried out.

**Control Unit (CU) :**

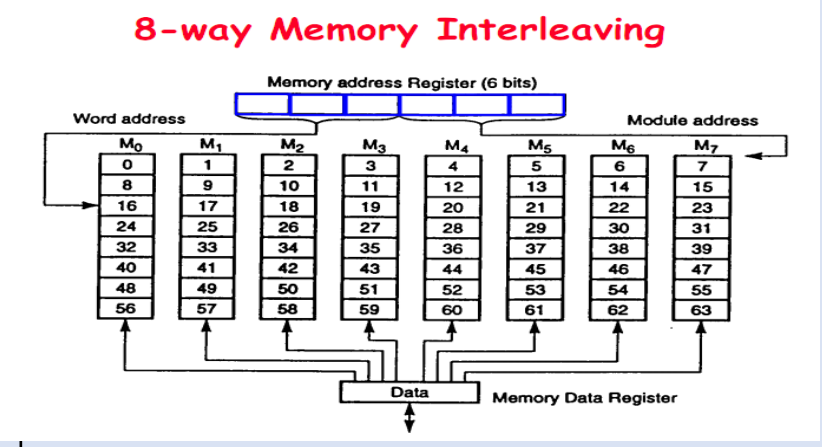
- The control unit controls the operation of the computer's ALU, memory and input/output devices, telling them how to respond to the program instructions it has just read and interpreted from the memory unit.

- The control unit also provides the timing and control signals required by other computer components.

**Buses:**

- Buses are the means by which data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory.
- A standard CPU system bus is comprised of a control bus, data bus and address bus.

| | |
|---|---|
| **Address Bus** | **Carries the addresses of data (but not the data) between the processor and memory** |
| **Data Bus** | Carries data between the processor, the memory unit and the input/output devices |
| **Control Bus** | Carries control signals/commands from the CPU (and status signals from other devices) in order to control and coordinate all the activities within the computer |




**Memory Unit**

- The memory unit consists of RAM, sometimes referred to as primary or main memory. Unlike a hard drive (secondary memory), this memory is fast and also directly accessible by the CPU.
- RAM is split into partitions. Each partition consists of an address and its contents (both in binary form).
- The address will uniquely identify every location in the memory.
- Loading data from permanent memory (hard drive), into the faster and directly accessible temporary memory (RAM), allows the CPU to operate much quicker.

**b) Write a short note on Interleaved and Associative Memory. 5 Marks**

***2 ½ marks for any 3-4 points on Interleaved memory and 2 ½ marks for any valid 3-4 points on Associative memory. If diagram provided can be consider for 1 mark.***

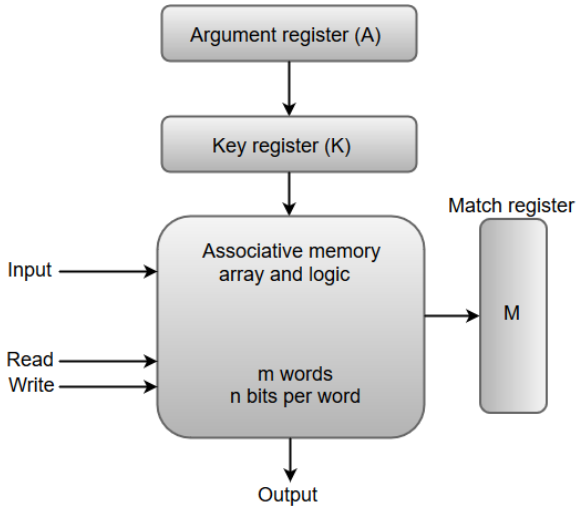
**Interleaved Memory:**

- It is a technique for compensating the relatively slow speed of DRAM(Dynamic RAM). In this technique, the main memory is divided into memory banks which can be accessed individually without any dependency on the other.
- **For example:** If we have 4 memory banks(4-way Interleaved memory), with each containing 256 bytes, then, the Block Oriented scheme(no interleaving), will assign virtual address 0 to 255 to the first bank, 256 to 511 to the second bank. But in Interleaved memory, virtual address 0 will be with the first bank, 1 with the second memory bank, 2 with the third bank and 3 with the fourt, and then 4 with the first memory bank again.

- Hence, CPU can access alternate sections immediately without waiting for memory to be cached. There are multiple memory banks which take turns for supply of data.
- Memory interleaving is a technique for increasing memory speed. It is a process that makes the system more efficient, fast and reliable.
- **For example:** In the above example of 4 memory banks, data with virtual address 0, 1, 2 and 3 can be accessed simultaneously as they reside in spearate memory banks, hence we do not have to wait for completion of a data fetch, to begin with the next.
- An interleaved memory with n banks is said to be **n-way interleaved**. In an interleaved memory system, there are still **two banks of DRAM** but logically the system seems one bank of memory that is twice as large.

**Associative Memory:**

- An associative memory can be considered as a memory unit whose stored data can be identified for access by the content of the data itself rather than by an address or memory location.
- Associative memory is often referred to as **Content Addressable Memory (CAM)**.
- When a write operation is performed on associative memory, no address or memory location is given to the word. The memory itself is capable of finding an empty unused location to store the word.
- On the other hand, when the word is to be read from an associative memory, the content of the word, or part of the word, is specified. The words which match the specified content are located by the memory and are marked for reading.




- From the block diagram, we can say that an associative memory consists of a memory array and logic for 'm' words with 'n' bits per word.
- The functional registers like the argument register **A** and key register **K** each have **n** bits, one for each bit of a word. The match register **M** consists of **m** bits, one for each memory word.
- The words which are kept in the memory are compared in parallel with the content of the argument register.
- The key register (K) provides a mask for choosing a particular field or key in the argument word. If the key register contains a binary value of all 1's, then the entire argument is compared with each memory word.

- Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus, the key provides a mask for identifying a piece of information which specifies how the reference to memory is made.

**c) Differentiate between hardwired control unit and Microprogrammed Control unit.**

5 Marks

***Every valid comparative point should be awarded with 1 mark. Minimum 5 valid points for 5 marks.***

| Sr. no. | Key points | Hardware Control Unit | Micro programmed |
|---|---|---|---|
| **1)** | Control signals | Generated using hardware | Generated using a micro program. |
| **2)** | Circuit design | Since h/w is used, the circuit design is rigid. | Since µ-program is used, design is flexible and can be changed by simply changing a few µ-inst. |
| **3)** | Debugging | Very difficult | Easier |
| **4)** | Speed | Faster | Slower |
| **5)** | Cost of implementation | More | Cheaper |
| **6)** | Subject ability to errors | More | Less |
| **7)** | Flexibility | Not flexible | Flexible |
| **8)** | Ability to handle large complex instruction sets | Somewhat difficult | Easier |
| **9)** | Design Process | complicated | Orderly, systematic and simple. |
| **10)** | Decoding & sequencing logic | Complex | Easier |
| **11)** | Applications | RISC µps. | CISC µps. |
| **12)** | Instruction set size | under 100 instructions | over 100 instructions. |
| **13)** | Control Memory (ROM) | Absent | Present |
| **14)** | Chip area required | More | Less |
| **15)** | Floating point functions | Cannot be realized efficiently. | Can be realized efficiently. |

**d) What is meaning of delayed branch and branch prediction? Write a difference between them. 5 marks**

***Meaning of delayed branch 1 marks and branch prediction 1 marks. 1 marks for each valid difference between them.***

**Delayed Branch:** A technique for minimizing the effect of control dependencies is to separate the point where the branch operation takes effect from the branch tests. The branch instruction performs a test on a branch condition. If the test succeeds, the PC is modified, but the modification does not take effect immediately. This delayed branch allows one or more instructions following the branch to be executed in the pipeline whether the branch is taken or not.

In the MIPS CPU, the branch operation is delayed by one instruction. The MAL assembler hides the delayed branch by inserting an instruction after each branch or jump. The instruction following a branch or jump is called the delay slot. By default the assembler inserts an instruction which does nothing, a no-op.

In the branch instruction, the PC was incremented when the branch was fetched and therefore the branch offset is relative to the instruction after the branch. The delayed branch means that the instruction following the branch is always executed before the PC is modified to perform the branch.

**Branch Predication:** In computer architecture, a branch predictor is a digital circuit that tries to guess which way a branch (e.g., an if–then–else structure) will go before this is known definitively. The purpose of the branch predictor is to improve the flow in the instruction pipeline. Branch predictors play a critical role in achieving high effective performance in many modern pipelined microprocessor architectures such as x86.

**Difference between Delayed branch and branch prediction:** Delayed branch and branch prediction are two different ways of mitigating the effects of a long execution pipeline. Without them, the pipeline needs to stall whenever a conditional branch is taken, because the instruction fetch mechanism can't know which instruction should be executed next after the branch instruction until the computations on which it depends are completed.

Delayed branch simply means that some number of instructions that appear *after* the branch in the instruction stream will be executed *regardless* of which way the branch ultimately goes. In many cases, a compiler can put instructions in those slots that don't actually depend on the branch itself, but if it can't, it must fill them with NOPs, which kills the performance anyway. This approach keeps the hardware simple, but puts a burden on the compiler technology.

Branch prediction is a more hardware-oriented approach, in which the instruction fetcher simply "guesses" which way the branch will go, executes instructions down that path, and if it later turns out to have guessed wrong, the results of those instructions are thrown away. Various systems have different ways of improving the accuracy of the guess. Sometimes the compiler puts a clue into the instruction stream, and sometimes the hardware keeps track of which way each branch has gone in the past.
Delayed branch executes instructions without branch and dependency after branch to avoid wasting cycles during branch and target address calculation. Branch prediction is to assume that a branch is not-taken (static branch prediction) or to predict based on history (dynamic branch prediction) to execute instructions after the branch according to the prediction. There is a difference between executing a part irrelevant to a branch and predicting a branch to execute according to prediction.

**e) Draw and explain instruction cycle state diagram. 5 Marks**

***Maximum 2 Marks for neat and correct diagram. 1 marks for brief description of each cycle state.***

## Instruction Cycle State Diagram:

Figure provides a more detailed look at the basic instruction cycle. The figure is in the form of a state diagram. For any given instruction cycle, some stales may be null and others may be visited more than once. The states can be described as follows:

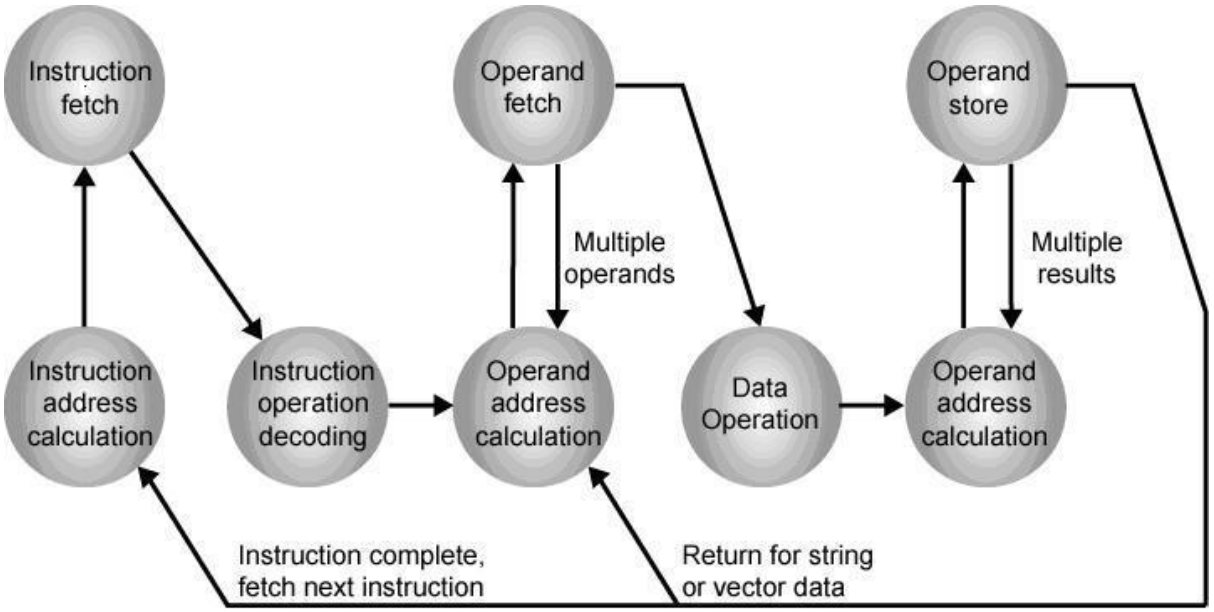



Figure : Instruction cycle state diagram

- Instruction address calculation (iac): Determine the address of the next instruction to be executed. Usually, this involves adding a fixed number to the address of the previous instruction. For example, if each instruction is 16 bits long and memory is organized into 16-bit words, then add 1 to the previous address. If, instead, memory is organized as individually addressable 8-bit bytes, then add 2 to the previous address.
- Instruction fetch (if): Read instruction from its memory location into the processor.
- Instruction operation decoding (iod): Analyse instruction to determine type of operation to he performed and operand(s) to be used.
- Operand address calculation (oac): If the operation involves reference to an operand in memory or available via I/O. then determine the address of the operand.
- Operand fetch (of): Fetch the operand from memory or read it in from I/O,
- Data operation (do): Perform the operation indicated in the instruction.
- Operand store (os): Write the result into memory or out to I/O

**f) Multiply (-10) and (-8) using Booth's algorithm. 5 Marks**

***Full step by step process for result carries 5 marks. Depending on the number of correct steps to results may award 1 mark to each evaluation step.***

- Multiplicand □ M□$(-10)_{10}$ □ $(10110)_2$ in 2's complement sign magnitude
- so, -M □ $(01010)_2$ in 2's complement sign magnitude
- Multiplier □Q□ $(-8)_{10}$ □ $(11000)_2$ in 2's complement sign magnitude
- Total 5 bit presentation, so counter□cnt□5

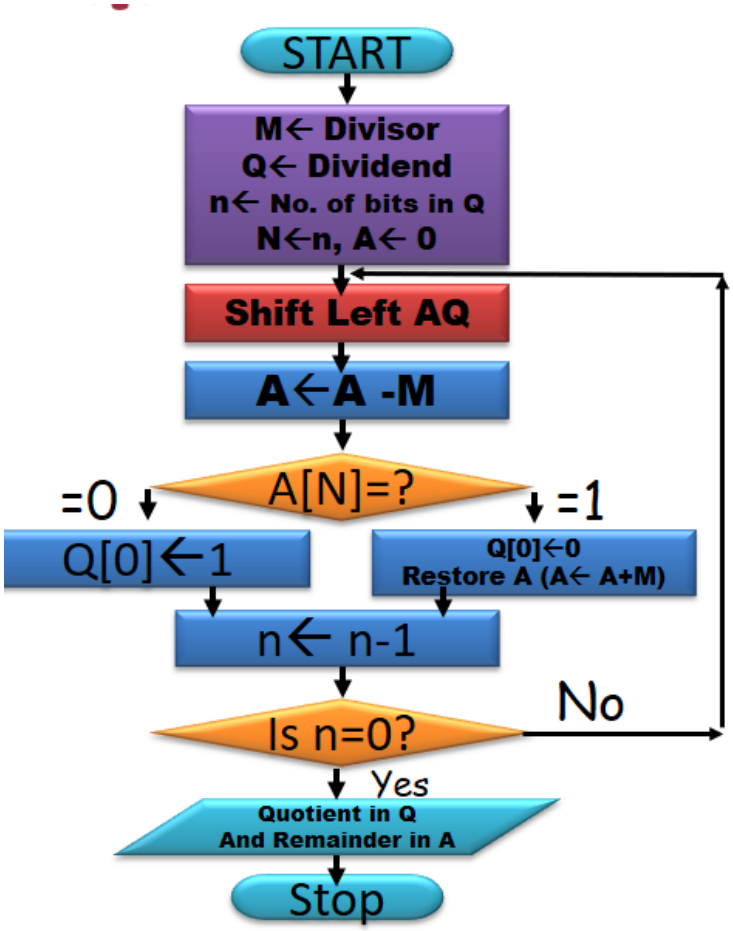
| Down Cnt | Test $Q_0Q_{-1}$ | Action | Accumulator (A) | Multiplier (Q) $Q_4Q_3Q_2Q_1Q_0$ | $Q_{-1}$ |
|---|---|---|---|---|---|
| 5 | -- | Initialization of A□0, Q□ Multiplier, $Q_{-1}$□0, Cnt-->5 | 00000 | 11000 | 0 |
| 4 | 00 | Arithmetic Shift Right | 00000 | 01100 | 0 |
| 3 | 00 | Arithmetic Shift Right | 00000 | 00110 | 0 |
| 2 | 00 | Arithmetic Shift Right | 00000 | 00011 | 0 |
| 1 | 10 | A□ A-M<br>A□A+ (-M)<br>00000<br>+ 01010<br>01010 | 01010 | 00011 | 0 |
| | | Arithmetic Shift Right | 00101 | 00001 | 1 |
| 0 | 11 | Arithmetic Shift Right | 00010 | 10000 | 1 |

Final product is in A & Q register in the form of 2's complement.
Result = 00010 10000 & MSB is 0, so number is positive and in the form of 2's complement.
Therefore $(= 00010\ 10000)_2 = (-80)_{10}$.

## Q.3 Solve any One out of Two.

**a) Draw the flowchart of Restoring Division Algorithm & perform 10 /3 using this Algorithm. 10 Marks**

***For flow chart allot 4 marks and for performing the mathematical task of 10/3 allot 6 marks***




10 (Dividend) / 3 (Divisor) = 3 (Quotient) 1 (Remainder)

M=$(3)_{10}$=$(00011)_2$, -M=11101, Q=Dividend=$(10)_{10}$=$(1010)_2$

| n | M | A | Q | Action |
|---|---|---|---|---|
| 4 | 00011 | 00000 | 1010 | Initialization |
| | 00011 | 00001 | 010? | Shift Left AQ |
| | 00011 | **1**1110 | 010? | A□A-M (00001+11101=11110) |
| | 00011 | **1**1110 | 0100 | A[N]=1, so Q[0]□0 (the most significant bit of the A is checked if it is 0 the least significant bit of Q is set to 1 otherwise if it is 1 the least significant bit of Q is set to 0 and value of register A is restored) |
| | 00011 | 00001 | 0100 | Restore A; n=4-1=3 |
| 3 | 00011 | 00010 | 100? | Shift Left AQ |
| | 00011 | 11111 | 100? | A□A-M (00010+11101=11111) |

| | | | | |
|---|---|---|---|---|
| | 00011 | **1**1111 | 1000 | A[N]=1, so Q[0]⍰0, |
| | 00011 | 00010 | 1000 | Restore A; n=3-1=2 |
| 2 | 00011 | 00101 | 000? | Shift Left AQ |
| | 00011 | 00010 | 000? | A⍰A-M (00101+11101=00010) |
| | 00011 | 00010 | 0001 | A[N]=0, so Q[0]⍰1; n=2-1=1 |
| 1 | 00011 | 00100 | 001? | Shift Left AQ |
| | 00011 | 00001 | 001? | A⍰A-M (00100+11101=00010) |
| | 00011 | 00001 | 0011 | A[N]=0, so Q[0]⍰1; n=1-1=0 |

**B) Explain with suitable diagrams - Flynn's Classification of Computer Architecture.**

**10 Marks**

***For what is Flynn's classification 1 Marks. 1 marks for covering suitable two-three points on each classification and 1 marks for suitable diagram of each classification. 1 marks for overall examples.***