SUGGESTIONS FOR SEMESTER EXAM

SUBJECT : COMPUTER SYSTEM ORGANIZATION (CST 207)

A. Choose the correct answer from the given alternatives : (1 Marks)

1. The three main components of a digital computer system are :

   (a) Memory, IO, DMA (b)ALU, CPU, Memory (c) CU, ALU, Register (d) CPU, Memory, IO

   ANSWER) (d) CPU, Memory, IO.

   EXPLANATION :

   The three main components of a digital computer system are the Central Processing Unit (CPU), memory, and input/output (IO) devices.

2. The second generation of computer used :

   (a) transistors (b) IC (c) vacuum tube (d) LSI.

   ANSWER) (a) transistors**.**

   EXPLANATION :

   The second generation of computers replaced vacuum tubes with transistors, which were smaller, more reliable, and energy-efficient.

3. Processors of all computers, whether micro, mini or mainframe must have :

   (a) ALU (b)Primary storage (c) Control Unit (d) all of these.

   ANSWER) (d) all of these**.**

   EXPLANATION :

   All processors require an ALU for calculations, primary storage for temporary data, and a control unit to manage operations.

4. In Which addressing mode is operand specified in the instruction itself ?

   (a) Register mode (b) Immediate mode (c) Direct Address mode (d) Index Addressing mode.

   ANSWER) (b) Immediate mode**.**

   EXPLANATION :

   In Immediate mode, the operand is directly specified in the instruction itself.

5. The instruction LOAD is a :

   (a) zero-address instruction (b) one-address instruction (c) two- address instruction (d) three-address instruction.

   ANSWER) (b) one-address instruction**.**

   EXPLANATION :

   The LOAD instruction typically involves one address to specify the memory location of the data to be loaded into a register.

6. The number of fetch operation to execute instruction in immediate mode is :

   (a) 0 (b) 1 (c) 2 (d) none of these.

   ANSWER) (b) 1**.**

EXPLANATION :

In Immediate mode, the operand is part of the instruction, so only one fetch operation is needed to retrieve the instruction.

7. The instruction 1111 111100001100 is a :

   (a) direct memory reference instruction (b) indirect memory reference instruction

   (c) register reference instruction (d) input outputinstruction.

   ANSWER) (c) register reference instruction

   EXPLANATION :

   The binary format indicates a direct memory reference, as it specifies the memory address directly for the operation.

8. 01110000 represents :

   (a) 0 (b) NaN (c) +∞ (d) -∞.

   ANSWER) (b) NaN

   EXPLANATION :

   In IEEE 754 format, 01110000 represents a positive infinity (+∞) value due to the combination of exponent and mantissa bits.

9. The largest floating point number that can be represented by 8 bit is :

   (a) 01111111 (b) 11111111 (c) 01101111 (d) 01111110.

   ANSWER) (a) 01111111**.**

   EXPLANATION : In an 8-bit floating-point representation, the largest number is achieved by setting all the exponent and fraction bits to 1, except for the sign bit (which must be 0 for positive numbers). "01111111" represents the maximum value under these conditions.

10. If n is number of bits in exponent, the bias number can be calculated as :

    (a) 2^n-1 (b) 2^n (c) 2^{n-1} (d) 2^{n-1}-1.

    ANSWER) (d) 2^{n-1}-1.

    EXPLANATION : The bias in a floating-point system is calculated as 2^{n-1}-1, where n is the number of bits in the exponent. This bias allows both positive and negative exponents to be represented using only non-negative numbers.

11. In Booth's algorithm, if the multiplier has n bits then the multiplicand should have :

    (a) 1 bit (b) n bits (c) n+1 bits (d) 2n bits.

    ANSWER) (d) 2n bits**.**

    EXPLANATION : In Booth's algorithm, the multiplicand is typically extended to 2n2n2n bits, where nnn is the number of bits in the multiplier. This extension ensures that the product, which can be as large as twice the size of the multiplier, is properly represented.

12. k-way set associative means :

    (a) k blocks are present in a set (b) k sets are present in ablock (c) k sets are present in the cache (d) none of these.

    ANSWER) (a) k blocks are present in a set**.**

EXPLANATION :

In a k-way set associative cache, each set contains kkk blocks (or slots) where data can be stored. This allows for more flexibility in storing data compared to direct-mapped caches, which have only one block per set.

13. A typical modem computer uses :

    (a) LSI chip (b) VLSI chip (c) valves (d) vacuum tube.

    ANSWER) (b) VLSI chip**.**

    EXPLANATION :

    A typical modern computer uses Very Large Scale Integration (VLSI) chips, which allow thousands or even millions of transistors to be integrated onto a single chip, providing the processing power needed for modern computing tasks.

14. A subtractor is not usually present in a computer because :

    (a) it is expensive (b) it is not possible to design it (c) the adder will take care of subtraction (d) none of the above.

    ANSWER) (c) the adder will take care of subtraction**.**

    EXPLANATION :

    Subtraction can be performed using an adder by converting the subtraction operation into addition. This is done by using the two's complement of the number to be subtracted, so computers typically use the adder to handle both addition and subtraction.

15. The instruction 0101 111100001100 is a :

    (a) direct memory reference instruction (b) indirect memory reference instruction (c) register reference instruction (d) input output instruction.

    ANSWER) (a) direct memory reference instruction**.**

    EXPLANATION :

    In a direct memory reference instruction, the instruction format typically includes the operation code (opcode) followed by a direct address in memory. The binary instruction "0101 111100001100" indicates an opcode (in the first part) and a direct address (in the latter part), meaning it refers directly to a memory location.

16. In auto increment addressing mode, the value of the register is incremented by 1 _____ the execution of the instruction :

    (a) before (b) after (c) during (d) not in the list.

    ANSWER) (b) after**.**

    EXPLANATION :

    In auto-increment addressing mode, the value of the register is incremented **after** the execution of the instruction. This means that the operand is fetched using the current value of the register, and then the register is incremented for the next use.

17. If we want an addition/subtraction circuit to do subtraction, the initial value of carry in should be :

    (a) -1 (b) 0 (c) 1 (d) 2.

    ANSWER) (c) 1**.**

    EXPLANATION :

To perform subtraction using an addition circuit, we need to use the two's complement of the number to be subtracted. This is achieved by setting the carry-in to 1, which allows the adder to add the two's complement of the subtracted value, effectively performing subtraction.

18. Virtual memory is :

    (a) primary memory (b) secondary memory (c) both a & b (d) none of these.

    ANSWER) (c) both a & b**.**

    EXPLANATION :

    Virtual memory is a memory management technique that uses both primary memory (RAM) and secondary memory (like hard drives or SSDs). It allows programs to access more memory than physically available by swapping data between primary and secondary memory.

19. Direct mapping is :

    (a) one to one mapping (b) many to many mapping (c) many to one mapping (d) all of these.

    ANSWER) (a) one to one mapping**.**

    EXPLANATION :

    In direct mapping, each block of memory is mapped to exactly one cache line, creating a one-to-one relationship between memory locations and cache slots. This means that a specific memory address always maps to a specific cache line.

20. In associative mapping technique, number of comparators required is :

    (a) 1 (b) 2 (c) equal to the number of blocks in main memory (d) equal to the number of lines in cache memory.

    ANSWER) (d) equal to the number of lines in cache memory**.**

    EXPLANATION :

    In associative mapping (also known as fully associative mapping), any block of memory can be placed in any cache line. Therefore, to check if a memory block is present in the cache, each cache line must be compared with the memory address. This requires as many comparators as there are cache lines.

21. Which of the following comment about the IR is true ?

    a) It is used to count the number of insouctions b) It is a cell in ROM c) During execution of the current instructions*, its* content changes d) Opcode fetched from memory stored in IR.

    ANSWER) (d) Opcode fetched from memory stored in IR.

    EXPLANATION :

    The Instruction Register (IR) stores the opcode of the instruction that is currently being executed. When an instruction is fetched from memory, the opcode is loaded into the IR for decoding and execution. The content of the IR changes with each new instruction being executed, but it does not count instructions or reside in ROM.

22. The speed imbalance between memory access & CPU operation can be reduced by :

    a) Cache memory b) memory interleaving c) reducing memory size d) virtual memory.

    ANSWER) (a) Cache memory**.**

    EXPLANATION :

    Cache memory is a small, high-speed memory located close to the CPU that stores frequently accessed data. By providing faster access to this data, cache memory helps reduce the speed imbalance between

memory access and CPU operation, as the CPU can access data from the cache much faster than from main memory.

23. The sequence of events that happen during a typical fetch operation is
a) PC—>MDR—>Memory—>IR b) PC—>MAR—>MEMORY—>MDR—>IR c) PC—>MEMORY—>IR d) MAR—>MDR—>IR.
ANSWER) (b) PC → MAR → MEMORY → MDR → IR**.**
EXPLANATION :
In a typical fetch operation:

1. The **Program Counter (PC)** holds the address of the next instruction to be fetched.
2. The address from the PC is loaded into the **Memory Address Register (MAR)**.
3. The memory is accessed, and the instruction at that address is placed in the **Memory Data Register (MDR)**.
4. Finally, the instruction from the MDR is loaded into the **Instruction Register (IR)** for decoding and execution.

24. Negative numbers can be represented in :
a) Sign-magnitude form b) 1's complemented form, c) 2's complemented form d) All of the above.
ANSWER) (d) All of the above.
EXPLANATION :
Negative numbers can be represented in sign-magnitude, 1's complement, and 2's complement forms, each using different methods to encode the sign and value.

25. The instruction ADD is_____address instruction :
a) 0 b) 1 c) 2 d) 3.
ANSWER) (b) 1**.**
EXPLANATION :
ADD is a **1-address instruction**, meaning it typically involves one operand, with the result being stored in a register.

26. The exponent of a floating point number is represented in excess-N code (biased) so that :
a) the dynamic range is large b) the precision is high c) the smallest number is represented by all zeroes d) overflow is avoided
ANSWER) (a) the dynamic range is large**.**
EXPLANATION :
Excess-N (biased) representation allows for a wide dynamic range by adjusting the exponent to represent both very large and very small numbers efficiently.

27. If negative numbers are stored in 2's complemented form, the range of numbers that can be stored in 8 bits is :
a) -128 to +128 b) -128 to +127 c) -127 to +128, d) -127 to +127.
ANSWER) (b) -128 to +127**.**
EXPLANATION :
In 2's complement representation, the range for 8-bit numbers is from -128 to +127, with 1 bit for the sign and 7 bits for the value.

28. The cost of storing a bit is minimum in :

a) Cache memory b) Register c) RAM d) Hard disk.

ANSWER) (d) Hard disk**.**

EXPLANATION :

The cost of storing a bit is least in hard disk storage compared to faster but more expensive options like cache, registers, and RAM.

29. Which of the following statement(s) is true ?

a) ROM is a read/write memory b) PC points to the last instruction that was executed c) Stack works on the principle of FIFO d) RAM is a Read/Write memory.

ANSWER) (d) RAM is a Read/Write memory.

EXPLANATION :

RAM is a Read/Write memory, meaning data can be both read from and written to it, unlike ROM, which is read-only.

30. Tera is 2 to to the power of :

a) 9 b) 20 c) 30 d) 40.

ANSWER) (d) 40**.**

EXPLANATION :

In binary, "Tera" refers to 2^{40}, representing a large quantity of data (1 TB = 2^{40} bytes).

31. In immediate addressing the operand is placed :

a) in the CPU register b) after OP code in the instruction c) in memory d) in stack

ANSWER) (b) after OP code in the instruction**.**

EXPLANATION :

In immediate addressing, the operand is directly specified in the instruction, following the opcode.

32. The number successful accesses to memory stated as a fraction is called as _____.

a) Access rate b) Success rate c) Hit rate d) Miss rate

ANSWER) (c) Hit rate**.**

EXPLANATION :

The hit rate refers to the fraction of successful memory accesses, where the requested data is found in the cache or memory.

33. The final addition sum of the numbers, 0110 & 0110 is _____.

a) 1101 b) 1111 c) 1100 d) 1010

ANSWER) c) 1100

EXPLANATION :

Adding the binary numbers 0110 and 0110 results in 1100:

```
  0110
+ 0110
------
  1100
```

34. What does CSA stands for?

a) Computer Service Architecture b) Computer Speed Addition c) Carry Save Addition d) None of these

ANSWER) (c) Carry Save Addition**.**

EXPLANATION :

CSA stands for Carry Save Addition, a technique used in binary addition to reduce the delay caused by carrying over during addition, commonly used in multipliers and other arithmetic operations.

35. Individual control word of the micro routine are called as :
a) Micro task b)Micro instruction c)Micro operation d)Micro Command
ANSWER) (b) Micro instruction**.**
EXPLANATION :
Micro instructions are the individual control words in a microroutine, specifying a single operation in a microprogram for controlling the hardware.

36. Which of the following circuit convert the binary data into a decimal ?
a) Decoder b) Encoder c) Code converter d) Multiplexer
ANSWER) (c) Code converter**.**
EXPLANATION :
A code converter is a circuit that converts binary data into other number systems, such as decimal.

37. The situation wherein the data of operands are not available is called _____
a) Data hazard b) Stock c) Deadlock d) Structural hazard
ANSWER) (a) Data hazard**.**
EXPLANATION :
A data hazard occurs when the required data for operands is not yet available, often due to dependencies between instructions.

38. What is the full form of CISC ?
a) Complex Instruction Sequential Compilation b) Complete Instruction Sequential Compilation c) Computer Integrated Sequential Compiler d) Complex Instruction Set Computer
ANSWER) (d) Complex Instruction Set Computer**.**
EXPLANATION :
CISC stands for Complex Instruction Set Computer, a type of CPU architecture that uses a large set of instructions to perform complex operations in a single instruction cycle.

39. The alternate way of writing the instruction, ADD #5,R1 is :
a) ADD [5],[R1] b) ADDI 5,R1 c) ADDIME 5,[R1] d) There is no other way
ANSWER) (b) ADDI 5, R1**.**
EXPLANATION :
The instruction "ADD #5, R1" adds the immediate value 5 to the contents of register R1. An alternate way to write this in some assembly languages is "ADDI 5, R1", where "ADDI" stands for "Add Immediate".

40. In order to read multiple bytes of a row at the same time, we make use of :
a) Memory extension b) Cache c) Shift register d) Latch
ANSWER) (b) Cache**.**
EXPLANATION :
Cache memory allows for fast access to multiple bytes of data, often storing them in parallel to speed up access. It helps read multiple bytes from a row at the same time.

41. In full adders the sum circuit is implemented using _____.
a) And & OR gates b) NAND gate c) XOR d) XNOR
ANSWER) (c) XOR**.**
EXPLANATION :
In a full adder, the sum is computed using an XOR gate, as it produces the correct sum for binary addition.

42. Computer address bus is :
a) Unidirectional b) Bidirectional c) Multidirectional d) None of the above
ANSWER) (a) Unidirectional**.**
EXPLANATION :
The address bus is unidirectional, meaning it only carries addresses from the CPU to memory or other devices, and does not carry data in the reverse direction.

43. Which of the following computer bus connects the CPU to a memory on the system board ?
a) Expansion bus b) Width bus c) System bus d) None of the above
ANSWER) (c) System bus**.**
EXPLANATION :
The system bus connects the CPU to memory and other components on the system board, allowing communication between them.

44. The instructions that are used for reading an input port and writing an output port respectively are :
a) MOV, XCHG b) MOV, IN c) IN, MOV d) IN, OUT
ANSWER) (d) IN, OUT**.**
EXPLANATION :
The "IN" instruction is used for reading data from an input port, while the "OUT" instruction is used for writing data to an output port.

45. Micro operation is shown as :
a) RI<—R2 b) R1+R2 c) Both d) None
ANSWER) (a) RI ← R2**.**
EXPLANATION :
A micro operation represents a simple operation on a register, such as transferring data. "RI ← R2" indicates that the contents of register R2 are transferred to register RI.

46. An interrupt that can be temporarily ignored is :

a) Vectored interrupt b) Non-maskable interrupt c) Maskable interrupt d) High priority interrupt

ANSWER) (c) Maskable interrupt**.**

EXPLANATION :

A maskable interrupt can be temporarily ignored or "masked" by the CPU, allowing it to prioritize other tasks until the interrupt is enabled again.

47. Von Neumann architecture is based on _____.

a) SISD b) SIMD c) MISD d) MIMD

ANSWER) (a) SISD**.**

EXPLANATION :

Von Neumann architecture is based on Single Instruction Single Data (SISD), where a single instruction operates on a single data element at a time.

48. Example for zero address instructions is _____.

a) push b) load A c) move R1, A d) store x.

ANSWER) (a) push**.**

EXPLANATION :

Zero address instructions, like "push", do not explicitly specify operands in the instruction itself; instead, the operands are implicitly defined, typically using the stack.

49. What is the full form of TLB ?
a) Translation Loop Buffer b) Translation Look- aside Buffer c) Time Loop Block d) None
ANSWER) (b) Translation Look-aside Buffer**.**

EXPLANATION :
TLB stands for Translation Look-aside Buffer, which is a cache used to store recent translations of virtual memory addresses to physical memory addresses.

50. What is Page Map Table ?
    a. It maps the virtual addresses to physical addresses
    b. It maps physical addresses to virtual addresses
    c. It maps the virtual addresses to Logical addresses
    d. Support all the above
    ANSWER) (a) It maps the virtual addresses to physical addresses**.**
    EXPLANATION :
    A Page Map Table (PMT) is used in virtual memory systems to map virtual addresses to physical addresses, helping in the translation between them.

51. The result of MOV AL, 65 is to store :
    a) 0100 0010 in AL b) 42H in AL c) 40H in AL d) 0100 0001 in AL
    ANSWER) (d) 0100 0001 in AL**.**
    EXPLANATION :
    The instruction "MOV AL, 65" stores the ASCII value of 65 (which is 'A') in AL, represented in binary as 0100 0001.

52. Which of the following is page fault?
    a. Page fault occurs when a program accesses a page of another program
    b. Page fault occurs when a program accesses a page in main memory
    c. Page fault occurs when there is an error in particular page
    d. Page fault occurs when a program accesses a page which is not present in main memory
    ANSWER) (d) Page fault occurs when a program accesses a page which is not present in main memory**.**
    EXPLANATION :
    A page fault happens when a program tries to access a page that is not currently loaded into main memory, triggering a memory management operation to load it.

53. Which of the following are the two main components of the CPU ?
    a) CU and registers b) Registers and main memory c) CU and ALU d) Registers and ALU.
    ANSWER) (c) CU and ALU**.**
    EXPLANATION :
    The two main components of the CPU are the **Control Unit (CU)**, which manages instructions and control signals, and the **Arithmetic Logic Unit (ALU)**, which performs calculations and logic operations.

54. The term that provides simultaneous data processing tasks are _____.
    a) parallel processing b) array processing c) vector processing d) distributed processing.
    ANSWER) (a) parallel processing**.**
    EXPLANATION :
    Parallel processing involves performing multiple data processing tasks simultaneously, using multiple processors or cores to enhance computational speed and efficiency.

55. A 16 x 8 Organisation of memory cells, can store upto _____
    a) 256 bits b) 1024 bits c) 512 bits d) 128 bits
    ANSWER) (a) 128 bits**.**
    EXPLANATION :
    In a 16 x 8 memory organization, there are 16 rows and 8 bits per row. Therefore, the total storage is 16×8=128 bits.

56. A processor can access a memory location by 32 bits. Then find the total memory size if all memory locations are available to the processor.
    a) 4 GB
    b) 4 MB
    c) 2 GB
    d) 4 Gb
    ANSWER) (a) 4 GB**.**
    EXPLANATION :
    If a processor can access a memory location by 32 bits (4 bytes), and all memory locations are available, the total memory size is calculated as: 2^{32} locations × 4 bytes = 4 GB.

57. Data transfer from Cache Memory to Processor is _____
    a) Word by Word b) Block by Block c) Block by Word d) Word by Block
    ANSWER) (b) Block by Block**.**
    EXPLANATION :
    Data transfer from Cache Memory to Processor typically occurs in blocks (also called cache lines), not word by word, to optimize efficiency in data retrieval.

58. In which of the following term the performance of cache memory is measured ?
    a. Chart ratio
    b. Hit ratio
    c. Cache ratio
    d. Data ratio
    ANSWER) (b) Hit ratio**.**
    EXPLANATION :
    The performance of cache memory is measured by the **hit ratio**, which indicates the fraction of memory accesses that are successfully found in the cache.

59. Which of these is NOT involved in the case of a memory write operation ?
    a) Data bus b) MDR c) MAR d) PC
    ANSWER) (d) PC**.**
    EXPLANATION :
    The Program Counter (PC) is not involved in a memory write operation; instead, the Data Bus, Memory Data Register (MDR), and Memory Address Register (MAR) are used for accessing and writing data to memory.

60. Which of the following memory unit communicates directly with the CPU?
    a. Auxiliary memory
    b. Main memory
    c. Secondary memory
    d. None of the above
    ANSWER) b. Main memory**.**
    EXPLANATION :
    Main memory (RAM) communicates directly with the CPU, providing quick access to data and instructions that the CPU needs for processing.

61. A computer system supports 2^{46} logical addresses and 2^{11} addresses per page. How many pages can be represented in secondary memory (virtual memory address space) ?
    a) 35 b) 2^{35} c) 2^{25} d) 257
    ANSWER) (a) 35.

EXPLANATION :
To find the number of pages, we divide the total number of logical addresses by the number of addresses per page.

2^{46} logical addresses / 2^{11} addresses per page = 2^{35} pages.

Thus, 35 pages can be represented in the secondary memory.

B. Fill in the blanks : (1 Marks)

1) _____ is used to store data, instructions and results permanently for future use.
ANSWER) Storage (or secondary storage)
2) _____ is generally used to increase the apparent size of physical memory.
ANSWER) Virtual memory
3) Gray Code is also called as _____.
ANSWER) Reflected Binary Code
4) Instruction register stores _____.
ANSWER) The current instruction being executed
5) A high speed memory is placed between the CPU and the primary memory is known as _____.
ANSWER) Cache memory
6) I/O address in 8086 is _____ bits.
ANSWER) 16 bits
7) Techniques that automatically move programs and data blocks into the physical memory when they are required for execution are called _____.
ANSWER) Paging and Segmentation
8) Hit ratio is maximum in _____ mapping.
ANSWER) Associative mapping
9) The bias value for single-precision floating point numbers is _____.
ANSWER) 127**.**
10) MOV AX, [2A50] is an example of _____ addressing mode.
ANSWER) direct addressing mode**.**
11) Loop unrolling is a technique to improve _____.
ANSWER) performance**.**
12) Page table resides in _____.
ANSWER) main memory (RAM)**.**
13) Microinstruction consists of _____.
ANSWER) control bits
14) The smallest entity of memory is called _____.
ANSWER) bit**.**
15) A source program is usually in _____ language.
ANSWER) high-level programming

C) Answer The Following Questions : (2 Marks)

1) What is a super computer ?
ANSWER) A **supercomputer** is a high-performance computing machine designed to perform complex and resource-intensive tasks at extremely fast speeds. It is capable of handling vast amounts of data and performing billions or even trillions of calculations per second. Supercomputers are typically used for tasks such as climate modeling, scientific simulations, cryptography, and research in physics, chemistry, and biology.
Supercomputers consist of thousands of processors working in parallel to achieve high computational power. Their performance is measured in FLOPS (Floating Point Operations Per

Second), and they are often used by governments, research institutions, and large enterprises.
In summary:

- **Supercomputers** are highly powerful systems designed for specialized, large-scale computations.
- They are used in research areas like simulations, weather forecasting, and scientific analysis.

2) How does a computer differ from a calculator ?

ANSWER) A **computer** and a **calculator** are both devices used for performing mathematical operations, but they differ significantly in terms of their capabilities and functions.

1. **Functionality**:
   - A **calculator** is a device designed primarily for performing basic arithmetic operations like addition, subtraction, multiplication, and division. Some advanced calculators may also handle more complex mathematical functions such as trigonometry or logarithms.
   - A **computer**, on the other hand, is a versatile device capable of performing a wide range of tasks, including not only arithmetic calculations but also data processing, running software applications, internet browsing, and handling multimedia tasks like video editing or gaming.
2. **Complexity and Processing Power**:
   - A **calculator** has a limited processing capability focused mainly on mathematical calculations.
   - A **computer** has a much more powerful processor and can execute a variety of operations, from complex algorithms to managing operating systems and running multiple applications simultaneously.

In summary:

- A **calculator** is specialized for mathematical calculations, while a **computer** is a general-purpose device capable of performing a wide range of tasks beyond just calculations.

3) What is bus ?

ANSWER) A **bus** in a computer system is a communication pathway used to transfer data and control signals between different components, such as the **CPU**, **memory**, and **input/output devices**. It consists of a set of physical wires or traces on a motherboard, through which information is passed.
There are typically three types of buses:

1. **Data Bus**: Transports data between the CPU, memory, and other components.
2. **Address Bus**: Carries the memory addresses specifying the location where data is to be read from or written to.
3. **Control Bus**: Carries control signals that coordinate and manage the operations of the CPU and other components.
   In summary:
   A **bus** is a system of pathways used for transferring data and instructions between different parts of the computer, enabling communication and coordination between the CPU, memory, and other devices.

4) What do you mean by Effective Address?

ANSWER) The **Effective Address (EA)** refers to the final memory location or address where data is stored or fetched during an instruction execution. It is calculated based on the addressing mode used in the instruction.
In most cases, the **Effective Address** is computed by combining the contents of various registers (such as the **base register** or **index register**) with an offset or displacement. The calculation of the EA depends on the type of addressing mode being used, such as **direct**, **indirect**, **indexed**, or **base-register** addressing.
For example:

- In **indexed addressing mode**, the effective address is calculated by adding the contents of the index register to a specified base address.
- In **indirect addressing mode**, the effective address is obtained by using the value in a register or memory location as a pointer to the final address.
  In summary:

- The **Effective Address** is the actual address in memory used to access data, determined by the addressing mode and calculation based on registers and offsets.

What is Program Counter?
ANSWER) The **Program Counter (PC)** is a special register in the CPU that holds the memory address of the next instruction to be executed. It is also referred to as the **instruction pointer** in some architectures.
Key Points:

1. **Function**: The Program Counter keeps track of the sequence of instructions in a program. After an instruction is fetched from memory, the Program Counter is automatically incremented (or modified in case of jumps or branches) to point to the next instruction.
2. **Control Flow**: It plays a crucial role in controlling the flow of execution. If a branch or jump instruction is encountered, the value of the Program Counter may be modified to the target address, allowing the program to execute non-sequentially.
3. **Operation**:
   - Initially, the PC holds the address of the first instruction.
   - After each instruction is fetched, the PC is updated to point to the next instruction.
   - In case of jumps or branches, the PC is updated to the new address specified by the instruction.

In summary:

- The **Program Counter** is responsible for keeping track of the address of the next instruction to be executed, ensuring the CPU follows the correct sequence of instructions.

5) What is I/O processor?
ANSWER) An I/O processor (Input/output processor) is a specialized processor used to manage the data flow between the CPU and peripheral devices, such as disks, keyboards, and printers. It offloads the CPU from routine I/O operations, allowing for more efficient data transfer and processing.

6) Write full form of RISC and CISC.
ANSWER)

- **RISC**: Reduced Instruction Set Computer
- **CISC**: Complex Instruction Set Computer

7) Write the IEEE format for single precision number.
ANSWER) The IEEE 754 single-precision format consists of 32 bits: 1 bit for the sign (S), 8 bits for the exponent (E), and 23 bits for the fraction (F). The format is:

```
S | Exponent | Fraction
```

8) Why is biased exponent used in computation?
ANSWER) A biased exponent is used to handle both positive and negative exponents uniformly in floating-point representation. It simplifies comparison and storage by representing negative exponents with positive values.

9) What is L2 cache?
ANSWER) L2 cache (Level 2 cache) is a memory cache located between the CPU and main memory (RAM). It is faster than main memory and stores frequently accessed data to improve the CPU's performance by reducing memory access time.

10) What is Opcode?
ANSWER) An Opcode (Operational Code) is part of a machine language instruction that specifies the operation to be performed, such as addition, subtraction, or data movement.

11) What is the minimization of pipeline?
ANSWER) Pipeline minimization refers to techniques aimed at optimizing the use of the pipeline stages to improve the throughput and reduce the latency of the processor by ensuring each stage operates as efficiently as possible.

12) What is write-through policy?
ANSWER) The write-through policy is a cache management technique where data written to the cache is also immediately written to the main memory, ensuring that the cache and memory are always synchronized.

13) What is TLB?
ANSWER) TLB (Translation Lookaside Buffer) is a cache used to store recent translations of virtual addresses to physical addresses, improving the speed of virtual memory access by reducing the time to access page tables.

14) What do you mean by vectored interrupt?
ANSWER) A vectored interrupt is an interrupt in which the address of the interrupt service routine (ISR) is provided directly by the interrupting device, allowing the CPU to quickly determine where to handle the interrupt.

15) What do you mean by Loop Buffer?
ANSWER) A loop buffer is a small, high-speed memory buffer used in computer systems to store the loop instructions, improving the efficiency of executing loops by reducing the need for repeated memory accesses.

16) What do you mean by Replacement Algorithm?
ANSWER) A replacement algorithm is used in cache memory or virtual memory systems to determine which data to replace when the memory is full. Common algorithms include Least Recently Used (LRU), First-In-First-Out (FIFO), and Optimal Replacement.

17) How control unit controls other units?
ANSWER) The control unit (CU) generates control signals that direct the operation of the CPU, including the ALU, registers, and memory. It fetches, decodes, and executes instructions by activating the appropriate components in the system.

18) Give an example of a 4-bit, 8-bit, 16-bit, and 32-bit microprocessor.
ANSWER)

- 4-bit: Intel 4004
- 8-bit: Intel 8080
- 16-bit: Intel 8086
- 32-bit: Intel 80386

19) What is MAR and MDR?
ANSWER)

- **MAR (Memory Address Register)**: Holds the address of the memory location to be accessed.
- **MDR (Memory Data Register)**: Holds the data being transferred to or from the memory.

20) What is register?
ANSWER) A register is a small, fast storage location within the CPU used to store data, addresses, or control information temporarily during processing.

21) What is interrupt?
ANSWER) An interrupt is a signal that temporarily halts the CPU's current operations to give priority to another task or event. After the interrupt is serviced, normal execution resumes.

22) What is non-volatile memory?
ANSWER) Non-volatile memory is a type of memory that retains stored data even when power is turned off. Examples include ROM, flash memory, and EEPROM.

23) What is logical address?
ANSWER) A logical address, also known as a virtual address, is an address generated by the CPU during a program's execution. It is mapped to a physical address in memory.

24) Which is an error-detecting code?
ANSWER) An error-detecting code is a method used to identify errors in data transmission or storage. Common examples include parity bits and CRC (Cyclic Redundancy Check).

25) What is the logic shift?
ANSWER) A logic shift is a bit manipulation operation that shifts bits to the left or right, filling the empty positions with zeros. It is commonly used for tasks like multiplying or dividing by powers of two.

26) What type of device converts digital signal into a form that is intelligible to the user?
ANSWER) A device that converts digital signals into a form understandable by the user is known as a **Digital-to-Analog Converter (DAC)**, such as speakers or display screens.

27) Which memory stores instruction which is required to start a computer?
ANSWER) The memory that stores instructions required to start the computer is known as **ROM (Read-Only Memory)**, particularly **BIOS** or **firmware**.

28) Define clock rate.
ANSWER) Clock rate is the frequency at which a processor's clock generates pulses, determining the speed at which the CPU performs operations. It is typically measured in Hertz (Hz).

29) What is the RAID system?
ANSWER) RAID (Redundant Array of Independent Disks) is a storage technology that combines multiple hard drives to improve performance, redundancy, or both. Common RAID levels include RAID 0, RAID 1, and RAID 5.

30) What are the three main elements of the control unit?
ANSWER) The three main elements of the control unit are:

1. **Decoder**: Decodes the instruction to determine the operation.
2. **Sequencer**: Controls the sequence of operations.
3. **Control Signals Generator**: Generates signals to activate different parts of the CPU.

31) What is Cache memory?
ANSWER) Cache memory is a small, high-speed memory located between the CPU and main memory. It stores frequently accessed data to reduce access time and improve processing speed.

32) What is control memory address?
ANSWER) Control memory address refers to the address used by the control unit to fetch control information, which dictates the operation of various CPU components during instruction execution.

33) What is the 2's complement representation of -6?
ANSWER) The 2's complement representation of -6 in 8-bit form is **11111010**.

34) What is clock signal in COA?
ANSWER) The clock signal in the context of Computer Organization and Architecture (COA) is a

periodic signal that synchronizes the timing of all operations within the CPU, ensuring orderly execution of instructions.

35) Is USB a bus?
ANSWER) Yes, **USB (Universal Serial Bus)** is a bus standard that allows peripheral devices to communicate with a computer. It serves as both a data transfer interface and a power supply for connected devices.

36) Draw the block diagram of the half adder.
ANSWER) The half adder consists of two inputs (A, B), an XOR gate for the sum output, and an AND gate for the carry output.

37) Draw a multiplication circuit diagram.
ANSWER) A multiplication circuit typically uses adders and logic gates to perform multiplication of binary numbers. It may involve multiple stages, including partial products and accumulation of results.

38) What's the difference between interrupt service routine and subroutine?
ANSWER) An **interrupt service routine (ISR)** is a special routine that handles interrupts and is triggered automatically by the system. A **subroutine** is a general-purpose code block that can be called by the main program or other routines.

39) What do you mean by the write-back policy?
ANSWER) In the **write-back policy**, data is written to the main memory only when it is replaced in the cache, rather than immediately when modified, to reduce memory write operations.

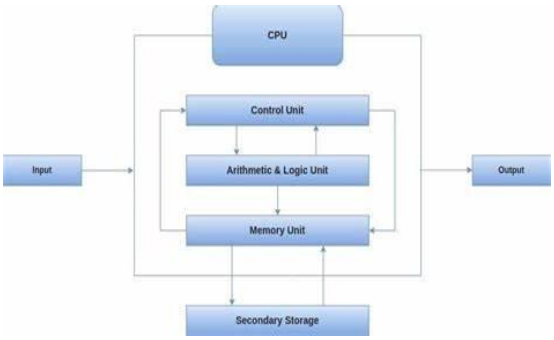
40) What is RISC Pipeline?
ANSWER) A RISC pipeline refers to the sequence of stages through which instructions pass in a Reduced Instruction Set Computer (RISC) architecture. Each stage performs a specific operation, improving instruction throughput.

41) What size of MUXs are needed?
ANSWER) The size of a **Multiplexer (MUX)** depends on the number of inputs and the width of the data being multiplexed. For example, a 4-to-1 MUX has 4 data inputs and 2 selection lines.

C. Long Answer Type Questions : (10 Marks)

1) Draw a block diagram of CPU and explain the function of each part. What are the functions of address bus,data bus and control bus ?
ANSWER) Block Diagram of CPU: The CPU (Central Processing Unit) is the core part of a computer that performs all the processing tasks. The main components of a CPU are:

1. **Control Unit (CU):** It coordinates and manages the operations of the CPU. It fetches instructions, decodes them, and signals the ALU or other components to execute the instructions. It also manages the flow of data within the CPU and between the CPU and memory.
2. **Arithmetic and Logic Unit (ALU):** The ALU performs all arithmetic and logical operations, such as addition, subtraction, multiplication, AND, OR, etc. It operates on the data provided by the CPU's registers.
3. **Registers:** These are small, fast storage locations used to store intermediate data, instructions, or results. They are located inside the CPU and used during instruction execution.



Bus System (Address, Data, and Control): The CPU communicates with memory and I/O devices through buses. These buses include:

- **Address Bus:** Transports the address of the memory location where data is to be read or written. It is unidirectional and defines the size of the memory that the CPU can access.
- **Data Bus:** Carries the actual data between the CPU, memory, and I/O devices. It is bidirectional and typically operates at a higher bandwidth.
- **Control Bus:** Transfers control signals from the control unit to coordinate operations between the CPU and other components. It is bidirectional.

2) What do you understand by Instruction Format ? Write two address instruction code to execute X=(A+B)*(C+D). What is Accumulator ?
ANSWER) Instruction Format: An instruction format is the layout that specifies the arrangement of fields within an instruction, such as the opcode (operation code), operand addresses, and sometimes additional control bits. It defines how an instruction is encoded to allow a CPU to fetch, decode, and execute it.

Two-address Instruction: In a two-address instruction set, there are two operands, and typically, one operand acts as both the source and the destination. For the expression `X = (A + B) * (C + D)`:

1. `ADD A, B` (Adds A and B, and stores the result in A)
2. `MUL A, C` (Multiplies A with C, stores the result in X)
3. `ADD C, D` (Adds C and D, stores the result in C)
4. `MUL C, D` (Completes the multiplication)

Accumulator: An accumulator is a register that stores intermediate results of arithmetic and logic operations. In many architectures, it serves as the destination for results from the ALU

3) What is biased exponent in floating point numbers ? Represent 0.1_2 in floating point representation using 8 bit. What are direct & indirect addresses ?
ANSWER) Biased Exponent in Floating Point Numbers: In floating-point representation, the exponent is stored with a bias value added to it. This bias ensures that the exponent can represent both negative and positive values without needing to use a sign bit for the exponent. The bias is a constant that depends on the number of bits used for the exponent. For instance, if the exponent is 8 bits, the bias is typically 127.
Floating Point Representation of 0.12 (8 bits): To represent 0.12 in floating point using 8 bits, we follow these steps:

1. Convert 0.12 to binary: `0.000111101000111...`
2. Normalize the value: `1.111010001111... × 2^-4`
3. Using a bias of 3 (if we assume an 8-bit exponent with a bias of 3), the exponent is adjusted to `-4 + 3 = -1` or `111` in binary.
   Direct Addressing: In direct addressing, the operand (data) is directly specified within the instruction. The instruction directly provides the address of the operand in memory.
   Indirect Addressing: In indirect addressing, the instruction contains a reference to a memory location that holds the actual address of the operand. The operand is found by fetching the value

at the given address.

4) Draw and explain the flow chart for Booth ’s Algorithm. Show the steps of multiplication performed by using Booth’s algorithm of -6x7.

ANSWER) Booth's Algorithm Flowchart: Booth’s algorithm is used for signed binary multiplication. It is based on examining two bits at a time (the current bit and the previous bit) and performs a set of shifts and additions/subtractions based on the bits' values.

1. Initialize the multiplier (Q), multiplicand (M), and accumulator (A) to 0.
2. Examine two bits at a time: Q_0 (least significant bit of the multiplier) and Q_{-1} (previous bit of the multiplier).
3. Based on the combination of Q_0 and Q_{-1}, perform the following operations:
    - `00`: No change, perform a shift.
    - `01`: Add M to A and shift.
    - `10`: Subtract M from A and shift.
    - `11`: No change, just shift.

Steps of Multiplication for -6 × 7 using Booth’s Algorithm:

- Multiplicand M = -6 = `1010` (in 4-bit two’s complement)
- Multiplier Q = 7 = `0111`
- Initial A = 0000, Q = 0111, Q_{-1} = 0

Step 1: Examine the last two bits of Q.

- Q_0 and Q_{-1} are `10`. This means subtract M from A and shift.

Continue shifting and adjusting until you have completed all bits. The final result is stored in the accumulator.

5) Write the algorithm of Restoring Division process. Show the restoring division steps of 12/3.

ANSWER) Restoring Division Algorithm: Restoring division is a method used for dividing binary numbers. The algorithm works by repeatedly subtracting the divisor from the current value of the dividend, restoring the previous state if the result is negative, and shifting.

Algorithm:

1. Initialize:
    - Quotient (Q) = 0
    - Dividend (D) = Dividend (12 in this case)
    - Divisor (M) = Divisor (3 in this case)
2. Shift the dividend and divisor left by 1 bit.
3. Subtract divisor from the leftmost part of the dividend.
4. If the result is negative, restore the previous value and set the corresponding quotient bit to 0.
5. If the result is positive or zero, set the quotient bit to 1 and shift.
6. Repeat the process for the required number of bits.

Steps for 12 ÷ 3:

1. Start with the dividend 12 (1100) and divisor 3 (0011).
2. Perform the steps of shifting and subtracting.
3. After the division, the quotient is 4 and the remainder is 0.

6) What is Cache memory ? What is the advantage of cache direct mapping ? It in a byte addressable system the main memory size is 32 GB, associative cache size is 32 KB and main memory block size is 1 KB, find the number of tag bits in the physical address.

ANSWER) Cache Memory: Cache memory is a small, high-speed storage area located close to the CPU. It stores frequently accessed data to speed up subsequent accesses. The data from the main memory is copied into the cache, reducing the time it takes for the CPU to fetch data.

Direct Mapping Cache: Direct mapping is a technique used to map the main memory to cache memory. In direct-mapped cache, each block of memory is mapped to a specific cache line. It is simple to implement, but it can lead to cache misses if multiple memory locations map to the same cache line.

Tag Bits Calculation: Given:

- Main memory size = 32 GB = $32 \times 1024 \times 1024 \times 1024$ bytes
- Cache size = 32 KB = 32×1024 bytes
- Block size = 1 KB = 1024 bytes

Total number of blocks in memory = $\frac{32 \times 1024 \times 1024 \times 1024}{1024}$ Total number of blocks in cache = $\frac{32 \times 1024}{1024}$

Now, calculate the number of tag bits based on the addressing structure. Tag bits depend on the number of memory blocks and cache lines, and the number of bits required for block addresses.

7) Differentiate between hardwired control and micro programmed control. What is horizontal microprogramming ? Explain the functions of control memory.

ANSWER) Hardwired Control:

- **Definition:** In hardwired control, control signals are generated by fixed logic circuits (e.g., combinational logic). It uses a set of gates, flip-flops, and other logic components to control the operations of the system.
- **Advantages:** It is fast and efficient for simple operations.
- **Disadvantages:** It is inflexible and difficult to modify or extend for new instructions.

Micro-programmed Control:

- **Definition:** In micro-programmed control, control signals are generated from a set of microinstructions stored in memory (control memory). A control unit fetches these microinstructions to produce the required control signals.
- **Advantages:** It is flexible and can be easily modified to add new instructions.
- **Disadvantages:** It is slower than hardwired control due to memory fetching.

Horizontal Microprogramming:

- In horizontal microprogramming, each microinstruction contains multiple bits, each controlling a specific part of the hardware in parallel. It uses wide control words that allow multiple operations to occur simultaneously in different parts of the system.

Control Memory: Control memory is used to store the control instructions (microprograms) that direct the operation of a system's control unit. It holds the microinstructions used by the control unit to control various parts of the computer's operation.

8) Explain Flynn's classification of computers. W hat is instruction pipeline ? Explain it with the flowchart.

ANSWER) Flynn's Classification of Computers: Flynn's classification divides computer

architectures based on the number of instruction streams and data streams that are processed simultaneously:

1. **SISD (Single Instruction Stream, Single Data Stream):** Traditional uniprocessor systems where one instruction is executed on one data element at a time.
2. **SIMD (Single Instruction Stream, Multiple Data Streams):** Parallel computing where one instruction is applied to multiple data elements simultaneously, such as in vector processors.
3. **MISD (Multiple Instruction Streams, Single Data Stream):** Rare in practice, where multiple instructions are applied to a single data stream.
4. **MIMD (Multiple Instruction Streams, Multiple Data Streams):** This architecture allows multiple processors to execute different instructions on different data, as seen in multi-core processors.

Instruction Pipeline: An instruction pipeline is a technique where multiple instruction phases (fetch, decode, execute, etc.) are overlapped. It divides the instruction cycle into discrete stages, each of which can be worked on concurrently.

Flowchart of Instruction Pipeline:

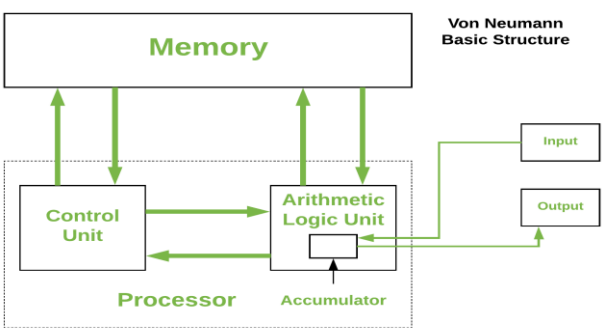
1. **Fetch Instruction**
2. **Decode Instruction**
3. **Execute**
4. **Write back result**

Each step is performed in parallel with others, allowing for a continuous flow of instructions through the processor.

9) a. Draw the block diagram of Von Neumann computer architecture and explain the function of each part.
b. Write Short note on : ALU.
ANSWER)

a. **Von Neumann Architecture:** The Von Neumann architecture is the foundational architecture for most computers today. It consists of the following components:

1. **CPU (Central Processing Unit):** The brain of the system responsible for executing instructions.
   - **ALU (Arithmetic and Logic Unit):** Performs mathematical and logical operations.
   - **Control Unit (CU):** Coordinates the execution of instructions.
2. **Memory (Main Memory):** Stores data and program instructions.
3. **Input Devices:** Devices such as keyboard, mouse, etc., to input data into the computer.
4. **Output Devices:** Devices like monitors or printers for displaying results.
5. **Bus System:** The communication pathways (data, address, and control buses) connecting the CPU, memory, and I/O devices.



b. ALU: The **ALU (Arithmetic and Logic Unit)** is a critical part of the CPU that performs arithmetic operations (addition, subtraction, etc.) and logical operations (AND, OR, NOT, etc.). It works with operands stored in the registers and provides the result to be stored back in the registers or memory.

10) What is the One address instruction?

ANSWER) In a one-address instruction set, only one operand is specified in the instruction. This operand is typically the accumulator, which is used for both the source and destination of the operation. The instruction format contains only the operation code (opcode) and the address of the operand. The result of the operation is stored back in the accumulator.

11) a. What is addressing mode?

b. Explain the different addressing modes?

ANSWER)

a. Addressing Mode: Addressing mode defines how the operands (data) are accessed in a computer instruction. It specifies the way in which the location of the operand is determined during instruction execution. Different modes allow flexibility in how the operands are specified in the instruction.

b. Common addressing modes include:

1. **Immediate Addressing Mode:** The operand is specified directly in the instruction.
   - Example: `MOV A, #5` (Move immediate value 5 into register A)
2. **Direct Addressing Mode:** The address of the operand is given explicitly in the instruction.
   - Example: `MOV A, [5000]` (Move the value at memory address 5000 into register A)
3. **Indirect Addressing Mode:** The instruction contains the address of a memory location that holds the address of the operand.
   - Example: `MOV A, [R0]` (Move the value at the address stored in R0 into register A)
4. **Indexed Addressing Mode:** The address of the operand is computed by adding a constant value (index) to a register value.
   - Example: `MOV A, [R1 + 10]` (Move the value at the address computed by adding 10 to R1 into register A)
5. **Register Addressing Mode:** The operand is a register, and the instruction directly specifies which register to use.
   - Example: `MOV A, R0` (Move the contents of register R0 into register A)

12) a. Convert and represent the decimal number 0.5624 in floating point representation using 8 bits.

b. Draw the Addition-Subtraction unit block diagram.

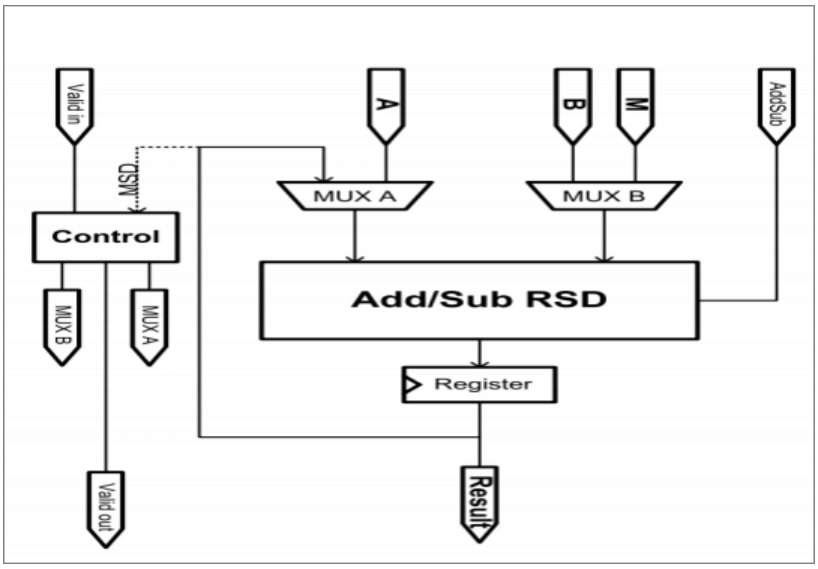
ANSWER)

a. Steps to convert:

1. Convert the decimal number into binary form: 0.5624 = `0.1001 0001 1001...`
2. Normalize the number: `1.0010001 × 2^(-1)`
3. Using 8 bits, we represent this as:
   - Sign bit = 0 (positive number)
   - Exponent = 127 (bias of 127) + (-1) = 126 = `01111110` (8 bits)
   - Mantissa = `0010001` (after normalization)

Thus, the 8-bit floating-point representation is: `0 01111110 0010001`

b. Addition-Subtraction Unit Block Diagram:



The block diagram of the Addition-Subtraction Unit consists of:

1. **Adder/Subtractor**: Performs the addition or subtraction of two numbers.
2. **Control Logic**: Determines whether the operation is addition or subtraction based on the instruction.
3. **Registers**: Stores operands and results for the operation.
4. **Output**: Provides the result of the operation.

13. a. Divide Q=l1010 (dividend) by M=l10 (divisor) using restoring division method.
b. Show the steps of multiplication performed by using Booth's algorithm of 1011x10011.

ANSWER)

a. Restoring Division Method:

Given:

- Dividend $Q=1010_2$ (decimal 10)
- Divisor $M=10_2$ (decimal 2)

Step 1: Initialize registers:

- $Q=1010_2$
- $M=10_2$
- $A=0000_2$ (Accumulator initialized to 0)
- $Q_0=0$ (the least significant bit of Q)

Step 2: Perform the division:

1. **Shift A and Q left**:
   A=0000, Q=1010, $Q_0=0$ → Shifted: A=0001, Q=0100, $Q_0=1$
2. **Subtract M from A** (restoring step if needed):
   A=0001−0010=−0001 (negative, so restore)
   - Restore: A=0000
   - Set $Q_0=0$ and repeat for next shift.

Final result: After all shifts, the quotient is $Q=0101_2$ (decimal 5), and the remainder is $A=0000_2$.

b. Booth's Algorithm for Multiplication:

Multiply 1011_2 by 10011_2.

1. **Initialize:**
   - $M = 1011_2$ (decimal 11)
   - $Q = 10011_2$ (decimal 19)
   - $Q_0 = 0$ (initial bit)
   - $A = 00000_2$ (initialized to 0)
2. **Steps:**
   - Check the last two bits of Q_0Q_1:
     - If 01: Add M to A
     - If 10: Subtract M from A
   - Shift A and Q left, update Q_0.
3. **Iterations** (not fully written out for brevity):
   - Iteration 1: $Q_0Q_1 = 01$, so add M to A, shift.
   - Iteration 2: $Q_0Q_1 = 10$, subtract M, shift.
   - Continue until you've completed 5 iterations (since there are 5 bits in Q).
4. **Final result**: After the shifts, the product is stored in A and Q. The final value is 111001_2, which equals 209 in decimal.

14) a. What is hit ratio? If numbers of hits are equal to 10 & if numbers of misses are equal to 4 then find the hit ratio ?

b. "Hit ratio is better in set-associative mapping than in direct mapping of cache" Is it correct ? Explain.

ANSWER)

a. Hit Ratio:

The hit ratio is the fraction of memory accesses that result in cache hits, meaning the data is found in the cache.

Formula for hit ratio:

$$\text{Hit Ratio} = \frac{\text{Number of Hits}}{\text{Total Memory Accesses}} = \frac{\text{Number of Hits}}{\text{Hits} + \text{Misses}}$$

Given:

- Number of hits = 10
- Number of misses = 4

$$\text{Hit Ratio} = \frac{10}{10+4} = \frac{10}{14} \approx 0.714 \text{ or } 71.4\%$$

b. Hit Ratio in Set-Associative vs. Direct Mapping:

Yes, it is generally correct that hit ratio is better in set-associative mapping than in direct-mapped cache.

Explanation:

- **Direct-mapped cache**: Each memory block maps to exactly one cache line. If multiple blocks map to the same line, a conflict occurs, which can reduce the hit ratio.
- **Set-associative cache**: Each memory block can be placed in one of several cache lines (a set), reducing conflicts and improving the chances of a hit.

Thus, **set-associative mapping** typically leads to fewer cache misses and a better hit ratio, especially in cases of high conflict.

15) Write Short Note on :
a. RISC pipeline.
b. Vector processing.

ANSWER)

a. RISC Pipeline:

A RISC (Reduced Instruction Set Computer) pipeline refers to the stages through which instructions pass in a RISC architecture, aimed at optimizing performance. RISC processors have a simplified instruction set, which allows for fast execution of instructions. The pipeline typically consists of stages such as:

1. **Fetch**: Instruction is retrieved from memory.
2. **Decode**: The instruction is decoded to determine the operation and operands.
3. **Execute**: The operation is performed, such as an arithmetic or logical operation.
4. **Memory Access**: Data is read or written from/to memory (if needed).
5. **Write-back**: The result is written back to the register file.
   By breaking down the execution into these stages, RISC pipelines improve the throughput, enabling higher instruction execution per cycle. The simplicity of RISC allows the use of pipelining to enhance performance and efficiency.

b. Vector Processing:

Vector processing involves the use of vector processors, which operate on entire vectors (arrays of data) in a single instruction, rather than on individual scalar values. This is particularly useful in applications requiring high-performance computing, such as scientific calculations and graphics processing.

Key features:

- **SIMD (Single Instruction, Multiple Data)**: A vector processor executes the same instruction on multiple data points simultaneously, allowing for parallel processing.
- **Vector registers**: Instead of scalar registers, vector processors use larger registers capable of holding multiple data elements.
- **Pipelined execution**: Like in RISC, vector processors often employ pipelined stages to improve throughput.
  Vector processing can significantly speed up tasks like matrix multiplication, image processing, and simulations, where the same operation is performed on large sets of data.

16) Describe the following addressing modes with suitable example.
(i) Direct addressing node, (ii) Indirect addressing mode, (iii) Indexed addressing mode, (iv) Based addressing mode, (v) Implied addressing mode.

ANSWER)

(i) Direct Addressing Mode:

In direct addressing mode, the operand is specified explicitly in the instruction itself.

- **Example**:
  - Instruction: `MOV A, 1000`
  - Here, the value from memory location 1000 is moved directly into register A.

The operand is directly the address in memory where the data is located.

(ii) Indirect Addressing Mode:

In indirect addressing mode, the instruction specifies a memory location that contains the address of the operand. The actual operand is found at that address.

- **Example**:
  - Instruction: `MOV A, @R0`
  - If register R0 contains the value 1000, the operand is located at memory address 1000, and the value from memory address 1000 is moved to register A.

  Here, `@R0` indicates that the operand is not at R0 directly but at the memory address stored in R0.

(iii) Indexed Addressing Mode:

In indexed addressing mode, the effective address of the operand is determined by adding a constant value (the index) to the contents of a register (the base).

- **Example**:
  - Instruction: `MOV A, 1000(R1)`
  - If register R1 contains 2000, the operand is at memory address 2000 + 1000 = 3000. The value at address 3000 is moved into register A.

  This mode is commonly used for accessing arrays, where the base address is in the register and the offset is the index.

(iv) Based Addressing Mode:

In based addressing mode, the address of the operand is computed by adding a base register value to a displacement (offset). This is similar to indexed addressing but typically involves a specific base register.

- **Example**:
  - Instruction: `MOV A, 50(R2)`
  - If register R2 contains 1500, the operand is at address 1500 + 50 = 1550. The value at memory address 1550 is moved to register A.

  Based addressing is useful when accessing variables or data structures relative to a known base address.

(v) Implied Addressing Mode:

In implied addressing mode, the operand is implicitly specified by the instruction, and there is no need to provide an explicit memory address or register.

- **Example**:
  - Instruction: `CLRA`
  - The `CLRA` instruction clears register A (sets it to zero). The operand (register A) is implied by the instruction itself.

  This mode is often used for operations like clearing registers or performing operations on a fixed register without needing to specify it.

17) Explain how priority of interrupt will be implemented ?

ANSWER)

The **priority of interrupts** is implemented to determine which interrupt should be serviced first when multiple interrupts occur simultaneously. This can be achieved through various methods:

1. Vectored Interrupts:

- In vectored interrupts, each interrupt source is assigned a unique vector or address in memory, where the interrupt service routine (ISR) for that interrupt is located. The interrupt controller can prioritize interrupts based on the vector number.
- **Priority Scheme**: Higher-priority interrupts have lower vector numbers, so the CPU will first look at the interrupt with the lowest vector number.

2. Interrupt Priority Levels:

- Many systems implement **priority levels** for interrupts, where each interrupt is assigned a priority level (e.g., from 0 to 7). When multiple interrupts occur, the one with the highest priority (lowest priority level number) is serviced first.
- **Example**: An interrupt with priority level 1 will be handled before one with priority level 2.

3. Interrupt Controller:

- The **interrupt controller** manages the priority and servicing of interrupts. Some common interrupt controllers, such as the **Programmable Interrupt Controller (PIC)** or **Advanced Programmable Interrupt Controller (APIC)**, support multiple interrupt priorities.
- The controller may have a priority register that defines which interrupt should be served first.

4. Polling vs. Vectoring:

- In systems with polling, the CPU frequently checks interrupt sources in a specific order. The first interrupt detected gets priority.
- In systems with vectoring, the interrupt source is checked based on predefined priorities, with higher-priority interrupts being serviced before lower-priority ones.

5. Nested Interrupts:

- **Nested interrupts** allow higher-priority interrupts to interrupt the servicing of lower-priority ones. For instance, if an interrupt of higher priority arrives while the CPU is processing a lower-priority interrupt, the CPU will pause the current ISR, service the higher-priority interrupt, and then resume the previous ISR.

By combining these methods, the system ensures that interrupts are handled efficiently based on their priority, preventing lower-priority interrupts from blocking critical operations.

18) a. Write down the flowchart of both restoring and non-restoring division of integer numbers.
b. Write down the features of RISC architecture.

ANSWER)

(a) Flowchart for Restoring and Non-Restoring Division of Integers :

Restoring Division Flowchart :

1. **Initialize:** Set Dividend in register A and B. Initialize A = 0, Q = Dividend, M = Divisor, and Count = n (bit length of the divisor).
2. **Shift Left:** Shift A and Q left by one position.
3. **Subtract Divisor (M):** A = A - M.
4. **Check Sign of A:**
    - If A ≥ 0, set Q0 = 1.
    - If A < 0, restore A by adding M (A = A + M) and set Q0 = 0.
5. **Decrement Count:** Reduce Count by 1.
6. **Repeat or End:** If Count > 0, go to Step 2. Otherwise, the division result is in Q, and the remainder is in A.

Non-Restoring Division Flowchart :

1. Initialize: Set Dividend in register A and B. Initialize A = Dividend, Q = Dividend, M = Divisor, and Count = n.
2. **Shift Left:** Shift A and Q left by one position.
3. **Check Sign of A:**

- If A ≥ 0, subtract Divisor (A = A - M).
- If A < 0, add Divisor (A = A + M).

4. Update Q0:

- If A ≥ 0, set Q0 = 1.
- If A < 0, set Q0 = 0.

5. Decrement Count: Reduce Count by 1.

6) Repeat or End: If Count > 0, go to Step 2. Otherwise, correct the remainder (if needed), and Q contains the result.

(b) Features of RISC Architecture :

- **Simple Instructions:** RISC processors use a small, highly optimized set of instructions.
- **Uniform Instruction Format:** Fixed-length instructions and a small number of instruction formats.
- **Load-Store Architecture:** Separate instructions for memory access (load/store) and arithmetic/logic operations.
- **Large Number of Registers:** Optimized for reduced memory access by having many registers.
- **Pipeline Execution:** Supports efficient pipelining due to simple instructions.
- **Few Addressing Modes:** Limited but efficient addressing modes for simplicity.
- **Hardwired Control Unit:** No microcode; control unit is hardwired for faster execution.

- **Reduced Complexity:** Emphasis on hardware simplicity and performance.

19) a. Explain different types of mapping technique used in cache memory.
b. How hit ratio depends on different types of mapping.
ANSWER)
(a) Types of Mapping Techniques Used in Cache Memory :
Cache memory is a small, fast storage that stores copies of frequently accessed data from the main memory to improve overall system performance. The mapping techniques define how data from main memory is placed in cache. There are three primary types of cache mapping:
Direct-Mapped Cache:

- **Description:** Each block of main memory maps to exactly one line in the cache. The cache uses a specific part of the memory address (usually the lower bits) to determine the cache line where the data will be placed.
- **Structure:** The address is divided into three parts:
  - **Tag:** Identifies the block of data.
  - **Index:** Maps to a specific cache line.
  - **Block offset:** Specifies the location of data within the block.
- **Advantages:** Simple and fast lookup.
- **Disadvantages:** High conflict misses (if two memory addresses map to the same cache line).

Fully Associative Cache:

- **Description:** A block from main memory can be stored in any cache line. The entire memory address is used as the tag, and the cache is searched fully for a match.
- **Structure:** The address only contains the **Tag** and **Block offset**, and the cache looks through all lines to find the data.
- **Advantages:** No conflict misses (any block can go into any line).

- **Disadvantages:** Slower access (due to full search) and more complex hardware.

Set-Associative Cache:

- **Description:** This is a compromise between direct-mapped and fully associative caches. The cache is divided into multiple sets, and each set contains multiple cache lines. A block of data can map to any line within a set.
- **Structure:** The address is divided into three parts:
  - **Tag:** Identifies the block.
  - **Set index:** Determines which set the data can be placed in.
  - **Block offset:** Specifies the data's position within the block.
- **Advantages:** Lower conflict misses compared to direct-mapped, more flexible.
- **Disadvantages:** More complex than direct-mapped, slower access than fully associative.

(b) The **hit ratio** in cache memory is the percentage of cache accesses that result in a cache hit (i.e., the requested data is found in the cache). The hit ratio depends on how data is mapped into the cache, and it varies across the different mapping techniques:
Direct-Mapped Cache:

- **Hit Ratio Impact:** The hit ratio tends to be lower in direct-mapped caches, especially for programs that exhibit a high level of conflict misses (when multiple frequently accessed memory addresses map to the same cache line).
- **Explanation:** Since each block of memory maps to a specific line in the cache, if two memory addresses that are frequently accessed map to the same cache line, one will replace the other, leading to a higher miss rate.

Fully Associative Cache:

- **Hit Ratio Impact:** The hit ratio is typically higher because any block can be placed in any cache line. This flexibility reduces conflict misses, which improves cache performance.
- **Explanation:** There is no fixed mapping between memory locations and cache lines, so multiple frequently accessed memory addresses can coexist in the cache without replacing each other. This reduces the miss rate significantly, leading to a higher hit ratio.

Set-Associative Cache:

- **Hit Ratio Impact:** The hit ratio lies between that of direct-mapped and fully associative caches, as there is a balance between flexibility and the overhead of searching within a set.
- **Explanation:** With multiple lines per set, the chance of conflict misses is reduced compared to direct-mapped caches. However, there is still a chance of conflict within a set, but it is less severe than in the direct-mapped case. Typically, increasing the number of lines per set (i.e., increasing the set-associativity) results in a higher hit ratio.

20) a. Explain the representation of floating point number in computer. Describe the IEEE format also.
b. What is NaN in floating point number representation? Give example.
ANSWER)
(a) Representation of Floating Point Numbers in Computers :
Floating-point numbers are used to represent real numbers in computers, particularly when precision and range are important. They are represented in a way that can handle very large or very small numbers by using a scientific notation approach.

Representation of Floating Point Numbers:

A floating-point number is typically represented in the following general form :

$$(-1)^s \times (1 + f) \times 2^{(e-bias)}$$

Where:

- **s**: Sign bit (0 for positive, 1 for negative).
- **f**: Fraction (or mantissa) representing the significant digits of the number.
- **e**: Exponent that determines the scale of the number.
- **bias**: A fixed value added to the exponent to allow for both positive and negative exponents.

Components of Floating-Point Representation:

1. **Sign bit (s)**: This bit determines the sign of the number (0 for positive, 1 for negative).
2. **Exponent (e)**: The exponent determines the range of values the number can take. It is typically stored using a "biased" representation to accommodate both positive and negative exponents.
3. **Mantissa (f)**: The mantissa (or significand) represents the precision of the number. In normalized form, the leading bit is assumed to be 1 (hidden bit), and only the fractional part is stored.

IEEE Floating-Point Format:

IEEE 754 is the standard used for representing floating-point numbers in computers. It defines two main formats: **single precision** (32 bits) and **double precision** (64 bits).

Single Precision (32 bits):

- **1 bit** for the sign (s).
- **8 bits** for the exponent (e).
- **23 bits** for the fraction (mantissa) (f).

Formula:

$$(-1)^s \times (1 + f) \times 2^{(e-127)}$$

- Bias = 127 (to handle both positive and negative exponents).
- The exponent is stored as an 8-bit unsigned integer, and the exponent value is represented as e - 127 (where 127 is the bias).

Example of a 32-bit IEEE 754 floating-point number:

0 10000001 01000000000000000000000

- Sign bit (s) = 0 (positive).
- Exponent (e) = 10000001 (binary) = 129 (decimal).
- Mantissa (f) = 01000000000000000000000 (binary) = 1.25 (normalized form).

Double Precision (64 bits):

- 1 bit for the sign (s).
- 11 bits for the exponent (e).
- 52 bits for the fraction (mantissa) (f).

Formula:

$$(-1)^s \times (1 + f) \times 2^{(e-1023)}$$

Bias = 1023.
Example of a 64-bit IEEE 754 floating-point number:

0 10000000001 0000000000000000000000000000000000000000000000000000

- **Sign bit (s) = 0** (positive).
- **Exponent (e) = 10000000001** (binary) = 1025 (decimal).
- **Mantissa (f) = 00000000000000000000000** (binary) = 1 (normalized form).

(b) NaN in Floating Point Number Representation :
NaN stands for **Not a Number**, and it is a special value used to represent undefined or unrepresentable values in floating-point arithmetic. NaN is commonly encountered in situations like dividing 0 by 0, taking the square root of a negative number, or performing other invalid operations.

Characteristics of NaN:

1. **Indicates an Invalid Operation**: NaN is used to represent results that do not have a valid numerical meaning (e.g., 0/0, infinity - infinity).
2. **Propagation**: If any arithmetic operation involves NaN, the result is also NaN, which helps in detecting errors in computations.

IEEE 754 Representation of NaN:

- In both single and double precision formats, NaN is represented with an exponent of all 1s and a non-zero fraction (mantissa).
- The specific pattern of bits for NaN in IEEE 754 format:
  - **Single precision (32 bits)**: Exponent is all 1s (255 in decimal), sign bit is either 0 or 1, and the fraction is any non-zero value.
  - **Double precision (64 bits)**: Exponent is all 1s (2047 in decimal), sign bit is either 0 or 1, and the fraction is any non-zero value.

Example of NaN (Single Precision):

0 11111111 10000000000000000000000

- **Sign bit (s) = 0** (positive, but this doesn't matter for NaN).

- **Exponent (e) = 11111111** (all 1s for NaN).
- **Mantissa (f) = 10000000000000000000** (non-zero for NaN).

Example of NaN (Double Precision):

0 11111111111 1000000000000000000000000000000000000000000000000000

- **Sign bit (s) = 0** (positive).
- **Exponent (e) = 11111111111** (all 1s for NaN).
- **Mantissa (f) = 10000000000000000000** (non-zero for NaN).

Example Use of NaN:

- Dividing 0 by 0:

$$\text{NaN} = \frac{0}{0}$$

- Square root of a negative number:

$$\text{NaN} = \sqrt{-1}$$

21) Explain the organization of control memory in Micro-programmed control with diagram.
ANSWER) Micro-programmed control is a method of implementing control logic in a digital system, especially in CPUs, using a set of micro-instructions stored in control memory. Instead of using fixed logic gates or hardware circuits to generate control signals, a micro-programmed control unit uses a sequence of micro-instructions to control the operation of the machine.|

Key Components of Micro-programmed Control:

1. **Control Memory**: This is a memory unit that stores the control words or micro-instructions. These micro-instructions specify the operation to be performed by the CPU at each step of an instruction cycle.
2. **Control Address Register (CAR)**: Holds the address of the current micro-instruction in the control memory.
3. **Control Buffer Register (CBR)**: Stores the micro-instruction retrieved from the control memory.
4. **Decoder**: A component that decodes the micro-instruction stored in the CBR and generates the control signals.
5. **Micro-Instruction Sequencer**: This logic unit generates the address for the next micro-instruction to be executed, based on the current instruction, the sequence of micro-operations, and sometimes conditions or flags.

Organization of Control Memory:

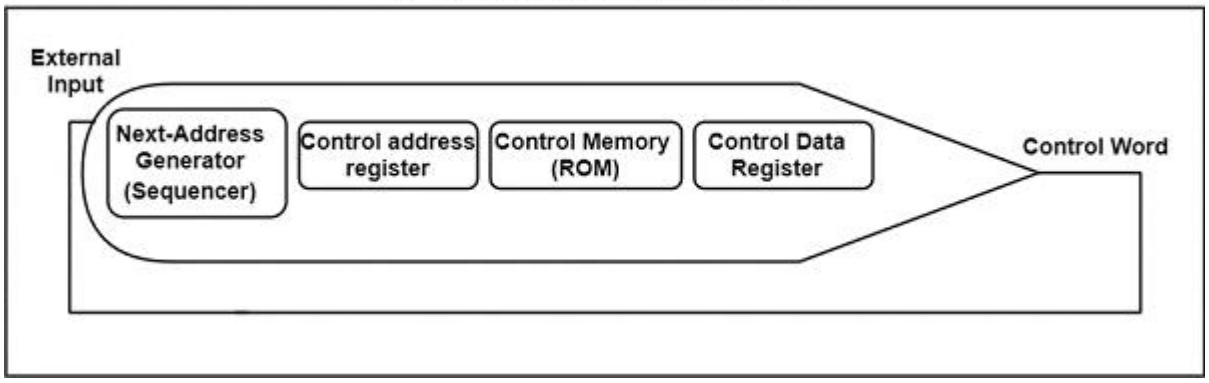
The control memory is typically organized as a **ROM (Read-Only Memory)** or **RAM (Random Access Memory)** that stores the sequence of micro-instructions needed for various operations. Each micro-instruction corresponds to one step in the execution of an instruction. These micro-instructions contain the control bits that tell the CPU which operations to perform (such as fetching, decoding, executing, and storing results).

The control memory is structured as follows:

- **Each micro-instruction** corresponds to a row in the control memory.
- **Each micro-instruction** consists of control bits that:
  - Specify which registers should be read or written.
  - Control ALU operations.
  - Define timing and sequencing of micro-operations.

Diagram :

Micro Programmed Control Organization



22) a. Describe DMA mode of data transfer in details with suitable diagram.
b. RISC architecture is more suitable for pipeline implementation-explain.
ANSWER)

(a) DMA Mode of Data Transfer :

DMA (Direct Memory Access) allows data to be transferred directly between I/O devices and memory without involving the CPU, freeing it to perform other tasks.

Steps in DMA:

1. **Initiation**: CPU sends a request to the DMA controller with source, destination, and data size.
2. **Bus Arbitration**: DMA controller gains control of the system bus.
3. **Data Transfer**: DMA controller transfers data between I/O and memory.
4. **Interrupt**: After completion, DMA sends an interrupt to the CPU to notify the transfer is done.

Types of DMA:

- **Burst Mode**: DMA transfers a block of data in one go.
- **Cycle Stealing**: DMA takes control for one cycle to transfer one piece of data.
- **Block Mode**: DMA transfers a block of data without releasing the bus until completion.
- **Demand Mode**: DMA transfers data as requested by I/O devices.

(b) RISC Architecture and Pipelining :

RISC (Reduced Instruction Set Computing) is highly suitable for pipelining because:

1. **Simple, Fixed-Length Instructions**: Easy to decode and execute in parallel.
2. **Load/Store Architecture**: Reduces memory access bottlenecks by separating memory and arithmetic operations.
3. **Large Register Set**: Minimizes memory access, helping smooth instruction flow.
4. **Fewer Instruction Formats**: Simplifies decoding and reduces pipeline hazards.

23) Write short notes on the following :
a) Generations of computer
b) Space time diagram
c) Array processor
d) Virtual memory
ANSWER)

a) Generations of Computers :

Computers have evolved through several generations, marked by technological advancements:

1. **First Generation (1940s–1950s)**: Used vacuum tubes, were large, expensive, and slow; programming was done in machine language. Example: ENIAC.
2. **Second Generation (1950s–1960s)**: Used transistors, which made computers smaller, faster, and more reliable; supported assembly language programming.
3. **Third Generation (1960s–1970s)**: Used integrated circuits (ICs), enabling better performance and multi-programming capabilities.

4. **Fourth Generation (1970s–present)**: Based on microprocessors, leading to personal computers and rapid advancements in hardware.
5. **Fifth Generation (Present and Beyond)**: Focused on AI, parallel processing, and quantum computing.

b) Space-Time Diagram

A space-time diagram represents the flow of instructions or data between processing elements in parallel computing. It shows:

- **Space** (horizontal axis): Represents multiple processors or tasks.
- **Time** (vertical axis): Represents the sequence of execution.

This diagram is used to visualize dependencies, synchronization, and performance in pipeline processing or distributed systems. It helps optimize task allocation and execution.

c) Array Processor

An array processor is a type of parallel processor that consists of multiple processing elements (PEs) operating simultaneously under the control of a single instruction stream (SIMD architecture).

- Used for tasks requiring repetitive computations, such as matrix operations and image processing.
- **Features**: High-speed computation, efficiency in handling vector operations, and suitability for scientific applications.

d) Virtual Memory

Virtual memory is a memory management technique that allows the execution of processes larger than the physical memory (RAM).

- It uses a combination of **RAM and secondary storage** (e.g., a hard disk) to create an illusion of a larger memory space.
- **Key Concepts**:
  - **Paging**: Dividing memory into fixed-size blocks.
  - **Swapping**: Moving data between RAM and disk as needed.
- **Advantages**: Allows multitasking, improves memory utilization, and enables large applications to run on limited RAM.

24) Explain the components of the Computer system and what is micro operation ?

ANSWER)

Components of a Computer System :

A computer system comprises five main components that work together to perform computations:

1. **Input Unit**:
   - Accepts data and instructions from the user.
   - Devices: Keyboard, mouse, scanner, etc.
2. **Output Unit**:
   - Presents processed data (results) to the user.
   - Devices: Monitor, printer, speaker, etc.
3. **Central Processing Unit (CPU)**:
   - The brain of the computer that executes instructions.
   - **Sub-components**:
     - **Control Unit (CU)**: Directs the flow of data and coordinates the operations of other components.
     - **Arithmetic Logic Unit (ALU)**: Performs arithmetic (addition, subtraction, etc.) and logical (comparison, AND, OR, etc.) operations.
     - **Registers**: Small, fast storage locations within the CPU for temporary data storage.
4. **Memory Unit**:

- Stores data and instructions.
- Types:
  - **Primary Memory (RAM/ROM)**: Temporary storage for running programs.
  - **Secondary Memory**: Permanent storage, e.g., hard drives, SSDs.
  - **Cache**: High-speed memory for frequently accessed data.

5. **Storage Unit**:
   - Permanent storage for data and programs.
   - Examples: HDD, SSD, optical disks, USB drives.
6. **Communication Unit** (optional in extended models):
   - Handles data exchange between the computer and other systems via networks.

Micro Operation :
A **micro-operation** is a low-level operation that involves the transfer or manipulation of data at the register level in a computer. It represents the basic steps needed to execute a machine instruction.

Types of Micro-Operations:

1. Register Transfer Micro-Operations: Move data between registers.
   Example: $R1 \leftarrow R2$ (data is moved from $R2$ to $R1$).
2. Arithmetic Micro-Operations: Perform basic arithmetic operations.
   Example: $R3 \leftarrow R1 + R2$.
3. Logical Micro-Operations: Perform bitwise logical operations.
   Example: $R1 \leftarrow R2 \text{ AND } R3$.
4. Shift Micro-Operations: Shift bits in a register left or right.
   Example: $R1 \leftarrow R1 >> 1$ (right shift).

25) Represent $(12.625)_{10}$ in 32 bit floating point representation and what is odd parity checker.
ANSWER)

Representation of $(12.625)_{10}$ in 32-bit Floating-Point Format

The IEEE 754 32-bit floating-point representation consists of three parts:

1. **Sign bit (1 bit)**: 0 for positive numbers, 1 for negative numbers.
2. **Exponent (8 bits)**: Stored in "excess-127" format (bias is 127).
3. **Mantissa (23 bits)**: Fractional part after normalizing the number.

Steps to Convert $(12.625)_{10}$:

1. Convert to Binary:
   - $12 = 1100_2$ (integer part).
   - $0.625 = 0.101_2$ (fractional part).
     Combining: $12.625 = 1100.101_2$.
2. Normalize the Binary Number:
   - $1100.101_2 = 1.100101 \times 2^3$.
3. Determine the Sign, Exponent, and Mantissa:
   - **Sign bit**: 0 (positive number).
   - **Exponent**: Add bias (127): $3 + 127 = 130 = 10000010_2$.
   - **Mantissa**: 100101 (omit the leading 1, pad with zeros to 23 bits): $10010100000000000000000_2$.
4. Final Representation:
   - Sign bit: 0
   - Exponent: 10000010
   - Mantissa: 10010100000000000000000

   Result: $0\ 10000010\ 10010100000000000000000$.

An **odd parity checker** is a combinational circuit used to verify whether the total number of 1s in a binary data stream is **odd**. Parity checking ensures data integrity in communication and storage.

How it Works:

- **Odd Parity Rule**: The data (including the parity bit) should have an odd number of 1s.
- If the received data has an even number of 1s, it indicates an error.

Example:

- Data: 1010, Parity bit: 1 → Total 1s: 3 (odd).
- If an error changes 1010 to 1110, the total 1s become 4 (even), and the error is detected.

Applications:

- Ensures reliable data transmission in networking, memory storage, and serial communication.

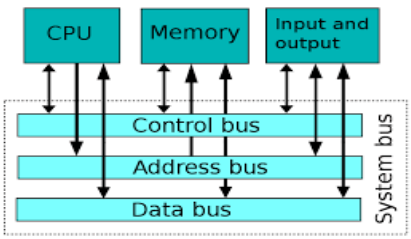
26) Explain the Bus Structure with examples.
ANSWER)
A **bus structure** is a communication pathway used for data transfer among the components of a computer system, such as the CPU, memory, and I/O devices. It consists of a set of parallel wires (or lines) that carry data, control signals, and addresses.

Types of Buses :

1. **Data Bus**:
   - Carries actual data between components (e.g., CPU to memory, memory to I/O devices).
   - Width (number of lines) determines how much data can be transferred at once (e.g., 8-bit, 16-bit).
   - Example: A 32-bit data bus can transfer 4 bytes of data simultaneously.
2. **Address Bus**:
   - Carries the addresses of memory locations or I/O devices that the CPU wants to access.
   - Unidirectional (from CPU to memory/I/O).
   - Width determines the maximum addressable memory space.
     - Example: A 16-bit address bus can address 216=65,5362^{16} = 65,536216=65,536 memory locations.
3. **Control Bus**:
   - Carries control signals like **read/write**, **interrupt requests**, and **clock signals** to coordinate and manage operations.
   - Bidirectional, as control signals can flow both ways.
   - Example: A "read" signal instructs memory to send data to the CPU, while a "write" signal instructs the CPU to send data to memory.

Single Bus vs. Multiple Bus Structures :

1. **Single Bus Structure**:
   - All components share the same bus.
   - **Advantages**: Simple and cost-effective.
   - **Disadvantages**: Performance bottlenecks due to bus contention when multiple devices try to communicate simultaneously.
2. **Multiple Bus Structure**:
   - Separate buses for different purposes (e.g., memory bus, I/O bus).
   - **Advantages**: Improved performance by reducing contention.
   - **Disadvantages**: Higher complexity and cost.



27) Describe the Flag Register of 8086 microprocessor.
ANSWER)
The **flag register** in the 8086 microprocessor is a 16-bit register used to indicate the status of the processor and the outcomes of arithmetic or logical operations. It consists of two categories of flags:

1. **Status Flags**: Indicate the result of an operation.
2. **Control Flags**: Control the execution of certain instructions.

Flags Description:

1. Status Flags:
    - Carry Flag (CF):
        - Set if there is a carry out or borrow into the most significant bit during arithmetic operations.
        - Example: $255 + 1 = 256$; CF = 1.
    - Parity Flag (PF):
        - Set if the number of 1s in the result is even.
        - Example: 00000111 (3 ones) → PF = 0.
    - Auxiliary Carry Flag (AF):
        - Set if there is a carry out from the lower nibble (4 bits) in arithmetic operations.
        - Example: $0x09 + 0x08 = 0x11$; AF = 1.
- Zero Flag (ZF):
    - Set if the result of an operation is zero.
    - Example: $5 - 5 = 0$; ZF = 1.
- Sign Flag (SF):
    - Set if the result of an operation is negative (MSB is 1).
    - Example: -3 (binary: 11111101); SF = 1.
- Overflow Flag (OF):
    - Set if the signed result of an operation exceeds the range representable by the data type.
    - Example: $127 + 1 = -128$ (overflow occurs); OF = 1.

2. Control Flags:

- Trap Flag (TF):
  - Enables single-step execution for debugging. If set, the processor generates an interrupt after executing each instruction.
- Interrupt Flag (IF):
  - Controls the enable/disable status of maskable interrupts.
  - IF = 1 → Interrupts enabled; IF = 0 → Interrupts disabled.
- Direction Flag (DF):
  - Determines string operation direction.
  - DF = 0 → Process strings from low to high memory (increment).
  - DF = 1 → Process strings from high to low memory (decrement).

27) Perform multiplication between 23 and 17 using fixed point multiplication algorithm.
ANSWER)
Fixed-point multiplication involves binary representation of numbers and a step-by-step process.

Here, we multiply 23 and 17 using the standard binary fixed-point method.
Convert the decimal numbers 23 and 17 to binary (assuming 8 bits for simplicity).

$$23_{10} = 10111_2$$

$$17_{10} = 10001_2$$

We align the binary numbers for multiplication. The algorithm uses the standard binary multiplication method, similar to decimal.

$$\begin{array}{r} 10111 \\ \times 10001 \\ \hline \end{array}$$

Each bit of the multiplier (10001_2) is used to multiply the multiplicand (10111_2), followed by shifts for alignment.

1. $10111 \times 1 = 10111$ (No shift for LSB).
2. $10111 \times 0 = 00000$ (Shift left by 1 bit).
3. $10111 \times 0 = 00000$ (Shift left by 2 bits).
4. $10111 \times 0 = 00000$ (Shift left by 3 bits).
5. $10111 \times 1 = 10111$ (Shift left by 4 bits).

Align the results and sum them:

$$\begin{array}{r} 10111 \\ +00000 \\ +00000 \\ +00000 \\ +101110000 \\ \hline 1111111_2 \end{array}$$

The result in binary is:

$$1111111_2 = 391_{10}$$

Thus, $23 \times 17 = 391$ using fixed-point multiplication.

Some More Important Questions, Solve By Yourself :

11) What are the key characteristics of micro-programmed control ? Explain different types of micro operation.

12) What is virtual memory? How does it work?

13) How can you interface RAM and the ROM EPROM to microprocessor 8086 ? What is the use of EPROM ?

14) a. Write the difference between Minimum Mode and Maximum Mode in 8086 Microprocessor.
b. Explain with example any four addressing modes available in 8086 microprocessors.

15) A typical computer system cache memory access time is 8 ns and its main memory access time is 80 ns. If hit ratio is 90%, what is the average memory access time?

16) Discuss in detail the architecture of 8086 microprocessor along with pin configuration diagram.

17) Write down the IEEE-754 format for single and double precision numbers ?

18) a. Describe the concept of Pipeline and its types ?
b. Write the different types of pipeline hazards.