First Section

a) What is the difference between ENIAC and stored-computer program?

1. **ENIAC (Electronic Numerical Integrator and Computer):**
   - Designed in the 1940s as one of the first electronic general-purpose computers.
   - Programmed through manual rewiring and plugboards.
   - It did **not** have the concept of a stored program.
2. **Stored-Computer Program:**
   - Proposed by John von Neumann, the stored-program architecture allows programs and data to be stored in the same memory.
   - Instructions are fetched and executed sequentially from memory.

Key Difference:
ENIAC lacked the stored-program capability, whereas modern computers use the stored-program concept for flexibility.

---

b) Discuss the use of PC register.

- The **Program Counter (PC)** register is used to store the address of the **next instruction** to be executed by the processor.
- It is incremented automatically during instruction execution.
- If a jump or branch occurs, the PC is updated to point to the target address instead.

Role:

- Keeps track of program execution flow.
- Essential for **sequential execution** and branching in programs.

---

c) What is the difference between instruction fetch and execution and interrupts?

1. **Instruction Fetch:**
   - Fetching the next instruction from memory based on the address stored in the Program Counter.
   - This step includes retrieving the binary code (opcode).
2. **Instruction Execution:**
   - Decoding the fetched instruction and performing the intended operation.
   - Operations could involve ALU computations, data transfer, or branching.
3. **Interrupts:**
   - An external or internal signal that temporarily **halts** the normal instruction execution.
   - Interrupt service routines (ISRs) execute to handle the interrupt.
   - Examples: hardware interrupts (e.g., I/O completion) or software interrupts.

Difference:

- **Fetch** and **Execution** are part of normal CPU operation, while **interrupts** cause deviation from the normal execution cycle.

---

d) Discuss the difference between ALU and Control Unit.

1. **ALU (Arithmetic Logic Unit):**
    - Performs arithmetic (addition, subtraction) and logic (AND, OR, NOT) operations.
    - It is a **computation unit**.
2. **Control Unit (CU):**
    - Directs the operations of the processor by generating control signals.
    - It **fetches**, decodes, and sends signals to coordinate data flow between CPU, memory, and I/O devices.

Difference:

- ALU performs computations, while the Control Unit manages the execution and control of instructions.

---

e) Define cache mapping function.

- A **cache mapping function** determines how memory blocks are mapped to cache lines.

Key Mapping Techniques:

1. **Direct Mapping:**
    - Each memory block is mapped to a single cache line using a modulo operation.
2. **Associative Mapping:**
    - A memory block can be placed in any cache line (fully flexible).
3. **Set-Associative Mapping:**
    - Combines both approaches. Cache lines are divided into sets, and a memory block can map to any line within a set.

Example (Given in question):

- A computer system uses **four-way set-associative mapping** with 128-byte cache and 32-bit addresses.
    - Divide address into **tag**, **set index**, and **block offset**.
    - Use these fields to determine cache location.

Diagram Explanation:

- The diagram would include blocks for tag comparison, set index lookup, and a block offset for byte access.

Second Section

a) Mention five steps of instruction execution cycles performed by CPU.

1. **Fetch:** Retrieve the instruction from memory.
2. **Decode:** Interpret the instruction (opcode and operands).
3. **Execute:** Perform the operation (ALU computation, data transfer, etc.).
4. **Memory Access:** Access memory if the instruction involves load/store operations.
5. **Write Back:** Write the result back to a register or memory.

---

b) Discuss the difference between user-visible registers and control-status registers.

1. **User-Visible Registers:**
   - Accessible directly by programs.
   - Examples: General-purpose registers, data registers, and address registers.
2. **Control-Status Registers:**
   - Used by the CPU to control operations and track status.
   - Examples: Program Counter (PC), Instruction Register (IR), Status Register (flags).

Difference:

- User-visible registers are for program data manipulation, while control-status registers are for CPU operation management.

---

c) Explain the concept of program execution.

- Program execution involves fetching, decoding, and executing instructions stored in memory.
- It follows a sequence of instruction cycles:
  - Fetch → Decode → Execute → Write Back.
- During execution, the **PC** advances, and operations are performed by ALU and registers.

---

d) Define cache mapping function and explain the differences among direct, associative, and set-associative mappings.

Cache Mapping Types:

1. **Direct Mapping:**
   - One memory block maps to one specific cache line.
   - Formula: `Cache Line = (Memory Block Address) MOD (Number of Cache Lines).`
2. **Associative Mapping:**
   - Memory block can go to any cache line.
   - Requires tag comparison for all cache lines.
3. **Set-Associative Mapping:**
   - Cache is divided into sets, and each set has multiple lines.
   - Formula: `Set Number = (Memory Block Address) MOD (Number of Sets).`

Difference Summary:

Mapping Type	Flexibility	Complexity	Cost
Direct	Low	Simple	Low
Associative	High	Complex	High
Set-Associative	Medium	Moderate	Moderate

---

e) What do you understand by the term pipelining?

- **Pipelining** is a technique used in CPUs to improve instruction throughput by overlapping instruction execution.
- Each instruction cycle is divided into stages (Fetch, Decode, Execute, etc.).
- Multiple instructions are executed simultaneously but in different stages.

Example:

Cycle	Instruction 1	Instruction 2	Instruction 3
1	Fetch		
2	Decode	Fetch	
3	Execute	Decode	Fetch

Benefit: Increases CPU performance by executing more instructions per unit time.