1. Explain the three major types of computer architecture - System Design, ISA (Instruction Set Architecture) & Microarchitecture.

   Von Neumann Architecture:

   - Structure: It uses a single shared memory space for both instructions (program code) and data. The system operates by fetching instructions and data over a single bus (a communication pathway for transmitting data).
   - Working: The CPU fetches instructions and data one at a time, in sequential order, which can cause delays due to the limitation of using a single bus. This sequential nature of instruction execution can lead to a performance bottleneck, commonly called the "Von Neumann bottleneck."
   - Instruction Set: Von Neumann architecture is commonly associated with general-purpose processors, such as those using the x86 or ARM instruction sets. These instruction sets are designed to handle a wide range of operations, from arithmetic and logic to control flow (branching, loops) and memory management. It Uses versatile, general-purpose instruction sets like x86 (CISC) and ARM (RISC), suitable for various applications.
   - Advantages: It simplifies hardware design, making it cheaper and easier to build. It's widely used in general-purpose computers.
   - Disadvantages: The need to fetch both data and instructions from the same memory slows down processing, especially for complex programs.

     Harvard Architecture:

   - Structure: Harvard architecture has physically separate memory for instructions and data. This allows the CPU to simultaneously access data and instructions over different buses, leading to faster execution.
   - Working: Because instructions and data are stored separately, the CPU can fetch an instruction and read/write data at the same time, improving efficiency.
   - Instruction Set: The Harvard architecture is often found in specialized systems like DSPs (Digital Signal Processors) and microcontrollers, which use more specialized instruction sets designed for specific tasks. These processors often use RISC architectures, which have simpler and more efficient instruction sets. It utilizes simpler, often application-specific instruction sets like RISC-V and DSP-specific sets, tailored for efficient, specialized processing.
   - Advantages: Faster execution due to simultaneous fetching of instructions and data, making it well-suited for specialised systems, such as DSP (Digital Signal Processing) and embedded systems.
   - Disadvantages: The hardware design is more complex and expensive due to the need for separate memory and buses.

     Modified Harvard Architecture:

   - Structure: This is a hybrid of Von Neumann and Harvard architectures. It generally uses separate memory caches for instructions and data (like Harvard), but allows shared memory access in certain conditions (like Von Neumann).
   - Working: It enhances performance by allowing parallel access to instructions and data while providing flexibility when accessing shared memory, leading to more efficient resource use.
   - Instruction Set: It combines general-purpose instruction sets with specialized instructions like SIMD (x86) or NEON (ARM) to balance flexibility and performance across tasks.
   - Advantages: Offers a balance between speed and design complexity. Modern CPUs (e.g., Intel and ARM) often use this approach because it optimizes performance and flexibility.

- Disadvantages: While more efficient, the modified design still introduces some complexity compared to pure Von Neumann architectures, though it's less than the traditional Harvard approach.

2. Explain Von Neumann and Harvard architecture in detail.
   Von Neumann Architecture:
   - Design: Named after mathematician John von Neumann, this architecture uses a single shared memory for both instructions (program code) and data. The processor fetches both instructions and data from this shared memory using the same bus (data communication pathway).
   - Working: The CPU executes instructions in a sequential manner—fetching an instruction, decoding it, fetching any required data, and then executing it. This process can lead to delays because the system cannot fetch instructions and data at the same time, causing what is known as the "Von Neumann bottleneck."
   - Simpler design but prone to bottlenecks, commonly used in general-purpose computers.

   Harvard Architecture:
   - Design: The Harvard architecture uses two separate memory spaces—one for program instructions and another for data. This allows the CPU to access instructions and data simultaneously using separate buses, which improves processing speed.
   - Working: The CPU can fetch instructions from instruction memory and data from data memory at the same time, significantly reducing the chances of delays. This architecture is commonly used in specialized systems like digital signal processors (DSPs) and embedded systems, where real-time performance is critical.
   - More efficient, with faster execution, commonly used in specialized systems like embedded devices and DSPs.

3. List down the pros & cons of Von Neumann and Harvard architecture.
   Von Neumann architecture
   Advantages
   - Simpler hardware design, which reduces costs.
   - Easier to program for general-purpose applications.

   Disadvantages

   - Slower execution due to the shared memory for instructions and data, leading to potential bottlenecks.
   - Cannot fetch data and instructions simultaneously, limiting performance.

   Harvard architecture

   Advantages

   - Faster execution due to simultaneous instruction and data access.
   - Minimizes the risk of bottlenecks, making it highly efficient for real-time tasks.

   Disadvantages

- More complex hardware design leads to higher costs.
- Programming can be more challenging due to the need for separate memory management.

4. What is Modified Harvard Architecture?
   This is a hybrid of Von Neumann and Harvard architectures. It generally uses separate memory caches for instructions and data (like Harvard), but allows shared memory access in certain conditions (like Von Neumann).
5. Explain RISC and CISC architecture in detail.
   RISC
   Design: RISC architecture focuses on a streamlined instruction set with a limited number of simple instructions. The goal is to execute each instruction in a single clock cycle, which simplifies the CPU design and improves performance.
   Characteristics:
   - Simple Instructions: Each instruction is designed to perform a small, basic task. This simplicity allows for faster execution and easier optimization.
   - Registers: RISC processors rely heavily on a large number of general-purpose registers to perform operations. This minimizes the need for frequent memory access, enhancing speed.
   - Load/Store Architecture: Only load and store instructions interact with memory; other instructions operate on registers. This separates data movement from data processing, which streamlines execution.

   CISC
   Design: CISC architecture uses a more extensive and complex set of instructions, allowing single instructions to perform multiple tasks or operations. This approach aims to reduce the number of instructions per program by providing more powerful, multifunctional instructions.
   Characteristics:
   - Complex Instructions: Each instruction can perform multiple operations, such as arithmetic, memory access, and control flow, within a single instruction.
   - Fewer Instructions: More complex instructions mean that fewer instructions are needed to accomplish the same task compared to RISC. This can reduce program size and potentially improve performance.
   - Microcode: Many CISC processors use microcode to break down complex instructions into simpler internal operations, making it easier to implement more complex functionalities.
6. List down the pros and cons of RISC Vs CISC.
   RISC
   Pros
   - Faster instruction execution due to simpler instructions.
   - More predictable performance and easier optimization by compilers.

   Cons
   - Requires more instructions to perform complex tasks, which can result in larger program sizes.
   - May need more advanced compiler support to optimize instruction sequences.

   CISC
   Pros
   - Reduced program size due to fewer, more powerful instructions.
   - Potentially more efficient for certain complex operations, as fewer instructions are needed.

   Cons
   - More complex and slower instruction decoding due to the diverse range of instructions.
   - This can lead to variable performance due to the complexity of instructions and execution.

7. What is a RISC-V processor?

RISC-V is an open-source instruction set architecture (ISA) based on the RISC principles. It provides a minimalistic and modular design, allowing for customization and extension.

8. Why do we use RISC-V?
   We use RISC-V for its flexibility, simplicity, and open-source nature, which enable cost-effective and customizable processor designs.
9. What is SIMD architecture in GPU?
   SIMD (Single Instruction, Multiple Data) architecture in GPUs allows the simultaneous execution of the same instruction on multiple data points, enhancing parallel processing efficiency.
10. What are the advantages of using SIMD over SISD?
    SIMD (Single Instruction, Multiple Data) provides higher throughput and efficiency than SISD (Single Instruction, Single Data) by executing the same operation on multiple data points simultaneously