**Quiz**

**1 .** Discuss the difference between logical level design and physical level design in relation to hardware software partitioning

1. **Logical Level Design:**
   * **Abstraction and Functionality:** Logical level design focuses on the high-level abstraction of the system and its functionality. It involves identifying and defining the tasks, functions, and algorithms that the system needs to perform.
   * **Algorithmic Decisions:** At this level, decisions regarding algorithms, control flow, and data flow are made. The emphasis is on what needs to be done, without specifying how it will be implemented in terms of hardware or software.
   * **Hardware-Independent:** Logical level design is largely hardware-independent. It aims to capture the logical structure and behavior of the system without delving into the specifics of the underlying hardware platform.
2. **Physical Level Design:**
   * **Implementation Details:** Physical level design deals with the concrete implementation details of the system. It involves mapping the functions and algorithms identified in the logical design onto specific hardware and software components.
   * **Hardware and Software Allocation:** This stage determines which functions will be implemented in hardware and which in software. It involves deciding whether a particular task is better suited for a dedicated hardware module (e.g., FPGA) or a software module running on a general-purpose processor.
   * **Performance Considerations:** Physical level design takes into account factors like execution speed, power consumption, and resource utilization. It involves making trade-offs between hardware and software to achieve optimal system performance.
3. **Relationship between Logical and Physical Levels:**
   * **Iteration:** The process of hardware-software partitioning often involves an iterative approach. Designers may need to go back and forth between logical and physical levels to refine the partitioning decisions based on performance analysis and other constraints.
   * **Co-Design:** Achieving an effective hardware-software partitioning requires a coordinated effort between logical and physical level design. Co-design methodologies aim to bridge the gap between these levels, ensuring that the system's overall design objectives are met.