4.3 Input-Output organization and multiprocessor:

Peripheral devices

Peripheral devices are hardware components external to the CPU and main memory, used for input, output, and storage operations. These devices can be classified into three categories:
Input Devices: Devices like keyboards, mice, and scanners that allow users to input data into the system.
Output Devices: Devices such as monitors, printers, and speakers that present processed data to the user.
Storage Devices: Includes hard drives, SSDs, and optical drives, which store data persistently.
Peripheral devices are connected to the CPU via interfaces and require drivers for communication. Data transfer between the CPU and these devices can occur synchronously or asynchronously, depending on the device's operating characteristics.

I/O modules Input-output interface

I/O modules serve as intermediaries between the CPU and peripheral devices, handling the complexities of communication and data transfer. They perform several functions:
Device Identification: Recognizing specific devices connected to the system.
Data Buffering: Temporarily storing data to handle speed mismatches between the CPU and the device.
Error Detection and Correction: Ensuring data integrity during transfer.
The I/O interface provides the hardware and software mechanisms necessary for communication between the CPU and peripherals.
Key types of interfaces include:
Programmed I/O: The CPU controls data transfer through explicit instructions.
Interrupt-Driven I/O: The device interrupts the CPU when it is ready for data transfer, reducing CPU idle time.
Direct Memory Access (DMA): A dedicated controller transfers data directly between memory and the device without CPU intervention.

Modes of transfer

Data transfer between the CPU and I/O devices can occur in several modes:
Synchronous Transfer: Data is transferred at fixed intervals, with synchronization between the device and CPU.
Asynchronous Transfer: Data is transferred without fixed timing, using handshaking signals for synchronization.
Interrupt-Driven Transfer: The device signals the CPU via an interrupt when it is ready, allowing the CPU to perform other tasks in the meantime.
DMA Transfer: A high-speed mode where a DMA controller handles the transfer, bypassing the CPU to improve efficiency.

Direct Memory access

DMA enables data transfer between peripherals and main memory without CPU intervention, freeing the CPU for other operations. The DMA controller takes over the bus, performs the transfer, and notifies the CPU upon completion via an interrupt.
Key steps in DMA operation:
The CPU initializes the DMA controller with the memory address, data size, and device information.
The DMA controller requests bus access from the CPU.
Upon gaining control of the bus, the controller transfers data directly between the memory and the device.
Once the transfer is complete, the controller generates an interrupt to inform the CPU.
The DMA transfer rate is higher than that of programmed or interrupt-driven I/O, making it ideal for high-speed peripherals like disk drives and network

adapters.

Characteristics of multiprocessors

Multiprocessors consist of two or more CPUs working together to execute tasks. They can be classified into:
Symmetric Multiprocessing (SMP): All processors share the same memory and are equally capable of executing instructions.
Asymmetric Multiprocessing (AMP): Processors have specialized roles, with one serving as the master and the others as slaves.
Key characteristics include:
Parallelism: Multiprocessors achieve high performance by executing multiple instructions simultaneously.
Scalability: Adding more processors can increase the system's processing capability.
Shared Resources: Memory, I/O devices, and the bus are shared among processors, requiring efficient resource management.

Interconnection Structure

The interconnection structure determines how processors, memory, and I/O devices communicate in a multiprocessor system. Common structures include:
Bus-Based Interconnection: All components are connected via a shared bus. While simple and cost-effective, it suffers from bandwidth limitations as the number of processors increases.
Crossbar Switch: Provides a direct connection between any pair of components, offering high performance at the cost of increased complexity.
Multistage Interconnection Networks: Use multiple switching stages to connect components, balancing cost and performance. Examples include Omega and Butterfly networks.

Inter-processor Communication and synchronization

Inter-processor communication is essential in multiprocessors to coordinate tasks and share data. Communication can occur via:

Shared Memory: Processors communicate by reading and writing to a common memory space. Mechanisms like semaphores and mutexes are used to prevent conflicts.
Message Passing: Processors exchange messages through a communication network. This method is common in distributed systems.
Synchronization ensures processors work together without conflicts. Techniques include:

Locks: Ensure only one processor accesses a critical section at a time.
Barriers: Processors wait until all others reach a certain point before proceeding.
Atomic Operations: Hardware-supported operations that execute without interruption, such as test-and-set or compare-and-swap.
Formulas used in synchronization include:
ð ‘‡ð ‘ =ð ‘‡1+ð ‘‡ð ‘ ð ‘¦ð ‘›ð ‘ â„ /ð ‘›Tsâ ‹ Where ð ‘‡ð ‘ is the synchronized execution time, ð ‘‡1 is the execution time on a single processor, T synch is the synchronization overhead, and n is the number of processors.