



SNS COLLEGE OF TECHNOLOGY

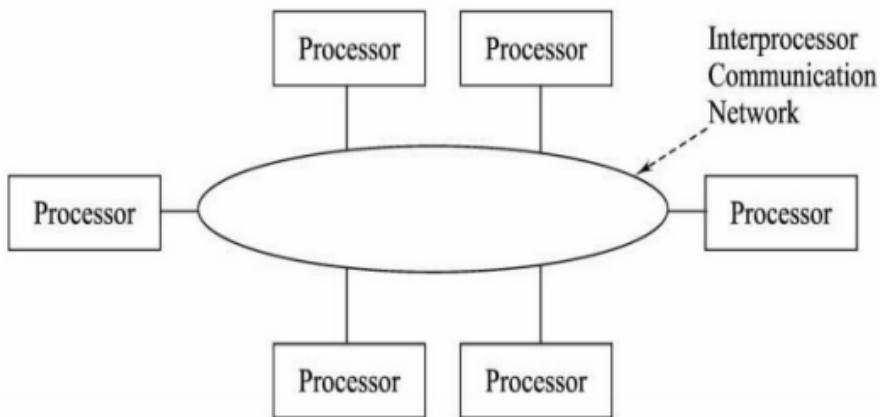
(An Autonomous Institution)
COIMBATORE – 641035



DEPARTMENT OF MECHATRONICS ENGINEERING

MULTIPROCESSORS

Multiprocessors refer to systems with two or more central processing units (CPUs) that work together to execute tasks. These processors can operate simultaneously, sharing tasks or executing separate tasks independently, improving system performance and efficiency. Multiprocessor systems are common in high-performance computing, data centers, and embedded systems requiring real-time, parallel processing.



Types of Multiprocessor Systems

1. Symmetric Multiprocessing (SMP):

- **Definition:** All processors share a single memory space and have equal access to all resources, including memory and I/O.
- **Characteristics:** Processors run the same operating system instance, and tasks can be distributed evenly across all CPUs.
- **Use Case:** General-purpose computing, servers, and embedded systems where load balancing is needed.

2. Asymmetric Multiprocessing (AMP):

- **Definition:** Each processor may have dedicated memory and tasks. One processor (master) controls the others (slaves), which perform specific tasks.

- **Characteristics:** Used in systems where certain tasks are more critical and require more processing power.
- **Use Case:** Specialized embedded systems where different processors handle different functions (e.g., one for real-time processing, another for data storage).

3. Massively Parallel Processing (MPP):

- **Definition:** A large number of processors operate independently, each with its own memory and resources, connected via a high-speed interconnect.
- **Characteristics:** Each processor runs its own operating system and has its own memory, with minimal sharing of resources.
- **Use Case:** Supercomputers and large-scale scientific simulations that require massive parallelism.

Key Concepts in Multiprocessor Systems

1. Parallelism and Concurrency:

- **Parallelism:** Refers to performing multiple tasks or computations simultaneously by distributing them across multiple processors.
- **Concurrency:** Refers to the ability of a system to handle multiple tasks at the same time, but not necessarily simultaneously; tasks can overlap in execution.

2. Task Scheduling:

- In multiprocessor systems, task scheduling becomes critical to balance the workload efficiently across processors.
- **Static Scheduling:** Tasks are assigned to processors at compile time, based on pre-defined rules.
- **Dynamic Scheduling:** Tasks are assigned during runtime based on system conditions, allowing for flexibility and better load balancing.

3. Memory Architecture:

- **Shared Memory:** All processors access the same memory space, which simplifies communication but can lead to contention issues (e.g., cache coherence problems).
- **Distributed Memory:** Each processor has its own local memory, reducing contention

but increasing the complexity of data sharing and communication between processors.

4. **Cache Coherence:**

- In systems with shared memory, each processor may have its own cache. Ensuring that all processors have the most up-to-date data is known as cache coherence.
- **Protocols:** Cache coherence protocols (e.g., MESI – Modified, Exclusive, Shared, Invalid) ensure that when one processor writes data, other processors' caches are updated accordingly.

Advantages of Multiprocessor Systems

1. **Increased Throughput:**

- By utilizing multiple processors, the system can handle more tasks simultaneously, increasing overall processing capacity and speed.

2. **Improved Reliability and Fault Tolerance:**

- If one processor fails, others can take over its workload, improving the system's reliability.

3. **Scalability:**

- Multiprocessor systems can be scaled up by adding more processors to increase performance without redesigning the entire system.

4. **Parallel Processing:**

- Parallelism allows for faster execution of large, complex tasks by breaking them into smaller subtasks that can be executed concurrently.

Challenges in Multiprocessor Systems

1. **Synchronization and Communication Overhead:**

- Multiple processors must coordinate their tasks, which introduces overhead for communication and synchronization. This can sometimes reduce the expected performance gains from parallelism.

2. **Load Balancing:**

- Ensuring that the workload is evenly distributed across all processors is critical. Poor load balancing can lead to some processors being overburdened while others remain

idle.

3. **Memory Contention:**

- In shared-memory systems, multiple processors accessing the same memory can cause contention, where processors have to wait for access, reducing overall efficiency.

4. **Programming Complexity:**

- Writing software that efficiently utilizes multiple processors can be complex. Developers must consider parallelism, synchronization, data consistency, and deadlock prevention.

Use Cases of Multiprocessor Systems

1. **High-Performance Computing (HPC):**

- Multiprocessor systems are the backbone of supercomputers used for scientific simulations, weather forecasting, and data-intensive tasks.

2. **Real-Time Embedded Systems:**

- In embedded systems like automotive control units, multiprocessors are used to handle multiple functions (e.g., safety systems, navigation, entertainment) concurrently.

3. **Server Farms and Data Centers:**

- Multiprocessor systems are essential in cloud computing environments, where high availability and parallel processing of numerous tasks are required.

4. **Mobile Devices and IoT:**

- Many smartphones and IoT devices use multiprocessor architectures (e.g., ARM big.LITTLE) to balance power efficiency and performance for different tasks.

Multiprocessor vs. Multicore

- **Multiprocessor Systems:** Consist of multiple separate CPUs on one or more motherboards, each with its own resources.
- **Multicore Systems:** A single CPU chip with multiple cores, where each core acts as an independent processor. Multicore processors are commonly used in personal computers, servers, and mobile devices due to their smaller size and lower power consumption compared to multiprocessor systems.

