# Lecture 3

# Introduction II

## Taxonomies

IKC-MH.57 *Introduction to High Performance and Parallel Computing* at October 27, 2023

Dr. Cem Özdoğan
Engineering Sciences Department
İzmir Kâtip Çelebi University

# Contents

# SIMD Architecture I

- The SIMD model of parallel computing consists of two parts:
  1. a front-end computer of the usual von Neumann style,
  2. a processor array.
- Each processor in the array has a small amount of local memory where the *distributed data resides* while it is being processed in parallel.
- The similarity between serial and data parallel programming is one of the strong points of *data* parallelism.
- Processors either do nothing or exactly the same operations at the same time.
- In SIMD architecture, parallelism is exploited by applying simultaneous operations across large sets of data.
- There are two main configurations that have been used in SIMD machines.

# SIMD Architecture II

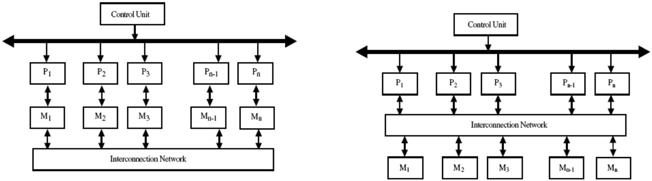
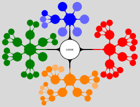



**Figure:** Two SIMD Schemes.

1. Each processor has its own local memory.
   - Processors can communicate with each other through the interconnection network.
   - If the interconnection network does not provide direct connection between a given pair of processors, then this pair can exchange data via an intermediate processor.
2. In the second SIMD scheme,
   - Processors and memory modules communicate with each other via the interconnection network.
   - Two processors can transfer data between each other via intermediate memory module(s) or possibly via intermediate processor(s).





# MIMD Architecture I

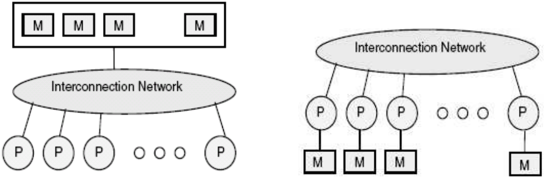



**Figure:** Two MIMD Categories; Shared Memory and Message Passing MIMD Architectures.

- It was apparent that distributed memory is the only way efficiently to increase the number of processors managed by a parallel and distributed system.
- If scalability to larger and larger systems (as measured by the number of processors) was to continue, systems had to use distributed memory techniques.

# MIMD Architecture II

Two broad categories, see Figure 2:

**Shared memory: Processors exchange information through their central shared memory**

- Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.

**Message passing: Also referred to as distributed memory. Processors exchange information through their interconnection network**

- There is no global memory, so it is necessary to *move data from one local memory to another by means of message passing.*
- This is typically done by a **Send**/**Receive pair** of commands, which must be written into the application software by a programmer
- Data copying and dealing with consistency issues.

# MIMD Architecture III

- Programming in the shared memory model was easier, and designing systems in the message passing model provided scalability.
- The distributed-shared memory (DSM) architecture began to appear in systems. In such systems,
  - memory is physically distributed; for example, the hardware architecture follows the message passing school of design,
  - but the programming model follows the shared memory school of thought.
  - Thus, the DSM machine is a *hybrid* that takes advantage of both design schools.

# Shared Memory Organization I

- A number of basic issues in the design of shared memory systems have to be taken into consideration.
- These include access control, synchronization, protection/security.
  - ***Access control*** determines which process accesses are possible to which resources.
  - ***Synchronization*** constraints limit the time of accesses from sharing processes to shared resources.
  - ***Protection*** is a system feature that prevents processes from making arbitrary access to resources belonging to other processes.
- The simplest shared memory system consists of one memory module that can be accessed from two processors.
- Requests arrive at the memory module through its two ports.

## Shared Memory Organization II

Depending on the interconnection network, a shared memory system leads to systems can be classified as:

- **Uniform Memory Access (UMA)**. A shared memory is accessible by all processors through an interconnection network in the same way a single processor accesses its memory.
  - Therefore, all processors have equal access time to any memory location.
- **Nonuniform Memory Access (NUMA)**. Each processor has part of the shared memory attached.
  - However, the access time to modules depends on the distance to the processor. This results in a nonuniform memory access time.
- **Cache-Only Memory Architecture (COMA)**. Similar to the NUMA, each processor has part of the shared memory in the COMA.
  - However, in this case the shared memory consists of cache memory.
  - A COMA system requires that data be migrated to the processor requesting it.

# Message Passing Organization I

- Message passing systems are a class of multiprocessors in which each processor has access to its own local memory.
- Unlike shared memory systems, communications in message passing systems are performed via send and receive operations.
- Nodes are typically able to store messages in buffers (temporary memory locations where messages wait until they can be sent or received), and perform send/receive operations at the same time as processing.
- The processing units of a message passing system may be connected in a variety of ways ranging from architecture-specific interconnection structures to geographically dispersed networks.

# Message Passing Organization II



Two important design factors must be considered in designing interconnection networks for message passing systems. These are the link bandwidth and the network latency.

1. The ***link bandwidth*** is defined as the number of bits that can be transmitted per unit time (bits/s).
2. The ***network latency*** is defined as the time to complete a message transfer.