Computer Architecture

Iolanthe II racing in Waitemata Harbour

MIMD Parallel Processors

Classification of Parallel Processors

- Flynn's Taxonomy
  - Classifies according to instruction and data stream
- Single Instruction Single Data
  - Sequential processors
- Single Instruction Multiple Data
  - CM-2 – multiple small processors
  - Vector processors
  - Parts of commercial processors - MMX, Altivec
- Multiple Instruction Single Data
  - ?
- Multiple Instruction Multiple Data
  - General Parallel Processors

MIMD systems

- It supports multiple simultaneous instruction streams operating on multiple data streams.
- Thus, MIMD systems typically consist of a collection of fully independent processing units or cores, each of which has its own control unit and its own ALU.
- Unlike SIMD systems, MIMD systems are **asynchronous**.
  - The processors can operate at their own place.
- In many MIMD systems, there is **no global clock** and there may be **no relation between the system times** on 2 different processors.
  - In fact, **unless the programmer imposes some synchronization**, even if the **processors** are executing **exactly the same sequence of instructions**, at any given instant they may be **executing different statements.**

MIMD Systems

- Thread and process-level parallel architectures are typically realised by MIMD (Multiple Instruction Multiple Data) computers.
- This class of parallel computers is the most general one since it permits **autonomous operations** on a set of data by a set of processors without any architectural restrictions.
- Instruction level data-parallel architectures should satisfy several constraints in order to build massively parallel systems. For example processors in array processors, systolic architectures and cellular automata should work synchronously controlled by a common clock.

MIMD Systems

- Generally the processors are very simple in these systems and in many cases they realise only a special function (systolic arrays, neural networks, associative processors, etc.).

MIMD Architectures

- In the early 80's small systems, incorporating only tens of processors were typical. The appearance of Transputer in the mid-eighties caused a great breakthrough in the spread of MIMD parallel computers and even more resulted in the general acceptance of parallel processing as the technology of future computers.
- By the end of the eighties mid-scale MIMD computers containing several hundreds of processors become generally available.
- The current generation of MIMD computers aim at the range of massively parallel systems containing over 1000 processors. These systems are often called **scalable parallel computers**

MIMD Architectures Concepts : 2 Choices

- Represents a natural generalisation of the **uniprocessor** von Neumann machine which in its simplest form consists of a single processor connected to a single memory module.

- Goal is to extend this architecture to contain multiple processors and memory modules basically two alternative choices are available :
  - Distributed Memory MIMD Architectures
  - Shared Memory MIMD Architectures

1. Distributed Memory MIMD Architectures

- The first possible approach is to replicate the processor/memory(cache) pairs and to connect them via an interconnection network. The processor/memory pair is called processing element(PE) and they work more or less independently of each other.
- Whenever interaction is necessary among the PEs they send messages to each other.
  - Or by using **special functions** that provide access to the memory of another processor.

1. Distributed Memory MIMD Architectures

- None of the PEs can ever access directly the memory module of another PE. This class of MIMD machines are called the Distributed Memory MIMD Architectures or Message-Passing MIMD Architectures.

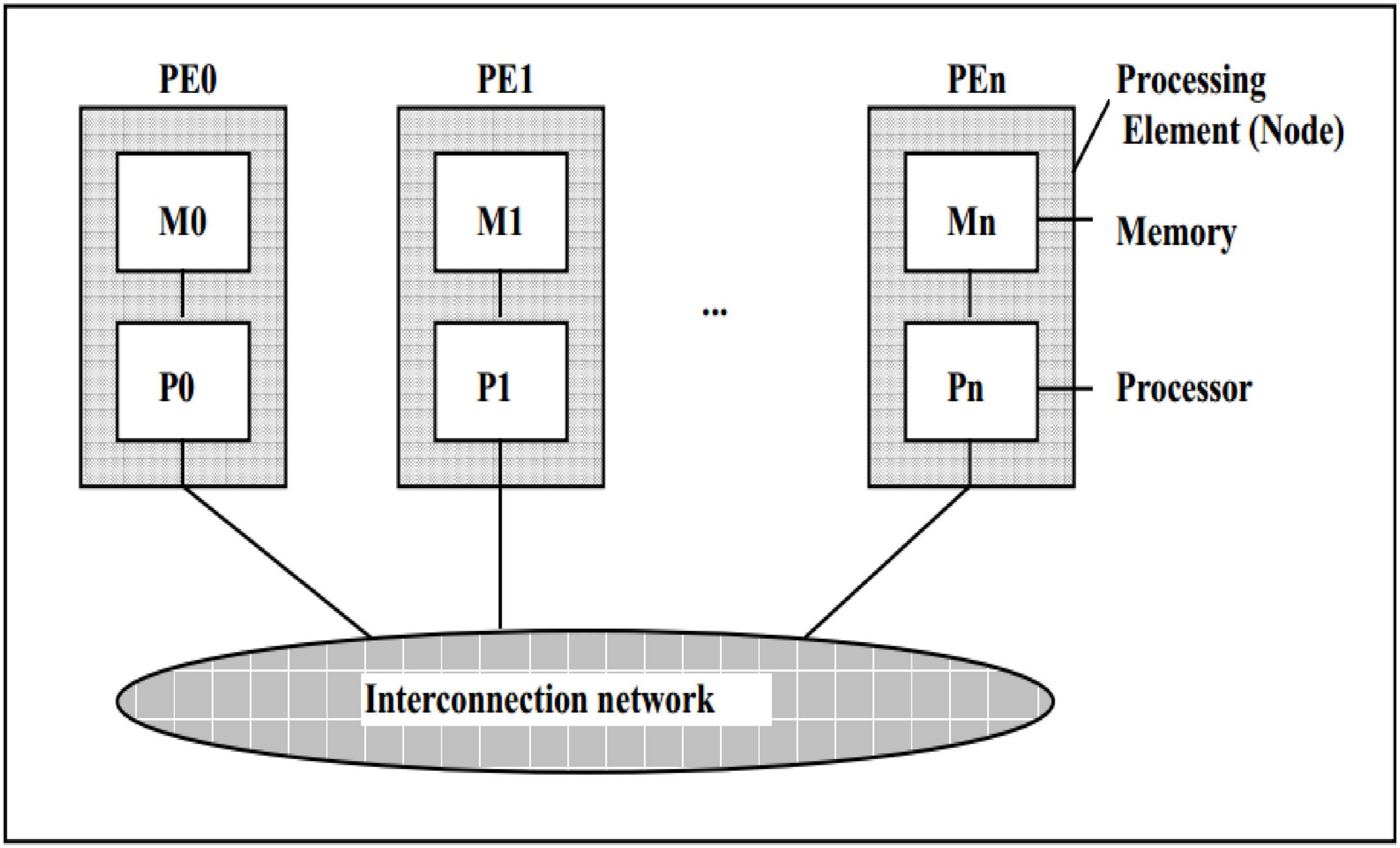

Figure 1. Structure of Distributed Memory MIMD Architectures

2. Shared Memory MIMD Architectures

- The second alternative approach is to create a set of processors and memory modules. Any processor can directly access any memory modules via an interconnection network as it is shown in Figure 2(next slide) . The set of memory modules defines a global address space which is shared among the processors.
- The name of this kind of parallel machines is Shared Memory MIMD Architectures and this arrangement of processors and memory is called the dance-hall shared memory system.

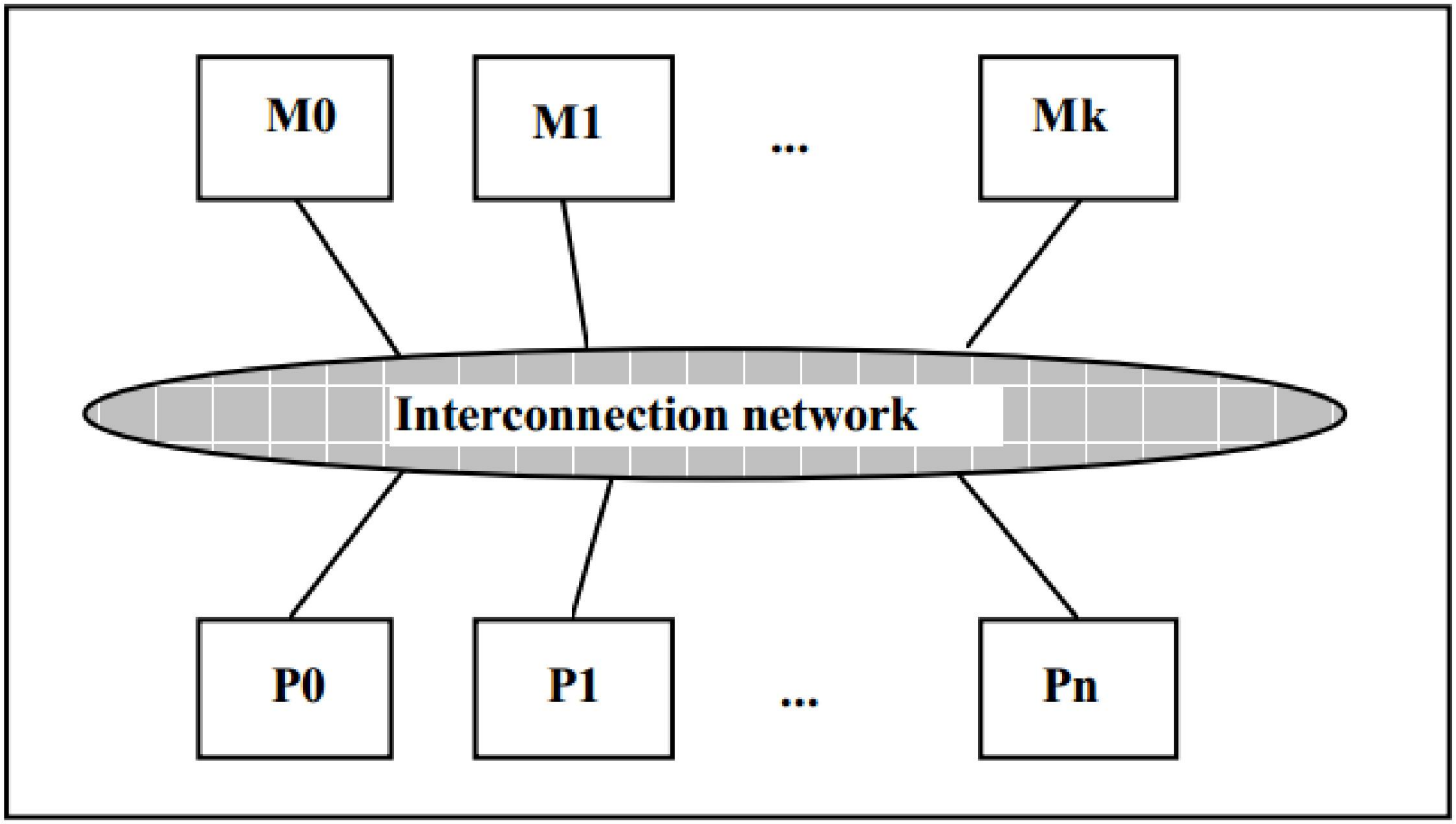

Figure 2. Structure of Shared Memory MIMD Architectures

1. Shared Memory MIMD Systems

Uniform memory access, or UMA, system

- In shared-memory systems with multiple multicore processors, the interconnect can connect all the processors directly to main memory.
- The time to access all the memory locations will be the same for all the cores.
- UMA systems are usually easier to program, since the programmer doesn't need to worry about different access times for different memory locations.

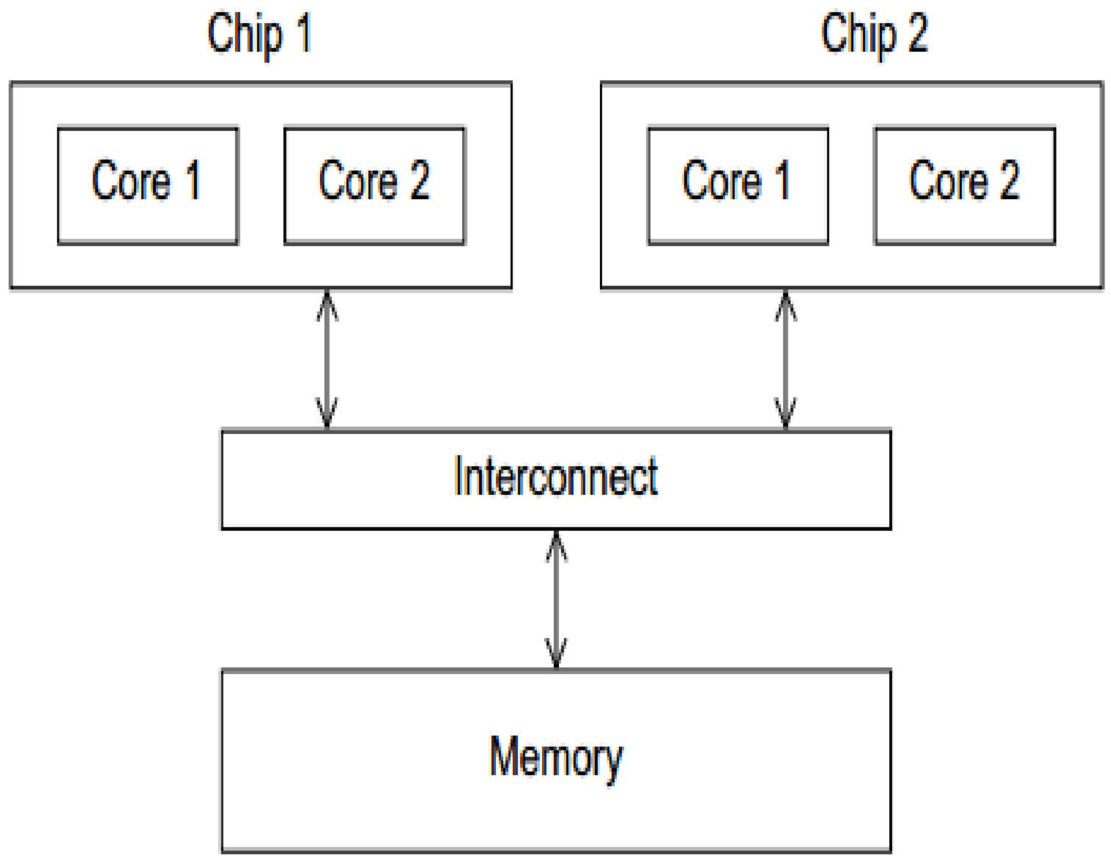

FIGURE 2.5

A UMA multicore system

2. Shared Memory MIMD Systems

nonuniform memory access, or NUMA, system

- In shared-memory systems with multiple multicore processors, each processor can have a direct connection to a block of main memory, and the processors can access each others' blocks of main memory through **special hardware** built into the processors.
- A memory location to which a core is directly connected can be accessed more quickly than a memory location that must be accessed through another chip.
- NUMA systems have the potential to use larger amounts of memory than UMA systems.

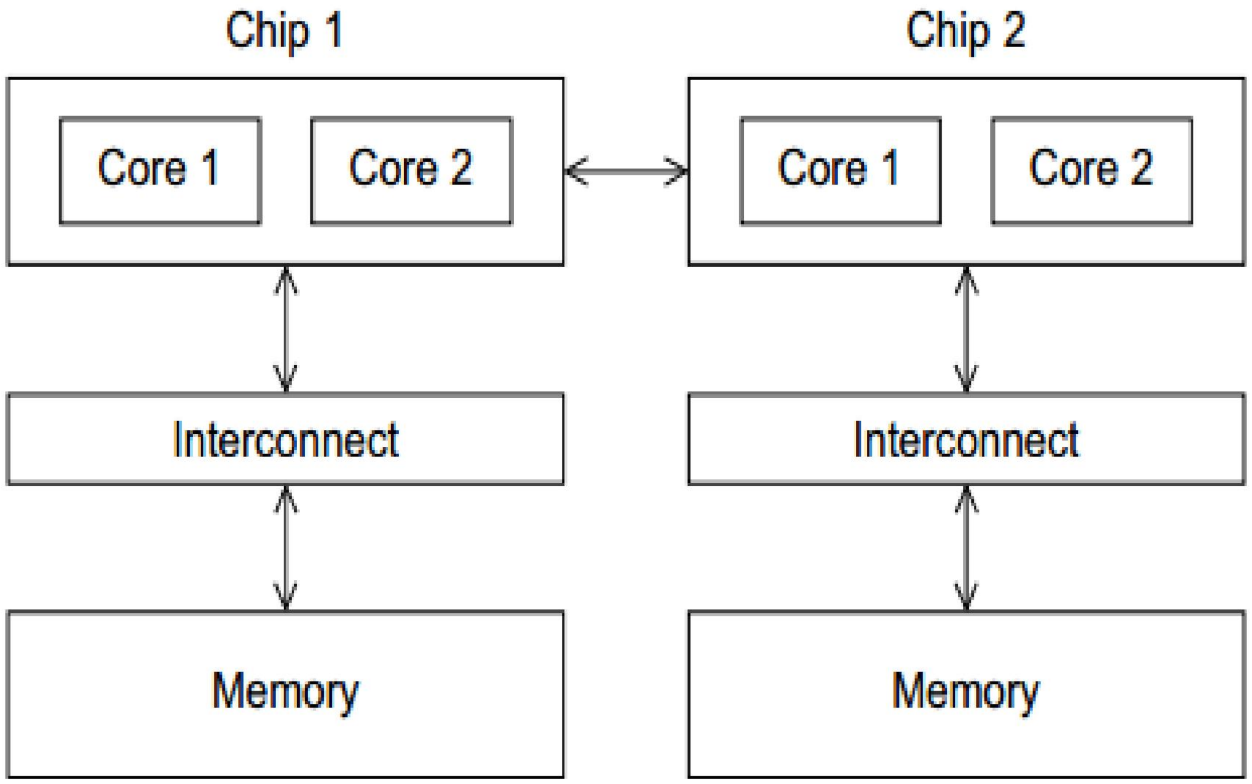

FIGURE 2.6

A NUMA multicore system

Multi-processor (shared memory system): Problems

- *Memory Access Time*
  - can be a bottleneck even in a single-processor system (unified cache)
- *Contention for Memory*
  - two or more processors want to access a location in the same block at the same time (hot spot problem).
- *Contention for Communication*
  - processors should share and use exclusively elements of the Interconnection Network

- Result: long latency-time, idle processors, nonscalable system

Distributed Memory MIMD Systems

Distributed Memory Systems

- The most widely available distributed-memory systems are called **clusters**. They are composed of a collection of commodity systems.

  for example, PCs—connected by a commodity *interconnection network*—for example, Ethernet.
- In fact, the nodes of these systems, the individual computational units joined together by the communication network, are usually shared-memory systems with one or more multicore processors.
- To distinguish such systems from pure distributed-memory systems, they are sometimes called hybrid systems.
- Nowadays, it's usually understood that *a cluster will have shared-memory nodes*.

- The grid provides the infrastructure necessary to turn large networks of geographically distributed computers into a unified distributed-memory system.
- In general, such a system will be heterogeneous, that is, the individual nodes may be built from different types of hardware.

- Distributed Memory MIMD Architectures are often simply called multicomputers while Shared Memory MIMD Architectures are shortly referred as multiprocessors.
- In both architecture types one of the main design considerations is how to construct the interconnection network in order to reduce message traffic and memory latency.

References

1. http://eclass.uoa.gr/modules/document/index.php?course=D186&download=/4c2b2d65qpn2/4c2b2d658h0l/4c2b2d65lr1o.pdf
2. TextBook