Parallel Platforms & Programming Models

Outline

1. Parallel platforms (~hardware) and programming models (~software)
   - Note: Parallel machine may or may not be tightly coupled to programming model

2. Data dependence

- Reading: Kumar – ch 1; Hwang – ch1; Foster – ch 1

Flavors of Parallelism

- Data parallelism:
  - Definition?
- Task parallelism:
  - Definition?

Data and Task Parallelization

- Data parallel:

```
for (i=0;i<1000;i++)
    a[i]=b[i]+c[i];
```

- Task parallel:

```
for (i=0;i<1000;i++)  /*block 1 */
    b[i+1]=b[i]+c[i]
…
for (j=0;j<5;j++) /*block 2*/
   a[j+1]=a[j]+d[j];
```

Parallel Platforms

- Basic components of any architecture:
  - Processors and memory (processing units)
  - Interconnect network
- Logic classification based on:
  - Control mechanism (Flynn's Taxonomy)
    - SISD (Single Instruction Single Datastream)
    - <u>SIMD</u> (Single Instruction Multiple Datastream)
    - MISD (Multiple Instruction Single Datastream)
    - <u>MIMD</u> (Multiple Instruction Multiple Datastream)
  - Address space organization
    - Shared Address Space
    - Distributed Address Space

Parallel Platforms based on Flynn's Taxonomy

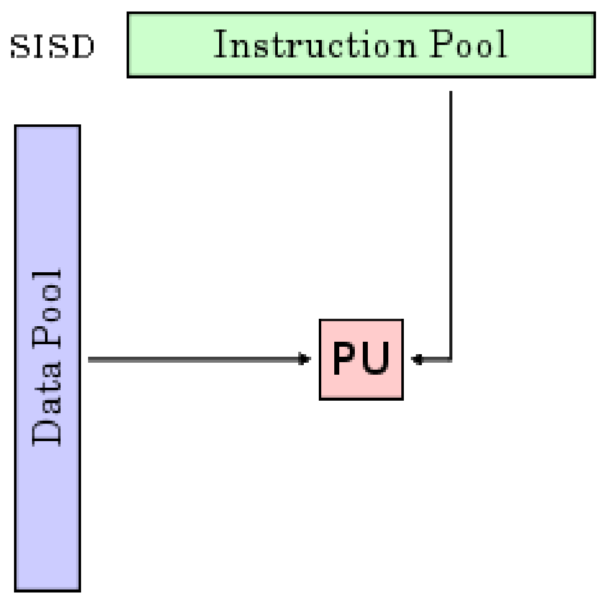
Flynn's Taxonomy

SISD **Single Instruction Stream** **Single Data Stream**	**SIMD** **Single Instruction Stream** **Multiple Data Stream**
MISD **Multiple Instruction Stream** **Single Data Stream**	**MIMD** **Multiple Instruction Stream** **Multiple Data Stream**

	Single Data	Multiple Data
Single Instruction	*SISD* typical thread	*SIMD* vector processors GPUs SSE instructions
Multiple Instruction	*MISD* rare possibly set of filters	*MIMD* cluster of computers

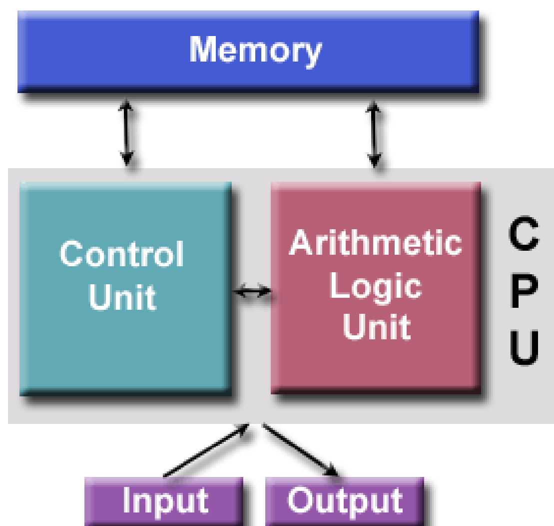
SISD Architecture

- Model of serial Von Neumann machine
- Logically, single control processor
- Includes some supercomputers, such as the 1963 CDC6600 (perhaps the first supercomputer)



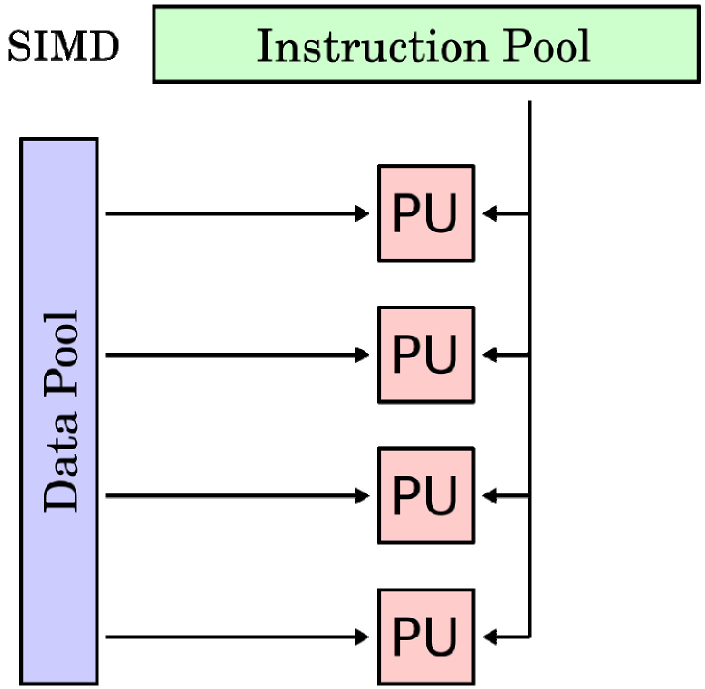
Von Neumann Architecture

- John von Neumann first authored the general requirements for an electronic computer in 1945
- Aka "stored-program computer"
  - Both program inst. and data are kept in electronic memory
- Since then, all computers have followed this basic design
- Four main components: memory, control unit, ALU, I/O



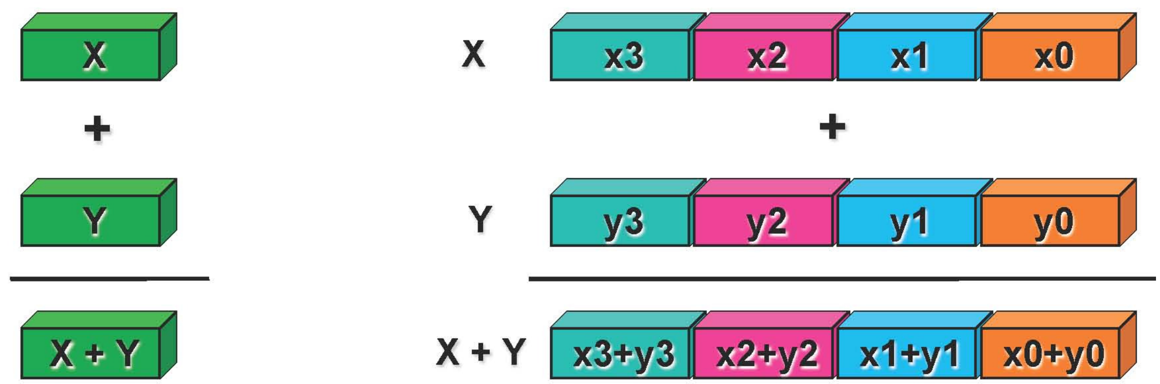
SIMD Architecture

- Parallelism achieved by dividing data among the processors
  - Multiple processors execute the same instruction
  - Data that each processor sees may be different
  - Individual processors can be turned on/off at each cycle ("masking")
- Examples:
  - Many early parallel computers like Illiac IV, Thinking Machines'CM-2, …
  - Today, GPU, vector units, and co-processors



Example of SIMD Vector Units

- Scalar processing
  - Traditional mode
  - One operation produces one result
- SIMD vector units
  - One operation produces multiple results



SIMD Drawbacks

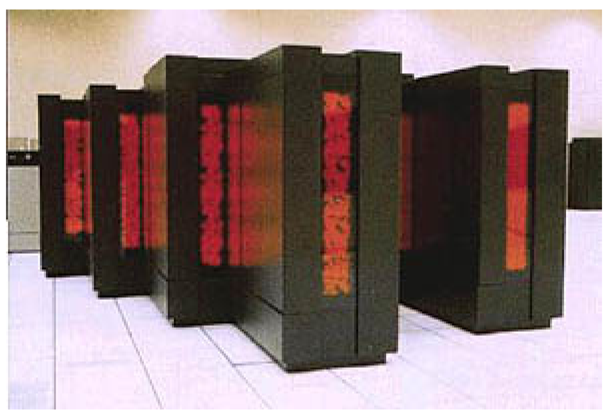
- Discussion?

The ill-fated Illiac IV

- Project started in 1965, predicted to cost $8M and provide 1000 MFLOP/S.
- Delivered to NASA Ames in 1972, cost $31M, ran first application in 1976, performed 15 MFLOP/S.
- 64 processors, 13-MHz clock, 1MB memory

Thinking Machine CM2

- CM2 (1990, built by Thinking Machines Corp) had 8,192 to 65,536 one-bit processors, plus one floating-point unit.
- Data Vault provides peripheral mass storage
- Single program - all unmasked operations happened in parallel.



Vector Processors

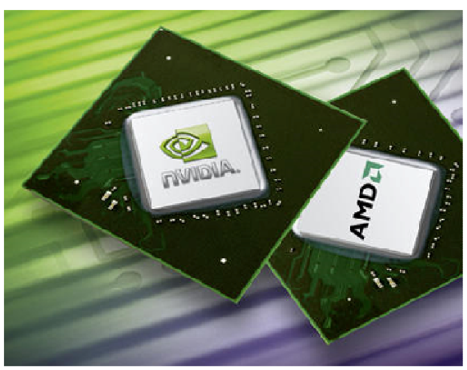
- Operate on arrays or vectors of data, while conventional CPU's operate on individual data elements or scalars
- Vector registers
  - Capable of storing a vector of operands and operating simultaneously on their contents
- Vectorized and pipelined functional units
  - The same operation is applied to each element in the vector
- Examples:
  - Cray supercomputers (X-MP, Y-MP, C90, T90, SV1, ...), Fujitsu (VPPxxx), NEC, Hitachi
  - Earth Simulator from Japan (on the TOP500 list)
  - many of these have multiple vector processors, but typically separate processors are used for separate jobs.

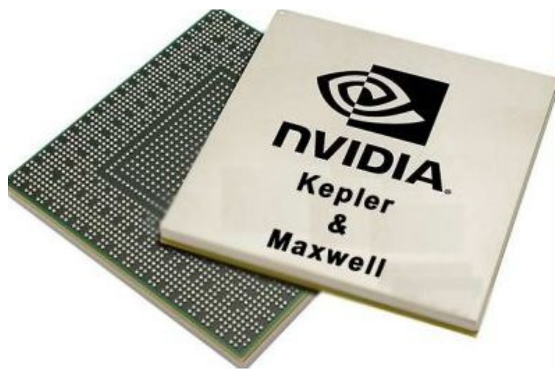
Vector Processors – Pros & Cons

- Discussion?

Graphic Processing Units (GPU)

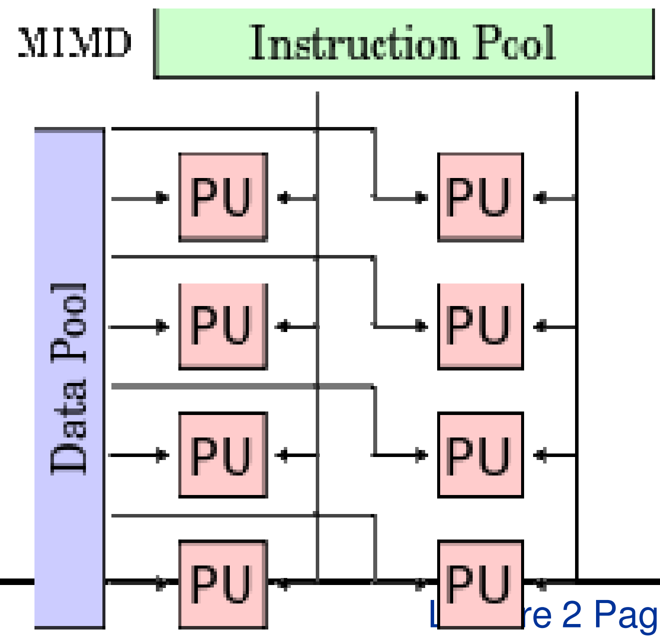
- Real time graphics application programming interfaces or API’s use points, lines, and triangles to internally represent the surface of an object
- A graphics processing pipeline converts the internal representation into an array of pixels that can be sent to a computer screen
- Several stages of this pipeline (called shader functions) are programmable
  - Typically just a few lines of C code





MIMD Architecture

- Supports multiple simultaneous instruction streams operating on multiple data streams
  - Each processor executes program independent of other processors
  - Processors operate on separate data streams
- Typically consist of a collection of fully independent PUs or cores, each of which has its own control unit and its own ALU.
- Examples:
  - Current generation systems



Examples

Tianhe-2

Mira

Titan

MISD Architectures

- Multiple Instruction Single Data
- Few (if any) actual examples of this class of parallel computer have ever existed.
- The term isn’t used (except when discussing the Flynn taxonomy) .
- Perhaps applies to pipelined computation, e.g. sonar data passing through sequence of special-purpose signal processors.

SIMD vs MIMD

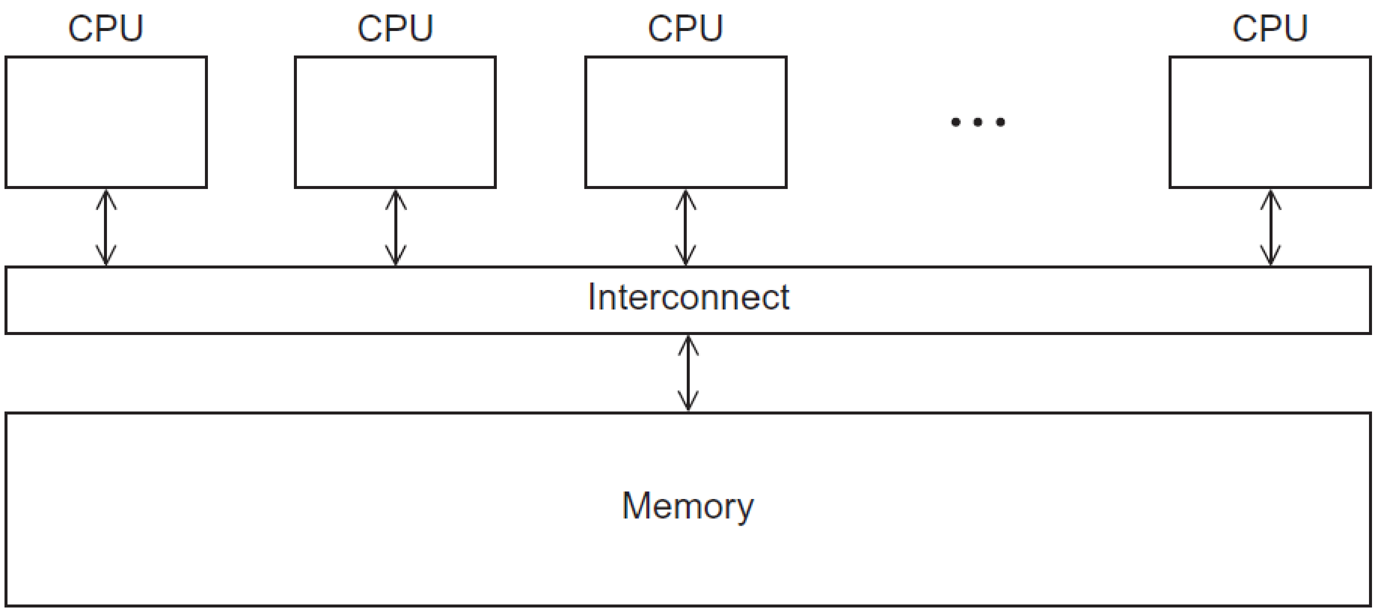
- SIMD platforms
  - ???

- MIMD platforms
  - ???

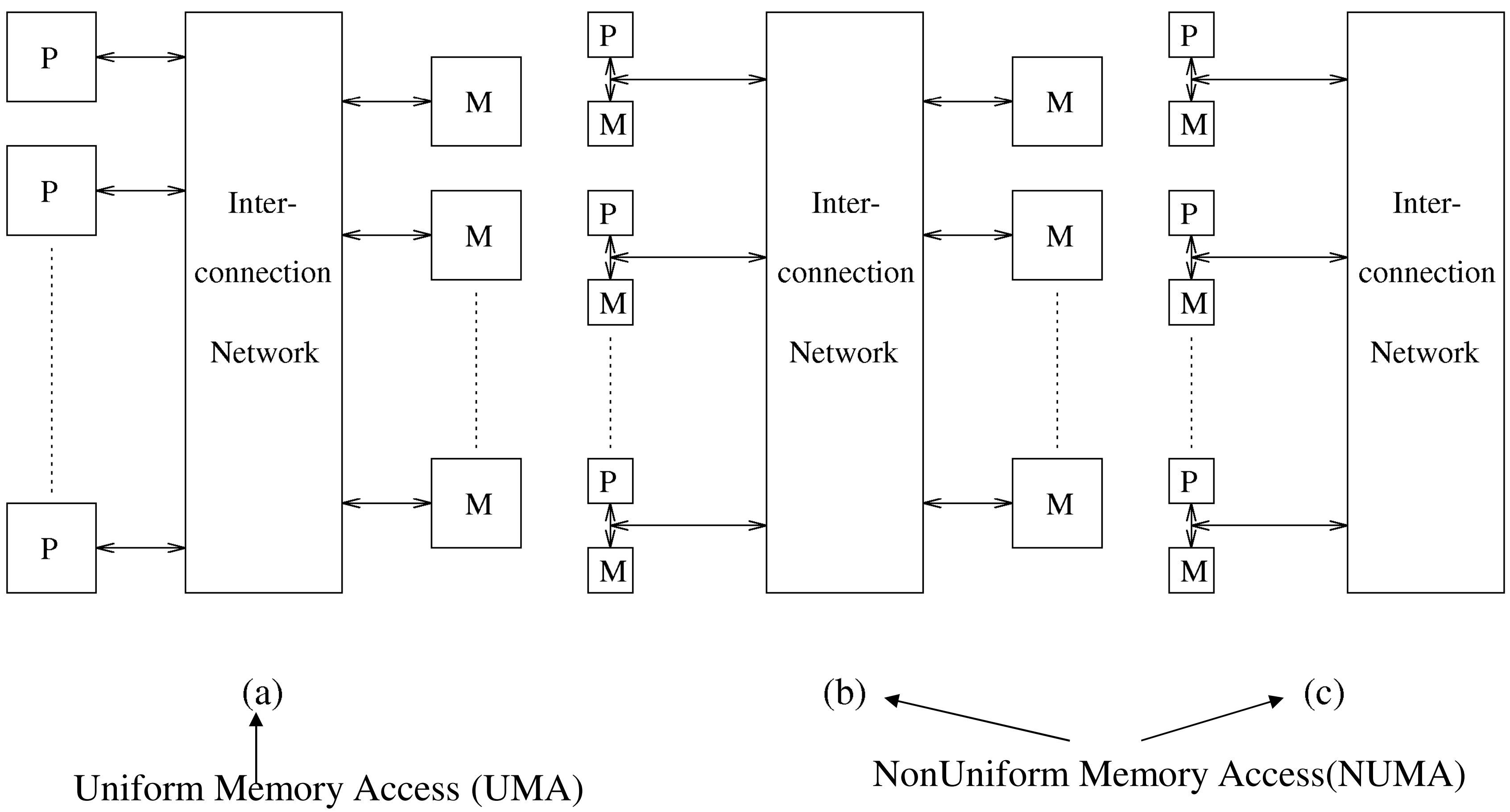
Parallel Platforms based on Address Space Organization

Shared Address Space

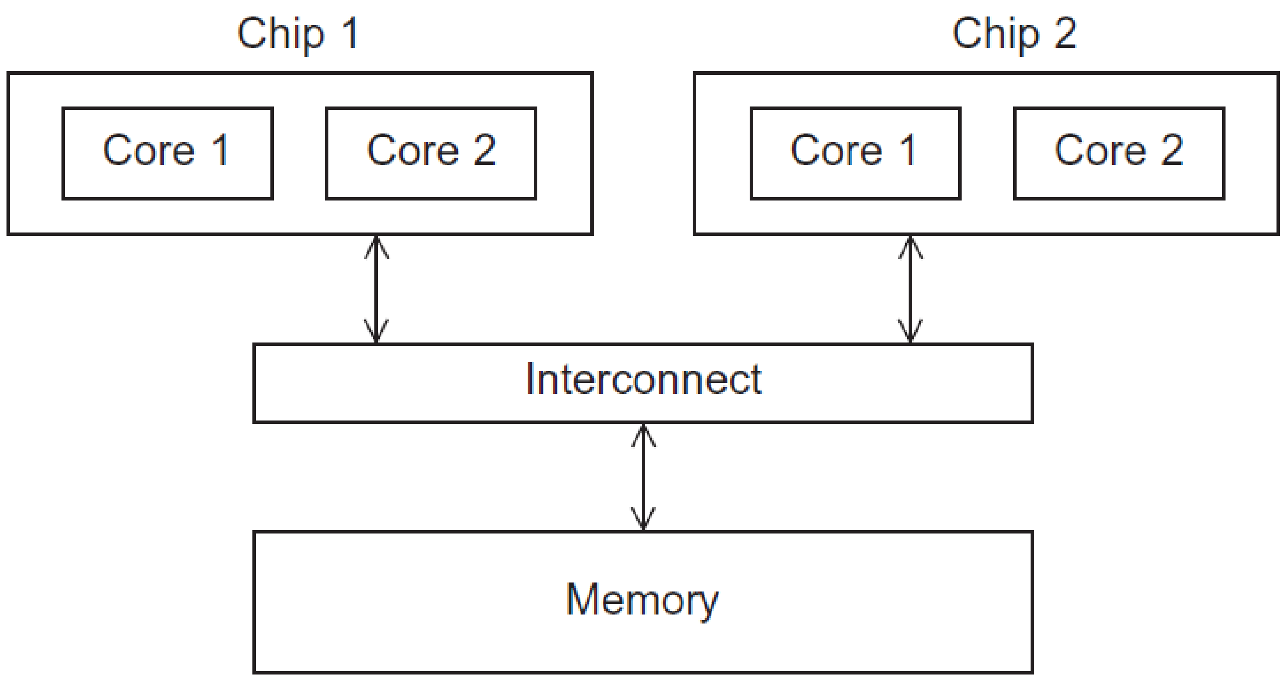
- Aka shared memory system
- Shared address space:
  - Processors can directly access all the data in the system
  - Inter-processor interaction ?
- Example: multi-core processors



Shared Address Space

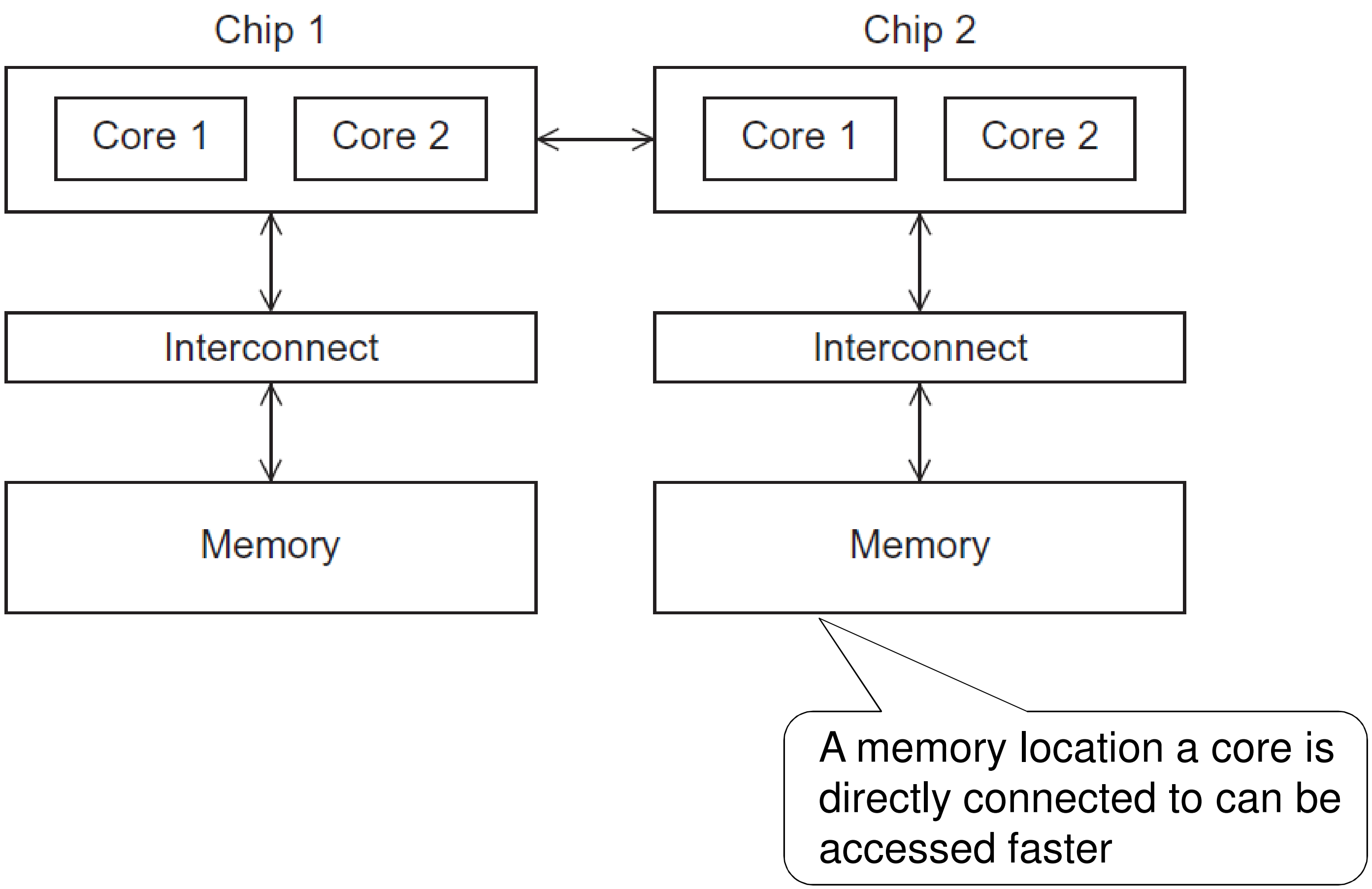


UMA Multicore System



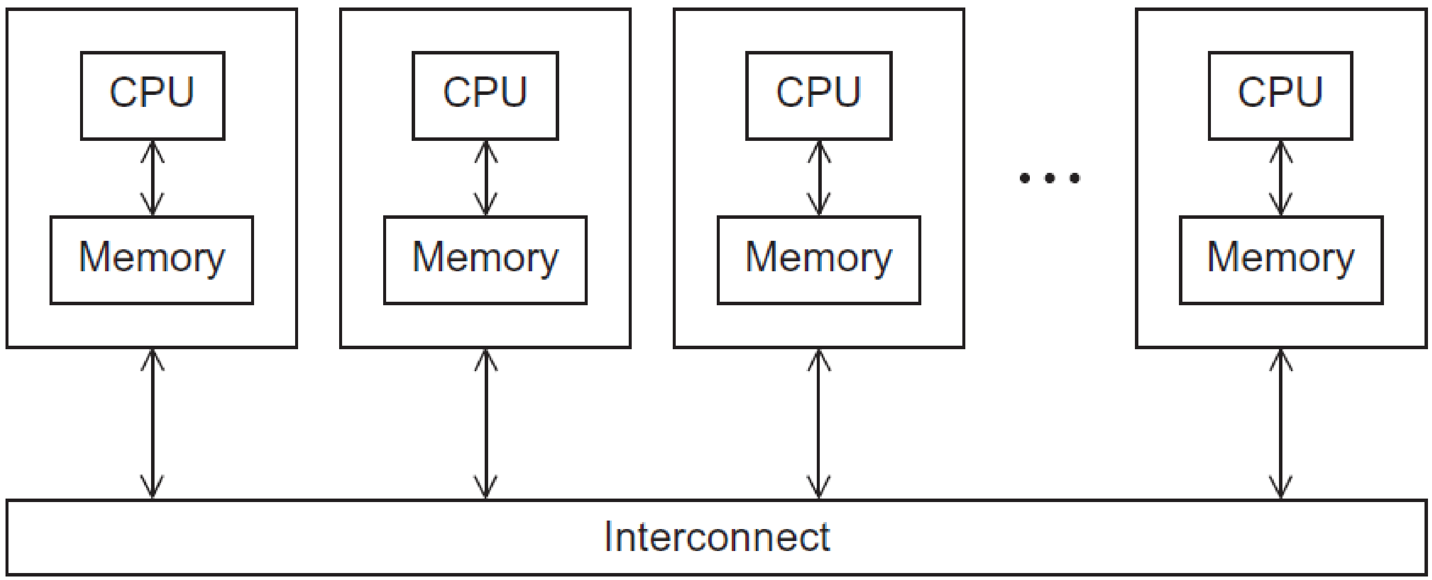
Time to access all the shared memory locations are the same for all the cores

NUMA Multicore System

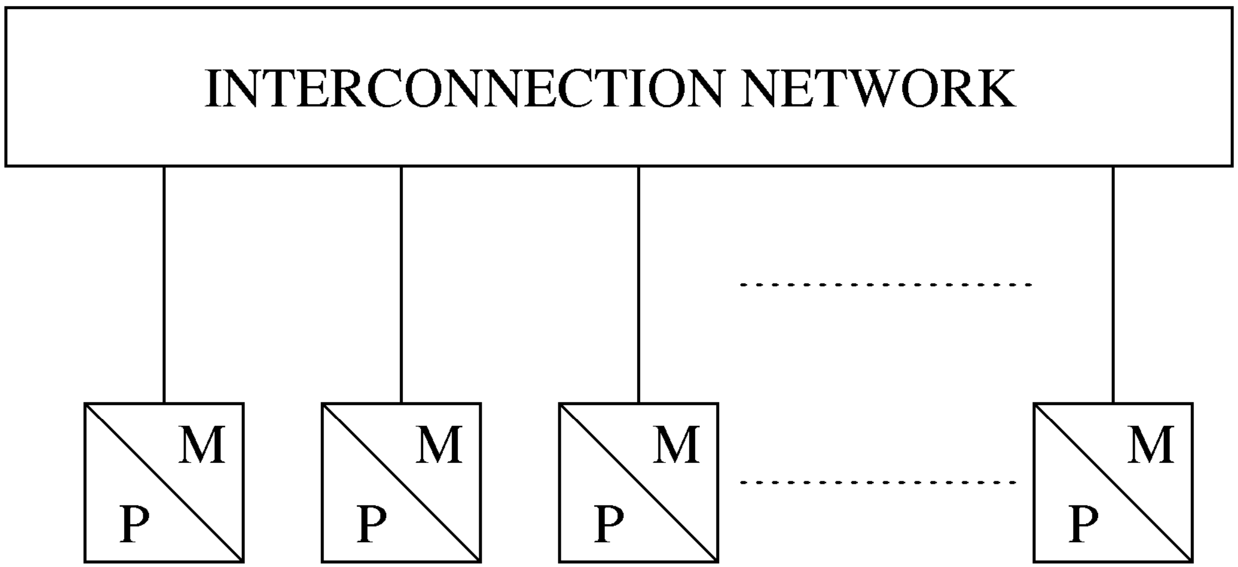


Distributed Address Space

- Aka distributed memory system
- Distributed address space:
  - "Shared nothing:" each processor has a private memory
  - Processors can directly access only local data
  - Inter-processor interaction ?



Distributed Address Space



P: Processor

M: Memory

Clusters

- A type of distributed address space machines
- A collection of commodity systems
- Connected by a commodity interconnection network
- Nodes of a cluster are individual computation units joined by a communication network

Shared vs. Distributed Address

- Shared address:
  - Pro. Vs Con.?
- Distributed address:
  - Pro. Vs. Con.?