Advanced Parallel Architecture

Annalisa Massini - 2014/2015

Multiprocessors

Parallel Architectures

- Parallel processors are computer systems consisting of multiple processing units connected via some interconnection network plus the software needed to make the processing units work together.
- There are two major factors used to categorize such systems: the **processing units** themselves, and the **interconnection network** that ties them together.

Parallel Architectures

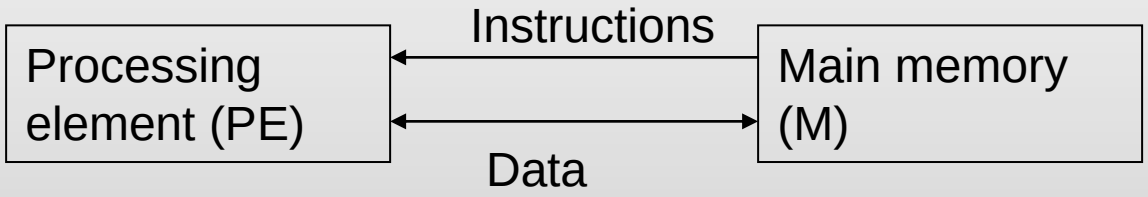
- A vast number parallel architecture types have been devised
- Various types of parallel architecture have **overlapping characteristics to different extents**
- It is not easy to develop a simple **classification** system for parallel architectures
- Parallel architecture can be distinguished under the following broad categories:
  - Flynn's classification
  - Classification based on memory arrangement
  - Classification based on interconnections among PEs and memory modules
  - Classification based on characteristic nature of PEs

Flynn's classification

- Based on the notion of a stream of information
- The **instruction stream** is defined as the sequence of instructions performed by the processing unit.
- The **data stream** is defined as the data traffic exchanged between the memory and the processing unit
- Either of the instruction or data streams can be single or multiple.
- Four distinct categories:
  - single-instruction single-data streams (SISD)
  - single-instruction multiple-data streams (SIMD)
  - multiple-instruction single-data streams (MISD
  - multiple-instruction multiple-data streams (MIMD)

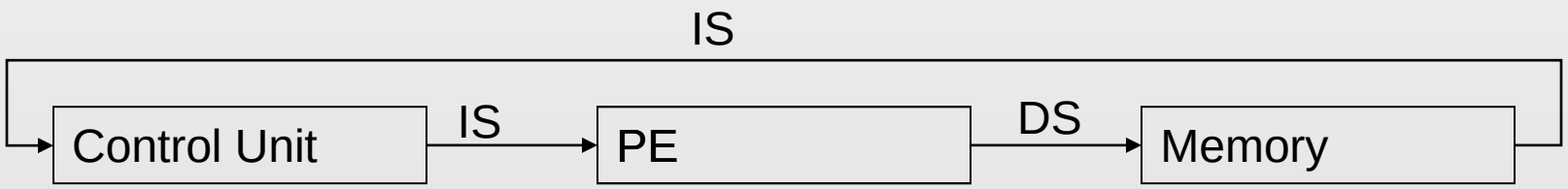
Flynn's classification

- During program execution the PE fetches instructions and data from the main memory, processes the data as per the instructions and sends the results to the main memory after processing has been completed.



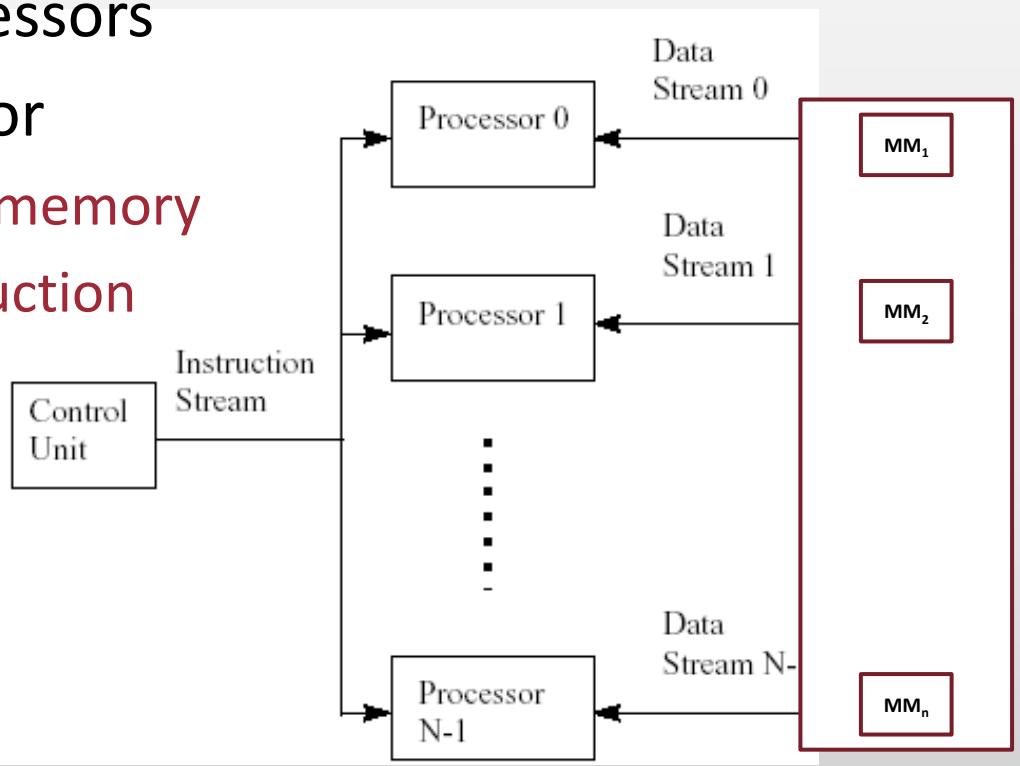
SISD

- The Von Neumann computer falls under the category
- of single-instruction single-data (SISD) stream.
- An alternative representation of the architecture indicating instruction and data flow

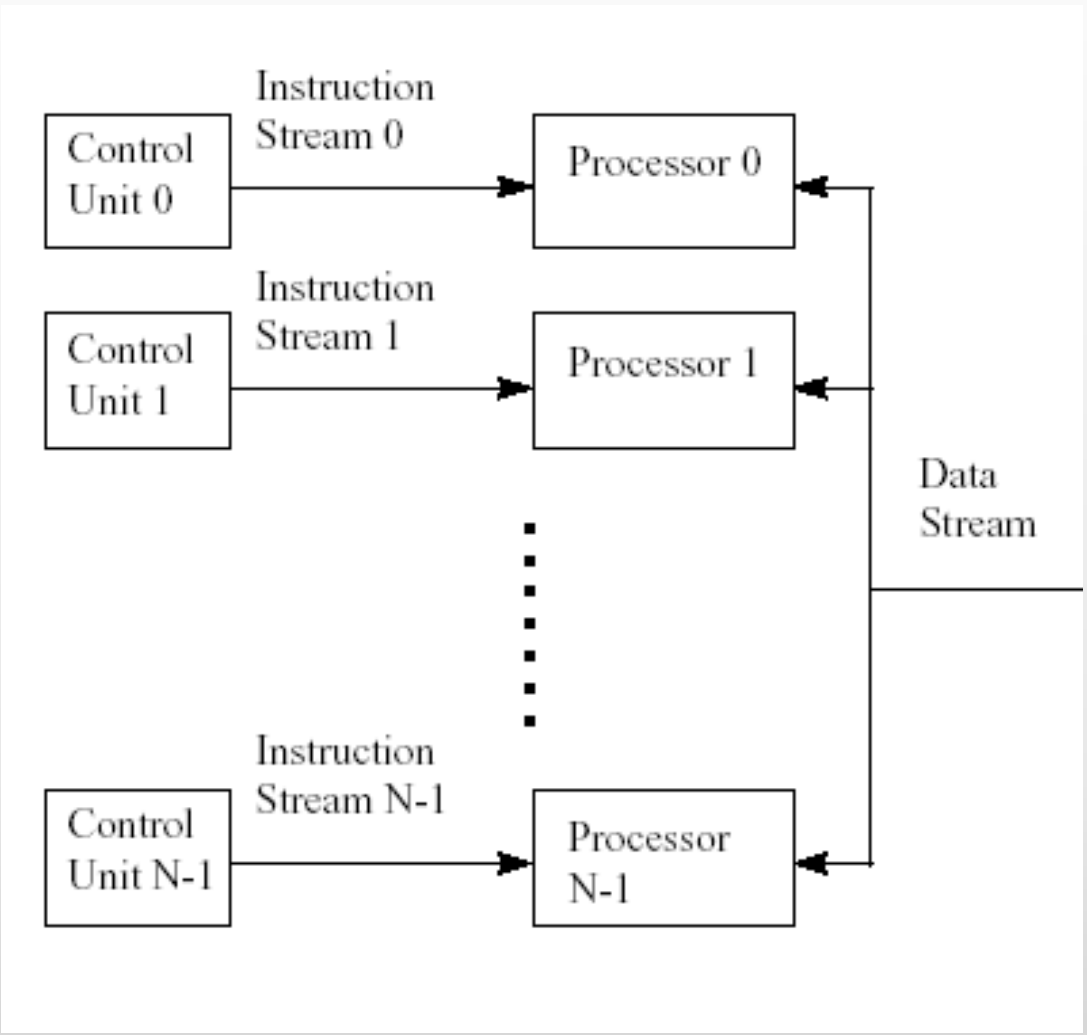


SIMD

- This architecture possesses a single instruction stream, to process the entire data structure (multiple-data stream)
- Asingle program control unit controls multiple execution units or execution processors
- Each execution processor
  - is equipped with a local memory
  - executes the same instruction
- Applications:
  - Image processing
  - Matrix manipulations
  - Sorting

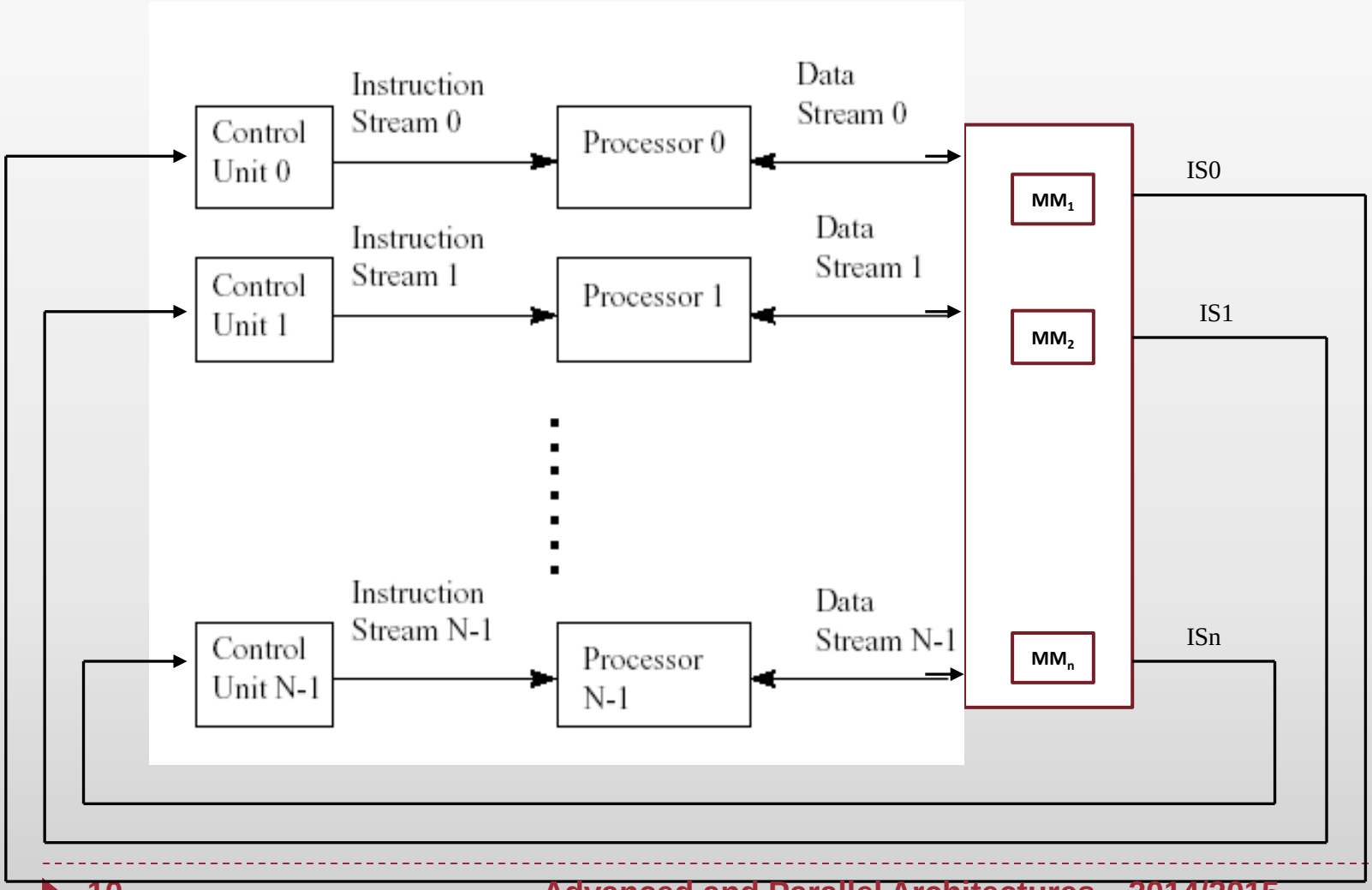


MISD



Applications:
- Classification
- Robot vision

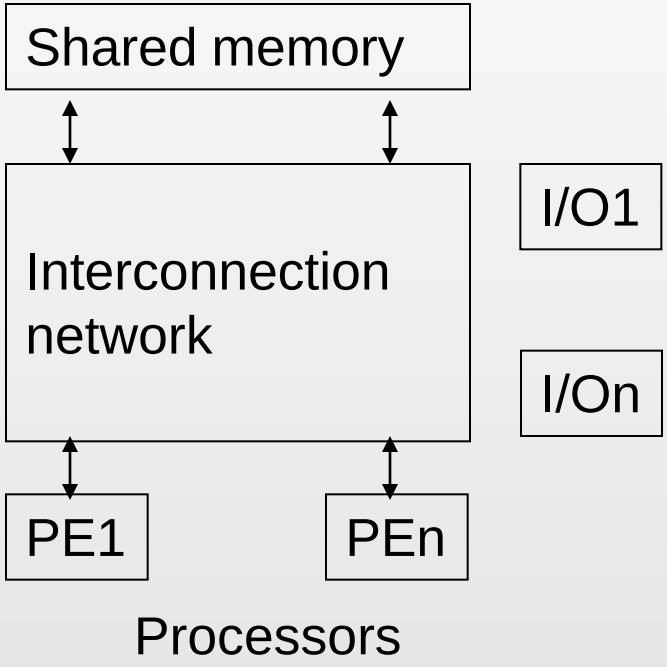
MIMD


Control Unit 0
Instruction Stream 0
Processor 0
Data Stream 0
Control Unit 1
Instruction Stream 1
Processor 1
Data Stream 1
Control Unit N-1
Instruction Stream N-1
Processor N-1
Data Stream N-1
MM1
MM2
MMn
IS0
IS1
ISn

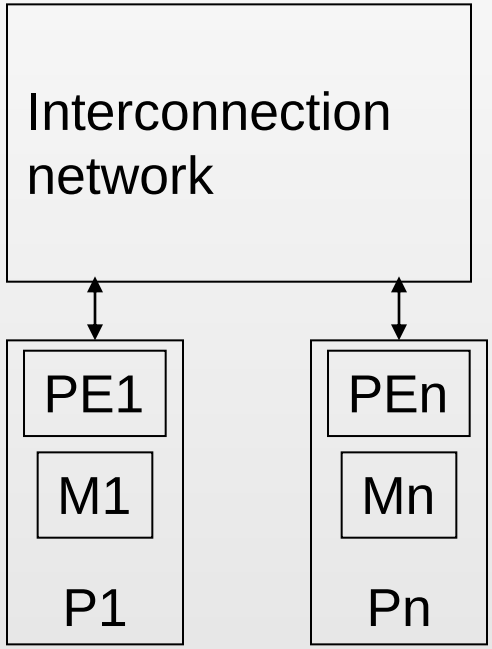
Flynn's classification

- Advantages of Flynn
  - Universally accepted
  - Compact Notation
  - Easy to classify a system (?)
- Disadvantages of Flynn
  - Very coarse-grain differentiation among machine systems
  - Comparison of different systems is limited
  - Interconnections, I/O, memory not considered in the scheme

Classification based on memory arrangement



Shared memory - multiprocessors



Message passing – multicomputers (Distributed memory)

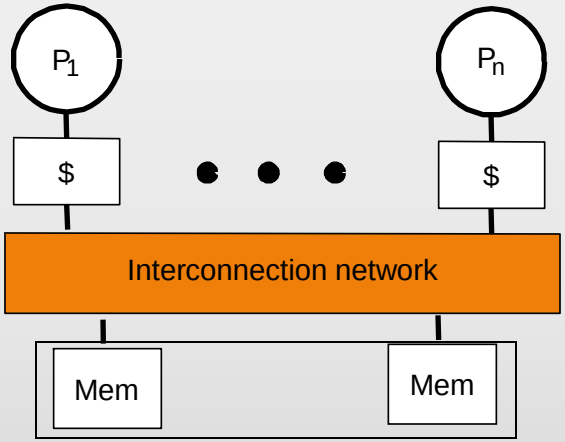
Shared-memory multiprocessors

- Uniform Memory Access (UMA)
- Non-Uniform Memory Access (NUMA)
- Cache-only Memory Architecture (COMA)

- Memory is common to all the processors.
- Processors easily communicate by means of shared variables.

The UMA Model

- Tightly-coupled systems (high degree of resource sharing)
- Suitable for general-purpose and time-sharing applications by multiple users.

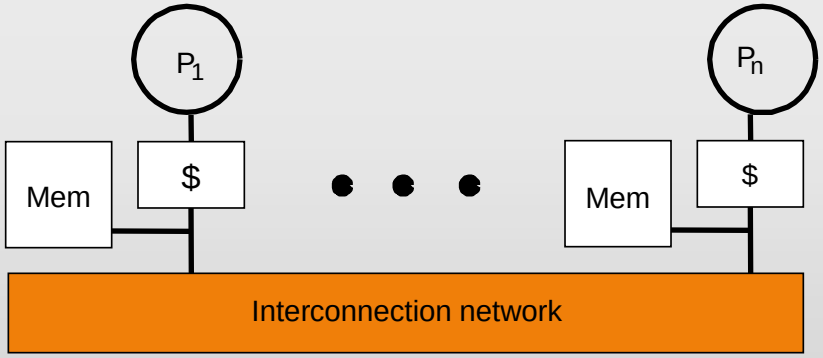


Symmetric and asymmetric multiprocessors

- Symmetric:
  - all processors have equal access to all peripheral devices.
  - all processors are identical.
- Asymmetric:
  - one processor (master) executes the operating system
  - other processors may be of different types and may be dedicated to special tasks.

The NUMA Model

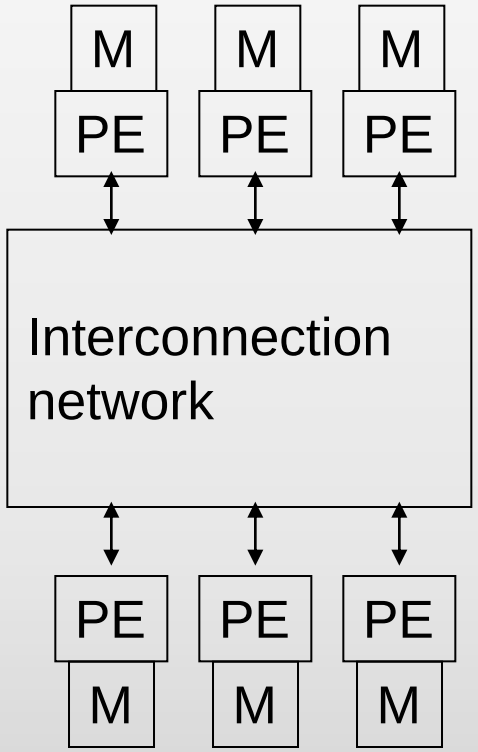
- The access time varies with the location of the memory word
- Shared memory is distributed to local memories
- All local memories form a global address space accessible by all processors



Distributed Memory (NUMA)

Distributed memory multicomputers

- Multiple computers- nodes
- Message-passing network
- Local memories are private with their own program and data
- No memory contention so that the number of processors is very large
- The processors are connected by communication lines, and the precise way in which the lines are connected is called the *topology* of the multicomputer.

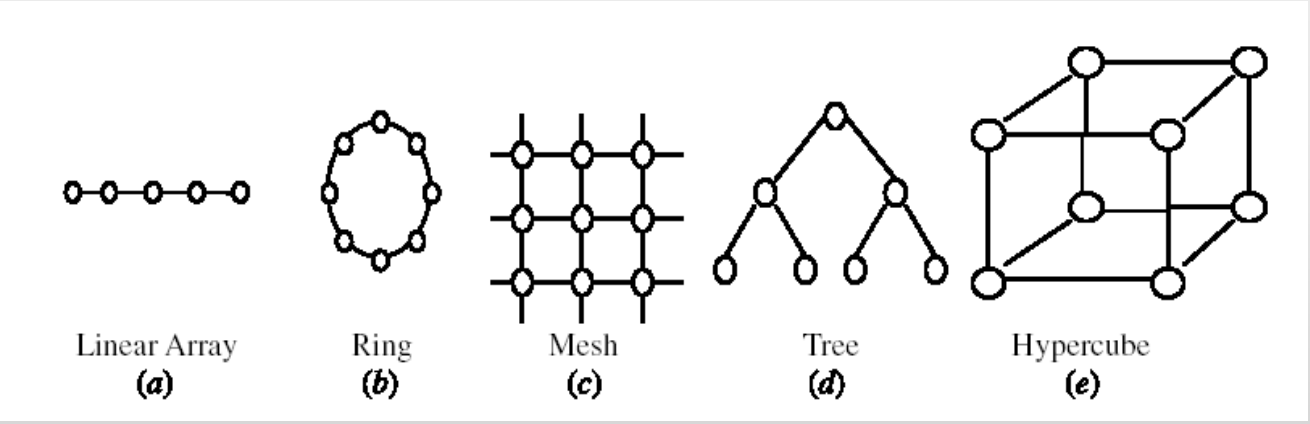


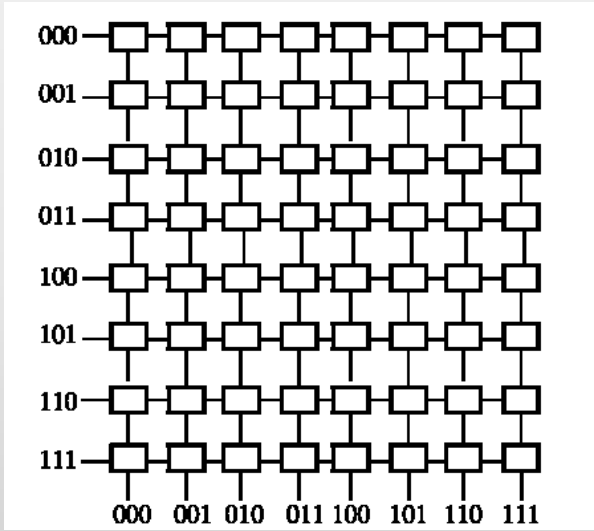
Classification based on type of interconnections

- This classification is quite specific to MIMD architectures as they, generally, comprises multiple PEs and memory modules
- The various interconnecting communication networks used for establishing communication schemes among the PEs of a parallel architecture include: **linear, shared single bus, shared multiple bus, crossbar, ring, mesh, star, tree, hypercube and complete graph**.

Classification based on type of interconnections

- Among these interconnecting networks, linear, mesh, ring, star, tree, hypercube and complete graph are **static** connection structures whereas shared single bus, shared multiple bus and crossbar are **dynamic** interconnection structures as they are reconfigurable under system control





Classification based on characteristic of PEs

- Parallel architectures are also classified in terms of the nature of the Pes comprising them
- An architecture may consist of either only one type of PE or various types of PEs
- The different types of processors that are commonly used to form parallel architectures are:
  - CISC Processors
  - RISC Processors
  - DSP and Vector Processors
  - Homogeneous and Heterogeneous Parallel Architectures

Interconnection Networks

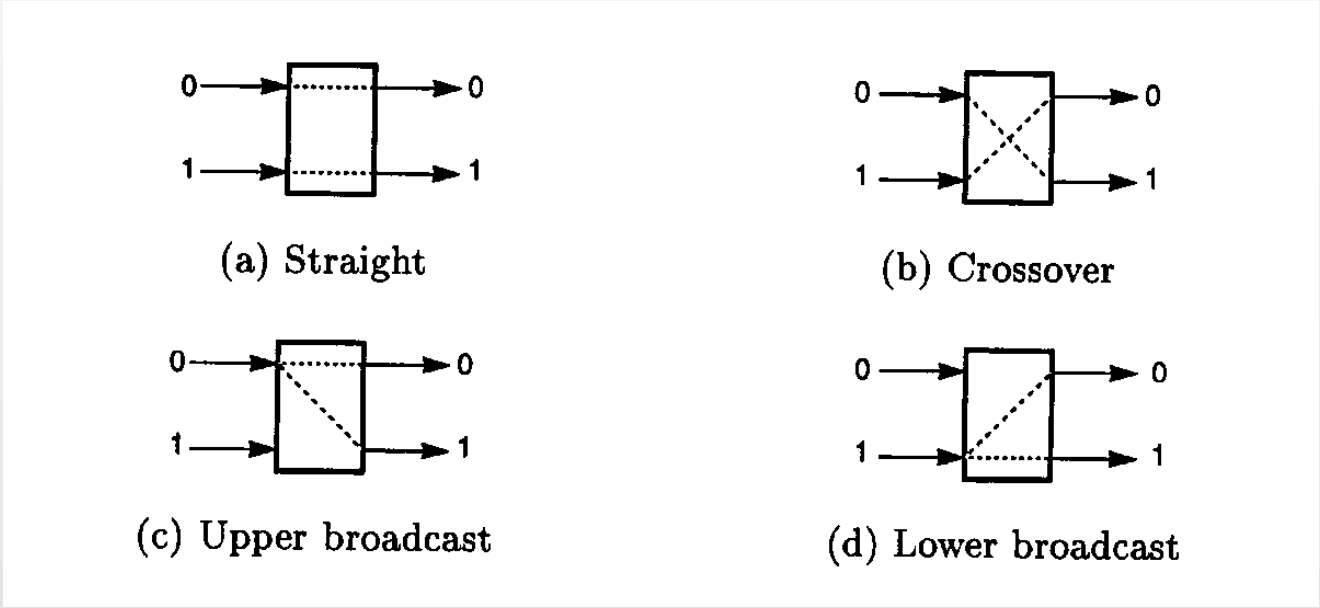
Criteria for classification

- Multiprocessors interconnection networks (INs) can be classified based on a number of criteria:
  - Mode of Operation (Synchronous vs. Asynchronous)
  - Control Strategy (Centralized vs. Decentralized)
  - Switching Techniques (Packet switching vs. Circuit switching)
  - Topology (Static Vs. Dynamic)

Dynamic Networks

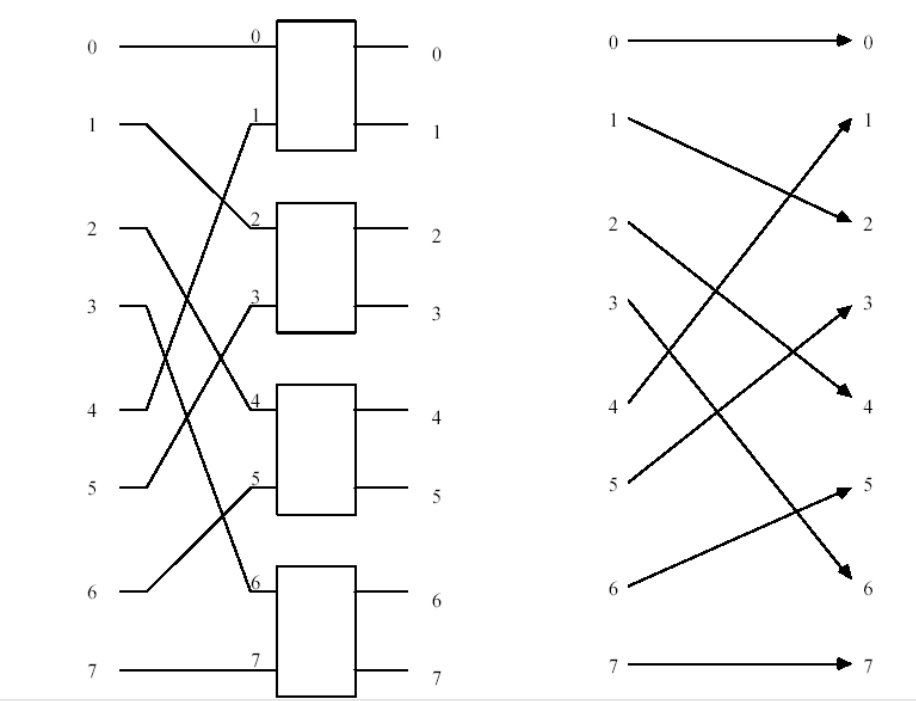
- Connections in a dynamic network are established on the fly as needed
- Dynamic networks can be classified based on interconnection scheme as **bus-based** or **switch-based**
- **Bus-based** networks can further be classified as single bus or multiple buses.
- **Switch-based** can be classified according to the structure of the interconnection network:
  - single-stage
  - multistage
  - crossbar networks

2 × 2 Switches


0
1
0
1
(a) Straight
0
1
0
1
(b) Crossover
0
1
0
1
(c) Upper broadcast
0
1
0
1
(d) Lower broadcast

Single-stage networks

- Single stage **Shuffle-Exchange** IN (left)
- Perfect **shuffle** mapping function (right)
- Perfect shuffle operation: cyclic shift 1 place left, eg 101 --> 011
- Exchange operation: invert least significant bit, e.g. 101 --> 100



Multistage Interconnection Networks

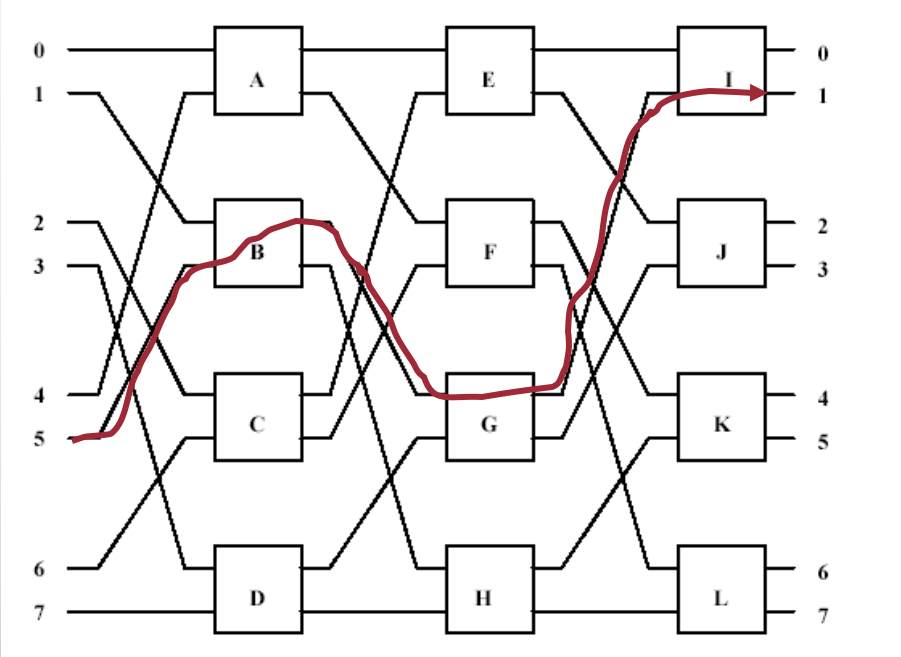
- The capability of single stage networks is limited
- If we cascade enough of them together, they form a Multistage Interconnection Network (MIN).
- Switches can perform their own routing or can be controlled by a central router

Multistage Interconnection Networks

- *Nonblocking*
  - A network is called strictly nonblocking if it can connect any idle input to any idle output regardless of what other connections are currently in process
- *Rearrangeable nonblocking*
  - In this case a network should be able to establish all possible connections between inputs and outputs by rearranging its existing connections.
- *Blocking interconnection*
  - A network is said to be blocking if it can perform many, but not all, possible connections between terminals.
  - Example: the Omega network

Omega networks

- A MIN using 2 × 2 switches and a perfect shuffle interconnect pattern between the stages
- There is one unique path from each input to each output
- No redundant paths → no fault tolerance, blocking



Example:

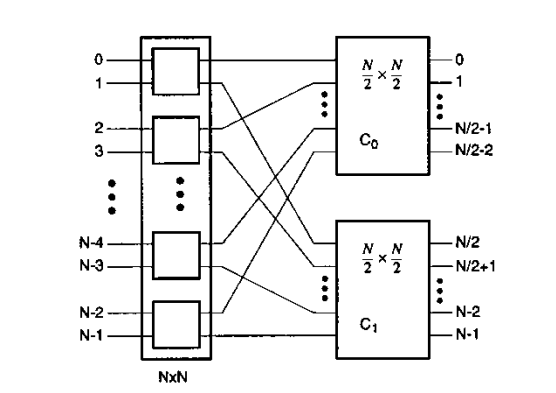
- Connect input 101 to output 001
- Use the bits of the destination address, 001, for dynamically selecting a path
- Routing:
  - 0 means use upper output
  - 1 means use lower output

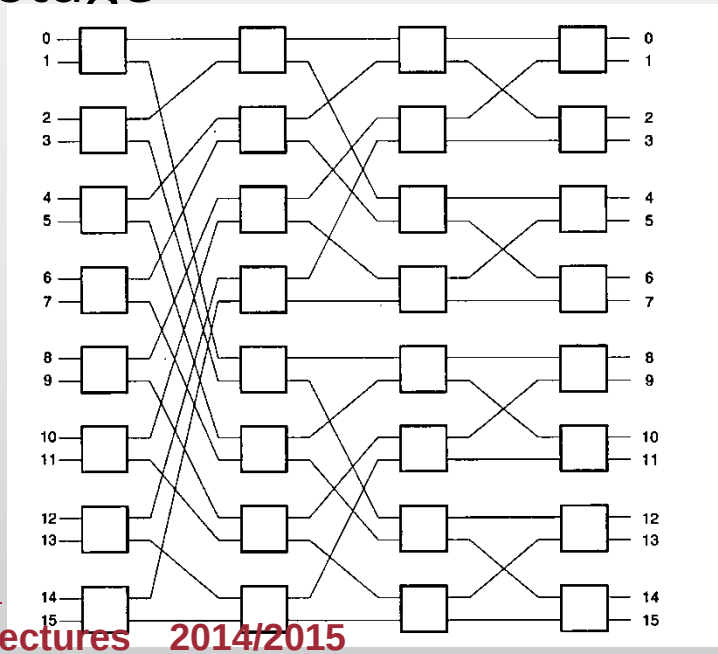
Omega networks

- $\log_2 N$ stages of 2×2 switches
- $N/2$ switches per stage
- $S=(N/2)\log_2(N)$ switches
- Number of permutations in a omega network 2^S

Baseline networks

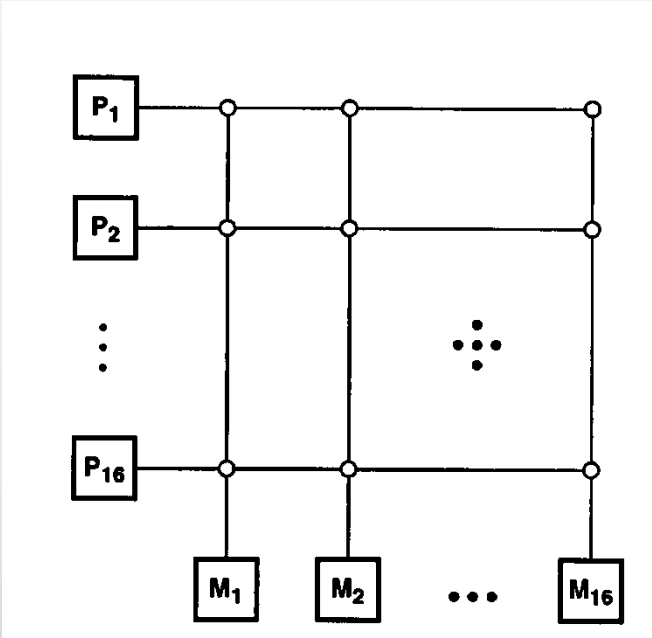
- The network can be generated recursively
- The first stage N × N, the second (N/2) × (N/2)
- Networks are topologically equivalent if one network can be easily reproduced from the other networks by simply rearranging nodes at each stage





Crossbar Network

- Each junction is a switching component – connecting the row to the column.
- Can only have one connection in each column



Crossbar Network

- The major advantage of the crossbar switch is its **speed**
- In one clock, a connection can be made between source and destination
- Blocking if the destination is in use
- Because of its complexity, the **cost** of the crossbar switch can become the dominant factor for a large multiprocessor system
- Crossbars can be used to implement the $a \times b$ switches used in MIN's
- In this case each crossbar is small so costs are kept down

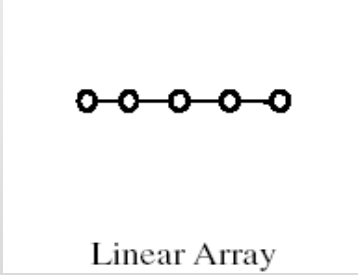
Static Networks

- Connections in a static network are fixed links
- Static networks can be further classified according to their interconnection pattern as:
  - one-dimension (1D)
  - two-dimension (2D)
  - hypercube (HC)

Static Networks

Linear Network

- A number of nodes are connected through buses in a linear format
- Every node, except the nodes at the two ends, in this configuration is directly connected to two other nodes
- To connect *n nodes in this configuration n− 1 buses are required and the maximum* internodes distance is *n− 1*

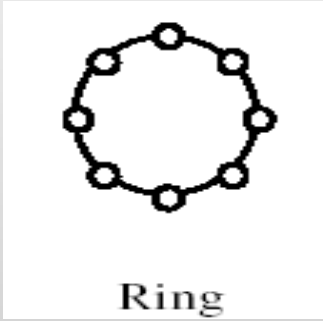


Linear Array

Static Networks

Ring Interconnection Network

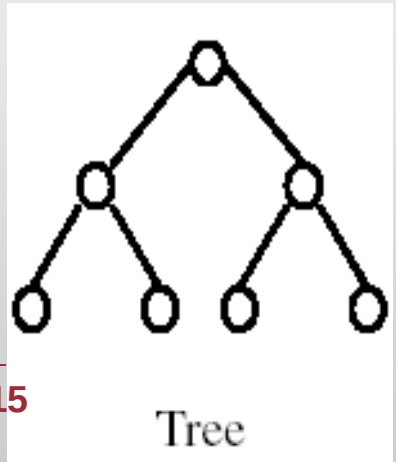
- This interconnection is very easy to implement
- In the case of ring interconnection *n buses are required to connect n nodes and the maximum* internodes distance is *n / 2*
- Commercial machines such as Hewlett-Packard's Exemplar V2600 and Kendal Square Research's KSR-2 have been designed using ring networks.



Static Networks

Tree Interconnection Network

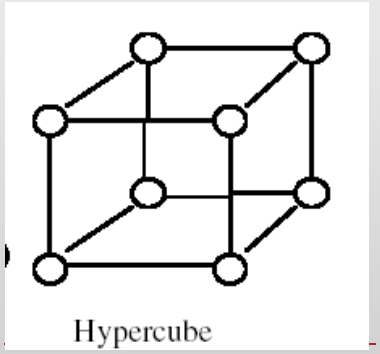
- In the Tree structure (*n -level tree)* any intermediate node acts as a medium to establish communication between its parents and children
- Communication could also be established between any two nodes in the structure
- A tree structure can be highly effective if a small portion of traffic goes through the root node otherwise due to bottleneck problems performance deteriorates



Static Networks

Hypercube Interconnection Network

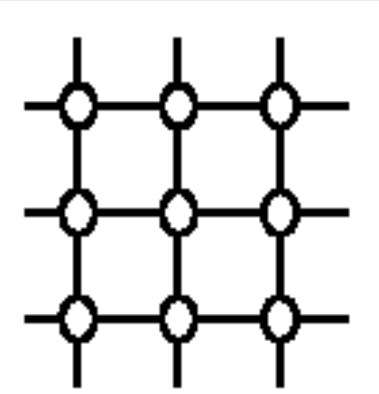
- Hypercube is a popular interconnection network architecture, especially for NUMA multiprocessors
- An *n -dimensional hypercube can connect* 2^n *nodes*
- The edges of the cube represent bi-directional communication links between two neighbouring nodes
- The nodes are labelled using binary addresses (addresses of the two neighbouring nodes differ by one bit position)
- Many commercial multiprocessors have used hypercube interconnections


Hypercube

Static Networks

Mesh and Torus Interconnection Network

- Mesh is a popular interconnection network structure used to connect large numbers of nodes.
- It came into being in the 1990s as an alternative to hypercube in large multiprocessors.
- To formulate a mesh structure comprising *n nodes 2(n − n0.5) buses are required and the* maximum internodes distance is $2(\sqrt{n} - 1)$.
- If a wraparound connection is made between the nodes at opposite edges then we have a Torus


Mesh

Network properties

- *Node degree d* - the number of edges incident on a node.
  - In degree
  - Out degree
- *Diameter* D of a network is the maximum shortest path between any two nodes.
- The network is *symmetric* if it looks the same from any node.
- The network is *scalable* if it expandable with scalable performance when the machine resources are increased.

References

- *Advanced Computer Architecture and Parallel Processing*

  H. El-Rewini, M. Abd-El-Barr, John Wiley and Sons, 2005

- *Parallel computing for real-rime signal processing and control* – **Ch. 2 *Parallel Architectures***

  M. O. Tokhi, M. A. Hossain, M. H. Shaheed, Springer, 2003