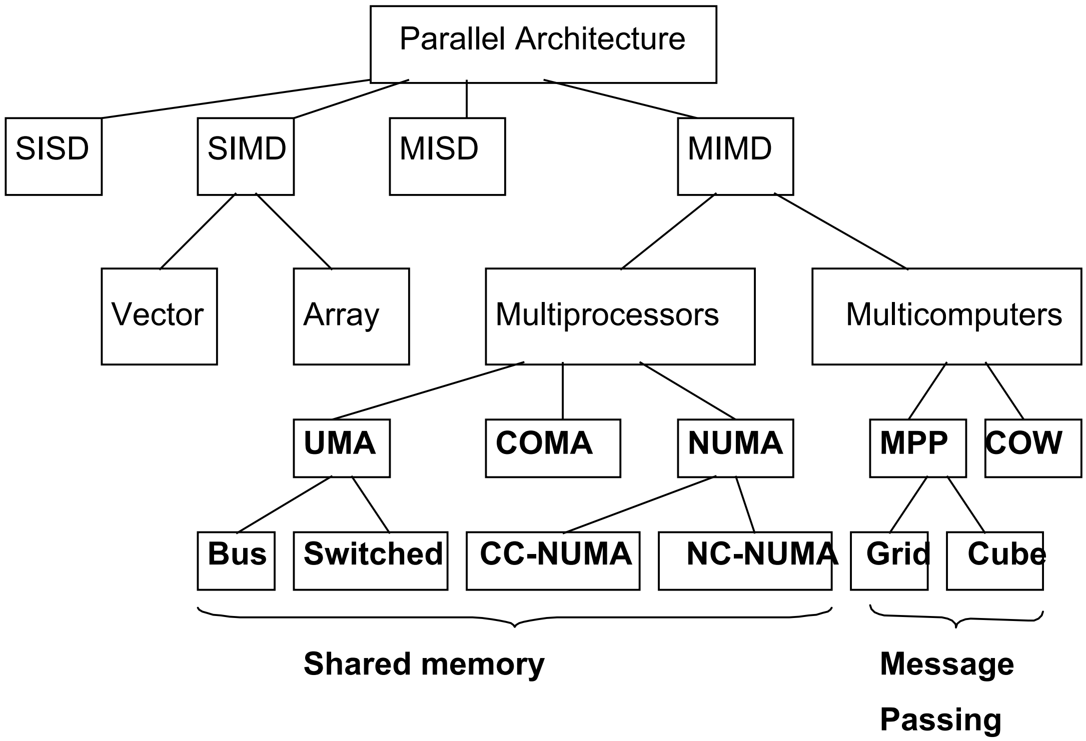
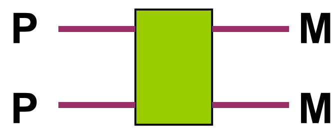
A Taxonomy of Parallel Computers

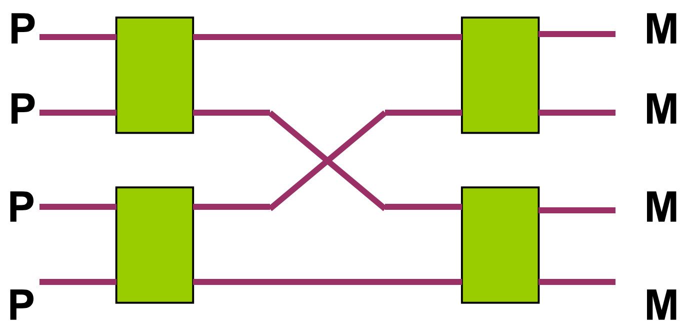


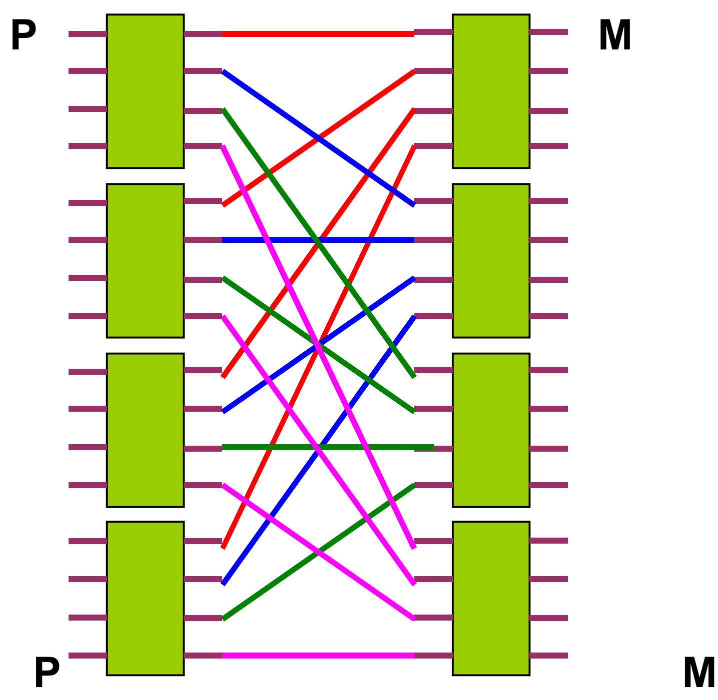
UMA	**Uniform Memory Access**
NUMA	**Non Uniform Memory Access**
COMA	**Cache Only Memory Access**
MPP	**Massively Parallel Processor**
COW	**Cluster Of Workstations**
CC-NUMA	**Cache Coherent NUMA**
NC-NUMA	**No Cache NUMA**

Processor-Memory Interconnection

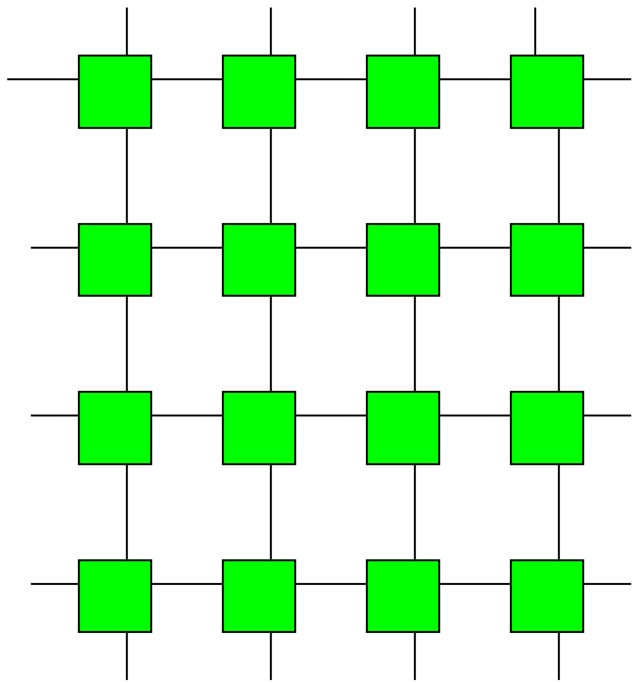
The Butterfly Switch



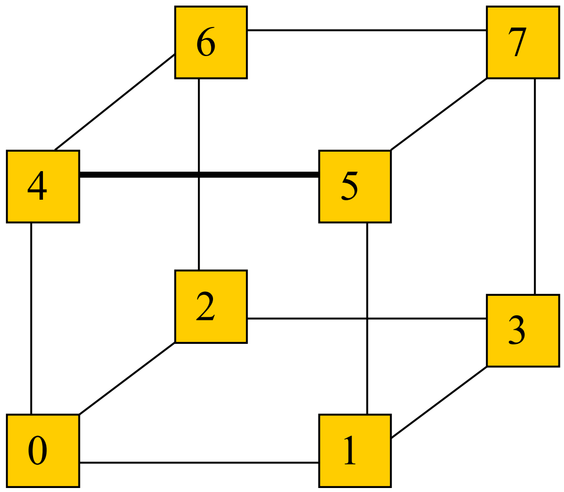


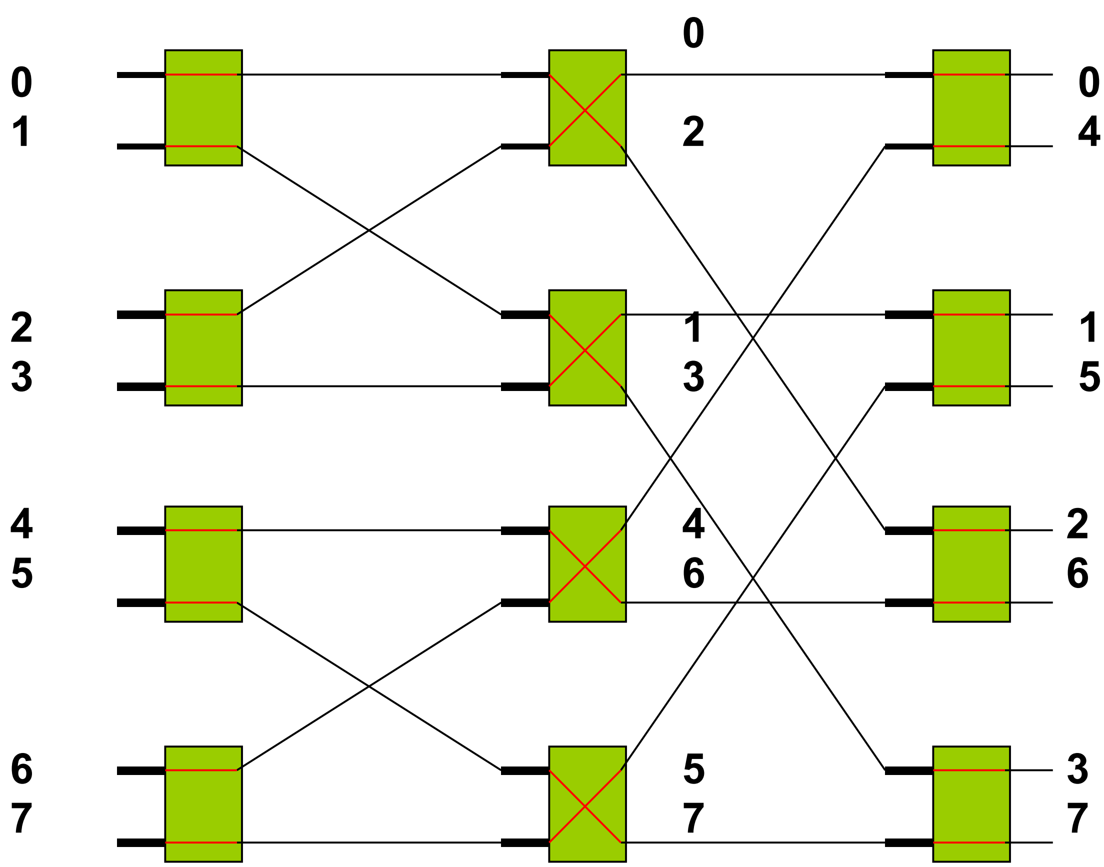


2D grid interconnection network



Cube Interconnection Network





<u>Contention Problems</u>

For some reason, two processors cannot always access the memory at the same time. Why?

1. ***Switch contention***

   Because the network is blocking.

   Use switches of higher base and add additional switch columns to minimize this.

2. ***Memory module contention***

   System library helps distribute the elements of vectors and matrices among different memory modules.

3. ***Memory location contention***

   Loop index variables

   Lock variables like semaphores

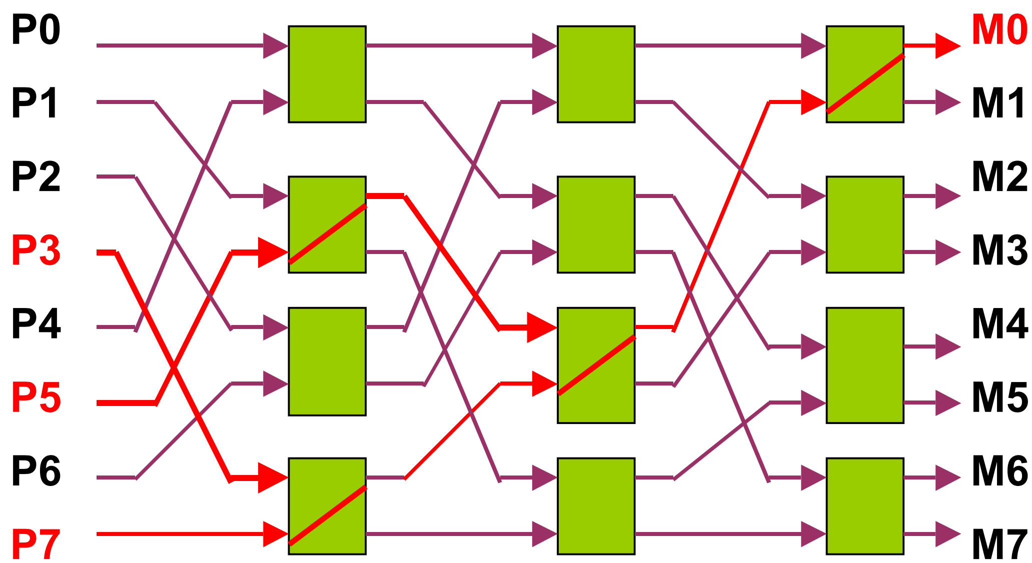
   (Recall the mutual exclusion or the barrier synchronization problems)

   How to avoid or resolve these contention problems?

Other potential contention-related problems

Hot-spots and Tree saturation

A hot spot is a memory bank that attracts significant amount of traffic.



M0 is the hot spot. But how difficult is it for P1 to access M1, if *store-and forward switching* is used?

(In store-and-forward switching, the pending requests for memory access are buffered in the intermediate switches. This can delay access to an otherwise cold spot by a not-so-active processor.**)**

Memory Location Contention

In addition to distributing the elements of vectors or matrices among different memory modules, sometimes solutions can be restructured to minimize memory location contention.

Example. $S = \sum X(j)$, initially P_j contains $X(j)$.

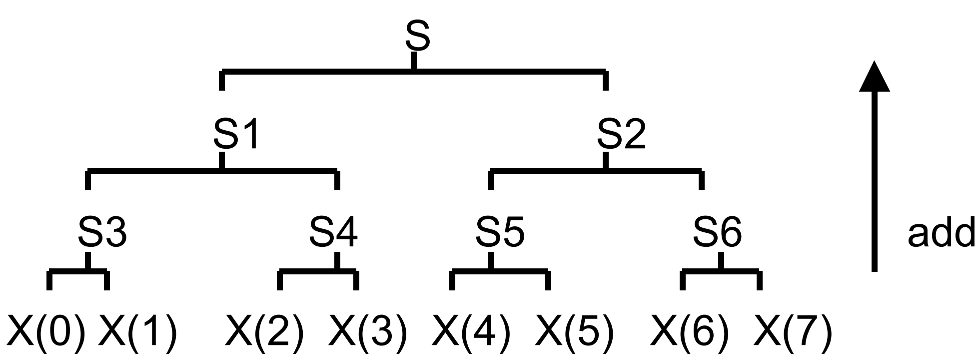
There are 8 processors 0..7

Solution 1.

```
S=0;
for (j=0; j < 8; j=j+1)
    S = S+j
```

Contention is a problem since S is a hot spot.

Solution 2. Divide and conquer.



Distribute S, S1, S2, S3, … among different modules.

Clusters

Independent machines connected through LAN.

A cluster with K machines has as much administrative overhead as K independent machines.

Due to modular construction, clusters are easier to maintain.

What is the impact of NUMA characteristics on the overall speed-up of the computation?

What is the price of implementing a CC-NUMA?
Consider the problem of adding 100,000 integers on a 64-node cluster.

DSM (Distributed Shared Memory) adds to the cost but simplifies programming.