LECTURE 8

Shared Memory : Advantages

- There is no need to partition either the code or the data; therefore uniprocessor programming techniques can easily be adapted in the multiprogramming environment. Neither new programming languages nor sophisticated compilers are needed to exploit shared memory systems.
- There is no need to physically move data when two or more processes communicate. The consumer process can access the data from the place where the producer stored it. As a result, communication between processes is very efficient.

Shared Memory: Disadvantages

- Synchronized access to shared data structures requires special synchronizing constructs such as semaphores, conditional critical regions, monitors and so on. Usually message passing synchronization is simpler to understand and apply.
- The main disadvantage of shared memory systems is lack of scalability due to the contention problem. When several processors want to access the same memory module they must compete for the right to do so. The winner can access the memory, while the losers must wait. The larger the number of processors, the higher the probability of memory contention. Beyond a certain number of processors, this probability is so high that adding a new processor to the system will not increase performance.

LOW SCALIBILITY: Solutions

- Use of a high throughput low latency interconnection network.
- Use of cache memories.
- The logically shared memory can be physically implemented as a collection of local memories. Their new architecture type is called a virtual shared memory of distributed shared memory architecture.

In distributed shared memory system, the local memories are components of a global address and any processor can access the local memory of any other processor.

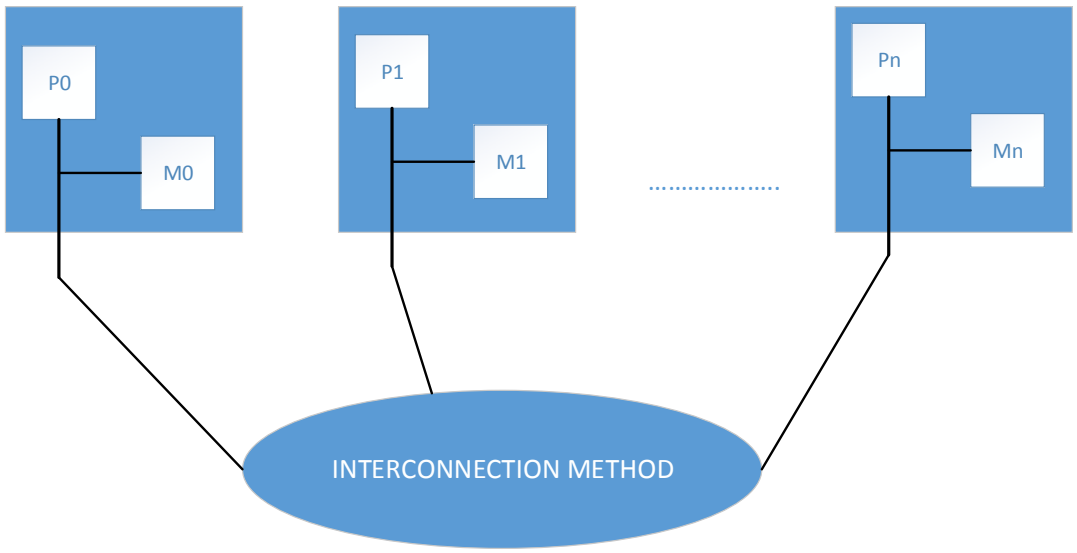
In distributed memory systems, the local memories have separate address spaces and direct address to the local memory of any other processor.

In distributed memory systems, the local memories have separate address spaces and direct access to the local memory of a remote processor is prohibited.

Distributed Shared Memory Systems: Classification

i. Non uniform memory access (NUMA) machines
ii. Cache-coherent non-uniform memory access (CC-NUMA) machines
iii. Cache only memory access (COMA) machines

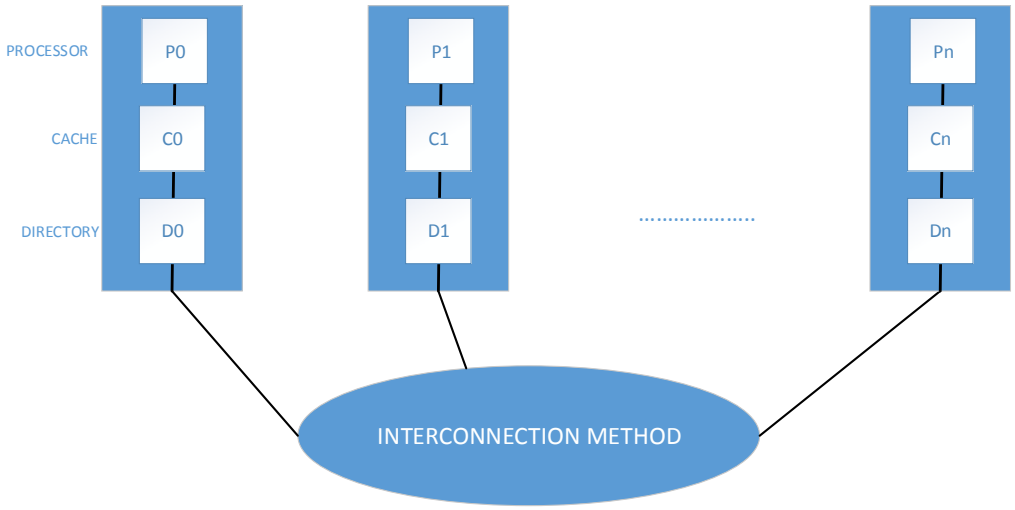
NUMA (non-uniform memory access) Machines



(Figure 1)

In NUMA machines, the shared memory is divided into as many blocks as there are processors in the system and each memory block is attached to a processor as a local memory with direct bus connection. As a result, whenever a processor addresses the part of the shared memory that is connected as local memory, access to that block is much faster than access to the remote ones.

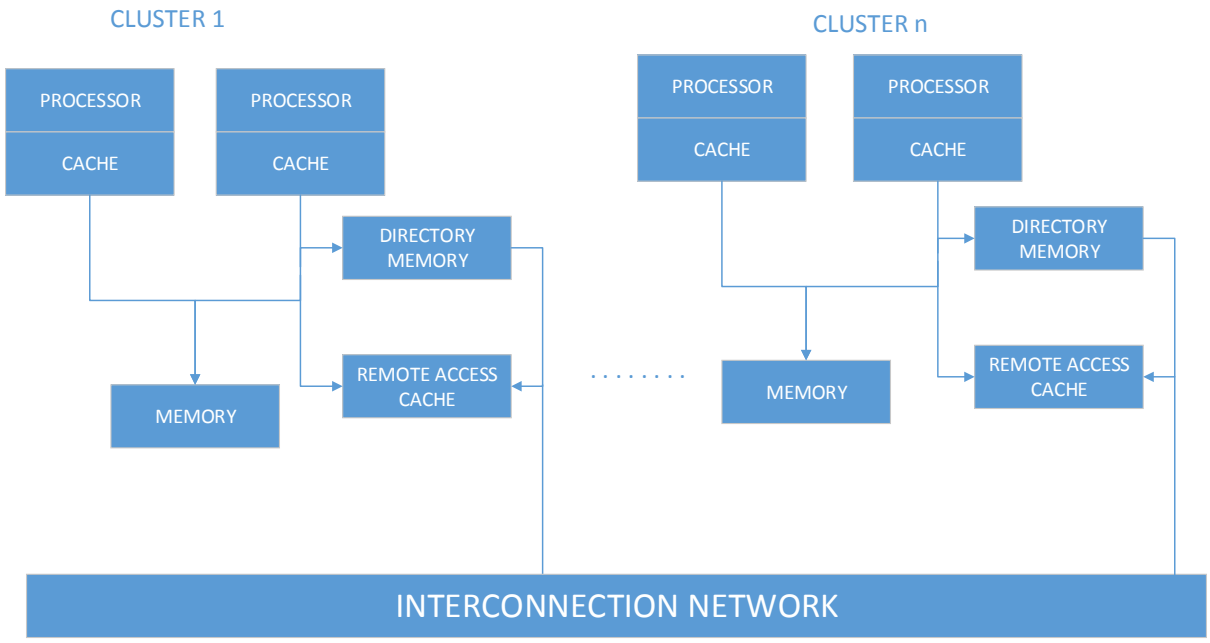
COMA (cache only memory access) Machines



(Figure 2)

The COMA model is a special case of a NUMA machine, in which the distributed main memories are converted to caches. There is no memory hierarchy at each processor node. All the caches form a global address space. Remote cache access is assisted by the distributed cache directories.

CC-NUMA (Cache coherent non-uniform memory access) Machines



(Figure 3 – Stanford DASH (A CC NUMA architecture))

CC-NUMA machines represent a compromise between the NUMA and COMA machines. Like the NUMA machines, the shared memory is constructed as a set of local memory blocks. However, in order to reduce the traffic in the interconnection network, each processor node is supplied with a large cache memory block.

The Dash architecture is a large scale CC-NUMA multiprocessor system. It consists of multiple microprocessor clusters connected through a scalable, low-latency interconnection network. Physical memory is distributed among the processing nodes in various clusters. The distributed memory forms a global address space.

For each memory block, the directory keeps track of remote nodes caching it. Directory memory and remote access cache facilitates pre-fetching and the directory based coherence protocol.

Shared Memory MIMD Architectures

Design Issues:

The three main design issues in increasing the scalability of shared memory systems are:

- Organizing shared memory
- Design of interconnection networks
- Design of cache coherence protocols

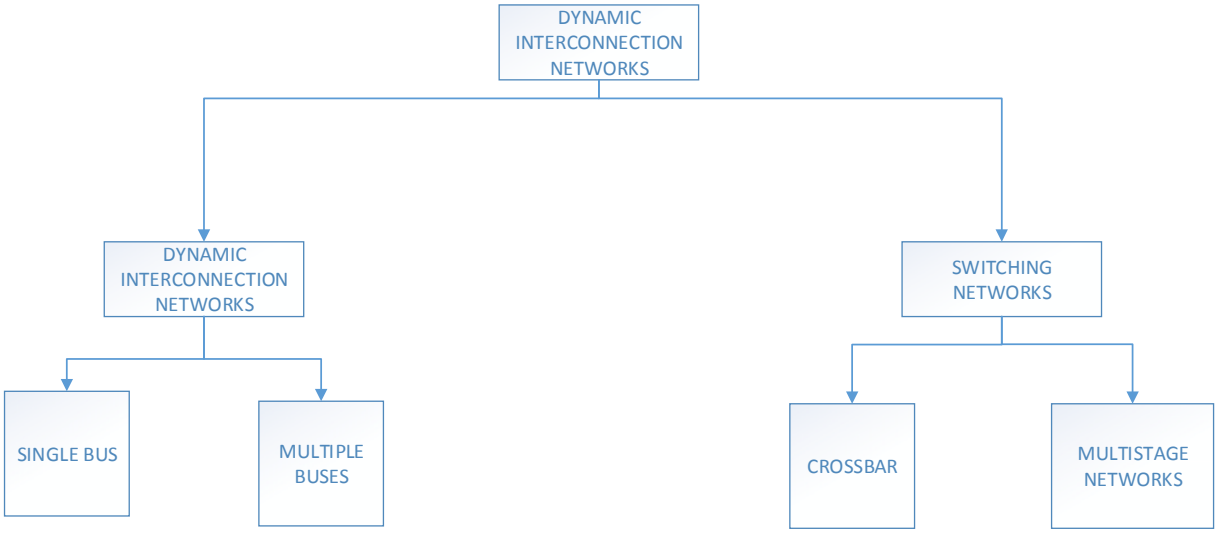
Organization of Memory

Shared memory systems are basically classified according to their memory organization since this is the most fundamental design issue. Accordingly, shared memory systems can be divided into four main classes:

- Uniform memory access (UMA) Machines
- NUMA
- CC-NUMA
- COMA

Shared Memory MIMD: Interconnection Networks

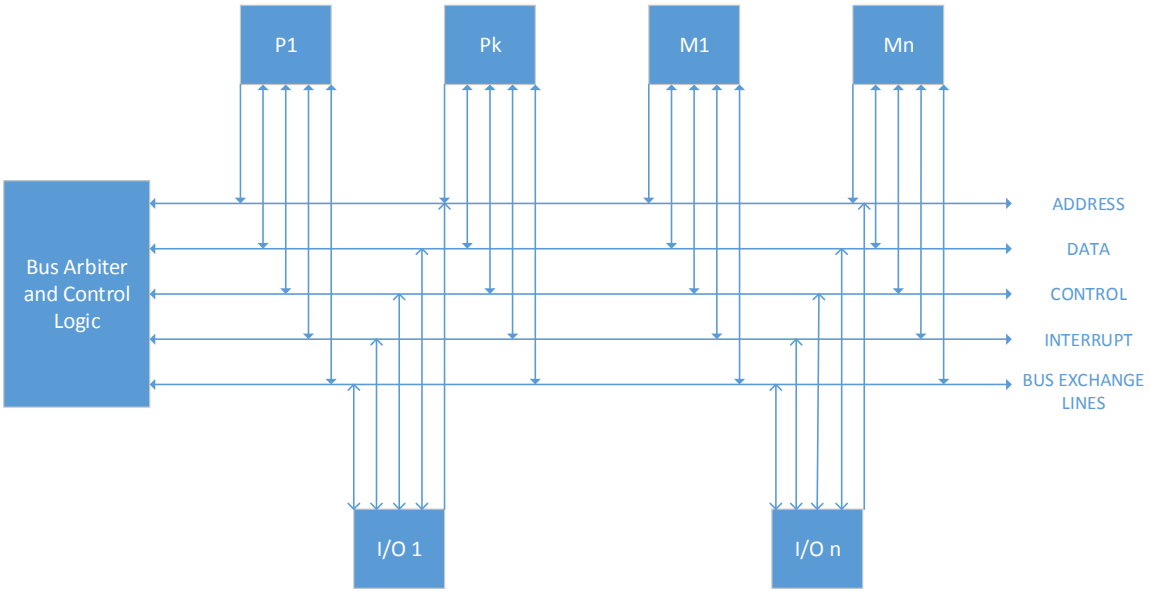
The quality of the interconnection network has a decisive impact on the speed, size and cost of the whole machine. Dynamic interconnection schemes are usually employed in multiprocessors.



(Figure 4)

Shared path networks provide continuous connection among the processors and memory blocks. The continuous connection is shared among processors which have to compete for its use.
Switching Networks do not provide a continuous connection among the processors and memory blocks. A switching mechanism enables processors to be temporarily connected to memory blocks.

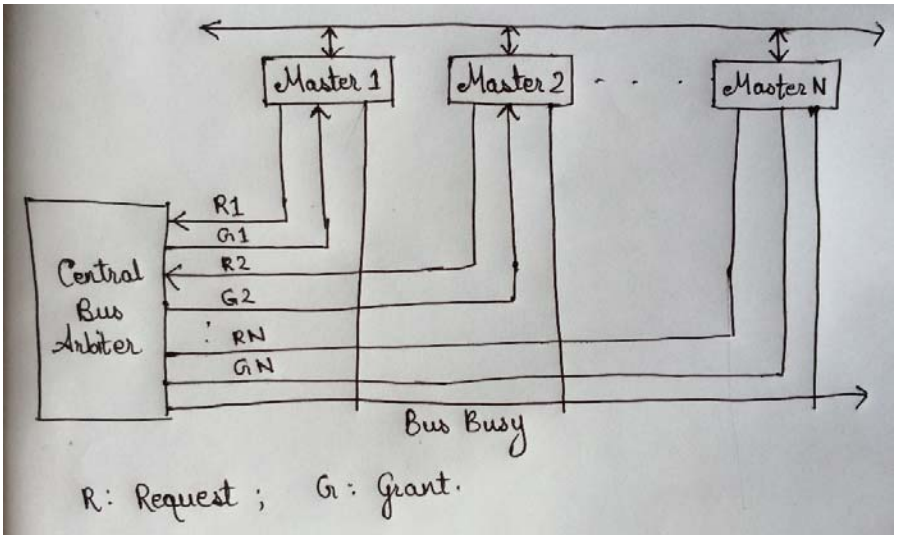
SINGLE: Shared Bus



(Figure 5: Structure of a single-bus multiprocessor without caches)

Bus Shelter Logic: Allocates the bus in the case of several simultaneous bus requests.

Bus Exchange Lines: Comprise typically one or more bus request lines by means of which the processors or other temporary bus masters can request bus allocation. According to the state of the bus request lines and the applied bus allocation policy, the arbiter grants one of the requests via the grant lines



(Figure 6)