LARGE COMPUTER SYSTEMS

Forms of Parallel Processing

Parallel processing is a term used to denote a large class of techniques that are used to provide simultaneous data-processing tasks for the purpose of increasing the computational speed of a computer system. Instead of processing each instruction sequentially as in a conventional computer, a parallel processing system is able to perform concurrent data processing to achieve faster execution time. For example, while an instruction is being executed in the ALU, the next instruction can be read from memory. The system may have two or more ALUs and be able to execute two or more instructions at the same time. Furthermore, the system may have two or more processors operating concurrently. The purpose of parallel processing is to speed up the computer processing capability and increase its throughput, that is, the amount of processing that can be accomplished during a given interval of time. The amount of hardware increases with parallel processing, and with it, the cost of the system increases. However, technological developments have reduced hardware costs to the point where parallel processing techniques are economically feasible.

There are a variety of ways that parallel processing can be classified. It can be considered from the internal organization of the processors, from the interconnection structure between processors, or from the flow of information through the system. One classification introduced by M. J. Flynn considers the organization of a computer system by the number of instructions and data items that are manipulated simultaneously. The normal operation of a computer is to fetch instructions from memory and execute them in the processor. The sequence of instructions read from memory constitutes an *instruction stream*. The operations performed on the data in the processor constitutes a *data stream*. Parallel processing may occur in the instruction stream, in the data stream, or in both. Flynn's classification divides computers into four major groups as follows:

Single instruction stream, single data stream (SISD)
Single instruction stream, multiple data stream (SIMD)
Multiple instruction stream, single data stream (MISD)
Multiple instruction stream, multiple data stream (MIMD)

SISD represents the organization of a single computer containing a control unit, a processor unit, and a memory unit. Instructions are executed sequentially and the system may or may not have internal parallel processing capabilities. Parallel processing in this case may be achieved by means of multiple functional units or by pipeline processing.

SIMD represents an organization that includes many processing units under the supervision of a common control unit. All processors receive the same instruction from the control unit but operate on different items of data. The shared memory unit must contain multiple modules so that it can communicate with all the processors simultaneously. MISD structure is only of theoretical interest since no practical system has been constructed using this organization. MIMD organization refers to a computer system capable of processing several programs at the same time. Most multiprocessor and multicomputer systems can be classified in this category.

Interconnection Networks

The components that form a multiprocessor system are CPUs, IOPs connected to input–output devices, and a memory unit that may be partitioned into a number of separate modules. The interconnection between the components can have different physical configurations, depending on the number of transfer paths that are available between the processors and memory in a shared memory system or among the processing elements in a loosely coupled system. There are several physical forms available for establishing an interconnection network. Some of these schemes are presented in this section:

1. Time-shared common bus
2. Multiport memory
3. Crossbar switch
4. Multistage switching network
5. Hypercube system

Time-Shared Common Bus

A common-bus multiprocessor system consists of a number of processors connected through a common path to a memory unit. A time-shared common bus for five processors is shown in Fig. 13-1. Only one processor can communicate with the memory or another processor at any given time. Transfer

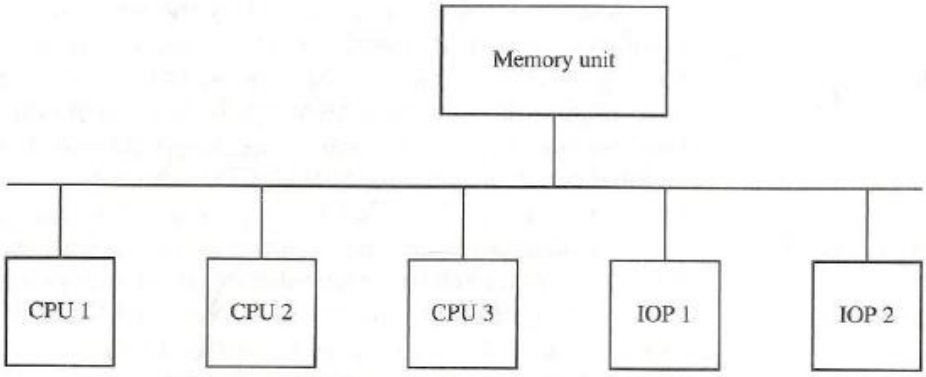


Figure 13-1 Time-shared common bus organization.

operations are conducted by the processor that is in control of the bus at the time. Any other processor wishing to initiate a transfer must first determine the availability status of the bus, and only after the bus becomes available can the processor address the destination unit to initiate the transfer. A command is issued to inform the destination unit what operation is to be performed. The receiving unit recognizes its address in the bus and responds to the control signals from the sender, after which the transfer is initiated. The system may exhibit transfer conflicts since one common bus is shared by all processors. These conflicts must be resolved by incorporating a bus controller that establishes priorities among the requesting units.

A single common-bus system is restricted to one transfer at a time. This means that when one processor is communicating with the memory, all other processors are either busy with internal operations or must be idle waiting for the bus. As a consequence, the total overall transfer rate within the system is limited by the speed of the single path. The processors in the system can be kept busy more often through the implementation of two or more independent buses to permit multiple simultaneous bus transfers. However, this increases the system cost and complexity.

Multiport Memory

A multiport memory system employs separate buses between each memory module and each CPU. This is shown in Fig. 13-3 for four CPUs and four memory modules (MMs). Each processor bus is connected to each memory module. A processor bus consists of the address, data, and control lines required to communicate with memory. The memory module is said to have four ports and each port accommodates one of the buses. The module must have internal control logic to determine which port will have access to memory at any given time. Memory access conflicts are resolved by assigning fixed priorities to each memory port. The priority for memory access associated with each processor may be established by the physical port position that its bus occupies in each module. Thus CPU 1 will have priority over CPU 2, CPU 2 will have priority over CPU 3, and CPU 4 will have the lowest priority.

The advantage of the multiport memory organization is the high transfer rate that can be achieved because of the multiple paths between processors and memory. The disadvantage is that it requires expensive memory control logic and a large number of cables and connectors. As a consequence, this interconnection structure is usually appropriate for systems with a small number of processors.

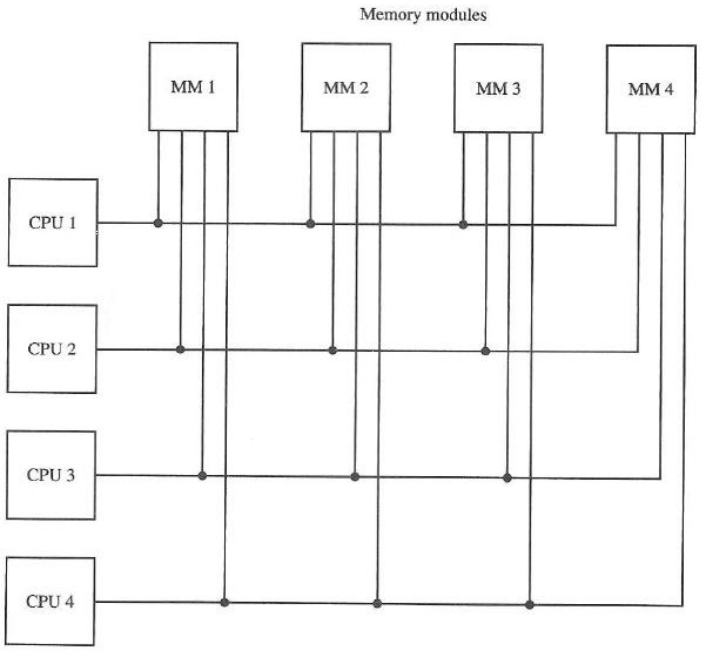


Figure 13-3 Multiport memory organization.

Crossbar Switch

The crossbar switch organization consists of a number of crosspoints that are placed at intersections between processor buses and memory module paths. Figure 13-4 shows a crossbar switch interconnection between four CPUs and four memory modules. The small square in each crosspoint is a switch that determines the path from a processor to a memory module. Each switch point has control logic to set up the transfer path between a processor and memory. It examines the address that is placed in the bus to determine whether its particular module is being addressed. It also resolves multiple requests for access to the same memory module on a predetermined priority basis.

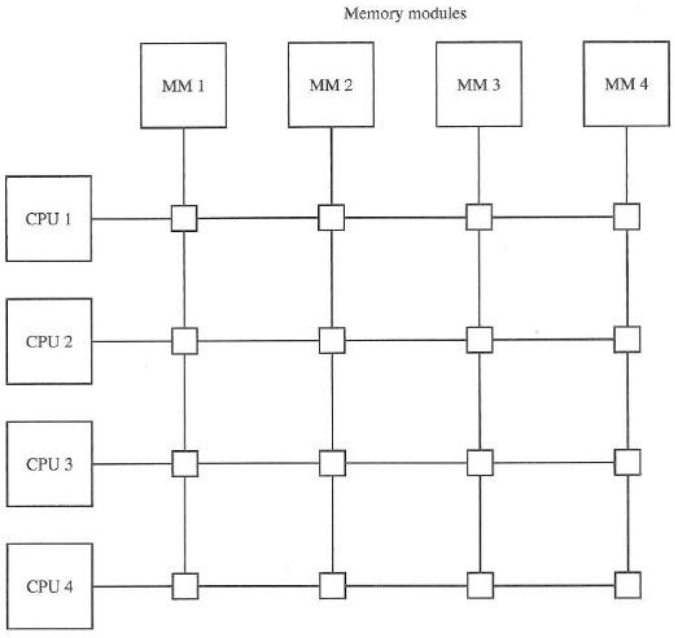


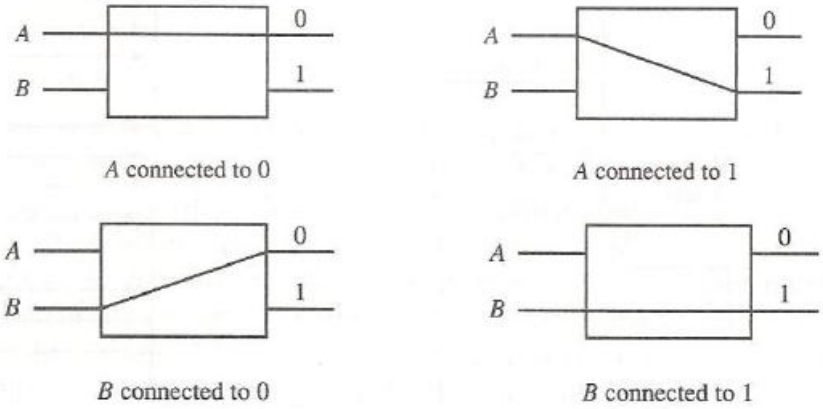
Figure 13-4 Crossbar switch.

A crossbar switch organization supports simultaneous transfers from memory modules because there is a separate path associated with each mo ule. However, the hardware required to implement the switch can becor quite large and complex.

Multistage Switching Network

The basic component of a multistage network is a two-input, two-output interchange switch. As shown in Fig. 13-6, the 2 × 2 switch has two input labeled *A* and *B*, and two outputs, labeled 0 and 1. There are control signal (not shown) associated with the switch that establish the interconnection between the input and output terminals. The switch has the capability connecting input *A* to either of the outputs.

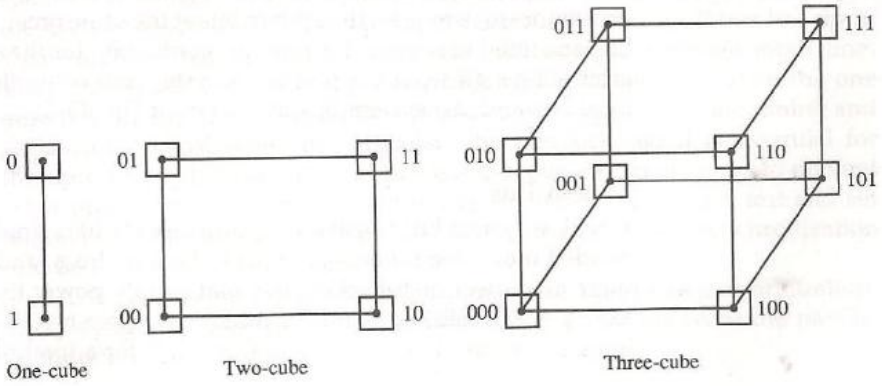
Figure 13-6 Operation of a 2 × 2 interchange switch.



Hypercube Interconnection

The hypercube or binary *n*-cube multiprocessor structure is a loosely coupled system composed of $N = 2^n$ processors interconnected in an *n*-dimensional binary cube. Each processor forms a *node* of the cube. Although it is customary to refer to each node as having a processor, in effect it contains not only a CPU but also local memory and I/O interface. Each processor has direct communication paths to *n* other neighbor processors. These paths correspond to the *edges* of the cube. There are 2^n distinct *n*-bit binary addresses that can be assigned to the processors. Each processor address differs from that of each of its *n* neighbors by exactly one bit position.

Figure 13-9 Hypercube structures for n = 1,2,3.



ARRAY PROCESSORS

* The SIMD form of parallel processing, also called array processing, was the first form of parallel processing to be implemented.

The following figure shows the structure of an array processor

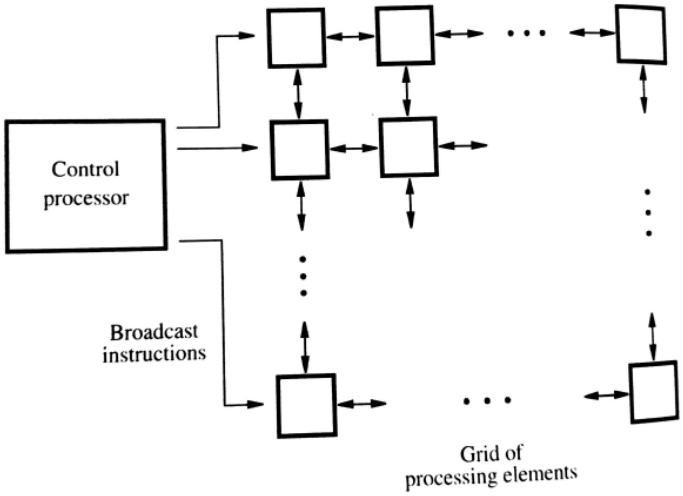


Figure 12.1 An array processor.

* A two-dimensional grid of processing elements execute an instruction stream that is broadcast from a central control processor.
* As each instruction is broadcast, all elements execute it simultaneously.
* Each processing element is connected to its four nearest neighbors for purpose of exchanging data.
* The capability needed in the array processor to perform such calculations is quite simple.
* Each element must be able to exchange values with each of its neighbors over the paths.
* Array processors are highly specialized machines as they are well suited to complex numerical problems.

THE STRUCTURE OF GENERAL PURPOSE MULTIPROCESSORS

The array processor architecture described in the previous section is a design for a computer system that corresponds directly to a class of computational problems that exhibit an obvious form of data parallelism.

In general cases in which parallelism is not so obvious, it is useful to have an MIMD architecture, which involves a number of processors capable of independently executing different routines in parallel.

Let us now discuss the three possible ways of implementing a multiprocessor system.

(1) A UMA multiprocessor:

* It is the most obvious scheme in which an interconnection network permits n processors to access K memories so that any of the processors can access any of the memories.

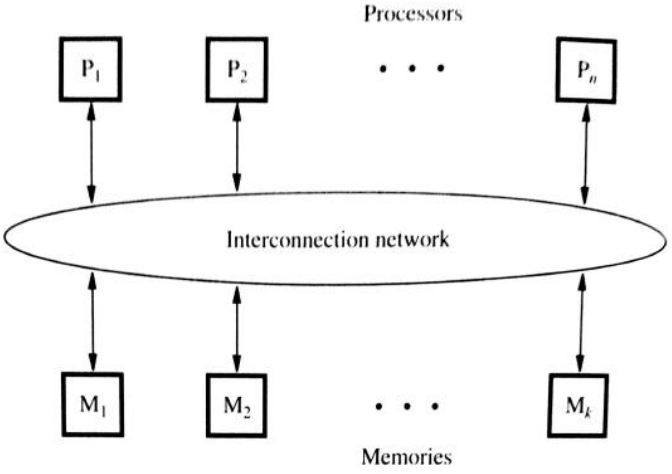


Figure 12.2 A UMA multiprocessor.

* The interconnection network may introduce considerable delay between a processor and a memory.
* If this delay is the same for all accesses to memory, which is common for this organization then such a machine is called a Uniform Memory Access (UMA) multiprocessor.

(II) A NUMA multiprocessor :

* An alternative, which allows a high computation rate to be sustained in all processors, is to attach the memory modules directly to the processors.

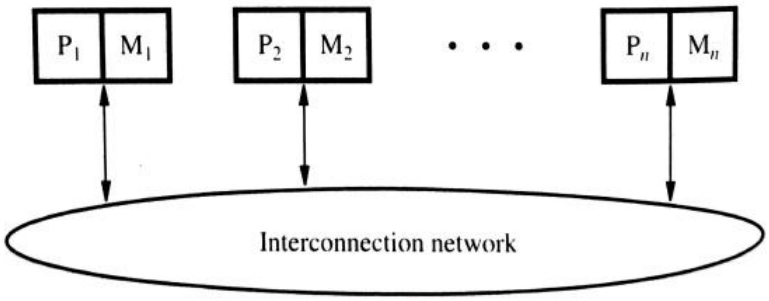


Figure 12.3 A NUMA multiprocessor.

* In addition to accessing its local memory, each processor can also access other memories over the network.
* Since the remode accesses pass through the network, these accesses take considerably longer than accesses to the local memory.
* Because of this difference in access times, such multiprocessors are called Non-Uniform Memory Access (NUMA) multiprocessor.

(III) A Distributed memory system:

* This organization provide a global memory, where any processor can access any memory module without intervention by another processor.

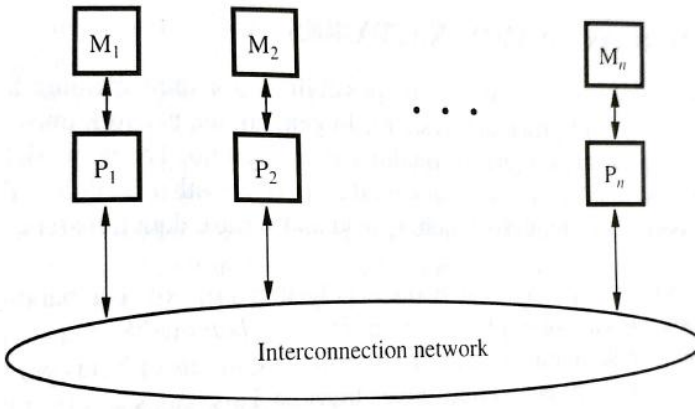


Figure 12.4 A distributed memory system.

* Here, all memory modules serve as private memories for the processors that are directly connected to them.
* A processor cannot access a remote memory without the cooperation of the remote processor.
* This co-operation takes place in the form of messages exchanged by the processors.
* Such systems are often called Distributed memory systems.

****

BTECH CSE
DEPARTMENT SPECIAL
BOOK ORGANISED BY
ADCC
A AD SIR'S INITIATIVE