MODULE 4
Microprocessor Architecture and Programming

Micro Processor Architecture and Programming

A number of processors (two or more) are connected in a manner that allows them to share the simultaneous execution of a single task. In addition, a multiprocessor consisting of a number of single uni-processors is expected to be more cost effective than building a high-performance single processor.

A multiprocessor is expected to reach faster speed than the fastest uni-processor. Multiprocessor characteristics are Interconnection Structures, Interprocessor Arbitration, Interprocessor Communication and Synchronization, Cache Coherence. Multiprocessing sometimes refers to the execution of multiple concurrent software processes in a system as opposed to a single process at any one instant. The 2 architectural models of Multiprocessor are:

- **Tightly Coupled Multiprocessor** are defined as Tasks and/or processors communicate in a highly synchronized fashion, Communicates through a common shared memory, Shared memory system
- **Loosely Coupled System** is defined as Tasks or processors do not communicate in a synchronized fashion, Communicates by message passing packets, Overhead for data exchange is high, Distributed memory system.

Functional Structures

Multiprocessors are characterized by 2 attributes:

- First a multiprocessor is a single computer that includes multiple processors
- Second processors may communicate and cooperate at different levels in solving a given problem.

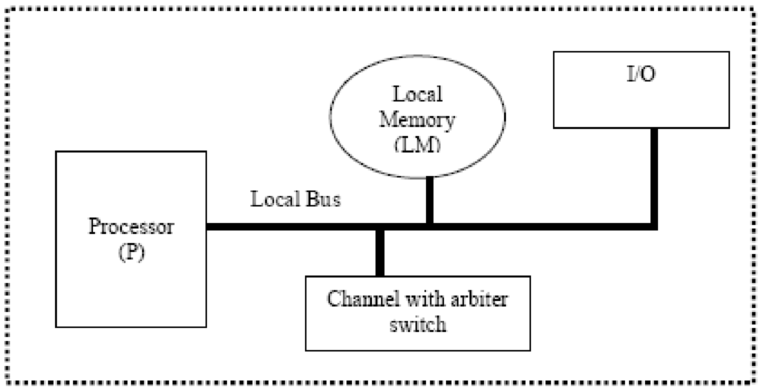
The communication may occur by sending messages from one processor to the other by sharing a common memory.

2 Different Architectural Models of Multiprocessor are

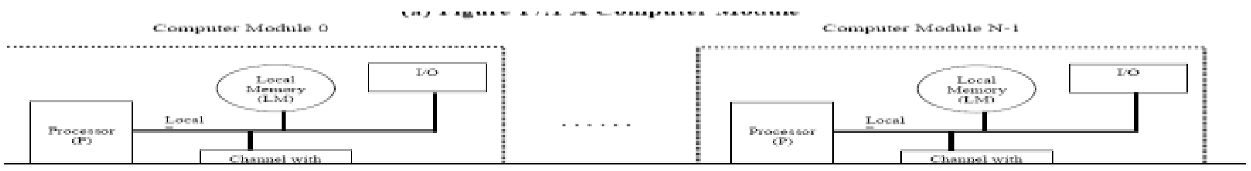
- Tightly Coupled System
  - Tasks and/or processors communicate in a highly synchronized fashion
  - Communicates through a common shared memory
  - Shared memory system
- Loosely Coupled System
  - Tasks or processors do not communicate in a synchronized fashion
  - Communicates by message passing packets
  - Overhead for data exchange is high
  - Distributed memory system

Loosely Coupled Multiprocessors

- In loosely coupled Multiprocessors each processor has a set of input-output devices and a large local memory from where instructions and data are accessed.
- Computer Module is a combination of
  - Processor
  - Local Memory
  - I/O Interface
- Processes which executes on different computer modules communicate by exchanging messages through a Message Transfer System (MTS).
- The Degree of coupling is very loose. Hence it is often referred as distributed system.
- Loosely coupled system is usually efficient when the interactions between tasks are minimal



(a) Figure 17.1 A Computer Module



It consists of a

- processor
- local memory
- Local Input-output devices
- Interface to other computer Modules
- The interface may contain a channel and arbiter switch (CAS)
- If requests from two or more different modules collide in accessing a physical segment of the MTs, the arbiter is responsible for choosing one of the simultaneous requests according to a given service discipline.

- It is also responsible for delaying requests until the servicing of the selected request is completed.
- The channel within the CAS may have a high speed communication memory which is used for buffering block transfers of messages.
- The message transfer system is a time shared bus or a shared memory system.

Tightly Coupled Multiprocessors (TCS)

The throughput of the hierarchical loosely coupled multiprocessor may be too slow for some applications that require fast response times. If high speed or real time processing is required the TCS may be used.
Two typical models are discussed:

In the first model, it consists of

- p processors
- l memory modules
- d input-output channels

The above units are connected through a set of three interconnection networks namely the

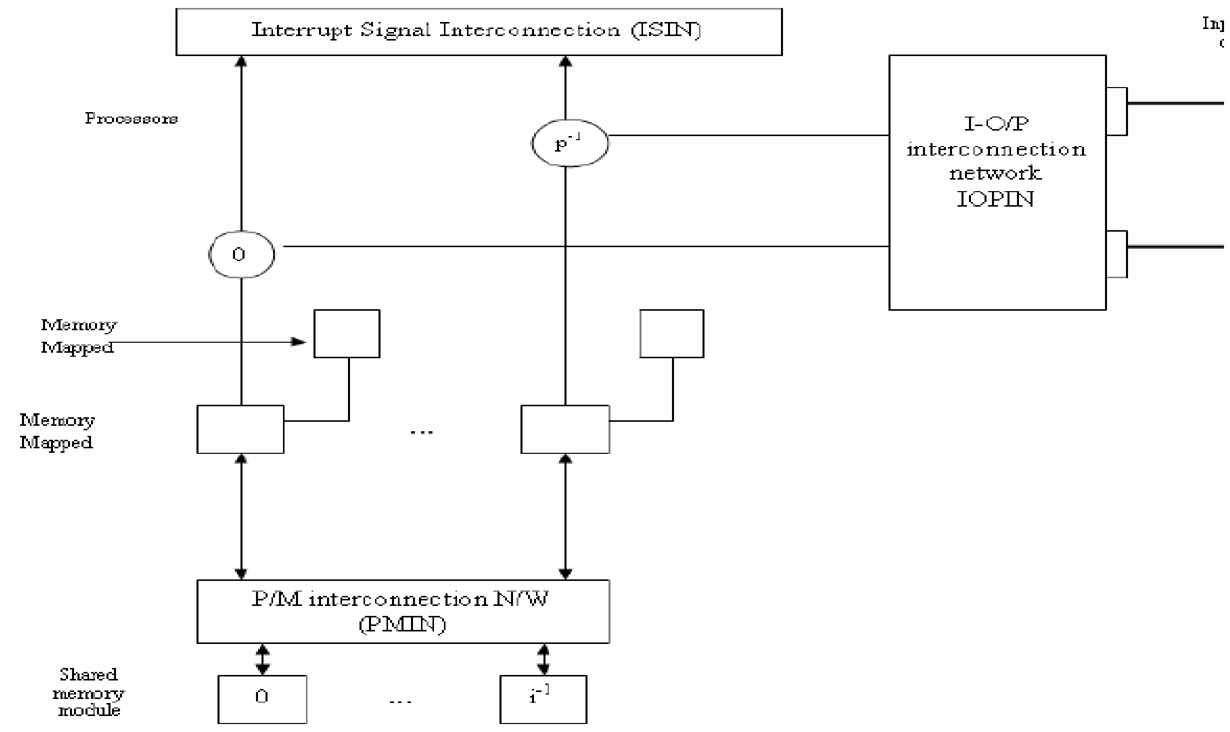
- processor-memory interconnection network (PMIN)
- I/O processor interconnection network (IOPIN)
- Interrupt Signal Interconnection Network (ISIN)

The PMIN is a switch which can connect every processor to every module. It has pl set of cross points. It is a multistage network. A memory can satisfy only one processor's request in a given memory cycle. Hence if more than one processors attempt to access the same memory module, a conflict occurs and it is resolved by the PMIN.

Another method to resolve the conflict is to associate a reserved storage area with each processor and it is called as ULM unmapped local memory. It helps in reducing the traffic in PMIN.Since each memory references goes through the PMIN, it encounters delay in the processors memory switch so this can be overcome by using a private cache with each processor.

A multiprocessor organization which uses a private cache with each processor is shown. This multiprocessor organization encounters the cache coherence problem. More than one consistent copy of data may exist in the system.

In the figure there is a module attached to each processor that directs the memory reference to either the ULM or the private cache of that processor. This module is called the memory map and is similar in operation to the Slocal.



Interrupt Signal Interconnection (ISIN)
Processors
p-1
0
I-O/P interconnection network IOPIN
Memory Mapped
Memory Mapped
...
P/M interconnection N/W (PMIN)
Shared memory module
0
...
i-1

Figure 17.4 Tightly Configured Multiprocessor configurations

(a) Without private cache

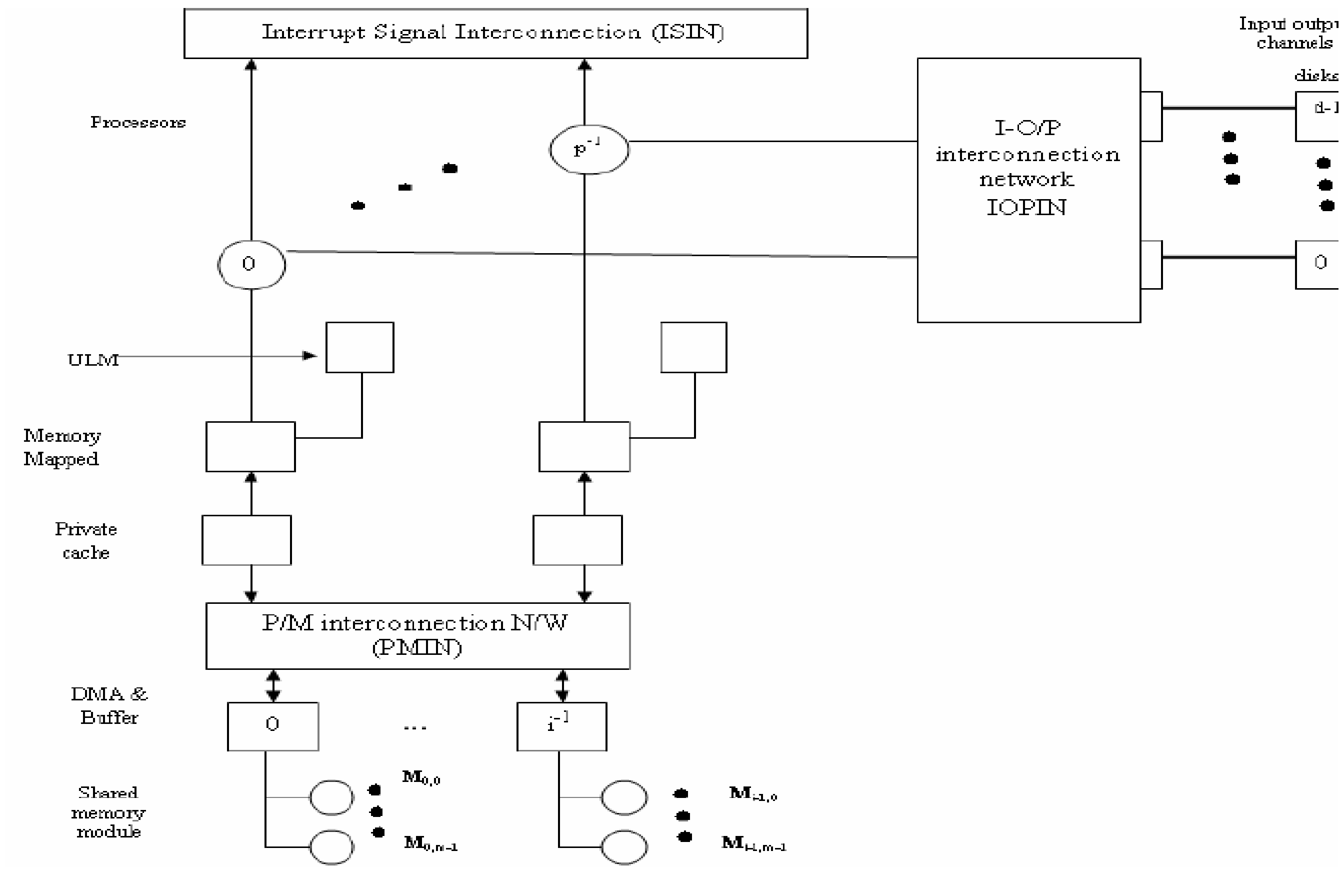


Figure 17.4 (Continued)

(b) With private cache

Example of Tightly Coupled Multiprocessor: The Cyber – 170 Architecture

This configuration has 2 subsystems.

- The central processing sub system
- The peripheral processing sub system

These subsystems have to access to a common central memory (CM) through a central memory controller, which is essentially a high speed cross bar switch. In addition to the central memory there is an optional secondary memory called the extended core memory (ECM) which is a low speed random access read-write memory. The ECM and CM form a 2 level memory hierarchy. Here the CMC becomes the switching center, which performs the combined functions of ISIN,IOPIN, and PMIN.

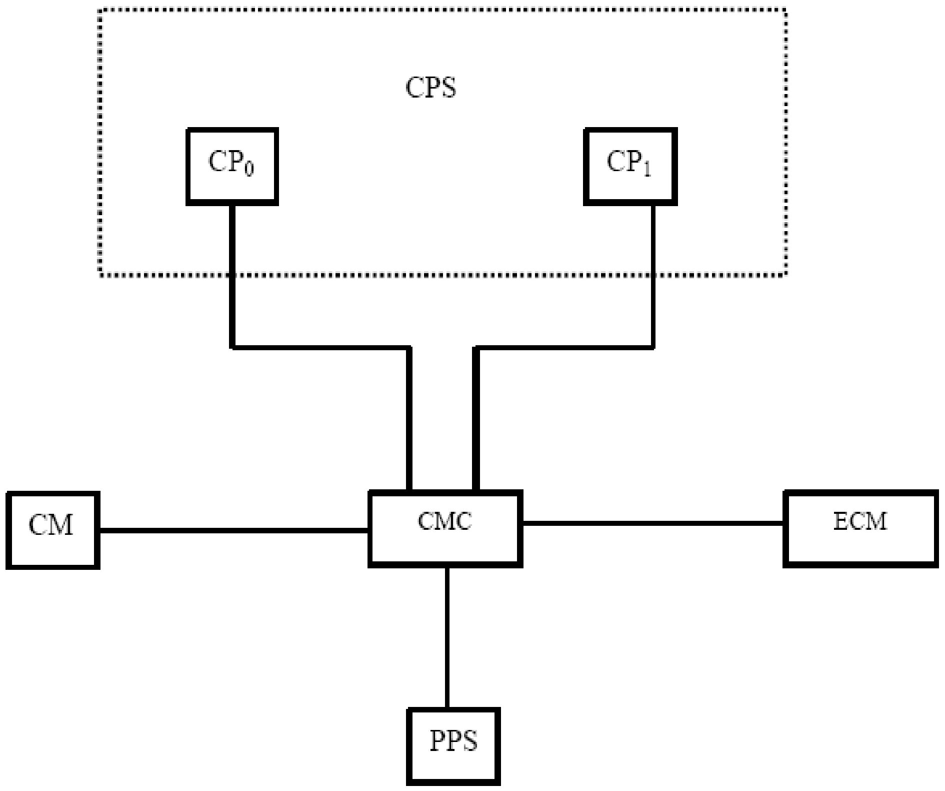


Figure 17.5 A Cyber-170 Multiprocessor configurations with the two processors

CM: Central Memory

CMC: Central Memory Controller

CP_i: ith central processor

CPS: Central Processing System

PPS: peripheral processing subsystem

Processor Characteristics for Multiprocessing

A number of desirable architectural features are described below for a processor to be effective in multiprocessing environment.

Processor recoverability

The process and processor are 2 different entities. If the processor fails there should be another processor to take up the routine. Reliability of the processor should be present.

Efficient Context Switching

A general purpose register is a large register file that can be used for multi-programmed processor. For effective utilization it is necessary for the processor to support more than one addressing domain and hence to provide a domain change or context switching operation.

Large Virtual and Physical address space

A processor intended to be used in the construction of a general purpose medium to large scale multiprocessor must support a large physical address space. In addition a large virtual space is also needed.

Effective Synchronization Primitives

The processor design must provide some implementation of invisible actions which serve as the basis for synchronization primitives.

Interprocessor Communication mechanism

The set of processor used in multiprocessor must have an efficient means of interprocessor mechanism.

Instruction Set

The instruction set of the processor should have adequate facilities for implementing high level languages that permit effective concurrency at the procedural level and for efficiently manipulating data structures.

Interconnection Networks

Interconnection networks helps in sharing resources, one is between the processor and the memory modules and the other between the processor and the I/O Systems. There are different physical forms available for interconnection networks.

They are:

- Time shared or common bus,
- Crossbar switch
- Multistage networks for multiprocessors.

The simplest interconnection system for multiprocessors is a communication path connecting all of the functional units. The common path is called as time shared or common bus.

The organization is least complex and easier to configure. Such an interconnection is often a totally passive unit having no active components such as switches. Transfer operations are completely controlled by the bus interfaces of the sending and receiving units. Since the bus is shared, a mechanism must provide to resolve contention. The conflict resolution methods include static or fixed priorities, FIFO queues and daisy chaining.

A unit that wishes to initiate transfer must first determine the availability of status of the bus, and then address the destination unit to determine its availability and capability to receive the transfer. A command is also issued to inform the destination unit what operation it is to perform with the data being transferred, after which the data transfer is finally initiated. A receiving unit recognizes its address placed on the bus and responds of the control signals from the sender.

Example of Time shared bus is PDP – 11.

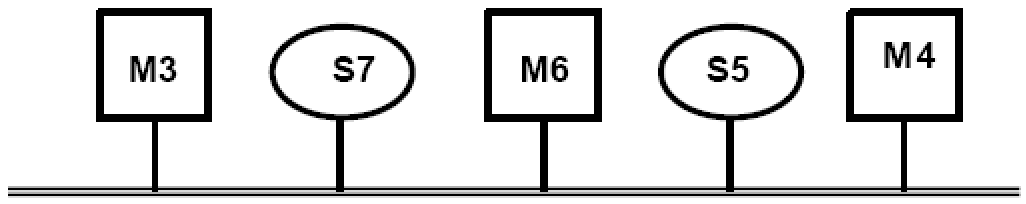


Figure 18.1 A single bus Multiprocessor organization

Multiprocessor with unidirectional buses uses both the buses and not much is actually gained

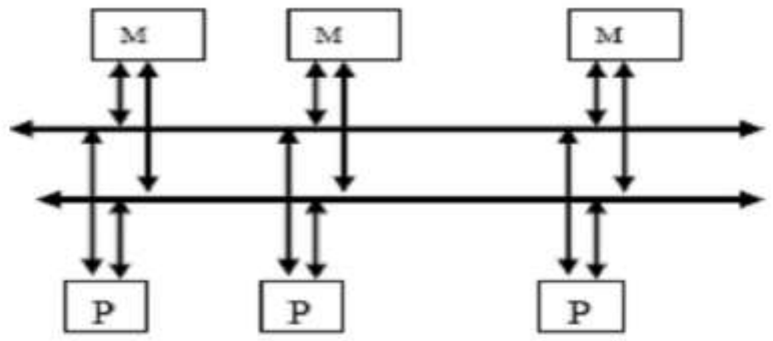


Figure 18.2 Multi-bus multiprocessor organization

This method increases the complexity. The interconnection subsystem becomes an active device.

Characteristics that affects the performance of the bus

- Number of active devices on the bus
- Bus arbitration Algorithm
- Centralization
- Data Bandwidth
- Synchronization of data transmission
- Error detection

Various Bus arbitration Algorithms are:

The static Priority Algorithm

When multiple devices concurrently request the use of bus, the device with highest priority is granted the bus. This method is called as daisy chaining and all the services are assigned static priorities according to their locations along a bus grant control line. The device closest to the bus control unit will get highest priority.

Requests are made through a common line, BRQ Bus request. If the central bus control accepts the request it passes a BGT Bus grant signal to the concerned device if the SACK Bus busy line is free or idle.

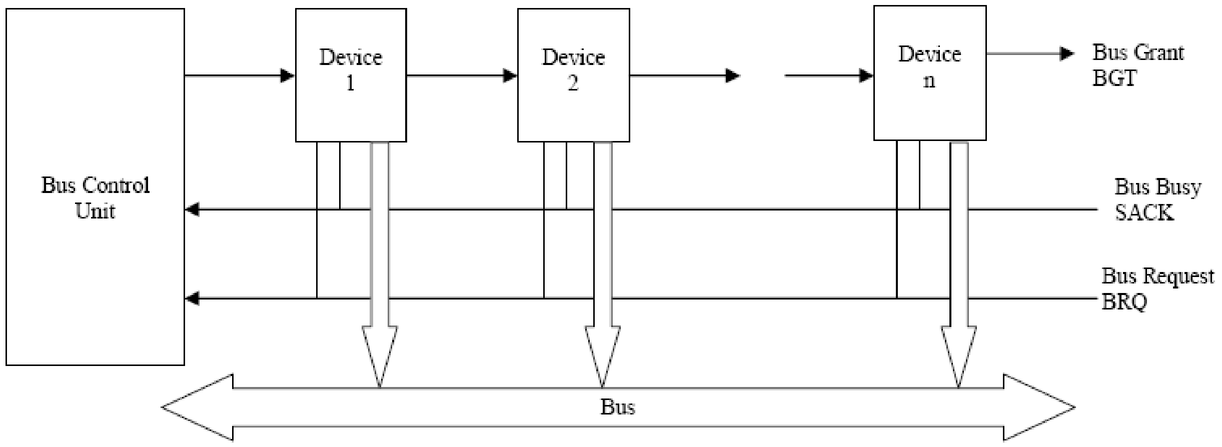


Figure 18.3 Static daisy chain implementation of a system bus

The Fixed Time Slice Algorithm

The available bus band widths divided into fixed time slices and then offered to various devices in around robin fashion. If the allotted device does not use the time slice then the time slice is wasted. This is called as fixed time slicing or time division multiplexing.

Dynamic Priority Algorithm

The priorities allocated to the devices are done dynamically and thus every device gets a chance to use the bus and does not suffer longer turnaround time.
The two algorithms are

- Least Recently used (LRU)
- Rotating daisy chain (RDC)

The LRU gives the highest priority to the requesting device that has not used the bus for the longest interval.

In a RDC scheme, no central controller exists, and the bus grant line is connected from the last device back to the first in a closed loop. Whichever device is granted access to the bus serves as a bus controller for the following arbitration (an arbitrary device is selected to have initial access to the bus). Each device's priority for a given arbitration is determined by that device's distance along the bus grant line from the device currently serving as a bus controller; the latter device has the lowest priority. Hence the priorities are dynamically with each bus cycle.

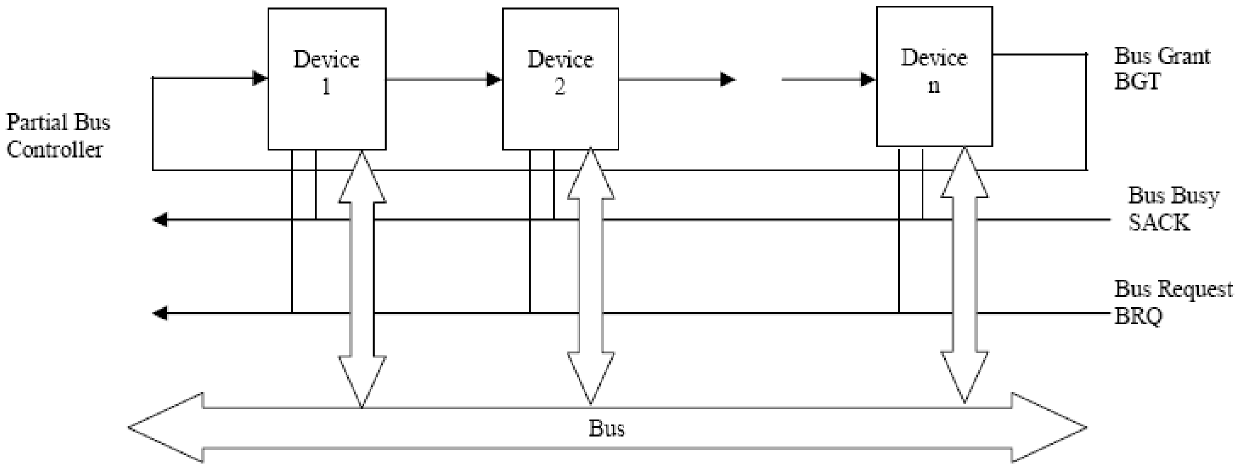


Figure 18.4 Rotating daisy chain implementation of a system bus

The first come First Served Algorithm

In the FCFS, requests are simply honored in the order received. It favors no particular processor or device on the bus. The performance measures will be degraded in case of FCFS since the device waiting for the bus for a longer period of time just because the request is made late.

Crossbar Switch and Multiport Memories

If the number of buses in a time shared bus is increased, a point is reached at which there is a separate path available for each memory unit. The interconnection networking is called as a nonblocking crossbar.

Multiport Memory Module

- Each port serves a CPU

Memory Module Control Logic

- Each memory module has control logic
- Resolve memory module conflicts fixed priority among CPUs

Advantages

- Multiple paths -> high transfer rate

Disadvantages

- Memory control logic
- Large number of cables and connections

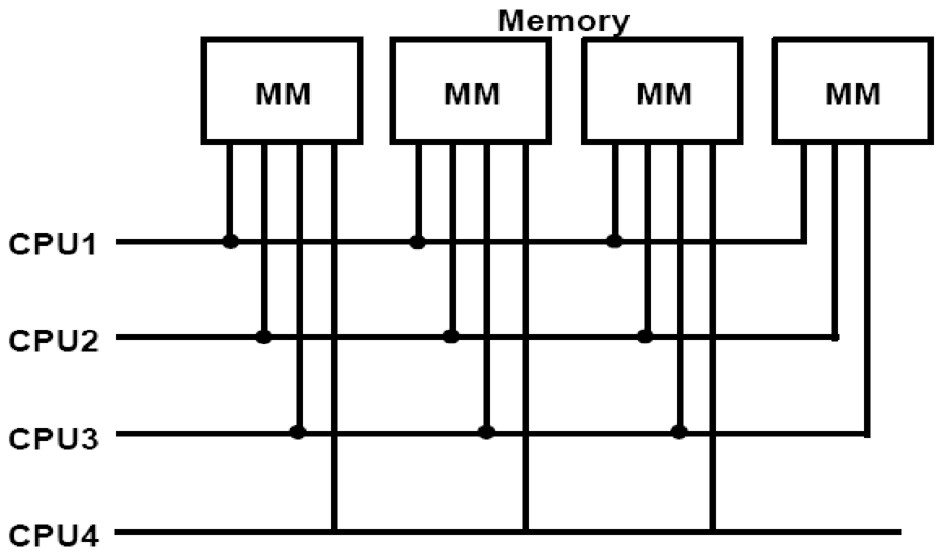


Figure 18.5 Cross Bar switch system organization for Multiprocessors

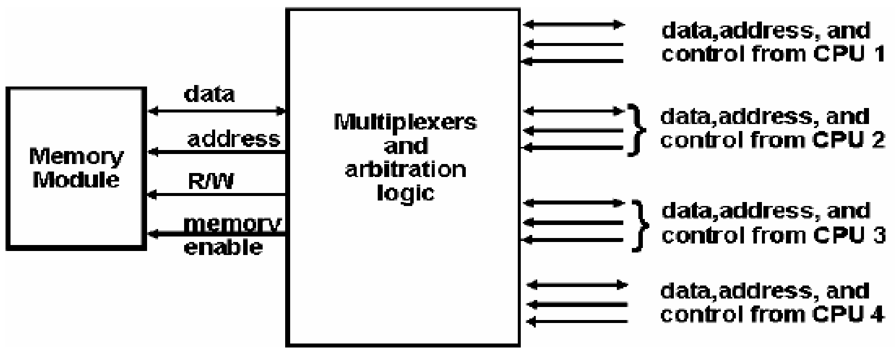

Memory Module
data
address
R/W
memory enable
Multiplexers and arbitration logic
data,address, and control from CPU 1
data,address, and control from CPU 2
data,address, and control from CPU 3
data,address, and control from CPU 4

Figure 18.6 Functional structure of a crosspoint in a crossbar network

Multistage Networks for Multiprocessors

In order to design multistage networks, the basic principles involved in the construction and control of simple crossbar has to be understood.

This 2 x 2 switch has the capability of connecting the input A to either the output labeled 0 or to the output labeled 1, depending on the value of some control bit C_A of the input A. If $C_A = 0$ the input is connected to the upper output, and if $C_A = 1$, the connection is made to the lower output. Terminal B of the switches behaves similarly with a control bit C_B. The 2 x 2 module also has the capability to arbitrate between conflicting requests. If both inputs A and B require the same output terminal, then only one of them will be connected and the other will be blocked or rejected.

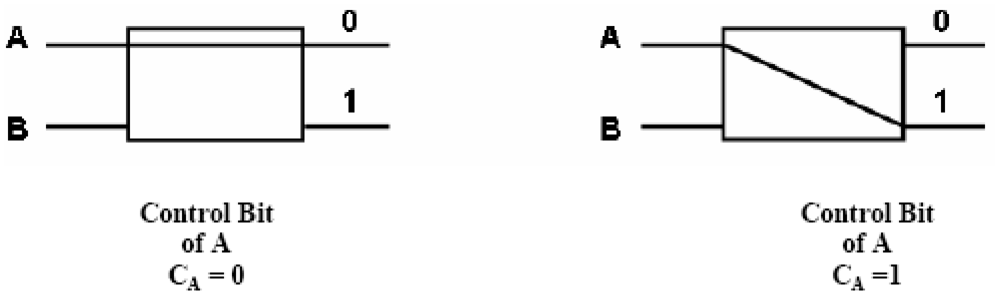


Figure 18.6 A 2 x2 Crossbar Switch

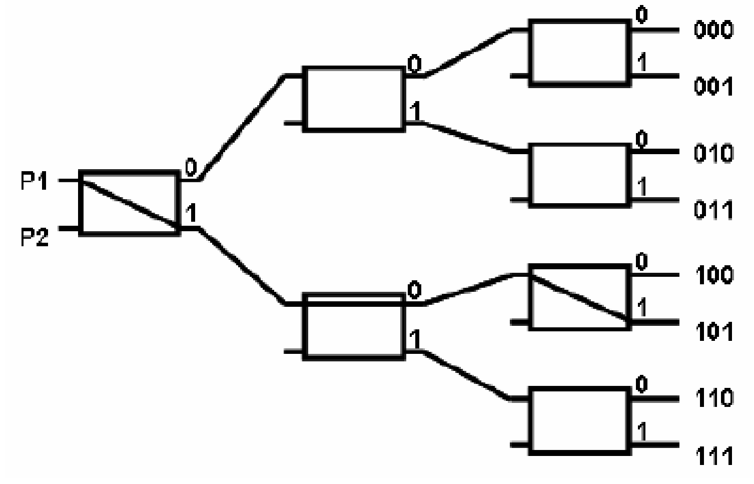


Figure 18.7 1-by-8 demultiplexer implemented with 2 x 2 switch boxes

Comparison of three multiprocessor hardware organizations.

Multiprocessor with Time Shared Bus

1. Lowest overall system cost for hardware and least complex
2. Very easy to physically modify the hardware system configuration by adding or removing functional units
3. Overall system capacity limited by bus transfer rate. Failure of the bus is a catastrophic failure.
4. Expanding the system by addition of functional units may degrade overall system performance
5. The system efficiency attainable is the lowest of all three basic interconnection systems.
6. This organization is usually appropriate for smaller systems only.

Multiprocessor with Crossbar Switch

1. This is the most complex interconnection system. There is a potential for the highest total transfer rate.
2. The functional units are the simplest and cheapest since the control and switching logic is in the switch
3. Because a basic switching matrix is required to assemble any functional units into a working configuration, this organization is usually cost effective for multiprocessors only.
4. System expansions usually improve overall performance.
5. Theoretically, expansion of the system is limited only by the size of the switch matrix, which can often be modularly expanded within initial design or other engineering limitations.
6. The reliability of the switch, and therefore the system can be improved by segmentation and / or redundancy within the switch.

Multiprocessors with Multiport Memory

1. Requires the most expensive memory units since most of the control and switching circuitry is included in the memory unit
2. The characteristics of the functional units permit a relatively low cost uniprocessor to be assembled from them.
3. There is potential for a very high total transfer rate in the overall system.
4. The size and configuration options possible are determined by the number and type of memory ports available; this design decision is made quite early in the overall design and is difficult to modify.
5. A large number of cables and connectors are required.