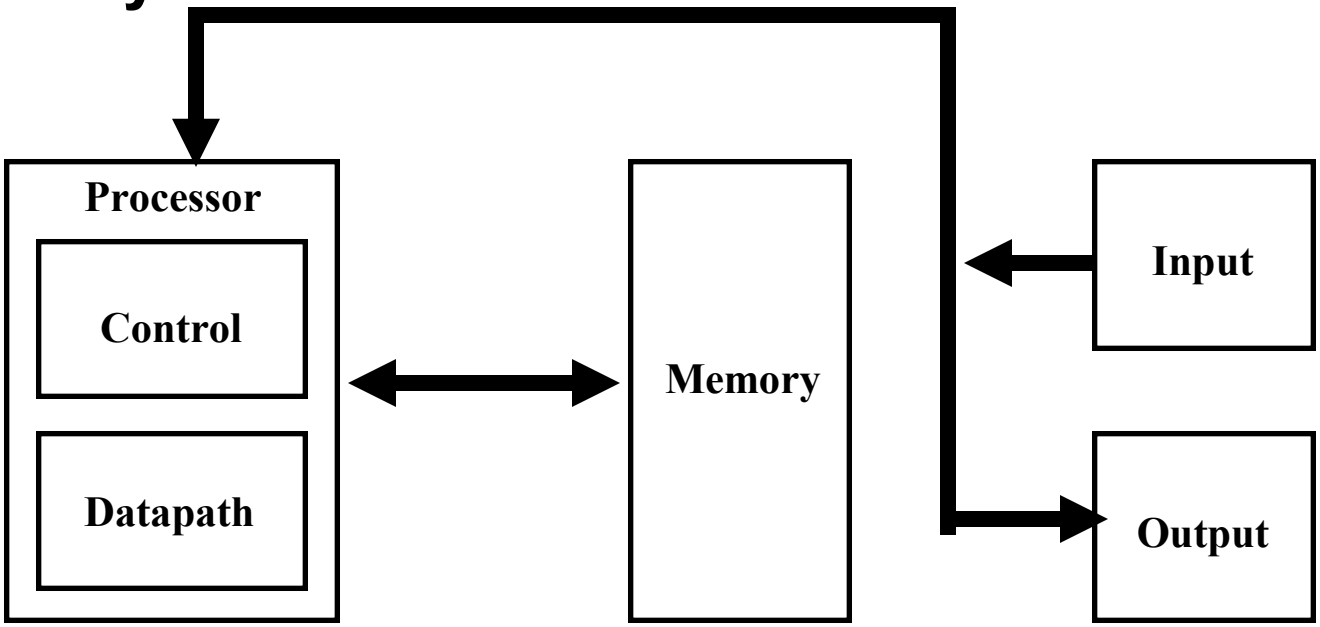
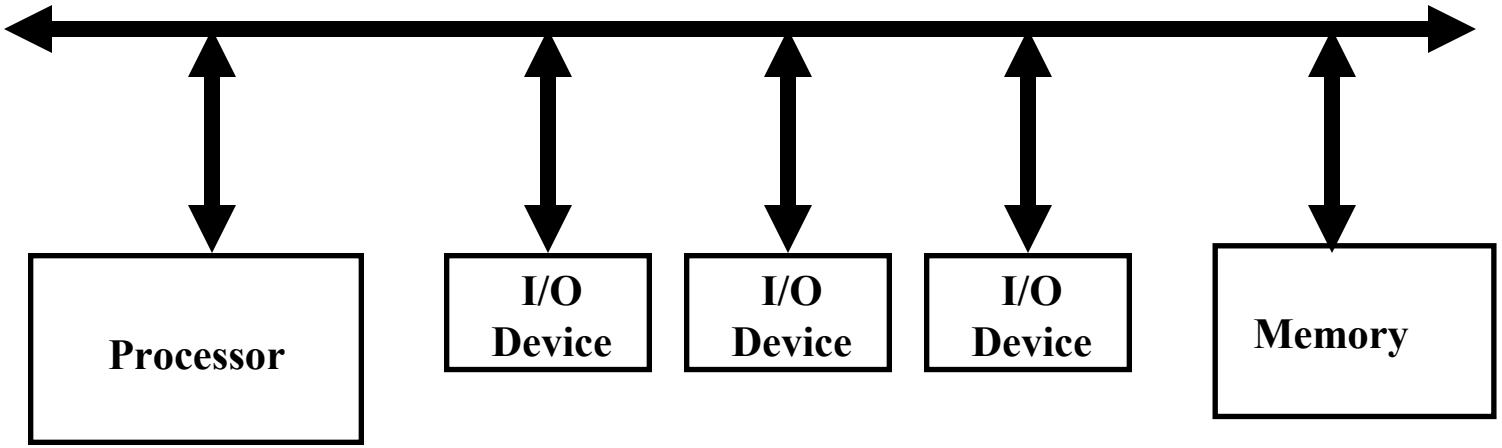
Buses: Connecting I/O to Processor and Memory



- A bus is a shared communication link
- It uses one set of wires to connect multiple subsystems

Advantages of Buses

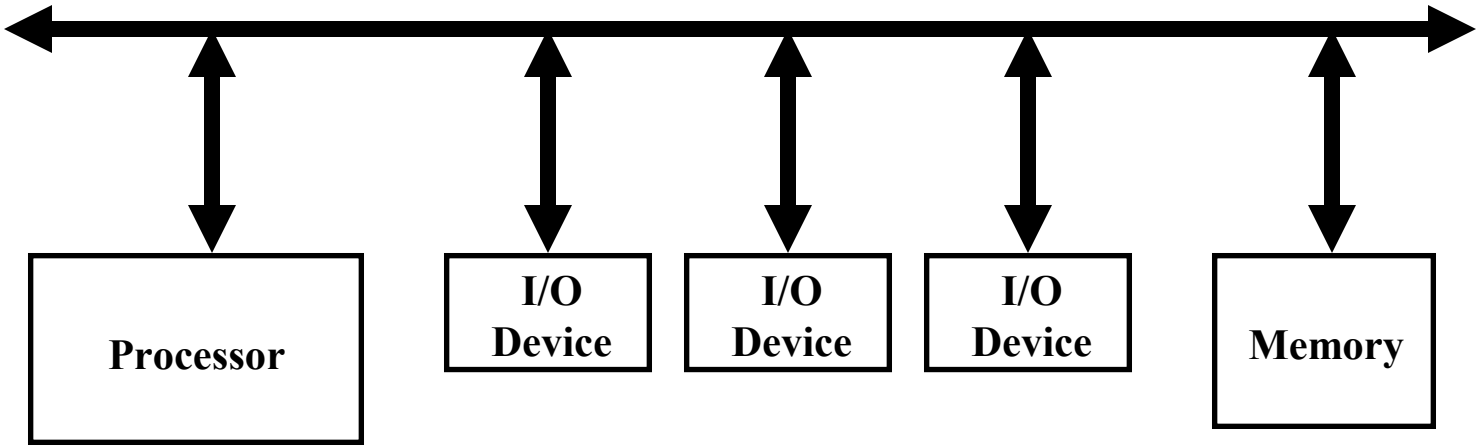


- New devices can be added easily
- Peripherals can be moved between computer systems that use the same bus standard

° Low Cost:

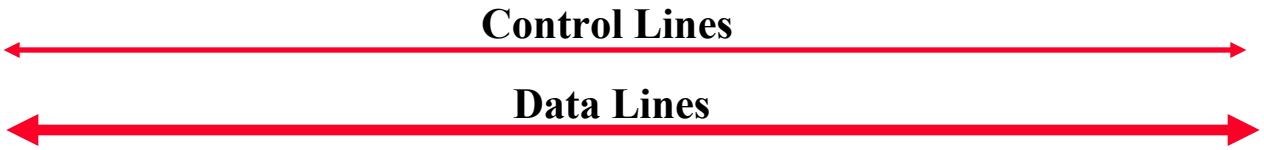
- A single set of wires is shared in multiple ways

Disadvantages of Buses



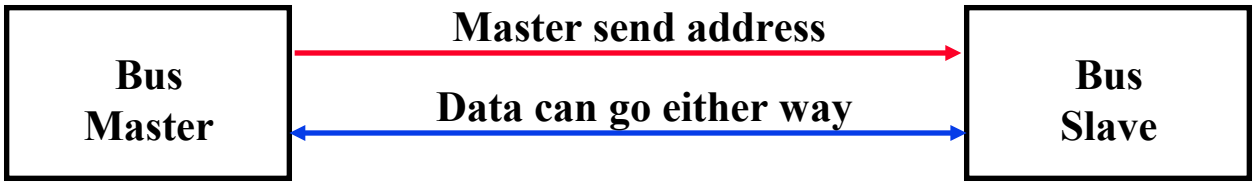
- It creates a communication bottleneck
  - The bandwidth of that bus can limit the maximum I/O throughput
- The maximum bus speed is largely limited by:
  - The length of the bus
  - The number of devices on the bus

The General Organization of a Bus



- Control lines:
  - Signal requests and acknowledgments
  - Indicate what type of information is on the data lines
- Data lines carry information between the source and the destination:
  - Data and Addresses
  - Complex commands
- A bus transaction includes two parts:
  - Sending the address
  - Receiving or sending the data

Master versus Slave

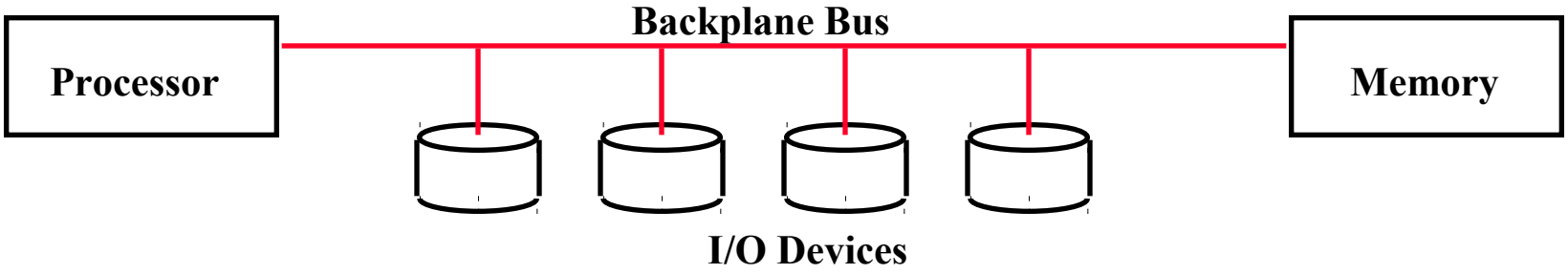


- Master is the one who starts the bus transaction by:
  - Sending the address
- Salve is the one who responds to the address by:
  - Sending data to the master if the master ask for data
  - Receiving data from the master if the master wants to send data

Types of Buses

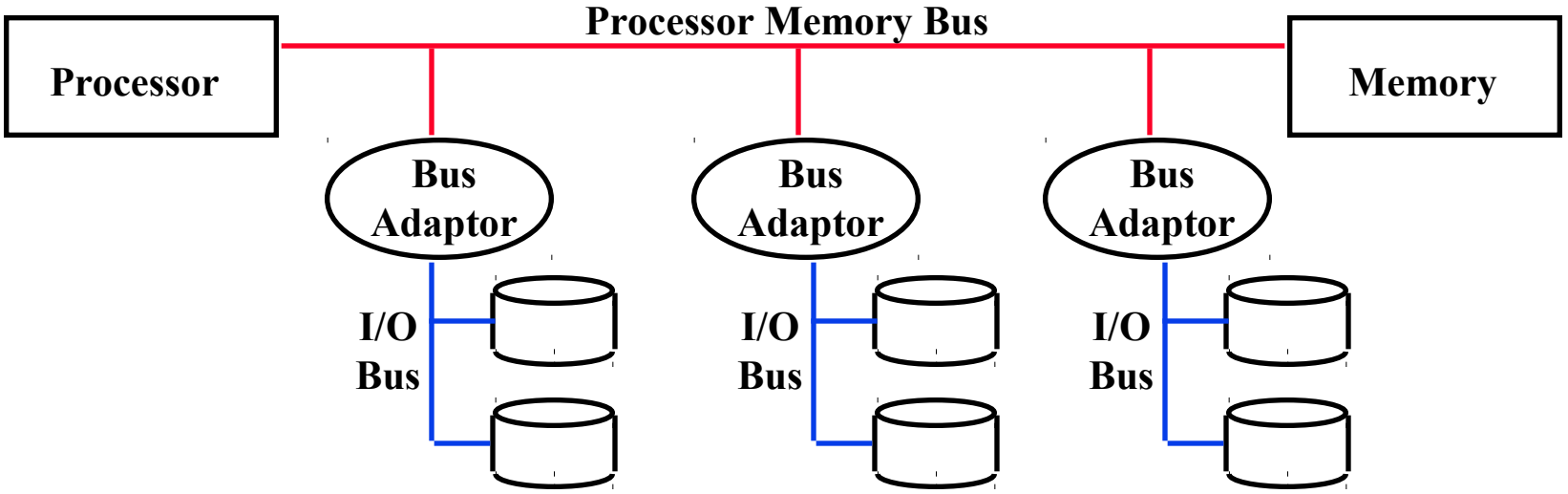
- Processor-Memory Bus
  - Short and high speed
  - Only need to match the memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to the processor
- I/O Bus
  - Usually is lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus
- Backplane Bus
  - Backplane: an interconnection structure within the chassis
  - Allow processors, memory, and I/O devices to coexist
  - Cost advantage: one single bus for all components

A Computer System with One Bus: Backplane Bus



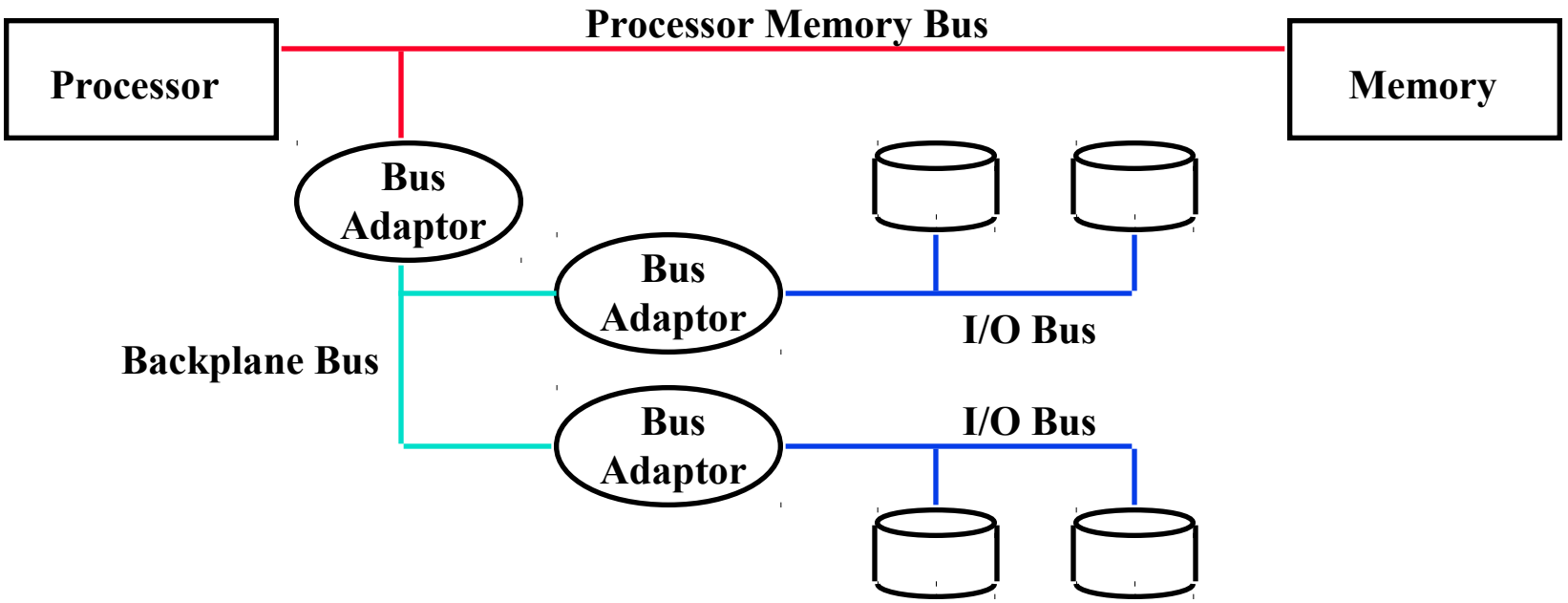
- A single bus (the backplane bus) is used for:
  - Processor to memory communication
  - Communication between I/O devices and memory
- Example: IBM PC

A Two-Bus System



- I/O buses tap into the processor-memory bus via bus adaptors:
  - Processor-memory bus: mainly for processor-memory traffic
  - I/O buses: provide expansion slots for I/O devices
- Apple Macintosh-II
  - NuBus: Processor, memory, and a few selected I/O devices
  - SCCI Bus: the rest of the I/O devices

A Three-Bus System



- A small number of backplane buses tap into the processor-memory bus
  - Processor-memory bus is used for processor memory traffic
  - I/O buses are connected to the backplane bus
- Advantage: loading on the processor bus is greatly reduced

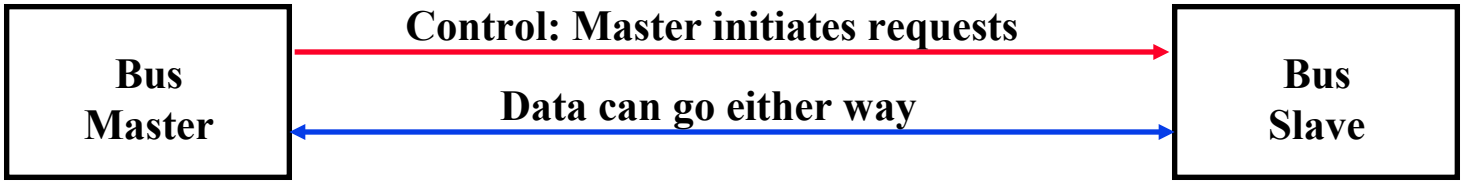
USB

- **Universal Serial Bus**
  - **Originally developed in 1995 by a consortium including**
    - **Compaq, HP, Intel, Lucent, Microsoft, and Philips**
  - **USB 1.1 supports**
    - **Low-speed devices (1.5 Mbps)**
    - **Full-speed devices (12 Mbps)**
  - **USB 2.0 supports**
    - **High-speed devices**
      - Up to 480 Mbps (a factor of 40 over USB 1.1)
    - **Uses the same connectors**
      - Transmission speed is negotiated on device-by-device basis

Synchronous and Asynchronous Bus

- **Synchronous Bus:**
  - **Includes a clock in the control lines**
  - **A fixed protocol for communication that is relative to the clock**
  - **Advantage: involves very little logic and can run very fast**
  - **Disadvantages:**
    - **Every device on the bus must run at the same clock rate**
    - **To avoid clock skew, they cannot be long if they are fast**
- **Asynchronous Bus:**
  - **It is not clocked**
  - **It can accommodate a wide range of devices**
  - **It can be lengthened without worrying about clock skew**
  - **It requires a handshaking protocol**

Obtaining Access to the Bus

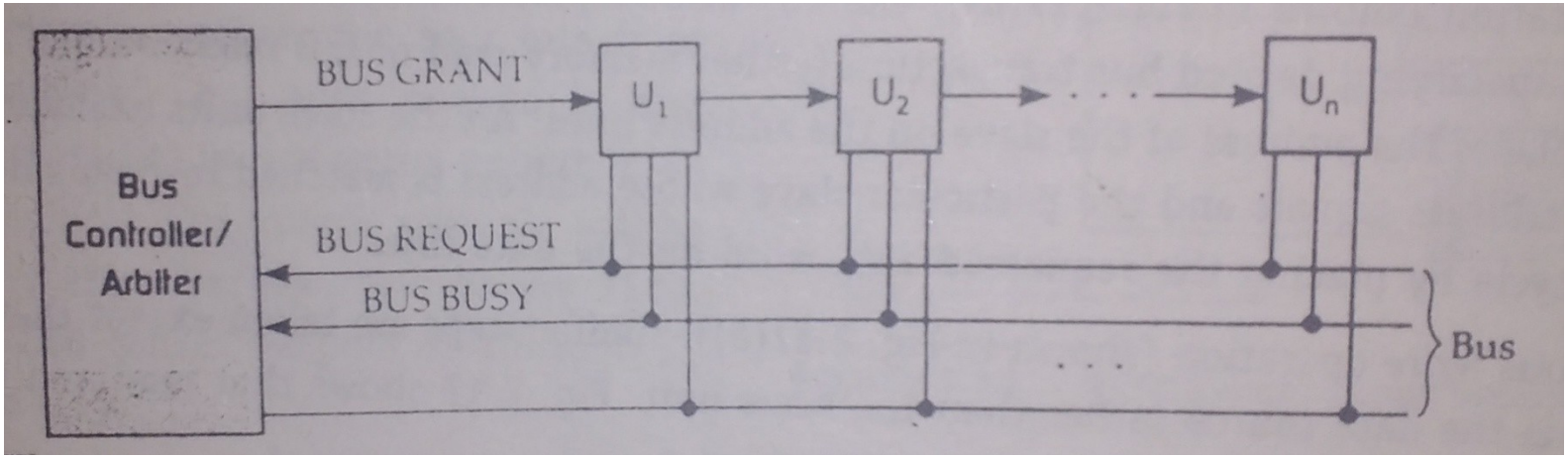


- One of the most important issues in bus design:
  - How is the bus reserved by a devices that wishes to use it?
- Chaos is avoided by a master-slave arrangement:
  - Only the bus master can control access to the bus:
    It initiates and controls all bus requests
  - A slave responds to read and write requests
- The simplest system:
  - Processor is the only bus master
  - All bus requests must be controlled by the processor
  - Major drawback: the processor is involved in every transaction

Bus Arbitration

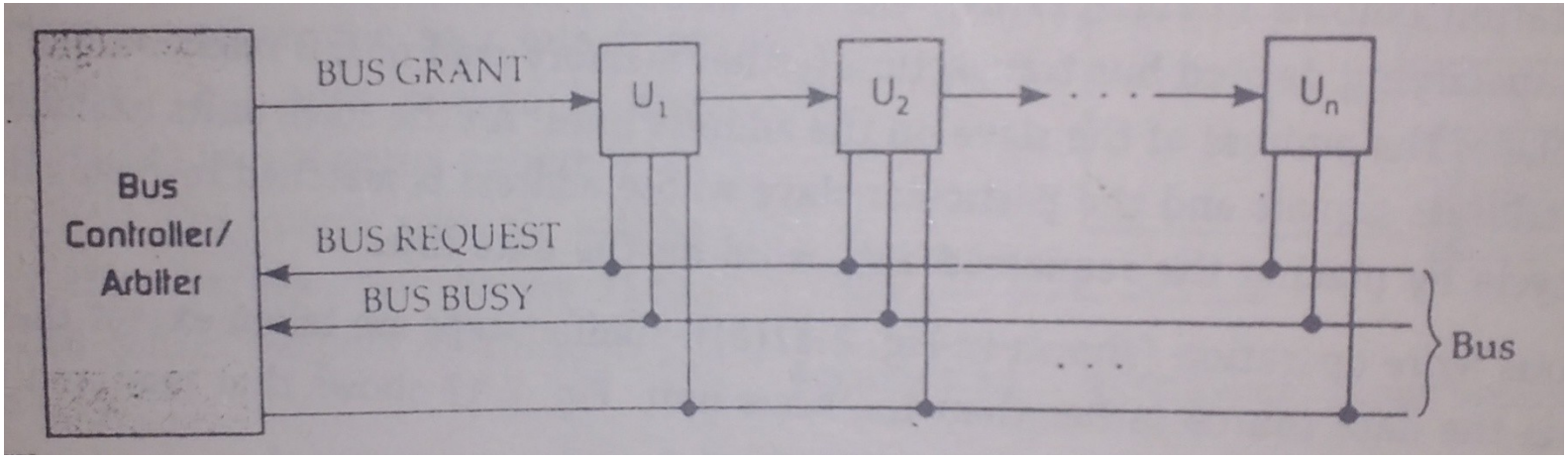
- The arbitration procedure comes into picture whenever there are more than one processors requesting the services of bus.
- Because only one unit may at a time be able to transmit successfully over the bus, there is some selection mechanism is required to maintain such transfers. This mechanism is called as *Bus Arbitration.*
- Bus arbitration decides which component will use the bus among various competing requests.
- A selection mechanism must be based on fairness or priority basis.
- Various methods are available that can be roughly classified as either centralized or distributed.
- There are three arbitration schemes-
  - Daisy – chaining
  - Polling
  - Independent requesting

The Daisy Chaining Scheme



- In this method, all requesting components are attached serially on to the bus.
- This method involves three control signals-
  - BUS REQUEST
  - BUS GRANT
  - BUS BUSY
- All the bus units are connected to BUS REQUEST line.
- When activated, it indicates that one or more devices are requesting to use the bus.
- Bus Controller responds to a BUS REQUEST only if BUS BUSY is inactive. When bus control is given to requesting device, it enables its physical bus connection and activates BUS BUSY.

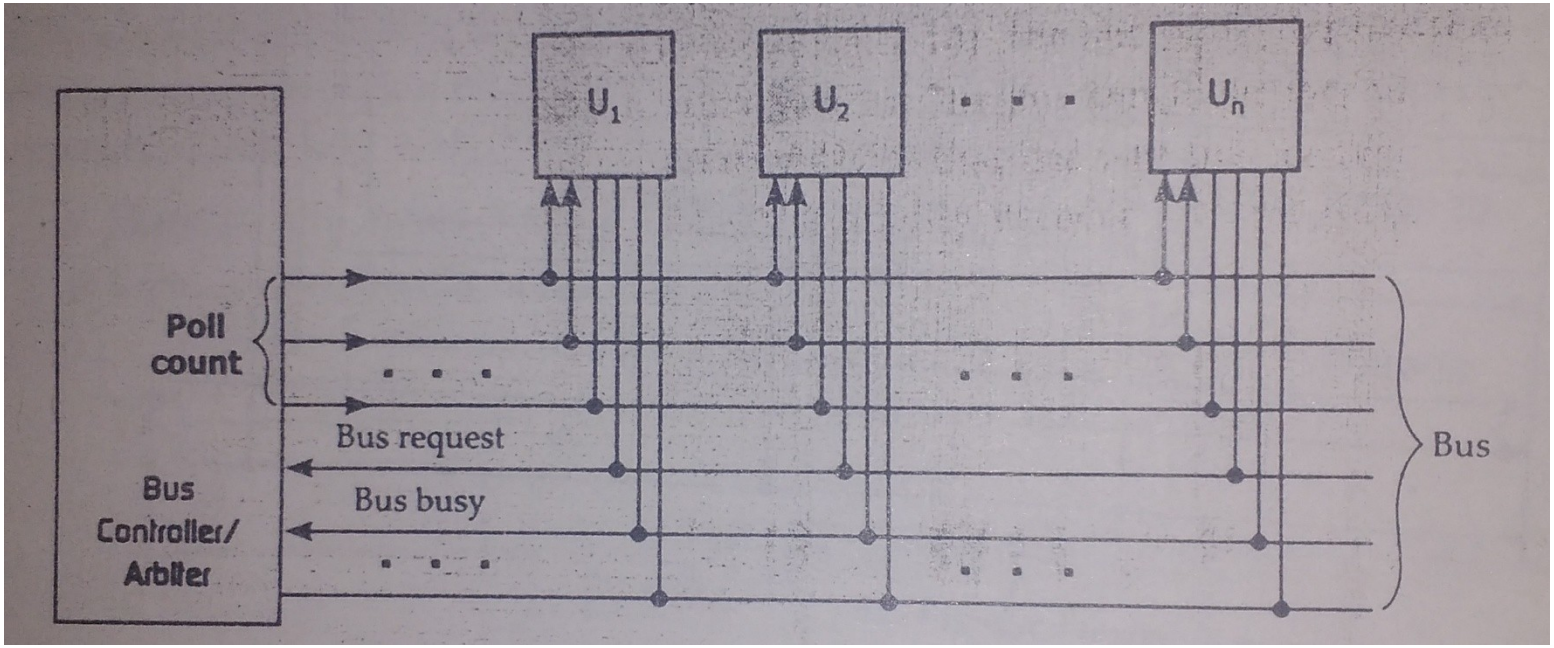
The



ons Sch eme

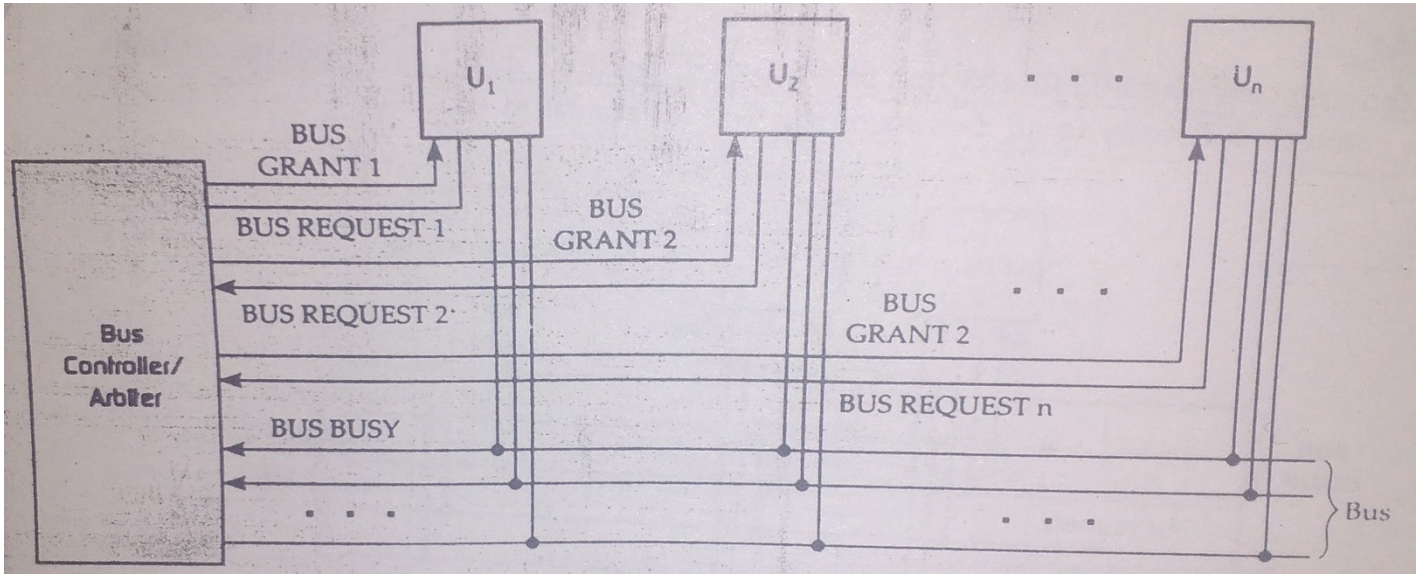
- When the 1st requesting device gets control of the bus and recievies BUS GRANT signal, it blocks further propagation of signals, activates BUS BUSY and begins to use bus.
- When a non requesting device receives BUS GRANT signal, it forwards the signal to next device.
- Thus if two devices simultaneously request bus access, the device that is closer to the bus controller receives BUS GRANT and receives the bus control.
- Means the devices that are closed to the bus controller are of higher priority than those of the other devices.

Polling



- This method replaces the BUS GRANT line of daisy chain method with a set of poll count lines that are connected directly to all devices on the bus.
- Devices request access to the bus via a common BUS REQUEST line.
- Bus controller generates a sequence of numbers on the poll count lines.
- Each device compares these numbers as their device address already assigned to them.
- When a requesting device finds that its address matches the numbers on the poll-count lines, the device activates BUS BUSY.
- The bus controller responds by terminating the polling process and the device connects to the bus.

Independent Requesting



- There are separate BUS REQUEST and BUS GRANT lines for every device that are sharing bus.
- In this, bus controller has the capability of immediate identifying all the requesting devices.
- Bus controller responds rapidly to the request by determining the highest priority device that has sent the bus request.
- This priority is programmable and is predetermined.