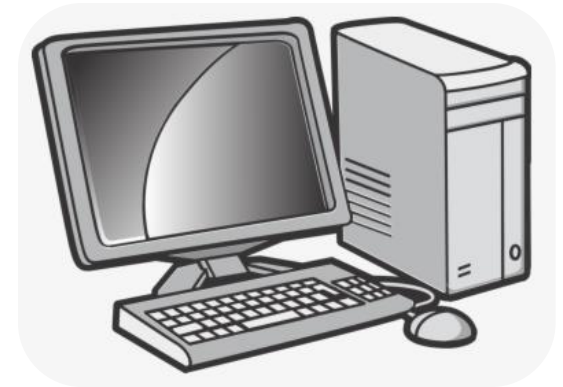
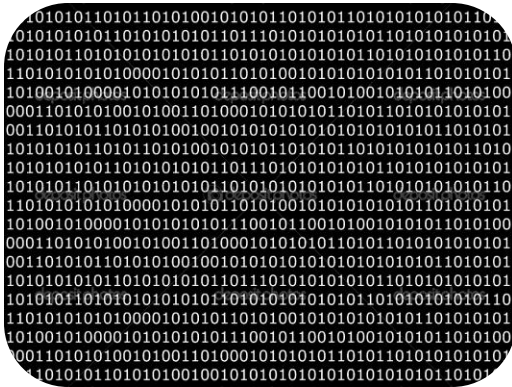


### Unit-8

# Input-Output Organization



# Topics to be covered

- Input-Output Interface,
- Asynchronous Data Transfer,
- Modes Of Transfer,
- Priority Interrupt,
- Input-Output Processor (IOP),
- CPU IOP Communication,
- Serial communication.

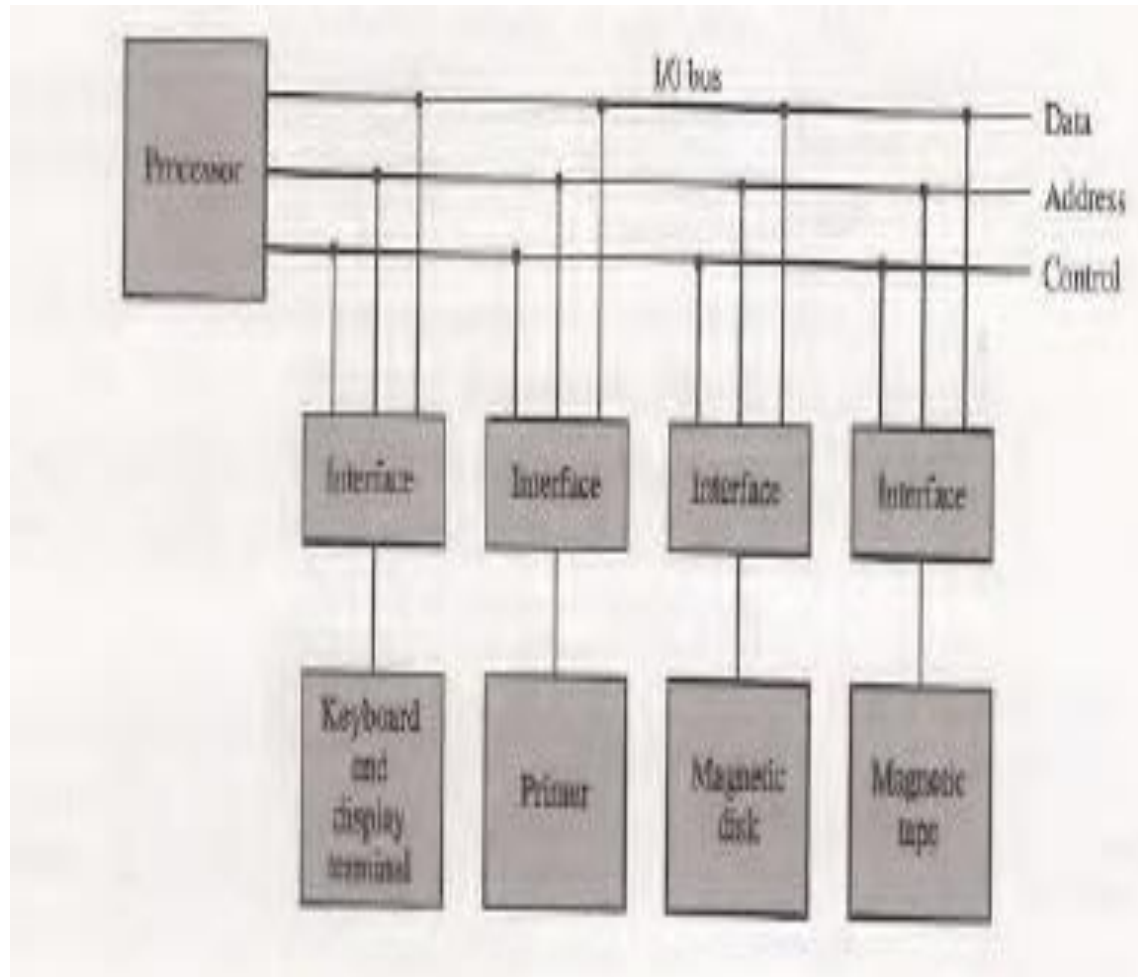
# Input-Output Interface

Input-output interface provides a method for transferring information between internal storage and external I/O devices. Peripherals connected to a computer need special communication links for interfacing them with the central processing unit. The purpose of the communication link is to resolve the differences that exist between the central computer and each peripheral. The major differences are:

1. Peripherals are electromechanical and electromagnetic devices and their manner of operation is different from the operation of the CPU and memory, which are electronic devices. Therefore, a conversion of signal values may be required.
2. The data transfer rate of peripherals is usually slower than the transfer rate of the CPU, and consequently, a synchronization mechanism may be needed.
3. Data codes and formats in peripherals differ from the word format in the CPU and memory.
4. The operating modes of peripherals are different from each other and each must be controlled so as not to disturb the operation of other peripherals connected to the CPU.

# Input-Output Interface

To resolve these differences, computer systems include special hardware components between the CPU and peripherals to supervise and synchronize all input and output transfers. These components are called interface units because they interface between the processor bus and the peripheral device.



# Input-Output Interface

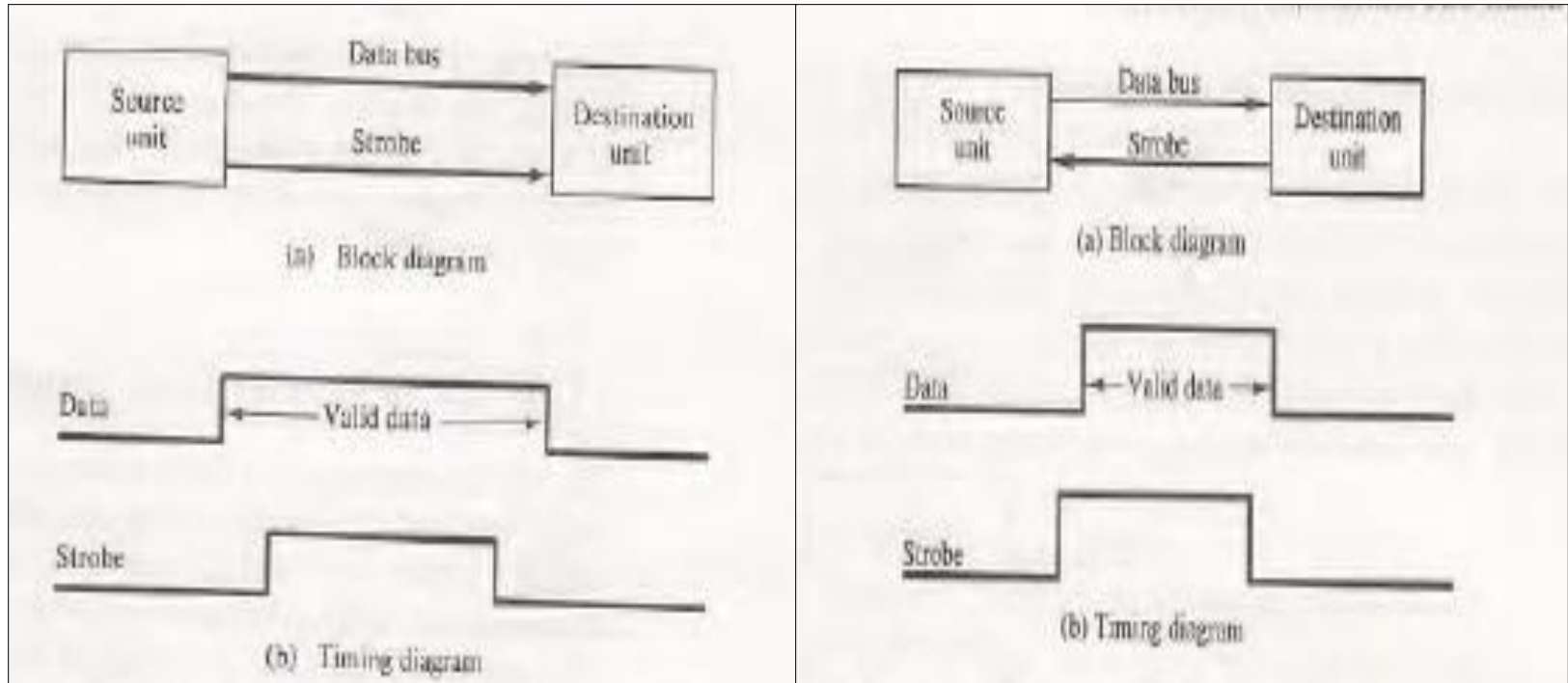
In addition to communicating with I/O, the processor must communicate with the memory unit. There are three ways that computer buses can be used to communicate with memory and I/O:

1. Use two separate buses, one for memory and the other for I/O.
2. Use one common bus for both memory and I/O but have separate control lines for each.
3. Use one common bus for memory and I/O with common control lines.

# Asynchronous Data Transfer

- Asynchronous data transfer between two independent units requires that control signals be transmitted between the communicating units to indicate the time at which data is being transmitted.
- One way of achieving this is strobe by means of a strobe pulse supplied by one of the units to indicate to the other unit when the transfer has to occur.
- Another method commonly used is to accompany each data item being transferred with a control signal that indicates the presence of data in the bus.
- The unit receiving the data item responds with another control signal to acknowledge receipt of the data. This type of agreement between two independent units is referred to as handshaking .

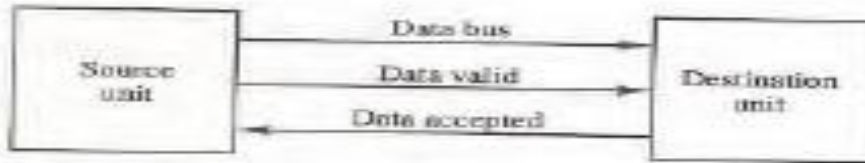
# Asynchronous Data Transfer



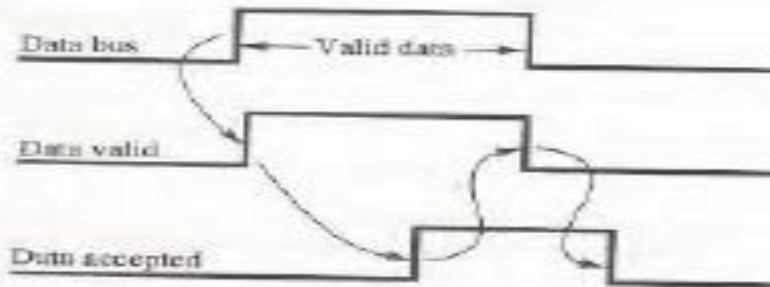
Source- initiated strobe for data transfer.

Destination- initiated strobe for data transfer.

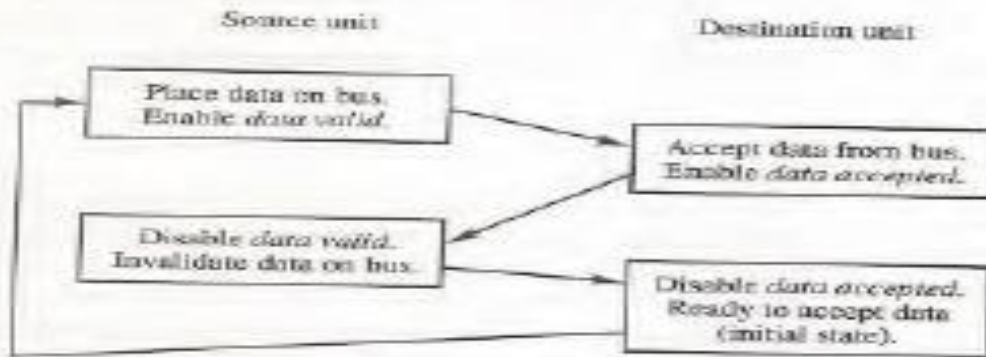
# Asynchronous Data Transfer



(a) Block diagram



(b) Timing diagram

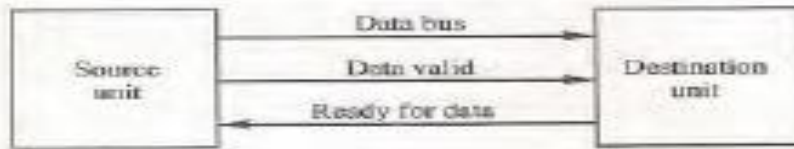


(c) Sequence of events

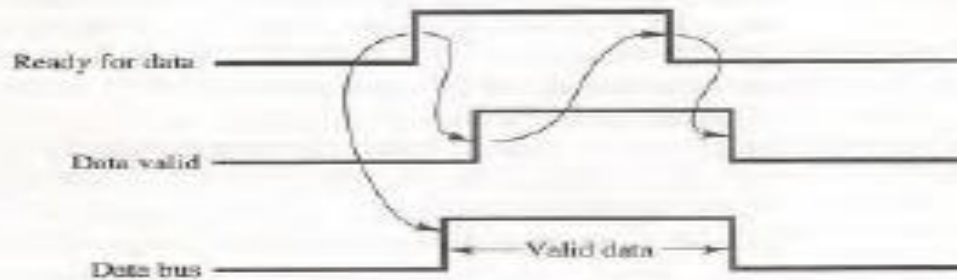
Source-  
initiated  
transfer  
using  
handshaking



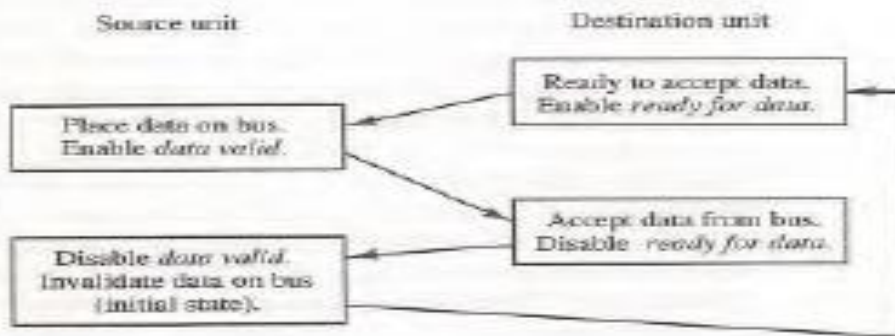
# Asynchronous Data Transfer



(a) Block diagram



(b) Timing diagram



(c) Sequence of events

Destination-initiated transfer using handshaking

# Modes Of Transfer

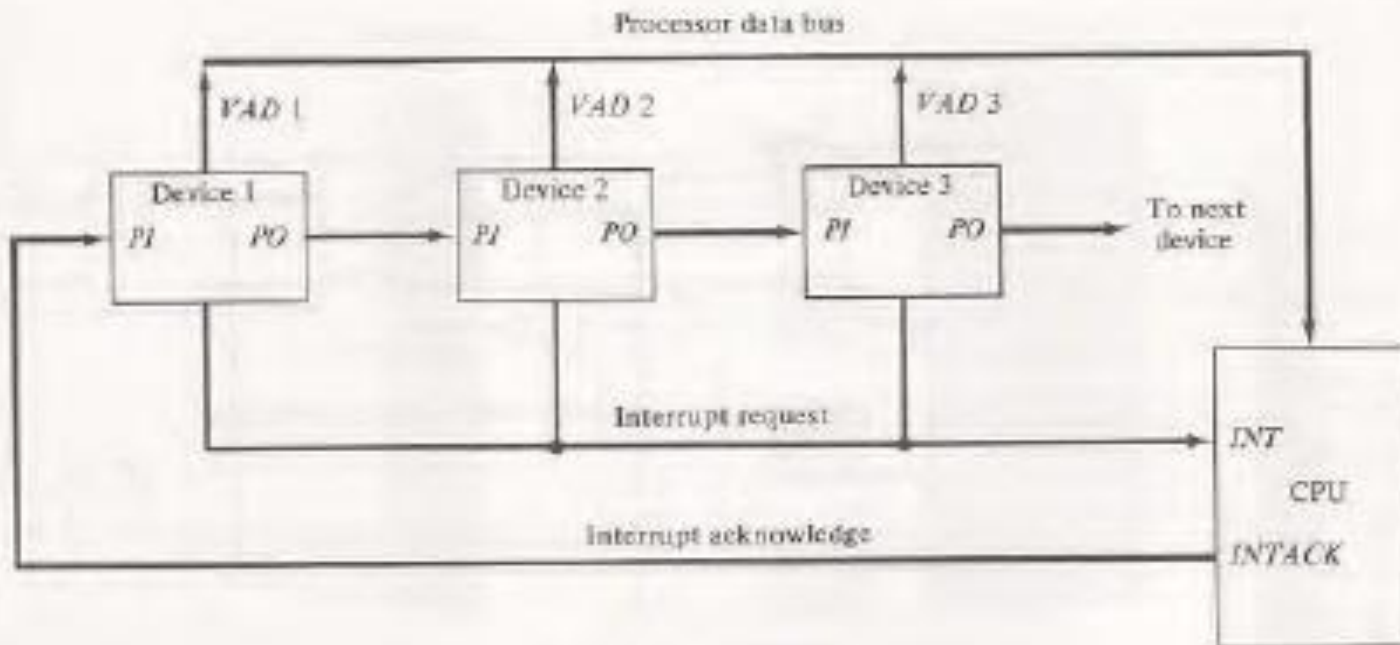
- Binary information received from an external device is usually stored in memory for later processing. Information transferred from the central computer into an external device originates in the memory unit.
- The CPU merely executes the I/O instructions and may accept the data temporarily, but the ultimate source or destination is the memory unit. Data transfer between the central computer and I/O devices may be handled in a variety of modes. Some modes use the CPU as an intermediate path; others transfer the data directly to and from the memory unit.
- Data transfer to and from peripherals may be handled in one of three possible modes:
  1. Programmed I/O
  2. Interrupt-initiated I/O
  3. Direct memory access (DMA).

# Priority Interrupt

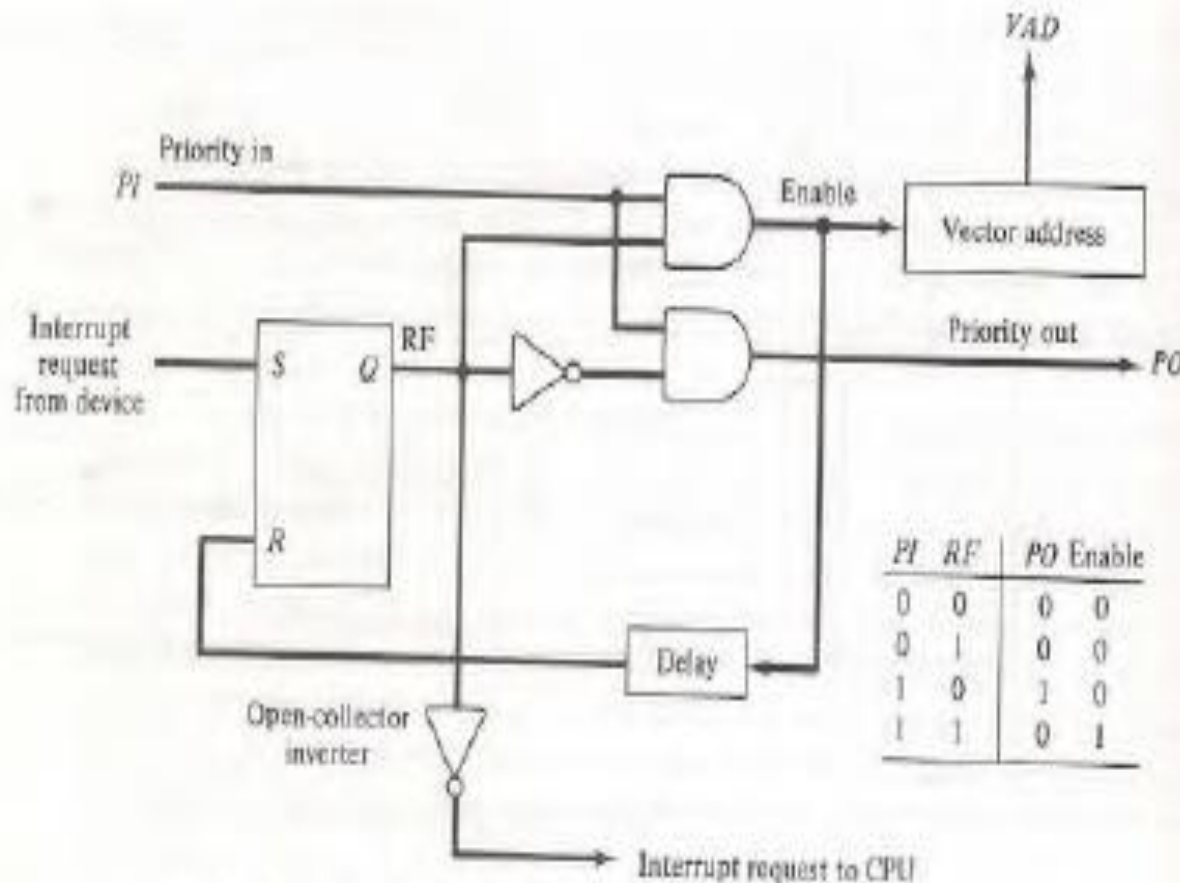
- A priority interrupt is a system that establishes a priority over the various sources to determine which condition is to be serviced first when two or more requests arrive simultaneously.
- The system may also determine which conditions are permitted to interrupt the computer while another interrupt is being serviced.
- Higher-priority interrupt levels are assigned to requests which, if delayed or interrupted, could have serious consequences.
- Devices with high speed transfers such as magnetic disks are given high priority, and slow devices such as keyboards receive low priority.
- When two devices interrupt the computer at the same time, the computer services the device, with the higher priority first.

# Priority Interrupt

- The daisy-chaining method of establishing priority consists of a serial connection of all devices that request an interrupt. The device with the highest priority is placed in the first position, followed by lower-priority devices up to the device with the lowest priority, which is placed last in the chain..



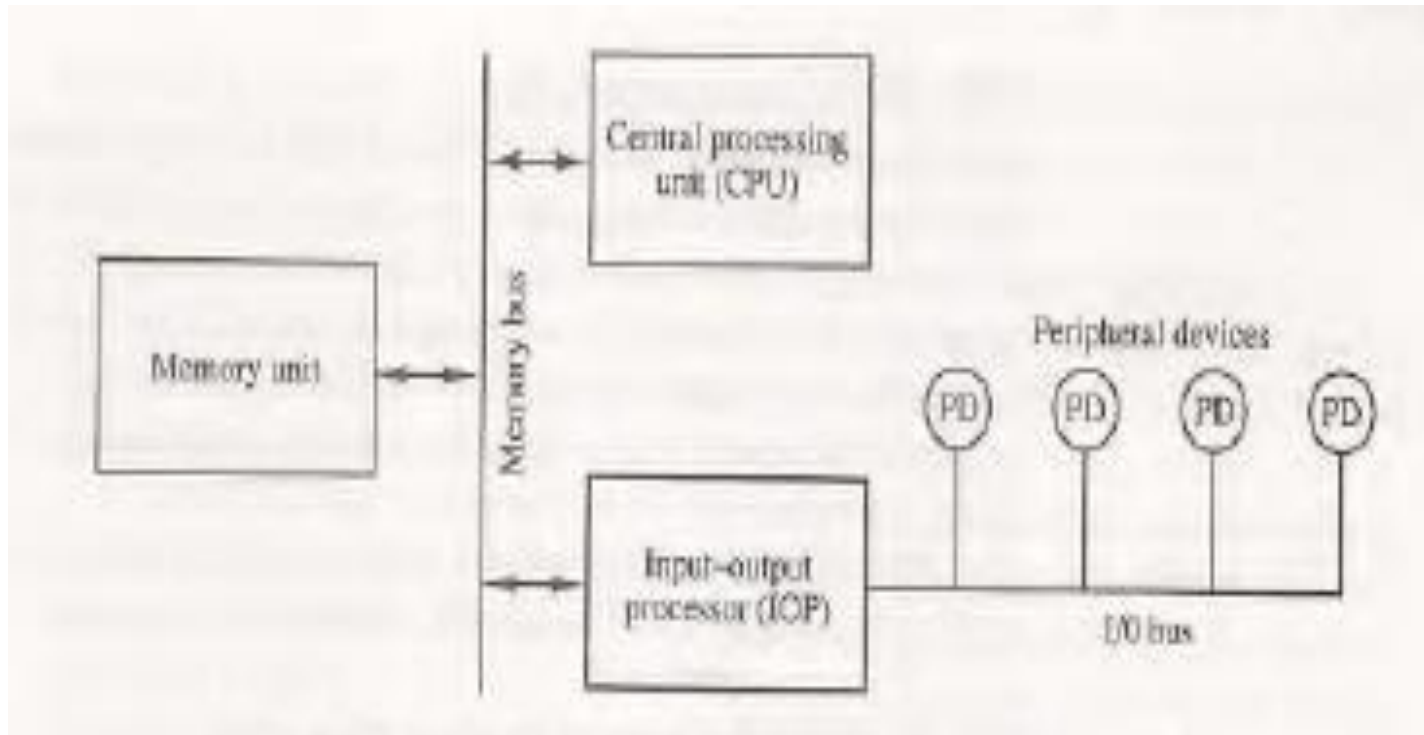
# Priority Interrupt



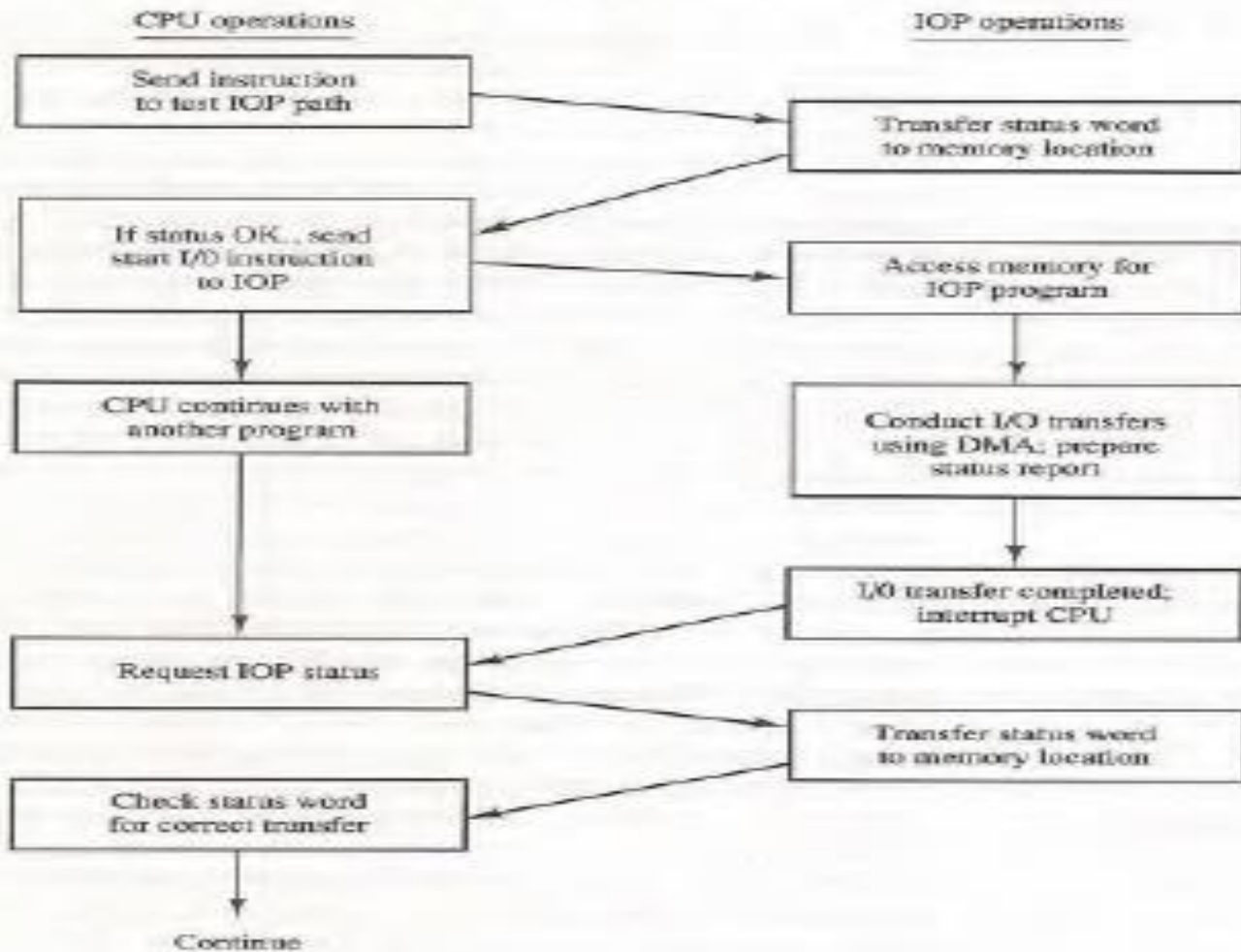
One stage of the daisy-chain priority arrangement

# Input-Output Processor (IOP),

- Block diagram of a computer with I/O processor.



# CPU IOP Communication



# Serial communication.

- A data communication processor is an I/O processor that distributes and collects data from many remote terminals connected through telephone and other communication lines. It is a specialized I/O processor designed to communicate directly with data communication networks.
- A communication network may consist of any of a wide variety of devices, such as printers, interactive display devices, digital sensors, or a remote computing facility.
- With the use of a data communication processor, the computer can service fragments of each network demand in an interspersed manner and thus have the apparent behavior of serving many users at once. In this way the computer is able to operate efficiently in a time-sharing environment.
- The most striking difference between an I/O processor and a data communication processor is in the way the processor communicates with the I/O devices. An I/O processor communicates with the peripherals through a common I/O bus that is comprised of many data and control lines.



# Serial communication.

- All peripherals share the common bus and use it to transfer information to and from the I/O processor. A data communication processor communicates with each terminal through a single pair of wires. Both data and control information are transferred in a serial fashion with the result that the transfer rate is much slower.
- The task of the data communication processor is to transmit and collect digital information to and from each terminal, determine if the information is data or control and respond to all requests according to predetermined established procedures. The processor, obviously, must also communicate with the CPU and memory in the same manner as any I/O processor.
- The communication lines, modems, and other equipment used in the
- transmission of information between two or more stations is called a data link. The orderly transfer of information in a data link is accomplished by means of a protocol .
- Protocols are divided into two major categories according to the message-framing technique used. These are character-oriented protocol and bit-oriented protocol.

# Serial communication.

- The character-oriented protocol is based on the binary code of a character set. The code most commonly used is ASCII (American Standard Code for Information Interchange). The character-oriented protocol was originally developed to communicate with keyboard, printer, and display devices that use alphanumeric characters exclusively. As the data communication field expanded, it became necessary to transmit binary information which is not ASCII text.
- The bit-oriented protocol does not use characters in its control field and is independent of any particular code. It allows the transmission of serial bit stream of any length without the implication of character boundaries

# References

- **Images , descriptive Tables , from Computer System Architecture, Morris Mano, 3<sup>rd</sup> edition Prentice Hall**
- **Note: These pdf/ppt notes are for purpose of teaching aids to classroom/online sessions study, and in no case imply for GTU syllabus or GTU exam. For GTU syllabus or exam related preparation, one may, however will need to attend college/online lectures and refer books given by GTU in their official syllabus.**