Module 4
INPUT-OUTPUT ORGANIZATION

Input-Output organization: Peripheral devices, I/O interface, Accessing I/O devices-Modes of transfer, interrupts-daisy chaining, priority interrupt, DMA

PHERIPHERAL DEVICES: -

- The ***input-output subsystem*** of a computer, referred to as I/O, provides an efficient mode of ***communication between the central system and the outside environment.***
- Input or output devices attached to the computer are also called ***peripherals***.
- The most common peripherals are keyboards***, display units (monitors)***, and ***printers***. Peripherals that provide auxiliary storage for the system are ***magnetic disks and tapes***.

INPUT-OUTPUT INTERFACE: -

Why do we need interface between the peripherals and internal devices?

- ***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 CPU.
    - 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/electromagnetic devices and CPU/memory 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. Therefore, 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.

- 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***.
- **<u>I/O bus and Interface modules:-</u>**
- A typical communication link between the processor and several peripherals is shown in figure.
- The I/O bus consists of data lines, address lines and control lines.

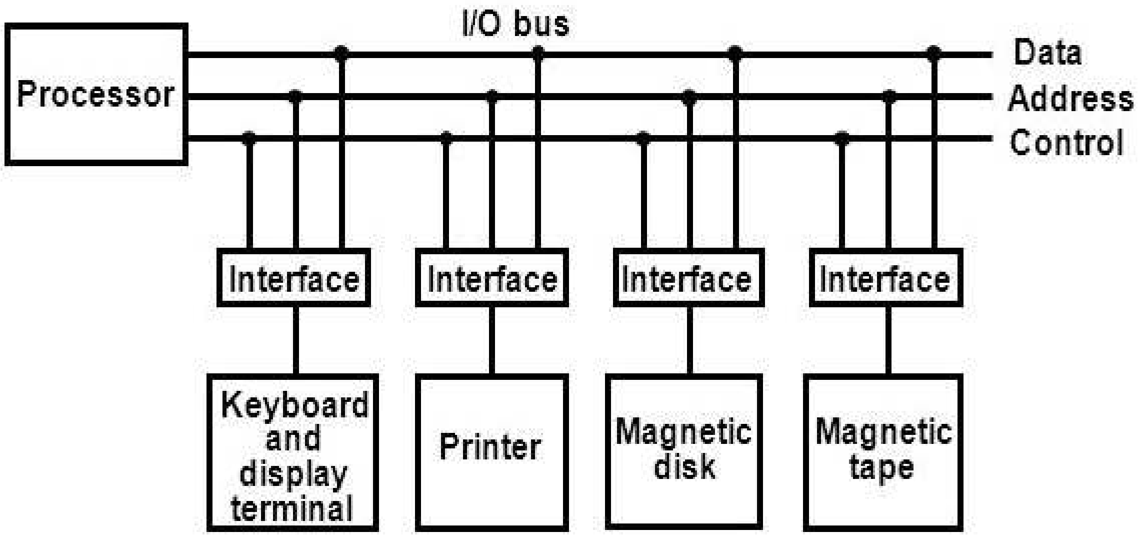


Fig: Connection of I/O bus to input-output devices

- Each peripheral device has associated with it ***an interface unit.***
- Each interface
  - ***Decodes the address and control received from the I/O bus.***
  - ***Interprets address and control information for the peripheral.***
  - ***Provides signals for the peripheral controller.***

- ***Synchronizes the data flow and supervises the transfer between peripheral and processor.***

WORKING OF AN INTERFACE OR ACCESSING I/O DEVICE:-

- The ***I/O bus*** from the processor is attached to all peripheral interfaces.
- To communicate with a particular device, the ***processor places a device address on the address lines*** and ***function code in the control lines.***
    - Each interface attached to the I/O bus contains an ***address decoder that monitors the address lines***.
    - When the interface detects its own address, ***it activates the path between the bus lines and the device*** that it controls.
    - All peripherals whose address does not correspond to the address in the bus are disabled by their interface.
- The ***function code*** is referred to as ***IO command*** and there are four types of commands that an interface may receive.
    - ***Control command** - It is issued to activate the peripheral and to inform it what to do.*
    - ***Status command** – It is used to test various status conditions in the interface and the peripheral.*
    - ***Output data –*** It causes the interface to respond by transferring data from the bus into one of its registers (from bus to the peripheral).
    - ***Input data –*** It causes the interface receives an item of data from the peripheral and places it in its buffer register (from the peripheral to the bus).

MODES OF TRANSFER

- Binary information received from an external device is usually stored in memory for later processing.

- ***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)
- ***Programmed I/O***
  - In the programmed I/O method, the I/O device ***does not have direct access to memory***. A transfer from an I/O device to memory requires the execution of several instructions by the CPU.
  - In short, Programmed I/O operations are the ***result of I/O instructions written in the computer program.***
  - Each ***data transfer is initiated by an instruction*** in the program.
  - E.g.: input instructions to transfer data from the device to the CPU, store instructions to transfer data from the CPU to memory, control instructions to verify the data availability and to count number of words transferred.
- ***Interrupt-initiated I/O***
  - In the ***programmed I/O method, the CPU stays in a program loop*** until the I/O unit indicates that it is ready for data transfer. This is a ***time-consuming process*** since it keeps the processor busy.

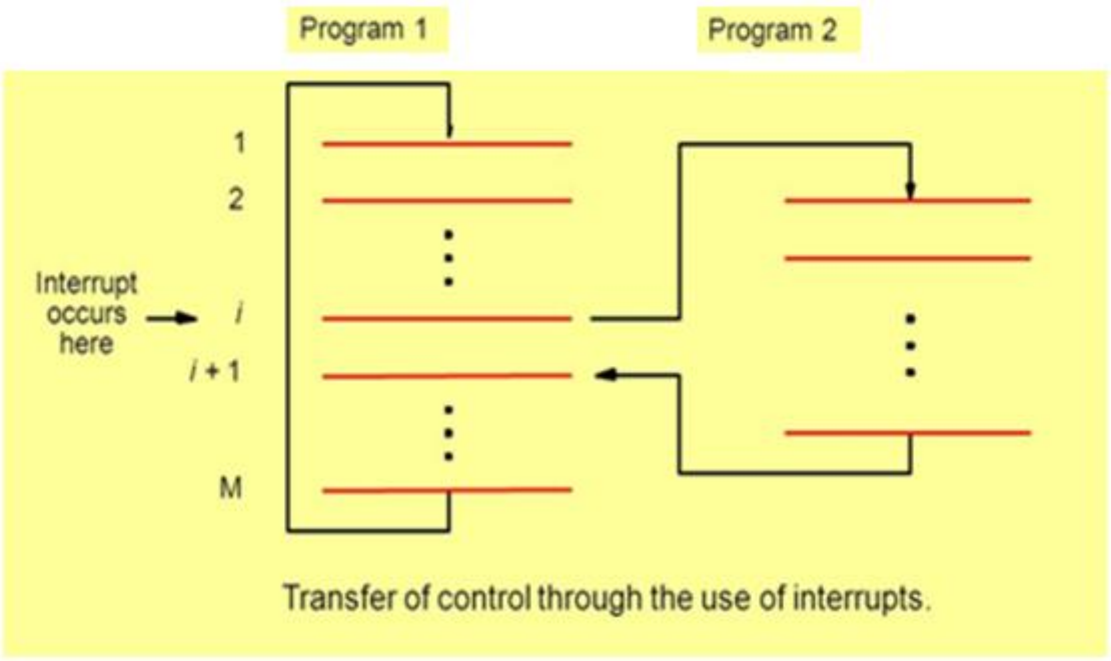
  - It can be avoided by using an ***interrupt facility*** and ***special commands to inform the interface to issue an interrupt request*** signal when the data are available from the device.
  - In the ***meantime the CPU can execute another program***.
  - **Method:** The ***interface keeps monitoring*** the device. When the interface determines that the ***device is ready*** for data transfer, ***it generates an interrupt request*** to the computer. Upon detecting the external interrupt signal, the ***CPU stops the task it is processing, branches to a service program*** to process the ***I/O transfer***, and then returns to the task it was originally performing.

- ***Direct memory access (DMA)***
  - In direct memory access (DMA), the interface transfers data into and out of the memory unit through the ***memory bus.***
  - The CPU ***initiates the transfer by giving the starting address and the number of words needed to be transferred to the interface*** and then proceeds to execute other tasks.
  - Thus we can allow the peripherals directly communicate with each other using the ***memory buses***, ***removing the intervention of the CPU.*** This type of data transfer technique is known as ***DMA or direct memory access.***
  - During DMA the CPU is idle or any other task and it has no control over the memory buses.
  - The DMA controller takes over the buses to manage the transfer directly between the I/O devices and the memory unit.

INTERRUPTS:-

- Interrupt refers to the transfer of program control from a currently running program to another service program as a result of an external or internal generated request.

- Control returns to the original program after the service program is executed.
- There are many situations where other tasks can be performed while waiting for an I/O device to become ready. It can do so by sending a ***hardware signal called an interrupt*** to the CPU.
- Bus control lines called interrupt-request line is used for this purpose.
- Consider the below example:
  - Consider a situation in which CPU is executing program1 and an interrupt request for program2 arrives during the execution ***of instruction i as in figure.***



Transfer of control through the use of interrupts.

  - The processor first completes execution of instruction i , then it loads the PC(program counter) with the address of the first instruction of the ISR(program2 here).
  - After the execution of ISR, the processor has to come back to the instruction i+1.
  - Therefore when an interrupt occurs, the current content of the PC, which points to instruction i+1 must be put in stack.

- Return from interrupt instruction at the end of ISR reloads the PC from the stack and resume at instruction i+1.
- As part of handling interrupts, the processor must inform the device that its request has been recognized and it is accomplished by an ***interrupt-acknowledge signal***.

DAISY CHAINING: -

- When a device is ready to communicate with the CPU, it generates an interrupt signal.
- A number of input-output devices are attached to the computer and each device is able to generate an interrupt request.
- The main job of the **interrupt system is to identify the source of the interrupt.**
- There is also a possibility that several devices will request simultaneously for CPU communication. Then, the interrupt system **has to decide which device is to be serviced first.**

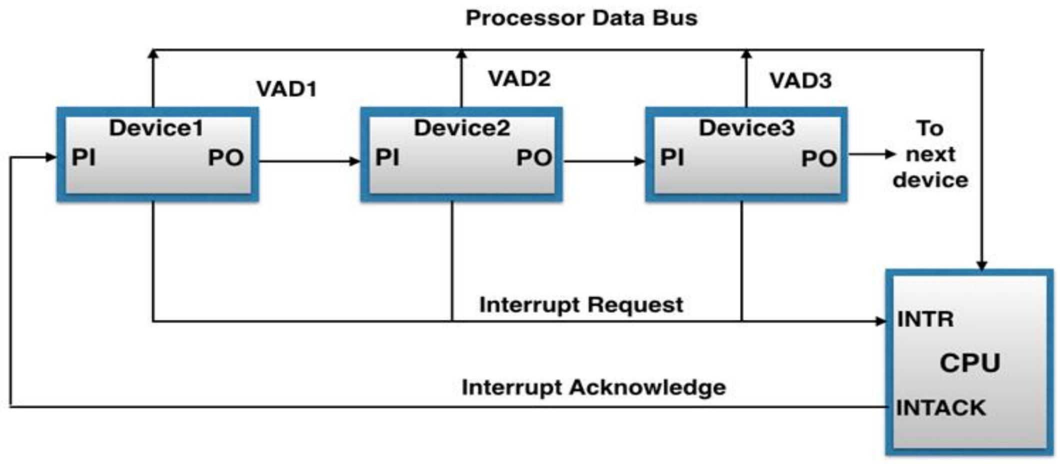
Priority Interrupt

- A priority interrupt is a system which decides the priority at which various devices, which generates the interrupt signal at the same time, will be serviced by the CPU.
- Generally, devices with high speed transfer such as *magnetic disks* are given high priority and slow devices such as *keyboards* are given low priority.
- When two or more devices interrupt the computer simultaneously, **the computer services the device with the higher priority first.**

Daisy Chaining Priority: -

- The daisy-chaining method involves connecting all the devices that can request an interrupt in a serial manner.
- This configuration is governed by the priority of the devices.

- The device with the highest priority is placed first followed by the second highest priority device and so on.



WORKING:

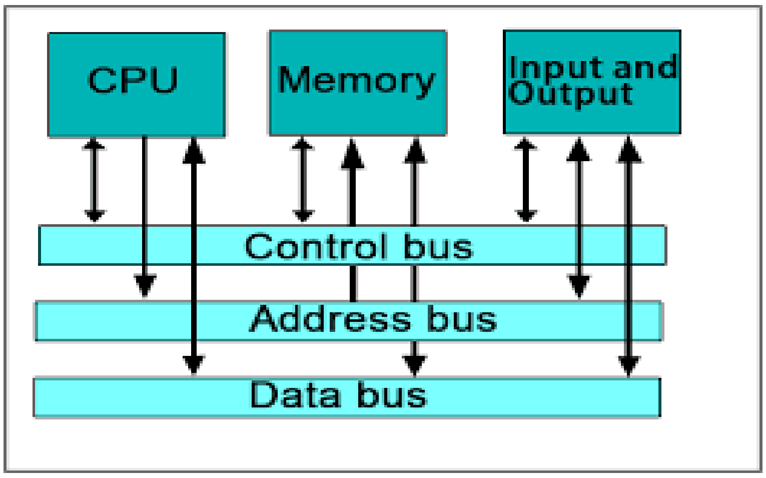
- There is an **interrupt request line** which is common to all the devices and goes into the CPU.
- When ***no interrupts are pending, the line is in HIGH state***. But if ***any of the devices raises an interrupt, it places the interrupt request line in the LOW state.***
- The CPU acknowledges this interrupt request from the line and then enables the **interrupt acknowledge line** in response to the request.
    - This signal (***Interrupt Acknowledge***) is received at the PI (Priority in) input of device 1.
    - If the device has not requested the interrupt, it passes this signal to the next device through its PO (priority out) output. (PI = 1 & PO = 1).
    - However, if the device had requested the interrupt, (PI =1 & PO = 0).
        - The device consumes the acknowledge signal and block its further use by placing 0 at its PO (priority out) output.
        - The device then proceeds to place its interrupt vector address (VAD) into the data bus of CPU.

- The device puts its interrupt request signal in HIGH state to indicate its interrupt has been taken care of.

NOTE: VAD is the address of the service routine which services that device.

BUSES: -

- Buses are actually ***communication path for the transfer of data*** between the devices.
- The processor, memory and I/O devices can be interconnected by means of buses.
- A ***bus protocol*** is the set of rules that govern the behaviour of various devices connected to the bus, when to place information on the bus, assert control signals and so on.
- The bus lines used for transferring data may be grouped into three types:



- **Data bus:** A data bus is a set of wires, ***that provides transportation for data.*** i.e., a group of electrical wires used to send information (data) between two or more components (memory or I/O and CPU). It is bidirectional.
- **Address bus:** An address bus is a group of wires or lines that are ***used to transfer the addresses of Memory or I/O devices***. It is ***unidirectional.***

- **Control bus:** A control bus is a group of wires ***used to process data, that is what to do*** with the selected memory location. Some control signals are ***Read, Write*** etc. This is a dedicated bus, because ***all timing signals*** are generated according to control signal. Normally Control bus is unidirectional.

Difference between a Synchronous and Asynchronous Bus:

Synchronous bus	***Asynchronous bus***
Transmitters and receivers are synchronized of clock.	Transmitters and receivers are not synchronized by clock.
Data bits are transmitted with synchronization of clock.	Bits of data are transmitted at constant rate.
Character is received at constant Rate.	Character may arrive at any rate at receiver.
Data transfer takes place in block.	Data transfer is character oriented.
Start and stop bits are required to establish communication of each character.	Start and stop bits are required to establish communication of each character.
Used in high – speed transmission.	Used in low – speed transmission.

DIRECT MEMORY ACCESS (DMA) :-

- ***DMA*** stands for "***Direct Memory Access***" and is a method of transferring data from the ***computer's RAM*** to another part of the computer ***without processing it using the CPU.***
- In DMA ***the interface transfer the data*** into and out of the memory unit ***through the memory bus.***
- ***Removing the CPU from the path*** and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This transfer technique is called ***Direct Memory Access (DMA).***
- During the DMA transfer, ***the CPU is idle and has no control of the memory buses.***

- ***A DMA Controller takes over the buses*** to manage the transfer directly between the I/O device and memory.
- The two control signals in the CPU that facilitates the DMA transfer.

1. ***The Bus Request (BR***) input is used by the DMA controller to request the CPU. When this input is active, the CPU terminates the execution of the current instruction and places the address bus, data bus and read write lines into a high Impedance state.)
2. ***The Bus Grant (BG***) The CPU activates the Bus Grant (BG) output to inform the external DMA that the Bus Request (BR) can now take control of the buses to conduct memory transfer without processor.

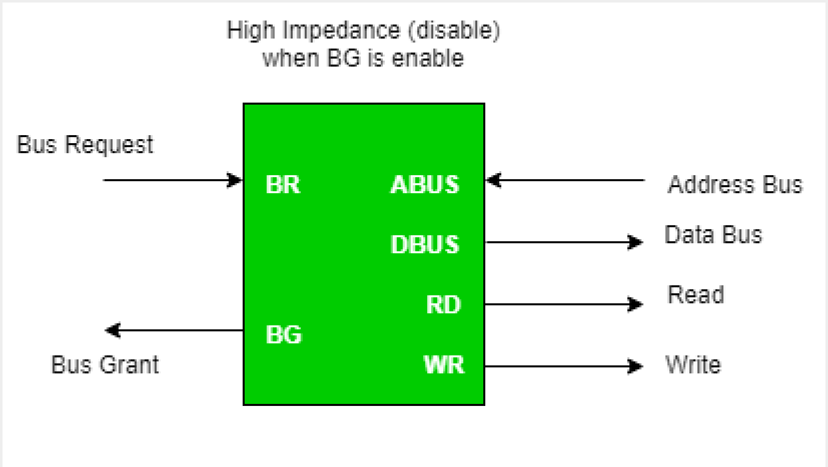


Figure - CPU Bus Signals for DMA Transfer

- When the DMA terminates the transfer, it disables the Bus Request (BR) line. The CPU disables the Bus Grant (BG), takes control of the buses and return to its normal operation.
- DMA the transfer can be made in several ways that are:

i. ***DMA Burst :*** In DMA Burst transfer, a block sequence consisting of a number of memory words is transferred in continuous burst while the DMA controller is master of the memory buses.

ii. ***Cycle Stealing :*** Cycle stealing allows the DMA controller to transfer one data word at a time, after which it must returns control of the buses to the CPU.

- **DMA Controller**
- The DMA controller needs the usual circuits of an interface to communicate with the CPU and I/O device.
- The DMA controller has three registers:
  i. ***Address Register :-*** Address Register contains an address to specify the desired location in memory.
  ii. ***Word Count Register :-*** WC holds the number of words to be transferred. The register is incremented/decremented by one after each word transfer and internally tested for zero.
  iii. ***Control Register :-*** Control Register specifies the mode of transfer.
- ***DMA Controller communicates with the CPU via the data bus and control lines***.
- ***DMA Transfer***
  - The CPU communicates with the DMA through the address and data buses as with any interface unit. Once the DMA receives the start control command, it can transfer between the peripheral and the memory.

.. END ....