Unit: 4
Input-Output Organization
Lecture-3
Modes of Data 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 the I/O devices may be handled in a variety of modes. Some modes use the CPU as an intermediate path, others transfer data directly to and from the memory unit.

Data transfer to and from peripherals may be handled in one of three possible modes:

- **Programmed I/O**
- **Interrupt-initiated I/O**
- **Direct Memory Access (DMA)**

Programmed I/O: It is due to the result of the I/O instructions that are written in the computer program. Each data item transfer is initiated by an instruction in the program. Usually the transfer is from a CPU register and memory.

- In this case, it requires constant monitoring by the CPU of the peripheral devices.

The transfer of data requires three instructions:

1. Read the status register.
2. Check the status of the flag bit and branch to step 1 if not set or to step 3 if set.
3. Read the data register.

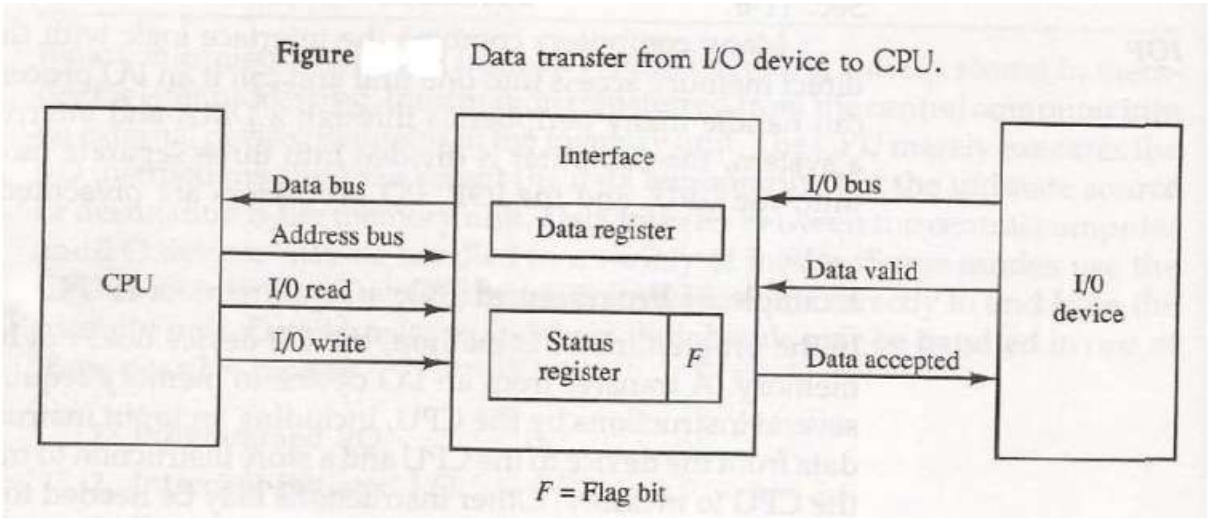

Figure Data transfer from I/O device to CPU.

Drawback of the Programmed I/O: The main drawback of the Program Initiated I/O was that the CPU has to monitor the units all the times when the program is executing. Thus 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 and the CPU time is wasted a lot in keeping an eye to the execution of the program.

To remove this problem an Interrupt facility and special commands are used.

Interrupt Initiated I/O:

In this method, an interrupt command is used to inform the device about the start and end of transfer. In the meantime the CPU executes other program. When the interface determines that the device is ready for data transfer it generates an Interrupt Request and sends it to the computer.

When the CPU receives such a signal, it temporarily stops the execution of the program and branches to a service program to process the I/O transfer and after completing it returns back to task, what it was originally performing.

The CPU responds to the interrupt signal by storing the return address from the program counter into a memory stack and then control branches to a service routine that processes the required I/O transfer.

The way that the processor chooses the branch address of the service routine varies from one unit to another. In principle, there are two methods for accomplishing this:

- **Vectored interrupt:** In a vectored interrupt, the source that interrupts supplies the branch information to the computer.
- **Non vectored interrupt:** In a non-vectored interrupt, the branch address is assigned to a fixed location in memory.

Direct Memory Access:

The data transfer between a fast storage media such as magnetic disk and memory unit is limited by the speed of the CPU.

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 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.