Mode of Data Transfer in Computer Architecture

Mode of data transfer in computer architecture plays an important role to transfer information between internal storage and external I/O devices.

There are three mode of data transfer in computer architecture. These mode of transfer are –

- Programmed I/O
- Interrupt- initiated I/O
- Direct memory access(DMA)

Today in this post we will cover all three mode of data transfer in computer architecture one by one with suitable diagram.

Mode of Transfer

- The binary information that is received from an external device is usually stored in the memory unit.
- The information that is transferred from the CPU to the external device is originated from the memory unit.
- CPU merely processes the information but the source and target is always the memory unit.
- Data transfer between CPU and the **I/O devices** may be done in different modes.

Data transfer to and from the peripherals may be done in any of the three possible ways

- Programmed I/O
- Interrupt- initiated I/O
- Direct memory access(DMA)

Now let’s discuss each mode one by one.

Programmed I/O

It is due to the result of the I/O instructions that are written in the computer program.

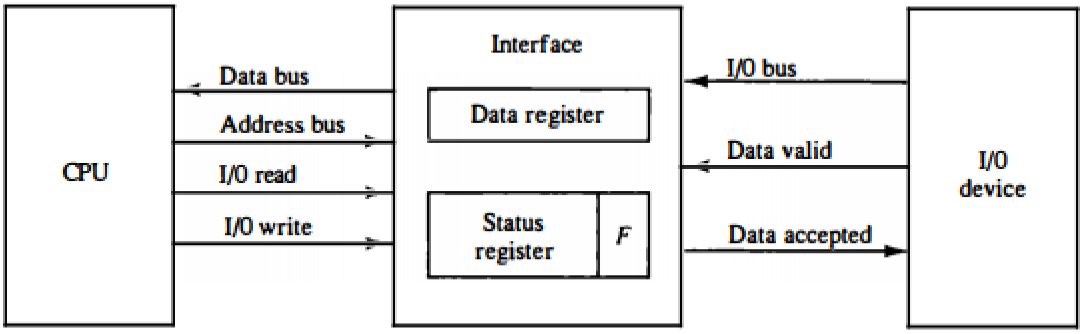
Each data item transfer is initiated by an instruction in the program. Usually the transfer is from a **CPU Register** and memory.

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

Example of Programmed I/O

- In Programmed Input Output mode of data transfer the I/O device does not have direct access to the memory unit.
- A transfer from I/O device to memory requires the execution of several instructions by the CPU, including an input instruction to transfer the data from device to the CPU and store instruction to transfer the data from CPU to memory.
- In programmed I/O, the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer.
- This is a time consuming process since it needlessly keeps the CPU busy. This situation can be avoided by using an interrupt facility.

Figure 10 Data transfer from I/O device to CPU.



F = Flag bit

Interrupt- initiated I/O

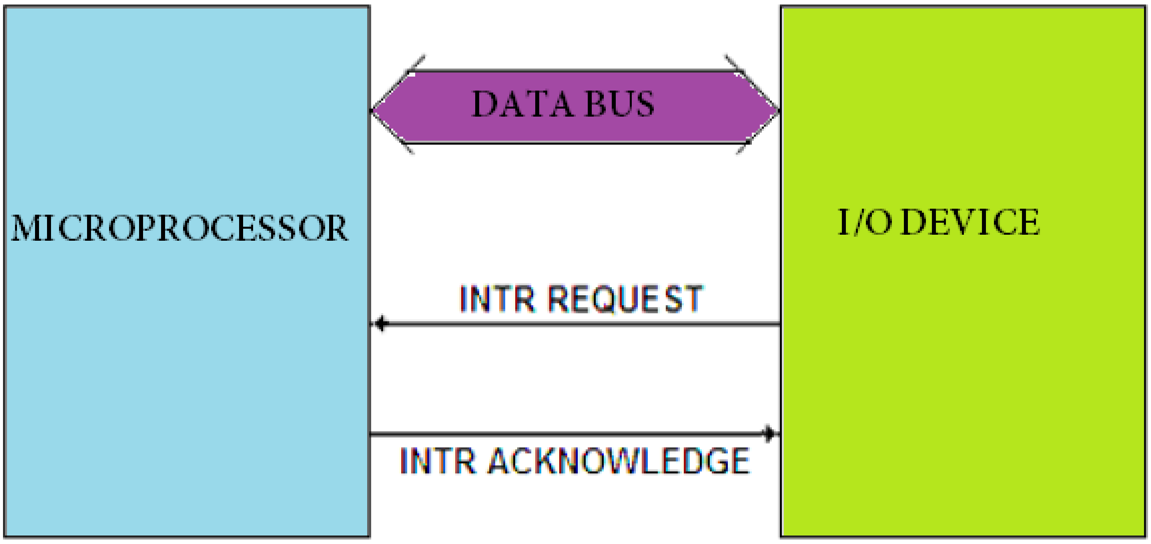
This mode uses an interrupt facility and special commands to inform the interface to issue the interrupt command when data becomes available and interface is ready for the data transfer. In the meantime CPU keeps on executing other tasks and need not check for the flag. When the flag is set, the interface is informed and an interrupt is initiated. This interrupt causes the CPU to deviate from what it is doing to respond to the I/O transfer. The CPU responds to the signal by storing the return address from the program counter (PC) into the memory stack and then branches to service that processes the I/O request. After the transfer is complete, CPU returns to the previous task it was executing. The branch address of the service can be chosen in two ways known as vectored and non-vectored interrupt. In vectored interrupt, the source that interrupts, supplies the branch information to the CPU while in case of non-vectored interrupt the branch address is assigned to a fixed location in memory.

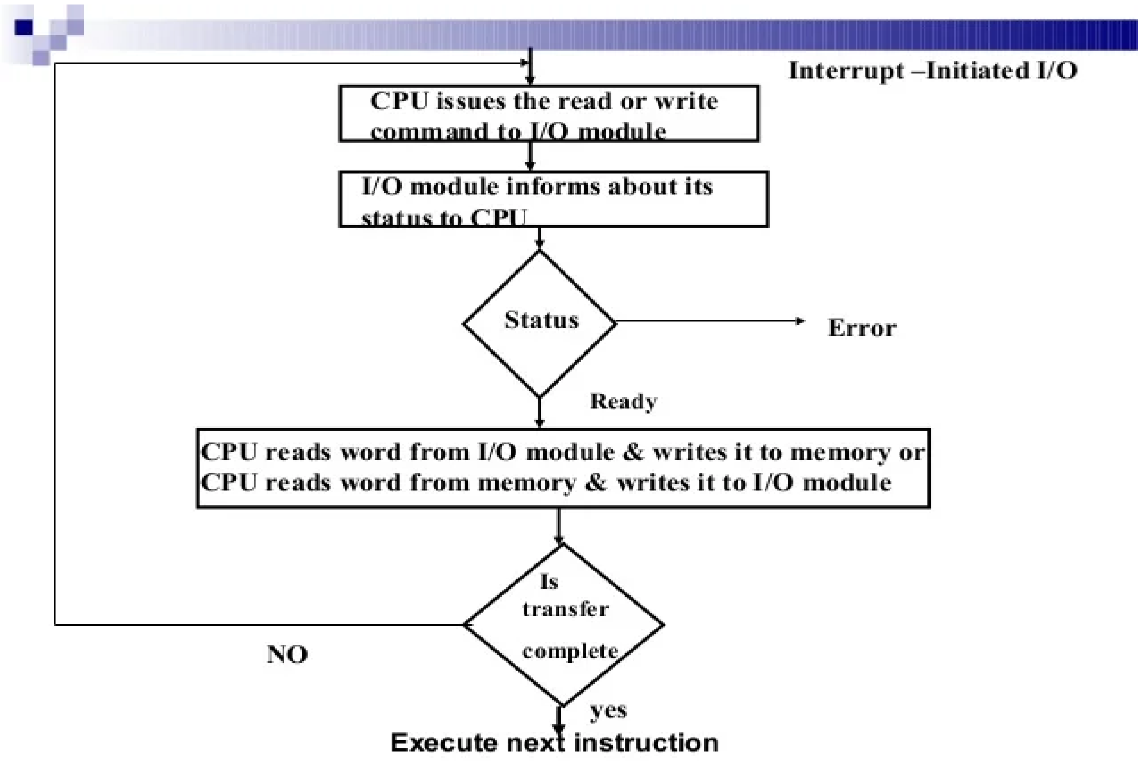
Drawbacks of Programmed Input Output and Interrupt Driven Input-Output

Both the methods programmed I/O and Interrupt-driven I/O require the active intervention of the processor to transfer data between memory and the I/O module, and any data transfer must transverse a path through the processor. Operating System and it’s Functions

Thus both these forms of I/O suffer from two inherent drawbacks.

- The I/O transfer rate is limited by the speed with which the processor can test and service a device.
- The processor is tied up in managing an I/O transfer; a number of instructions must be executed for each I/O transfer.




Interrupt –Initiated I/O
CPU issues the read or write command to I/O module
I/O module informs about its status to CPU
Status
Error
Ready
CPU reads word from I/O module & writes it to memory or CPU reads word from memory & writes it to I/O module
Is transfer complete
NO
yes
Execute next instruction

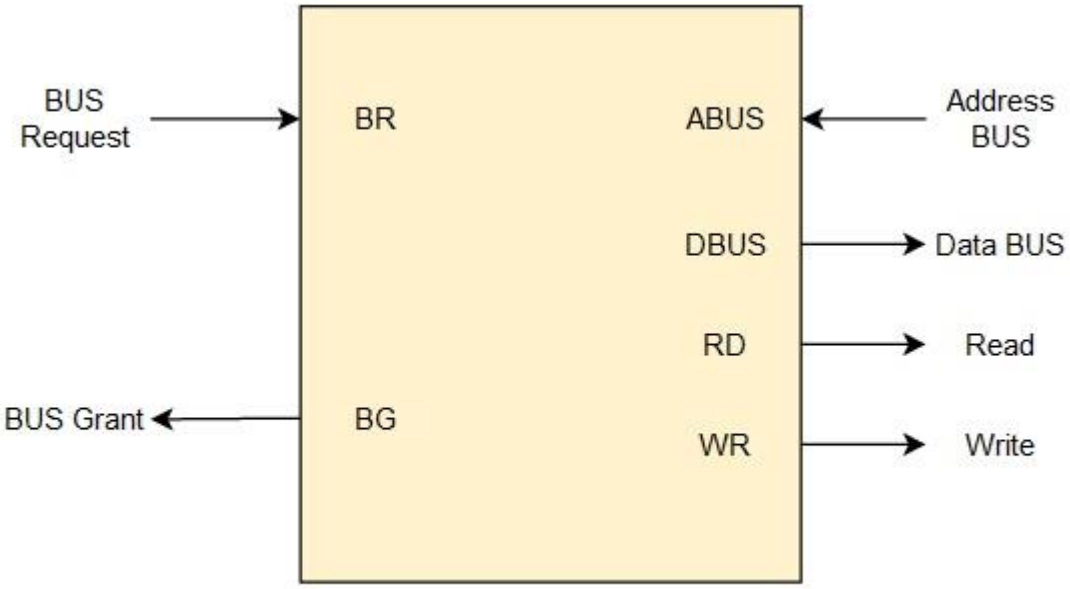
Direct Memory Access

- 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.

Bus Request : It is used by the DMA controller to request the CPU to relinquish the control of the buses.

Bus Grant : It is activated by the CPU to Inform the external DMA controller that the buses are in high impedance state and the requesting DMA can take control of the buses.

Once the **DMA** has taken the control of the buses it transfers the data.



CPU Bus Signal in Direct Memory Access Mode

This transfer can take place in many ways

Conclusion and Summary

In this tutorial we have discussed and explained different mode of data transfer in computer architecture, their advantages and drawbacks along with their suitable diagram.

I hope after reading this tutorial students will be able to answer the questions related to mode of Data Transfer.