Unit 7

**Data Transfer Technique**

**Programmed I/O Data Transfer scheme of 8085 microprocessor:**

Programmed I/O Data Transfer scheme of 8085 microprocessor is a simple parallel data transfer scheme. This method of data transfer is generally used in the simple microprocessor systems. It is obvious that where speed is unimportant. This method uses instructions to get the data into or out of the microprocessor. Programmed I/O Data Transfer scheme of 8085 microprocessor can be work on synchronous or asynchronous mode. The data transfer can be synchronous or asynchronous it completely depends upon the type and the speed of the I/O devices.



**Synchronous Data Transfer Technique :**

Synchronous means “at the same time” .Synchronous type of data transfer can be used when the speed of the I/O devices matches with the speed of the 8085 microprocessor. So for synchronization established between I/O device and microprocessor we need common clock pulse. This common clock pulse synchronizes the microprocessor and the I/O devices. Synchronous type of data transfer scheme because of the matching of the speed, the microprocessor does not have to wait for the availability of the data. The microprocessor immediately sends data for the transfer as soon as the microprocessor issues a signal.

**Asynchronous Data Transfer Technique:**

Asynchronous means “at a regular interval”. The asynchronous data transfer method is used when the speed of the I/O devices is slower than the speed of the microprocessor. Because of the mismatch of the speed, the internal timing of the I/O device is independent from the microprocessor. That is why two units are said to be asynchronous to each other. The asynchronous data transfer is normally implemented using ‘handshaking’ mode. Now question is what is handshaking mode? In the handshaking mode some signals are exchanged between the I/O device and microprocessor before the data transfer takes place.

 

By this handshaking the microprocessor has to check the status to the input/output device. Now if the device is ready for the data transfer or not.

* First step of microprocessor is initiates the I/O device to get ready.
* Then status of the I/O device is continuously checked by the microprocessor.
* This process remain continues until the I/O device becomes ready.
* After that microprocessor sends instructions to transfer the data.
* Flow chart for this mode of data transfer is shown above.
* Now form this bellow figure, the microprocessor sends a ready signal to I/O device. When the device is ready to accept the data, the I/O device sends an ‘ACK’ (Acknowledge) signal to microprocessor. By sending ACK, it indicating that the I/O device has acknowledged the ‘Ready’ signal. Now finally it is ready for the transfer of data.



* Again in bellow figure shows the asynchronous handshaking process to transfer the data from the I/O device to microprocessor. In this case I/O device issues the ready signal to microprocessor indicating that I/O device is ready to send the data to microprocessor. In response to this signal, valid data signal is sent by the microprocessor to I/O device and then the valid data is put on the data bus for the transfer.
* 

**Interrupt Driven I/O Data Transfer Technique :**

We saw that in the[**programmed I/O data transfer method**](http://www.electronicsengineering.nbcafe.in/programmed-i-o-data-transfer-scheme-8085-microprocessor/), microprocessor is busy all the time in checking for the availability of data from the slower I/O devices. And it also busy in checking if I/O device is ready for the data transfer or not. In other words in this data transfer scheme, some of the microprocessor time is wasted in waiting while an I/O device is getting ready. To overcome this problem interrupt driven I/O data transfer introduced.

 The interrupt driven I/O data transfer method is very efficient because no microprocessor time is wasted in waiting for an I/O device to be ready. In this interrupt driven I/O data transfer method the I/O device informs the microprocessor for the data transfer whenever the I/O device is ready. This is achieved by interrupting the microprocessor. As we know that the interrupt is hardware facilities provided on the microprocessor.



 Now come to the working process of interrupt driven I/O data transfer. So the beginning the microprocessor initiates data transfer by requesting the I/O device ‘to get ready’ and then continue executing its original program rather wasting its time by checking the status of I/O device. Whenever the device is ready to accept or supply data, it informs the processor through a control signal. This control signal known as interrupt (INTR) signal. In response to this interrupt signal, the microprocessor sends back an interrupt acknowledge signal to the I/O device. By sending acknowledgement it indicating that it received the request. It then suspends its job after executing the current instruction. It saves the contents and status of program counter to stack and jumps to the subroutine program.

This subroutine program is called Interrupt Service Subroutine (ISS) program. The ISS saves the processor status into stack; and after executing the instruction for the data transfer, it restores the processor status and then returns to main program.

As already discussed, several input/output devices may be connected to microprocessor using Interrupt Driven Data Transfer Scheme. Following interrupt request configuration may arise while interfacing the I/O devices to microprocessor.

1. Single Interrupt system
2. Multi Interrupt System

**Single Interrupt System**

When only one interrupt line is available with the microprocessor and several I/O devices are to be connected, then the method is known as Single Interrupt System.

**Multi Interrupt System**

When the microprocessor has several interrupt terminals and one I/O device is to be connected to each interrupt terminal, then it is known as multi interrupt system. In this scheme, the number of I/O devices to be connected to the interrupt lines should be equal to or less than the number of interrupt terminals. In this way one device is connected to each level of interrupt. So when a device interrupts the microprocessor, it immediately knows which device has interrupted. Such an interrupt scheme is known as vectored interrupt.

**Direct Memory Access (DMA) Data Transfer :**

As we discussed earlier that in programmed I/O or interrupt driven I/O methods of data transfer between the I/O devices and external memory is via the accumulator. Now think for bulk data transfer from I/O devices to memory or vice-versa, these two methods discussed above are time consuming and quite uneconomical even though the speed of I/O devices matches with the speed of microprocessor. Because in those methods the data is first transferred to accumulator and then to concerned device.

To overcome those problem direct memory access data transfer method is introduced. The Direct Memory Access (DMA) data transfer method is used for bulk data transfer from I/O devices to microprocessor or vice-versa. In this method I/O devices are allowed to transfer the data directly to the external memory without being routed through accumulator. For this reason the microprocessor relinquishes the control over the data bus and address bus, so that these can be used for transfer of data between the devices.

**Working principle of direct memory access data transfer**

So now come to working principle of direct memory access data transfer. For the data transfer using DMA process, a request to the microprocessor in form of HOLD signal, by the I/O device is sent. When microprocessor receipt of such request, the microprocessor relinquishes the address and data buses and informs the I/O devices of the situation by sending Acknowledge signal HLDA. The I/O device withdraws the request when the data transfer between the I/O device and external memory is complete.



If we discuss in brief about working principal of DMA controller. Then we should mention that DMA controller is used with the microprocessor that helps to generate the addresses for the data to be transferred from the I/O devices. The peripheral device sends the request signal (DMARQ) to the DMA controller and the DMA controller in turn passes it to the microprocessor (HOLD signal). On receipt of the DMA request the microprocessor sends an acknowledge signal (HLDA) to the DMA controller. On receipt of this signal (HLDA) the DMA controller sends a DMA acknowledge signal (DMACK) to the I/O device. The DMA controller then takes over the control of the buses of microprocessor and controls the data transfer between RAM and I/O device. When the data transfer is complete, DMA controller returns the control over the buses to the microprocessor by disabling the HOLD and DMACK signals.

Now question is how many way DMA can work? It may be mentioned here that DMA transfer the data of the following types:

* Memory to I/O device
* I/O device to memory
* Memory to memory
* I/O device to I/O device