Computer organization and microprocessor

Unit -5 **24/4/2020**

Data transfer techniques

Why is Data Transfer needed?

We can connect several I/O devices and memory peripherals to a microprocessor. However, since different technologies are involved, there will be differences in the speed of operation and of data transfer.

Usually, when memory is connected with the microprocessor, there isn't a stark difference in the processing speed since semiconductor memories are generally easily compatible with microprocessors. However, this might not always be the case.

But the problem often arises when external peripherals are connected as I/O devices. A slow I/O device won't be able to transfer data at a satisfactory rate whenever the microprocessor requests for data transfer. And usually, the peripheral devices *do* have slower transfer rates than the processor. Maybe the processor might send two units of information per second, but the external device might only accept 1 unit of information per second. Or conversely, perhaps the external device expects quicker transfers, but our microprocessor might be sending information a bit slower.

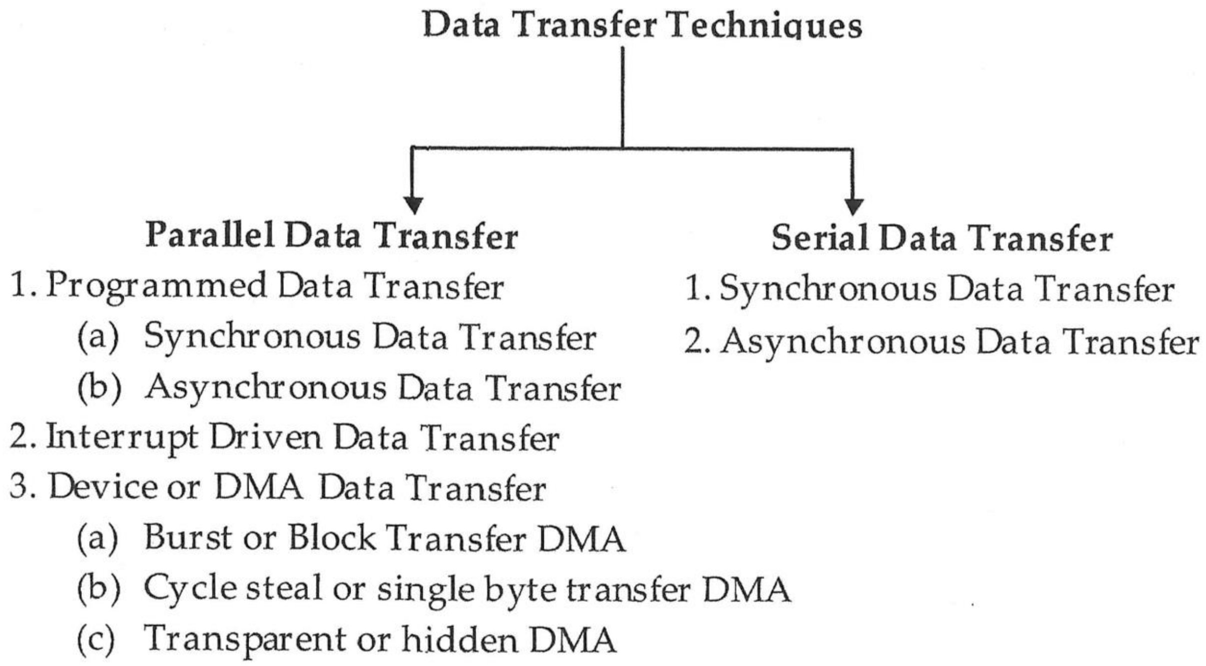
This might lead to severe data losses, or the devices might get damaged, or there might even be chances for the system to slow down all over, thus affecting the overall efficiency of the system.

To avoid this problem, a number of ***data transfer techniques*** have been devised.

Since constant communication is required between the external device and the processor, these data transfer techniques play a crucial role in the efficient functioning of the system with the externally connected devices.

Classification of Data Transfer Schemes

We can broadly classify the data transfer schemes into two modes – Serial Data Transfer and Parallel Data Transfer.


Data Transfer Techniques
Parallel Data Transfer
1. Programmed Data Transfer
(a) Synchronous Data Transfer
(b) Asynchronous Data Transfer
2. Interrupt Driven Data Transfer
3. Device or DMA Data Transfer
(a) Burst or Block Transfer DMA
(b) Cycle steal or single byte transfer DMA
(c) Transparent or hidden DMA
Serial Data Transfer
1. Synchronous Data Transfer
2. Asynchronous Data Transfer

Data Transfer

As discussed earlier, we can broadly classify data transfer schemes into parallel data transfer techniques and serial data transfer techniques. Let us understand the differences between them.

SERIAL DATA TRANSFER TECHNIQUES	PARALLEL DATA TRANSFER TECHNIQUES
Under this scheme, the data is transferred one bit at a time.	Under this scheme, the data is transferred several bits the same time.
It is a slower mode of data transfer.	Data is transferred much quicker.
Serial data transfer is preferred when data is to be sent over a long distance and the cost of cables would be too expensive.	Parallel data transfer is the preferred technique for sho distance communication.
For this mode, the transmitter first performs parallel – to – serial conversion and the serial – to – parallel conversion at the receiver.	No such conversions are required at both the transmission and reception endpoints.
This mode requires a single line to transfer information.	This mode requires multiple lines for data transfer.
Noise and errors are much lesser.	As multiple bits are transmitted at the same time, there scope for more error and noise.
Cables used for serial communication are much thinner, longer, and very economical.	Here, the cables are much shorter, and thicker compar to the Serial communication cables.
This is a very reliable, inexpensive, and straightforward	It is considered a little unreliable, expensive, and

	process.	complicated process.

Parallel Data Transfer Techniques

We know that under the parallel data transfer scheme, multiple data bits can be transmitted at the same time. Thus, for the Intel 8085, 8 bits of data are sent all together using eight parallel lines. Let us go through the different types of parallel data transfer schemes. We have:

- Programmed I/O Data Transfer
- Interrupt Driven I/O Data Transfer
- Device or Direct Memory Access (DMA) Data Transfer

Let us study each of these transfer schemes in detail.

Programmed I/O Data Transfer

- This is a straightforward scheme under parallel data transfer mechanisms. This mode is generally preferred for simple, small microprocessor systems where speed is critical.
- This method can work under the synchronous and asynchronous mode, depending on the speed and architecture of the I/O devices.
- We also prefer this method when there is a small amount of information to be exchanged between the microprocessor and other devices that are placed near to the microprocessor. Example: Computer, printer, etc.

- Using the IN andand OUT instructions, data transfer is carried out between the microprocessor and I/O devices.
- The processor reads the data from an input port or input device using the IN command. The processor sends data out from the CPU to the output port or the output device using the OUT instruction.
- Thus, as the speeds of the processor and external device match, the data transferring process is carried out using the IN and OUT instructions.

Synchronous Data Transfer Method

- The word '**Synchronous**' means 'taking place at the same time.'
- Thus, to establish communication between our processor and the device, we need to set a common clock pulse. This common pulse ***synchronizes*** the peripheral device with the 8085 microprocessor.

- This method is used when the speed of the microprocessor, Intel 8085, in this case, and the external peripheral device match with each other.
- If the device is ready to send data, it can indicate via the **READY** pin of 8085. Once the speeds match, the data transfer immediately begins, once a signal is issued by the microprocessor to begin transferring. The microprocessor need not wait for an extended period because of the matching speeds.
- This technique of data transfer is seldom used to communicate with I/O devices though. Because I/O devices compatible with the microprocessor's speed are usually not found.
- Hence, this method of data transfer is most commonly employed for communicating with compatible memory devices.

Asynchronous Data Transfer Method

- When the speed of the I/O device is slower than that of the microprocessor, we prefer the Asynchronous Data Transfer Method. As the speeds of both the devices differ, the I/O device's internal timing is entirely independent of the microprocessor.
- Thus, they are termed to be '***asynchronous***' from each other. The term asynchronous means 'at irregular intervals.'