UNIT STRUCTURE

2.1 LEARNING OBJECTIVES

After going through this unit, you will be able to :

- learn about the input and output devices
- learn how data transfer takes place between computer and peripherals
- know about the interrupts
- describe input-output processors

2.2 INTRODUCTION

In the previous unit, we have learnt about the various digital components. In this unit, we will discuss how the data is fed to the computer and the way they are being processed, that is, how the data is being transferred to and from the external devices to the CPU and the memory.

2.3 INPUT- OUTPUT DEVICES

The computer communicates with the users with the help of input and output devices. Input devices enter data and instructions into the computer for processing and the output devices display the result obtained for the users to see. The input-output devices attached to the computer are also known as peripherals. There are three types of peripherals: input, output and input-output peripherals. Most common peripherals are :

- Keyboard,
- Monitor,
- Printers,
- The auxiliary storage devices such as magnetic disks and tapes.

A brief description of their functions are given below :

Keyboard and Monitor :

Keyboard allows the user to enter alphanumeric information. On pressing a key, a binary coded character, typically 7 or 8 bits in length is sent to the computer. The most commonly used code is a 7-bit code referred to as ASCII (American Standard Code For Information Interchange). Each character in this code is represented by a unique 7-bit binary code; thus a total of 128 different characters can be represented as shown in table 2.1

Table 2.1 The American Standard Code for Information Interchange(ASCII)

	0 **0** **0**	**0** **0** **1**	**0** **1** **0**	**0** **1** **1**	**1** **0** **0**	**1** **0** **1**	**1** **1** **0**	**1** **1** **1**
0000	NUL	DLE	SP	0	@	P		p
0001	SOH	DC$_1$	!	1	A	Q	a	q
0010	STX	DC$_2$	“	2	B	R	b	r
0011	ETX	DC$_3$	#	3	C	S	c	s
0100	EOT	DC$_4$	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

The **Control Character** are :

NUL - Null

SOH - Start of Heading

STX - Start of Text

ETX - End of Text

EOT - End of Transmission

ENQ - Enquiry

ACK - Acknowledge

BEL - Bell

BS - Backspace

HT - Horizontal Tab

LF - Line feed

VT - Vertical Tab

FF - Form feed

CR - Carriage Return

SO - Shift Out

SI - Shift In

DLE - Data Line Escape

DC1, DC2, DC3, DC4 - Device Controls

NAK - Negative Acknowledge

SYN - Synchronous Idle

ETB - End of Transmission Block

CAN - Cancel

EM - End of Medium

SUB - Substitute

ESC	- Escape
FS	- File Separator
GS	- Group Separator
RS	- Record Separator
US	- Unit Separator
DEL	- Delete

Video monitors may be of different types but most commonly used one is the CRT (Cathode Ray Tube) monitors. Another one that becomes popular now-a-days is LCD (Liquid Crystal Display) monitor.

The monitor displays a cursor on the screen which marks the position on the screen where the next character is to be inserted.

Printers :

Printers produce a hard copy of the output data. Based on the technology used printers may be either impact or non-impact. Impact printers operates by using a hammer to strike a character against an inked ribbon; the impact then causes an image of the character to be printed. Types of impact printers are dot-matrix, daisy wheel, line printers. Non-impact printers do not touch the paper while printing. Non-impact printers like Laser printers uses a rotating photographic drum to imprint the character images and then transfer the pattern onto the paper. Inkjet, deskjet and thermal printers also fall under this category.

Magnetic Tapes and Disks :

The magnetic tape drives and disk drives are used to read and write data to and from the magnetic tapes and the disks. The surfaces of these output devices are coated with magnetic material so as to store data in them. These surfaces can be magnetized and the presence of a magnetic field represents a ‘1’ bit and the absence of a magnetic field represents a ‘0’ bit. The magnetic disk is a circular metal disk that is coated on both sides to store data. The magnetic tape stores data in a sequential manner. To process a data in a tape ,it must begin searching at the beginning and check each record until the desired record is found.

In case of magnetic disks, the data can be stored and accessed directly and so it is fast. Floppy disks, hard disks are examples of magnetic disks.

CHECK YOUR PROGRESS

1. Fill in the blanks :

   (a) The Keyboard is a ____________ device.

   (b) Daisy Wheel is a ________ printer and the Deskjet is a __________ printer.

   (c) Phosphor on being struck by ______ emits light on the monitor screen.

   (d) Magnetic tape stores data on its _________ sides.

2.4 I/O INTERFACE

The I/O Interface is responsible for exchange of data between the peripherals and internal storage of the computer. Instead of directly connecting the peripherals to the system bus, an I/O Interface is used in between because,

1. The mode of operation of the different peripherals is different from the operation of the CPU and memory.
2. The data transfer rate of the peripherals is much slower compared to that of CPU and memory.
3. Peripherals use different data formats than the word format in the CPU and memory.

The interface units are hardware components between the CPU and the peripherals that supervise and synchronize all the I/O transfers.

Two main types of interfaces are CPU Interface that corresponds to the system bus and input-output interface that is tailored to the nature and operation of the peripheral devices. The major functions of the I/ O Interface units are :

1. Data Conversion; conversion between digital and analog signals.
2. Synchronization; synchronizes speeds of the CPU and other peripherals.
3. Device Selection; selection of the I/O device by the CPU in a queue manner.

Input-Output devices are attached to the processor with the help of the I/O Interface as shown in the figure below:

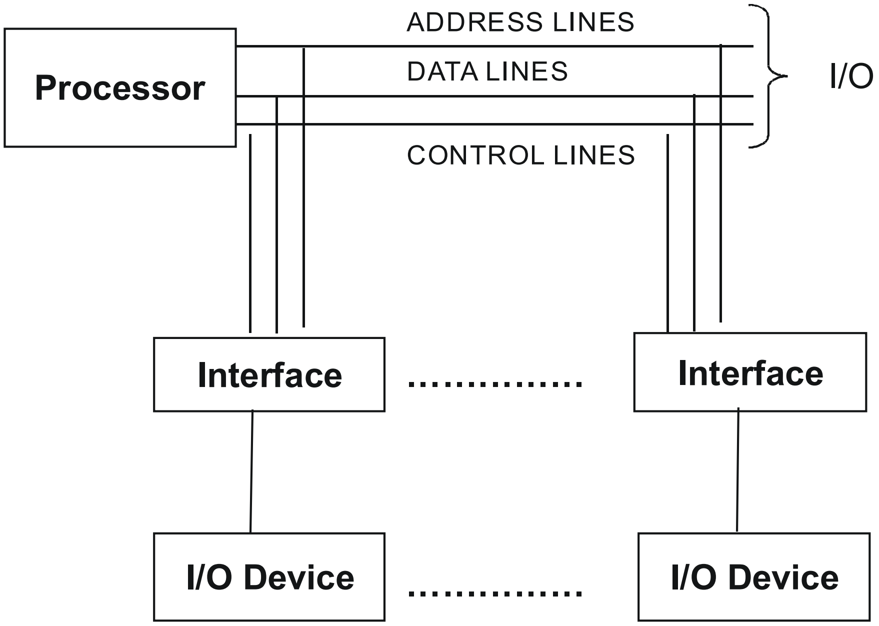

Fig. 2.1 : Processor and the I/O Interface

The I/O interface circuit also has an address decoder, control circuits and data and status registers. Whenever the CPU has to communicate with a particular device, it places its address in the address lines; the address decoder in the address lines then helps in recognizing the address of its connected I/O device. The I/O interface can also generate addresses associated with the device that it controls. The control circuits accept the detailed control information from the CPU. The I/O interface also holds the responsibility of data buffering. The data registers in it holds the data being transferred to and from the interface. This is necessary as the data transfer rate into and out of the CPU or memory is quite high as compared to the peripheral devices. The status registers provides the current status information relevant to the input-output device operation. The data and status registers are connected to the data bus.

The input-output interface can detect errors and then after detection reports them to the CPU.

CHECK YOUR PROGRESS

2. Write True or False :

   (a) The I/O Interface is a hardware component in the computer.

   (b) The Control lines of the System Bus carries the addresses of the peripherals.

   (c) Main memory can transfer data at high speed compared to the peripherals.

   (d) The Status register in the I/O Interface holds the data that is being transferred.

2.5 DIFFERENT I/O TECHNIQUES

There are different I/O techniques in the computer. Some may involve the CPU for transferring data between the computer and the input-output devices and others may directly transfer data from and to the memory units. The three possible techniques for I/O operations are :

- Programmed I/O,
- Interrupt - Driven I/O,
- Direct Memory Transfer (DMA)

2.5.1 Programmed I/O

In the Programmed I/O, data transfer takes place between the peripherals and the processor during program execution.

When the CPU encounters an I/O instruction, while executing a computer program, it immediately sends the address of the particular I/O device in the address lines of the I/O bus and also at the same time it issues a command to the particular I/O interface in the control lines of the I/O bus.

There are four types of commands that an interface may receive:

- Control command
- Status command
- Read command
- Write command

A *Control command* is used to activate the peripheral and tell it what to do. This command depends on the type of the particular peripheral.

A *Status command* is used to test the various status of the peripheral and the I/O interface.

A *Read command* causes the interface to get the data from the peripheral and place it in its data registers. The interface then places the data in the data lines of the bus for the processor when it requests for it.

A *Write command* causes the interface to transfer the data from the bus into its data registers. When it completes transferring data to its data registers, it then transmits that data to the particular I/O device.

In Programmed I/O, the CPU must constantly check the status of the I/O interface. That is, on issuing a command to the I/O interface, the processor must wait until the I/O operation is complete. This is a time consuming process as it keeps the CPU needlessly busy.

2.5.2 Interrupt- Driven I/O

In Programmed I/O the processor has to constantly monitor the status of the interface to check when it is ready for data transfer. This causes wastage of the CPU time. An alternative to this method is the Interrupt-Driven method, where after issuing a command to the interface, the CPU can switch to execution of other programs in the computer. When the interface is ready for the transfer, it will immediately send an interrupt signal to the CPU. The CPU will then do the data transfer and resume back to what it was doing before it received the interrupt signal. One of the bus control lines, called an Interrupt Request Line, is dedicated for this purpose.

In this method, the advantage is that the CPU need not spend time waiting for an I/O operation to be completed, thereby increasing its efficiency.

The routine that is executed when there is an interrupt request is called Interrupt Service Routine (ISR). When the CPU gets an interrupt signal, while it is executing an instruction of a program, say instruction I, it first finishes execution of that instruction. It then stores the current contents of the Program Counter (PC), which now points to the next instruction i+1, and stores it in a memory stack. This is its return address after execution of the ISR. The control then branches to the service routine that processes the required I/O transfer.

Determining the I/O device requesting Interrupt

When the CPU receives an interrupt signal over the interrupt request line, it must determine which I/O interface has sent it. There are different techniques for determining this. They are:

1. Software Poll
2. Vectored Interrupt
3. Bus arbitration

1. Software Poll

When the CPU detects an interrupt, it branches to an interrupt-handler. This interrupt-handler poll each I/O interface to determine which interface caused the interrupt. The interface which signaled the interrupt responds positively to it. This is time-consuming.

2. Vectored Interrupt

This is a more efficient technique which uses a daisy chain. It provides a hardware poll. The interrupts are signaled through a common Interrupt request line. The interrupt acknowledge line is daisy-chained through the I/O interface. On receiving an interrupt, CPU sends an acknowledge signal through it. The interface which sent the interrupt responds by placing its vector on the data lines.

3. Bus Arbitration

At a time only one interface can send an interrupt signal. When the CPU detects the interrupt signal, it acknowledges through the interrupt acknowledge line. The interface signaling the interrupt then places its vector on the data lines.

Handling Multiple Interrupts

There may be multiple interrupts simultaneously. In those cases the CPU has to decide which one to service first. Priorities are thus assigned to the I/O interfaces to determine the order in which the interrupts are serviced. The devices having higher speeds such as the disks are given higher priorities, while the slower devices such as keyboards are given low priority. Also higher priority is given to the interrupts which if delayed may cause serious consequences. So, when two device interfaces sends interrupt requests to the processor at the same time, the device having higher priority will be serviced first.

2.5.3 Direct Memory Access

When large blocks of data are to be transferred between memory and high-speed I/O devices, a more efficient approach is used. This alternative technique is the Direct Memory Access (DMA) technique. It eliminates the continuous intervention of the CPU in transferring data.

DMA transfers are performed by a DMA controller, which is a control unit in the I/O interface. This unit carries out the functions of the processor when accessing the main memory during the data transfers. In DMA transfers, the controller takes control of the memory buses from the CPU and does the transfer between the peripheral and the memory.

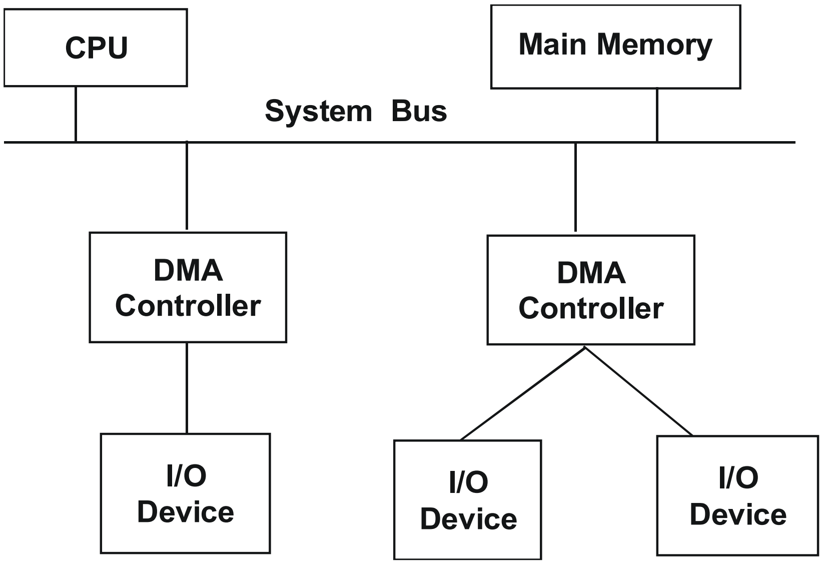

Fig. 2.2 : DMA Controllers in a Computer System

When the CPU gets an instruction that involves huge amount of data transfer while executing a computer program , it issues a command to the DMA Controller sending the address of the I/O device, the number of words to be transferred and whether a read/write operation is to be performed. Thus the DMA, CPU just initializes the data transfer and become free for other works. i.e., the data transfer operation is completed by the DMA controller without the continuous involvement of the CPU.

The transfer may be in two ways / modes:

Burst Transfer : By Burst Transfer a block of words is transferred in a continuous burst. This mode is needed for fast devices where data cannot be stopped until the entire block is transferred. Then the controller informs the CPU by sending an interrupt signal.

Cycle Stealing : The controller transfers one word at a time and return the bus control to the CPU. The CPU then temporarily suspends its operation for one memory cycle to allow the DMA transfer to steal its memory cycle.

The DMA controller uses a register for storing the starting address of the word to be transferred, a register for storing the word count of the transfer and another register contains the status and the control flags. The status and the control registers are shown below

IRQ	IE		R/W	Done

IRQ	:	Interrupt Request
IE	:	Interrupt Enabled
R/W	:	Read or Write operation

The IRQ bit is set to 1 when the controller has requested an interrupt. When the data transfer is completed, it sets the IE to 1 thereby to raise an interrupt. When it is ready for the next command, after completion, the Done bit is set to 1. When the R/W bit is 1, the controller performs a Read operation, otherwise a Write operation is performed.

Whenever a word is transferred, the DMA Controller increments its address register to point to the next word to be transferred and decrements its word count register. When finally the word count becomes zero, the controller stops any further transfer and informs the CPU by raising an interrupt.

CHECK YOUR PROGRESS

3. Write True or False :

   (a) There are only 2 techniques for the I/O operations in the computer.

   (b) Read command in Programmed I/O causes the interface to get the data from I/O devices and place it in its data registers.

   (c) Interrupt Driven I/O wastes the CPU time needlessly.

   (d) Interrupt Service Routine is an interrupt signal.

   (e) In Direct Memory Access, the data transfer takes place directly between the peripheral and the memory.

   (f) The DMA Controller carries out the functions of the processor when accessing the main memory during the data transfers.

2.5.4 I/O PROCESSOR

An Input – Output processor (IOP) is a processor in addition to the CPU. It directly communicates with the input and output devices. The IOP need not interfere with the tasks of the CPU as they themselves can fetch and execute the instructions. Other than fetching and executing I/O instructions the Input Output processor can also do processing tasks like arithmetic, logic and branching.

Data from the devices are first collected into the IOP and then transferred to the memory directly by stealing one memory cycle from the CPU. Similarly data is transferred from memory to IOP