# UNIT 8: MEMORY AND I/O INTERFACING

## UNIT STRUCTURE

8.1 Learning Objectives

8.2 Introduction

8.3 I/O Processor (IOP)

8.4 Priority Encoder

8.5 Interfacing with the I/O Devices

8.6 Let Us Sum Up

8.7 Further Readings

8.8 Answers to Check Your Progress

8.9 Model Questions

## 8.1 LEARNING OBJECTIVES

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

- describe I/O processor i.e. IOP
- learn about the communication way of IOP with memory unit, CPU and peripheral devices
- define Priority Encoder
- describe Interfacing with the I/O Devices, such as Keyboard, Printer and Display Interfaces

## 8.2 INTRODUCTION

In the previous few units we have discussed the different types of memory mapping techniques like direct and set associative mapping etc along with concepts like replacement algorithm. We have also discussed about memory read and write operations in detail in the previous unit. In this unit, we will discuss the basics of the I/O processor including the communication way among of IOP with memory unit, CPU and peripheral devices. Then we will learn about the basic concept of Priority Encoder. An effort is made here to discuss in detail the concept of Interfacing with the I/O Devices, such as keyboard, printer and display interfaces. In the next unit, the concepts of I/O processor and interfacing with I/O devices will be explored.

## 8.3 I/O PROCESSOR (IOP)

A I/O processor is a processor which communicates with I/O devices with direct input-output memory access capability. Regarding the setup of this configuration, the computing system can be arranged as follows:

1. A memory unit
2. The CPU
3. One or more IOPs

Each IOP releases the CPU from ordinary I/O transfer process by taking care of I/O tasks. So CPU will get more time to be busy with other important tasks.

Instructions of IOP are specially designed to perform I/O operations. IOP can fetch and even execute its own instructions. The IOP is efficient itself for performing processor's operation such as all arithmetic operations, logical operations, branching or looping operations, code translation

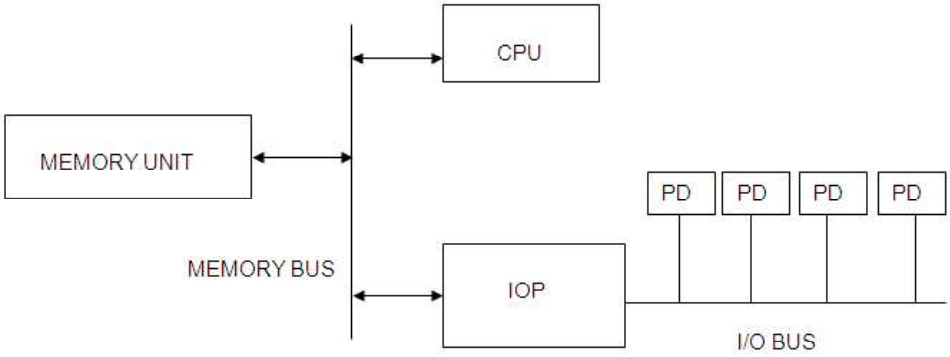



**Figure 8.1: Block Diagram of a typical IOP**

Memory unit has interface with the CPU and IOP both. CPU is there for processing the data of computational tasks. The IOP is providing a path for various peripheral devices and memory unit for data transfer purpose. Initialization of I/O program will be done by the CPU. After initialization of I/O program, IOP will execute other necessary actions for data transfer between peripheral devices and memory.

As memory and CPU have their own data formats, peripheral devices also have own data formats and the data formats of peripheral devices may not be same as the data formats of memory and

CPU.Regarding these data formats IOP is responsible for structuring the data words coming from different sources.

The communication of IOP with memory unit, CPU and peripheral devices can be summarized as follows:

1. Communication of IOP with memory unit is equivalent to the DMA (Direct Memory Access) method.
2. Communication of IOP with CPU differs from computing system to computing system. Basically, we can say that CPU acts like a master and IOP is like a slave.
3. Communication of IOP with peripheral devices is similar to the program control method of transfer.



**CHECK YOUR PROGRESS**

**Choose the correct answer:**

Q1. Instructions of IOP are specially designed / not specially designed to perform I/O operations.

Q2. Initialization of I/O program will be done by the CPU / IOP.

Q3. Data formats of peripheral devices same / may not same with the data formats of memory and CPU Operation.

## 8.4 PRIORITY ENCODER

To implement the priority function, there is a circuit which is called **Priority Encoder**. When two or more input requests come simultaneously, then the input request having the highest priority will be performed. This is the logic behind the working principle of priority encoder. The truth table for a priority encoder having four inputs is given below:

**Table 8.1: Truth Table of Priority Encoder with Four Inputs**

| INPUTS | | | | OUTPUTS | | |
|---|---|---|---|---|---|---|
| I0 | I1 | I2 | I3 | x | y | IST |
| 1 | X | X | X | 0 | 0 | 1 |
| 0 | 1 | X | X | 0 | 1 | 1 |
| 0 | 0 | 1 | X | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | X | X | 0 |

Here $I_0$, $I_1$, $I_2$, $I_3$ input lines and x and y are two output status. IST is the interrupt status. X stands for don't care condition.

The characteristics of a priority encoder with four inputs are drawn from the above truth table as follows:

1. $I_0$ has the priority. When $I_0$ is high, then regardless of the status of other input the status output xy will be 00.
2. Next priority level is achieved by $I_1$. For high status of $I_1$, then regardless of the status of $I_2$, $I_3$ input the status output xy will be 01 and so on.
3. The interrupt status IST will be set when one or more inputs is high.
4. When all inputs are low, output xy will be 00 and interrupt status IST=0.

A priority encoder with eight inputs will generate an output with three bits as shown below:

**Table 8.2: Truth Table of Priority Encoder with Eight Inputs**

| **INPUTS** | | | | | | | | **OUTPUTS** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **$I_0$** | **$I_1$** | **$I_2$** | **$I_3$** | **$I_4$** | **$I_5$** | **$I_6$** | **$I_7$** | **X** | **Y** | **Z** | **IST** |
| 1 | X | X | X | X | X | X | X | 0 | 0 | 0 | 1 |
| 0 | 1 | X | X | X | X | X | X | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | X | X | X | X | X | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | X | X | X | X | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | X | X | X | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | X | X | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | X | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | X | X | X | 0 |

## 8.5 INTERFACING WITH THE I/O DEVICES

There must be good communication between internal storage system and external I/O devices for transferring information. This communication is provided by input-output interface. Again, the external devices connected to computer, which are known as peripheral devices, need special

communication links for interfacing them with the brain of the computer i.e. the CPU. To provide this special communication links and for the purpose of supervising and synchronizing all input and output transfers, some special hardware components are there between CPU and peripherals. These special hardware components are called **interface** units.

There are two main types of interface:

1. CPU interface : It corresponds to the system bus
2. Input-Output interface: It depends on the nature of input-output devices.

Data conversion, synchronization, and device selection are the main functions of input-output interface circuit. A typical communication link between the processor and the different peripherals (e.g. keyboard, printer display devices etc) is shown below in figure 8.2.

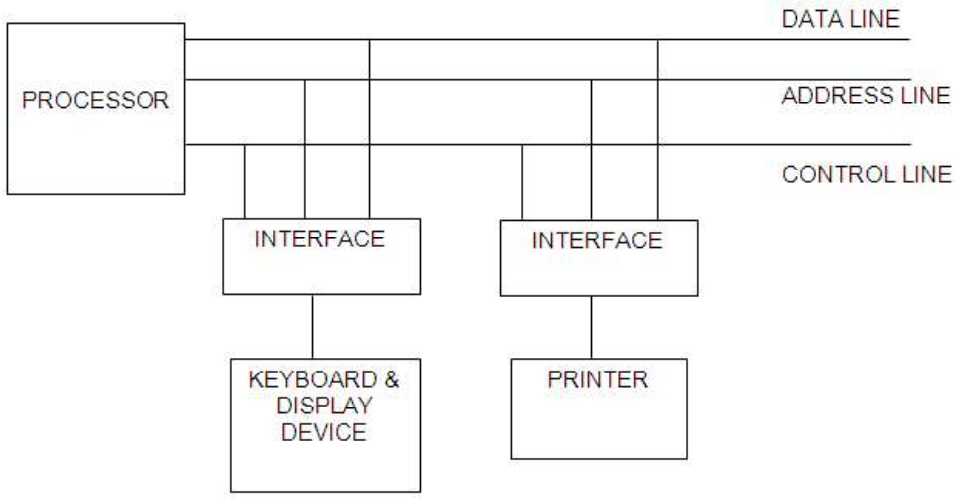



**Figure 8.2: A Typical Communication Link between Processor and Peripherals**

There are three lines for I/O bus organization, namely, Data line, Address line and Control line. Each peripheral device shown in the above figure i.e. keyboard, display device, printer etc. is associated with an interface unit. The functional characteristic of each interface unit is to decode the address and the control received from the I/O bus, to interpret them for the peripheral, and to provide signal for the peripheral controller. It is also

responsible for synchronizing the data flow and for supervising the data transfer between peripheral and processor. To operate the particular electromechanical device, each and every peripheral device has its own controller.

The communication between a particular device and the processor happens as shown below:

1. Processor has to put the address of the device with which it wants to make communication on the address line.
2. An address decoder is attached with the each Interface connected with the I/O bus. This address decoder has to monitor the address lines.
3. When the decoder detects its own address, then it sends the report of detecting its own address to its interface to activate the path between the bus lines and the device that it controls.

After receiving its own address, the interface has to receive again a function code from the control lines sent by the processor. This function code is referred to as I/O command. After receiving the I/O command, the interface has to proceed to execute it. There are four types of commands that an interface may receive. These are:

1. Control Command: Its responsibility is to activate the interface and to inform the interface about the task to be performed.
2. Status Command: There are various status conditions in interface and peripheral. To test these various statuses, status command is used.
3. Data Output Command: After receiving this command, the interface has to respond by transferring the data from the bus to the I/O device.
4. Data Input Command: In this case, interface has to receive the data from the peripheral and transfer it to the CPU.



**CHECK YOUR PROGRESS**

Q4. _____ circuit is used to implements the priority function.

Q5. In Priority Encoder circuit, IST stands for what?

Q6. A priority encoder with eight inputs will generate an output with ___ bits.

Q7. The external devices connected to computer are known as ____.

Q8. To provide special communication links between CPU and peripherals, some special hardware components are used, and they are called ____.

Q9. The names of three lines for I/O bus organization are ____.

Q10. The functional characteristic of each interface unit is to decode / encode the address. (Which one is correct?)

Q11. There are ____ types of commands that an interface unit may receive.

Q12. Address decoder has to monitor the ____ lines



## 8.6 LET US SUM UP

- Each IOP releases the CPU from ordinary I/O transformation by taking care of I/O tasks.
- The IOP is efficient itself for performing the processor's operation such as all arithmetic operations, logical operations, branching or looping operations, code translation etc.
- As memory and CPU have their own data formats, peripheral devices also have own data formats, and data formats of peripheral devices may not be same as the data formats of memory and CPU.
- Communication of IOP with memory unit is equivalent to the DMA (Direct Memory Access) method.
- Communication of IOP with CPU differs from computing system to computing system. Basically, we can say that CPU acts like a master and IOP is like a slaver.
- Communication of IOP with peripheral devices is similar to the program control method of transfer.
- To implement the priority function, there is a circuit which is called Priority Encoder. When two or more input requests come, then the input request having the highest priority will be performed.
- The external devices connected to the computer, which are known as peripheral devices, need special communication links for interfacing them with the brain of the computer i.e. the CPU.
- In daisy chaining technique the device which is electrically closest to the processor will acquire the highest priority.



## 8.7 FURTHER READINGS


1) Mano, M. M. (2006). Computer systems architecture.

2) Hamacher, V. C., Vranesic, Z. G., Zaky, S. G., Vransic, Z., & Zakay, S. (1984). Computer organization (Vol. 3). New York et al.: McGraw-Hill.




## 8.8 ANSWERS TO CHECK YOUR PROGRESS

**Ans to Q No 1:** Special Device

**Ans to Q No 2:** CPU

**Ans to Q No 3:** May not same

**Ans to Q No 4:** Priority Encoder

**Ans to Q No 5:** Interrupt Status

**Ans to Q No 6:** Three

**Ans to Q No 7:** Peripheral Devices

**Ans to Q No 8:** Interface Unit

**Ans to Q No 9:** Data Line, Address Line, Control Line

**Ans to Q No 10:** Decode

**Ans to Q No 11:** Four

**Ans to Q No 12:** Address



## 8.9 MODEL QUESTIONS

Q1. What is IOP?

Q2. What is Priority Encoder? Explain the functional characteristics of a Priority Encoder with eight inputs.

Q3. Explain the working principle and functional characteristics of a typical IOP with two processors.

Q4. Explain the typical communication link between the Processor and the peripherals in detail.

Q5. How does the communication between a particular device and the processor happen?

Q6. Explain all the types of commands that an interface may receive.