UNIT 7: INPUT - OUTPUT ORGANIZATION

UNIT STRUCTURE

7.1 Learning Objectives

7.2 Introduction

7.3 Input Output Organization

7.4 Different I/O techniques

- 7.4.1 Programmed I/O
- 7.4.2 Interrupt-Driven I/O
- 7.4.3 Direct Memory Access (DMA)

7.5 Priority and Daisy Chaining Technique

7.6 Let Us Sum Up

7.7 Further Readings

7.8 Answers to Check Your Progress

7.9 Model Questions

7.1 LEARNING OBJECTIVES

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

- describe the different input-output techniques
- describe interrupt driven and programmed I/O techniques
- define direct memory access (DMA)
- describe daisy chaining technique

7.2 INTRODUCTION

In the previous units, we have discussed the different types of sequential and combinational circuits and data representation in binary number system, decimal number system, BCD, ASCII systems etc. We have also learnt about the different instruction formats and types in the previous unit along with different types of addressing modes and assembly language notation. In this unit, we will explore the different concepts related to input output organization. Different I/O techniques like programmed I/O, interrupt driven I/O and DMA are covered in detail in this unit. Priority and

daisy chaining technique are also covered in this unit. In the next unit, we discuss about cache and virtual memory.

7.3 INPUT OUTPUT ORGANIZATION

The transfer of information between the main memory and the outside world is through the input-output devices. An I/O system is composed of I/O devices (peripherals), I/O control units, and software to carry out the I/O transaction(s) through a sequence of I/O operations. An I/O transaction example is reading a block of data from disk to memory. It is simply a sequence of I/O operations (instructions) to transfer data between the peripheral devices and main memory, and to enable the central processing unit (CPU) to control the peripheral devices connected to it. Thus, I/O operations are of two classes: control operations and data transfer operations. There are different techniques for communication between the memory and the I/O devices which will be discussed in the following sections.

7.4 DIFFERENT I/O TECHNIQUES

The different I/O techniques are categorized into three types based on how information is transferred between the main memory and the input/output devices, that is, whether CPU interaction is used continuously or the CPU is interrupted by the device only at the time of transfer. They are:

1. Programmed Input/Output
2. Interrupt driven Input/Output
3. Direct memory Access

The **I/O module,** also called an I/O interface works as the mediator between the I/O devices and the CPU. The information from the I/O devices to the CPU and vice-versa is conveyed by the I/O module. An I/O module is connected to the system bus at one end and is connected to the number of I/O devices on the other end.

The I/O module is used for many reasons:

a. Due to the speed mismatch between the CPU and the peripherals, it is improper to use a slow peripheral on a high speed system bus.

b. Many peripherals might be required to be connected to the same system bus but it may be difficult to do so.
c. Data format of the peripherals might be different from that used by the CPU.

The I/O module functions are as given below:

i. Provide control and timing signals
ii. Communicate with CPU
iii. Communicate with I/O devices.

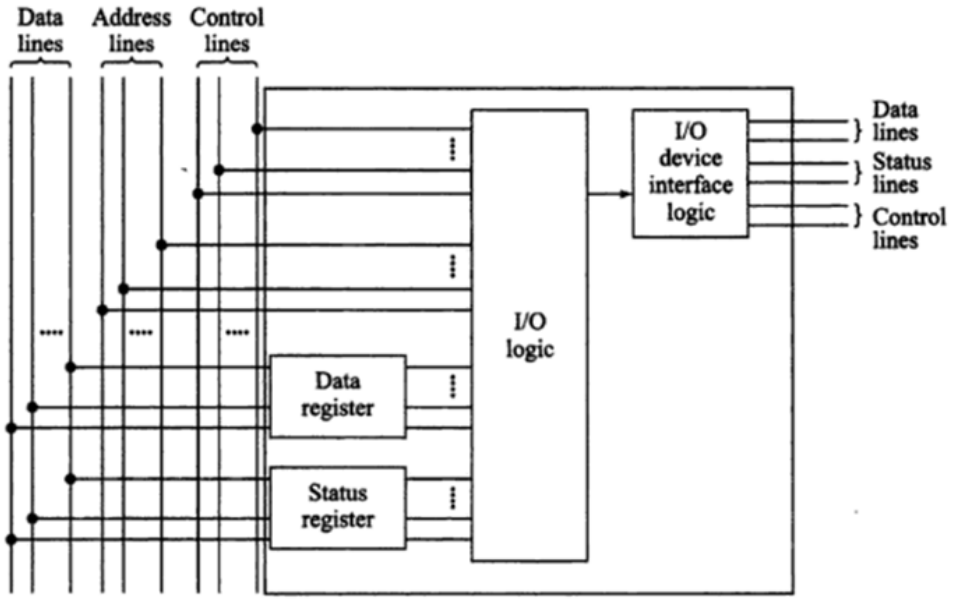


Figure 7.1: General structure of the I/O module

7.4.1 Programmed I/O

The programmed I/O is the simplest type of I/O technique for the exchange of data or any type of communication between the processor and the external devices. With programmed I/O, data are exchanged between the processor and the I/O module i.e. data may be transferred from I/O device to the CPU as input or from CPU to the I/O device as output.

At the time of executing an I/O instruction the CPU issues a command, then waits for I/O operations to be complete. As the CPU is faster than the I/O module, the CPU has to wait a long time for the concerned I/O module to be ready for either reception or transmission of data. The

CPU, while waiting, must repeatedly check the status of the different I/O modules one after another serially and this process is known as **polling**. As a result, the level of the performance of the entire system is severely degraded.

Programmed I/O basically works in these ways:

- CPU requests I/O operation
- I/O module performs operation
- I/O module sets status bits
- CPU checks status bits periodically
- I/O module does not inform CPU directly
- I/O module does not interrupt CPU
- CPU may wait or come back later

Programmed I/O Mode Input Data Transfer:

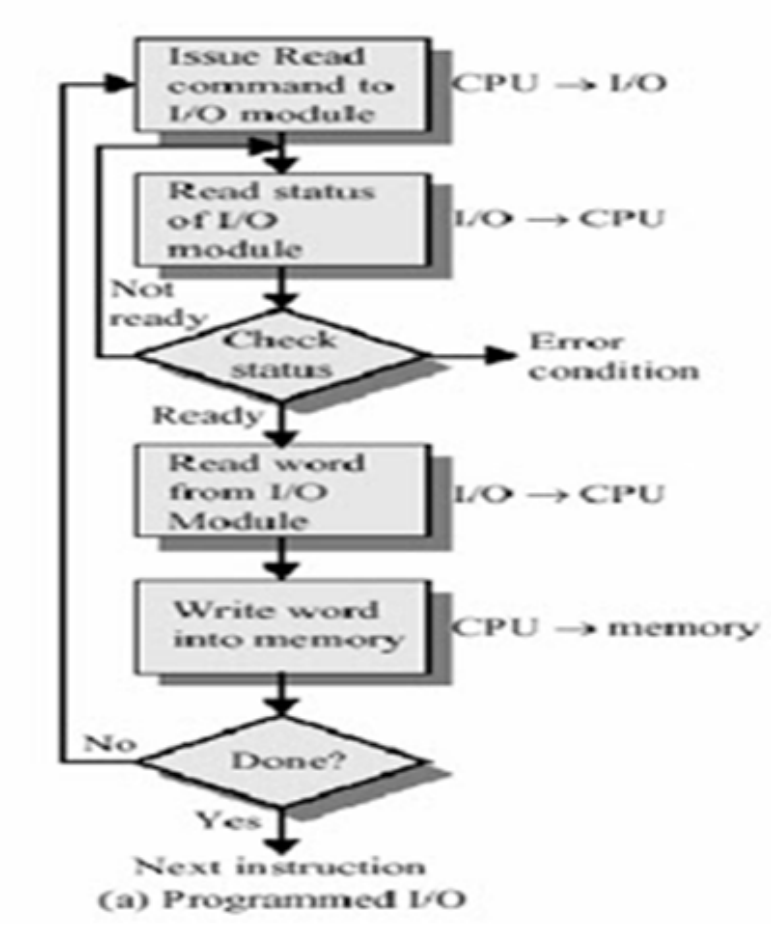


Figure 7.2: Prgorammed I/o mode Input Data Transper

Advantages & Disadvantages of Programmed I/O

Advantages	- simple to implement, suitable for very small amount of data transfer
	- very little hardware support
Disadvantages	- busy waiting
	- ties up CPU for long period with no useful work

7.4.2 Interrupt-Driven I/O

Interrupt driven I/O is another technique which is dealing with I/O operations. It is a way of controlling input/output activity in which the peripheral that needs to send or receive data sends a signal.

The problem with Programmed I/O is that the processor has to wait for a long time for the concerned I/O module to be ready for either reception or transmission of data. The processor must continuously check the status of the I/O module and as such the performance of the system is affected.

An alternative to this is for the CPU to issue a command to the I/O module and to proceed to do some other useful task until the device requesting the service sends an interrupt to the CPU when it is ready to exchange data. The processor then executes the data transfer and on completion resumes its former processing.

The interrupt technique requires more complex hardware and software, but makes far more efficient use of the computer's time and capacities.

The basic operations of Interrupt are as follows:

1. CPU issues read command.
2. I/O module gets data from peripheral while the CPU does other work.
3. I/O module interrupts CPU.
4. CPU requests data.
5. I/O module transfers data.

Advantages & Disadvantages of Interrupt Drive I/O

Advantages	-	fast
	-	efficient, no wastage of CPU time
Disadvantages	-	can be tricky to write if using a low level language
	-	can be tough to get various pieces to work well Together
	-	usually done by the hardware manufacturer / OS maker, e.g. Microsoft

In interrupt driven I/O implementation, there are two design issues. First, there may be multiple I/O modules and the issue is how the processor will determine which device issued the interrupt. Secondly, if multiple interrupts have occurred how the processor can decide which interrupt to process? To solve this, there are four categories of techniques that can be used.

- **Multiple Interrupt Lines**

The straightforward approach is to provide multiple interrupt lines between the processor and the I/O modules. This allows multiple modules to be handled at the same time. However, it is not practical to assign many bus lines and processor pins to interrupt lines. One of the reasons is that there might be more than one I/O module attached to a single line. This defeats the purpose of this technique.

- **Software Poll**

Whenever an interrupt is detected by the processor, it branches to an interrupt service routine which will poll each and every I/O module to determine the exact interrupting module. The processor raises a poll which could be in the form of a command line. The address of the respective I/O module which is interacted by the poll will be then placed on the address line. The module will respond positively if it is responsible for setting the interrupt. Alternatively, every I/O module has an addressable status register which can be read by the processor to determine the interrupting module. The drawback of this technique is that it is time consuming, as the processor

has to poll all the devices connected to it one after another or in some specified order.

- **Daisy Chain (Hardware Poll)**

Daisy chain is a hardware poll. Whenever there is an interrupt, the processor sends out an interrupt acknowledge which will propagate throughout the series of I/O modules. This process will continue until it reaches a requesting module which will then respond by placing a word on the data lines. The processor subsequently directs the module to its specific device-service routine. This method is also known as vectored interrupt.

- **Bus Arbitration**

This method involves the I/O module gaining control over the bus before requesting for the interrupt. It is limited to only one module at a time. The processor sends an acknowledge signal whenever it detects an interrupt. The requesting module then places its data on the data lines.



CHECK YOUR PROGRESS

Fill in the blanks:

Q1. The _____ works as the mediator between the I/O devices and the CPU.

Q2. It is improper to use a slow peripheral on a _____ system bus due to the speed mismatch between the CPU and the peripherals.

Q3. In programmed I/O, the CPU issues a _____ and then waits for I/O operations to be complete.

Q4. The processor must continuously check the status of the I/O module in _____.

Q5. The interrupt I/O technique requires more ____ hardware and software.

Q6. _____ allows multiple modules to be handled at the same time.

Q7. Every I/O module has an addressable _____ which can be read by the processor to determine the interrupting module.

Q8. Daisy chain is a _____ poll.

Q9. _____ is limited to only one module at a time.

Q10. The CPU, in programmed I/O, must repeatedly check the status of the I/O module, and this process is known as _____.

7.4.3 Direct Memory Access (DMA)

We have till now discussed two different methods of data transfer which requires active intervention of the CPU to transfer data between the memory and the I/O module. However, both these two approaches suffer from two drawbacks.

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

Thus, to transfer large blocks of data at a high speed, a special control unit may be provided to allow the transfer to be direct between the main memory and the I/O device. This technique, where the data is directly transferred from the I/O device to the memory or vice versa without continuous involvement of the processor, is called Direct Memory Access or DMA. The CPU has just to initiate the transfer by initializing some parameter in DMA transfers. It is a sophisticated I/O technique in which a ***DMA controller*** replaces the CPU and takes care of both the I/O and the memory. By using DMA, the data transfer rates became faster. The parameters that are to be initialized by the CPU in DMA transfer are:

- Starting address of data from or to which the data are to be transferred;
- number of memory words (count) to be transferred whether it is a read or write operation.

The figure 5.3 below shows how the transfer takes place using a DMA Controller.

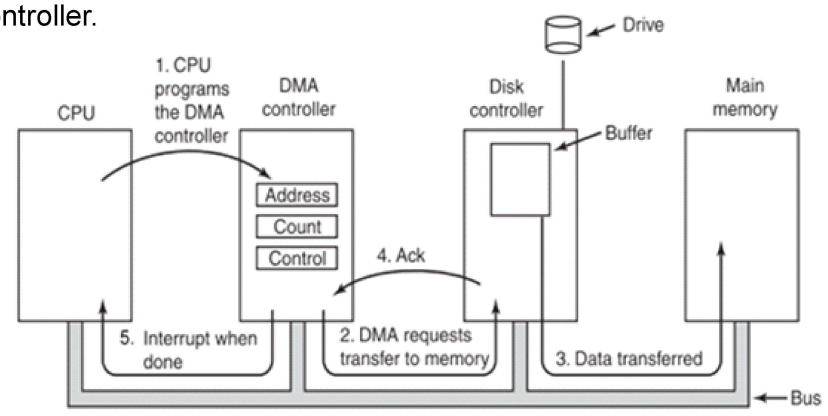


Figure 5.3: Operation of a DMA transfer

Step 1: The CPU first programs the DMA controller by setting its registers so that the DMA controller knows what to transfer. It also issues a command to the disk controller telling it to read data from the disk into its internal buffer and verify the checksum.

Step 2: The DMA controller initiates the transfer by issuing a read request over the bus to the disk controller. This read request looks like any other read request, and the disk controller does not know whether it came from the CPU or from a DMA controller.

Step 3: The memory address where the data is to be written is on the bus' address lines; so, when the disk controller fetches the next word, it knows where to write the word.

Step 4: When the write is complete, the disk controller sends an acknowledgement signal to the DMA controller. The DMA controller then increments the memory address to use and decrements the byte count. If the byte count is still greater than 0, steps 2 through 4 are repeated until the count reaches 0.

On completion of the transfer, the DMA controller interrupts the CPU to let it know that the transfer is complete. System performance improves by separate processing of the transfers to and from the peripherals. For example, between camera memory and USB port.

7.5 PRIORITY AND DAISY CHAINING TECHNIQUE

We have already come to know that at **Interrupt driven I/O,** priority of the Interrupt signal will play a vital role. Now, the question is: what is priority of signal? Priority means the weightage or importance. If the priority of the interrupt signal is more than the current execution, then the processor is to stop the current execution and to show response to the recently coming interrupt request. The processor will continue its current execution if the priority of the current execution is more than that of interrupt signal. For a particular processor, different interrupt signals have different types of priority. For example, Intel 8085 processor has five number of interrupt signals, namely, INTR, RST5.5, RST6.5, RST7.5 AND TRAP. There is an interrupt acknowledgement whose name is INTA. Among these five interrupt signals, INTR has the lowest priority and trap has the highest priority.

Say, the processor is receiving INTR and TRAP interrupt request at a time from device 1 and from device 2 respectively as shown below:

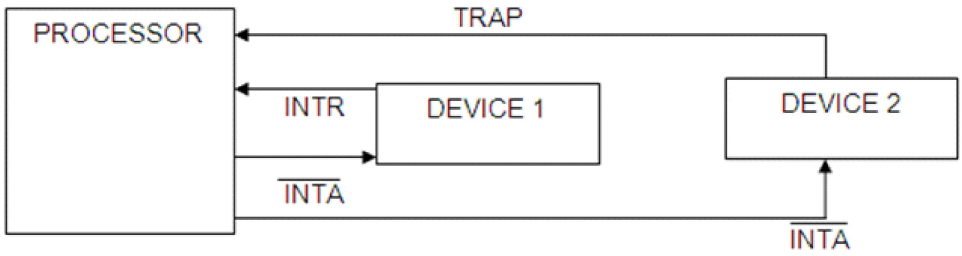


Figure 7.4: Schematic Representation of Interrupt signal INTR and TRAP

Then the processor will compare the priorities of INTR and TRAP with the priority of the current execution. If both INTR and TRAP have more priority than the current execution, then it will again check the priority between the two interrupt requests. And the processor will give response to the device 2 as TRAP has more priority than INTR sent by device 2.

But what will happen if both the devices send the interrupt requests having the same priority, say INTR! Then the concept of Daisy Chaining will come. In Daisy Chaining technique, arrangement will be as shown below:

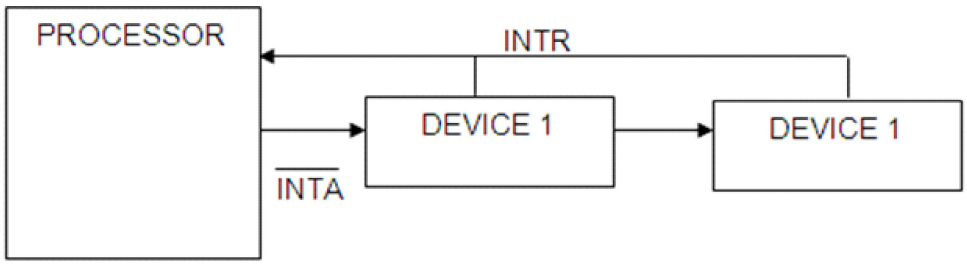


Figure 7.5: Schematic Representation of Interrupt signal INTR only

In this type of situation, interrupt acknowledgement will receive device 1 at first. That means, in daisy chaining technique the device which is electrically closest to the processor will acquire the highest priority. If device 1 has pending request then it will not pass the INTA signal to the device 2 until the completion of the pending task. If device 1 has no pending request then it will pass the INTA to the device 2.



CHECK YOUR PROGRESS

Fill up the blanks:

Q11. To transfer large blocks of data at a high speed, the technique used is called _____.

Q12. The _____ initiates the transfer by issuing a read request over the bus to the disk controller.

Q13. When the write is complete, the _____ sends an acknowledgement signal to the DMA controller.

Q14. If the priority of the interrupt signal is more than the current execution, what the processor will do?

Q15. Write the name of Interrupt signals of a typical processor, say Intel 8085. Arrange them in ascending order according to their priority.



7.6 LET US SUM UP

- The different I/O techniques are categorized into three types -
  - Programmed Input/Output
  - Interrupt driven Input/Output
  - Direct memory Access.
- The I/O module works as the mediator between the I/O devices and the CPU.
- The CPU issues a command and then waits for I/O operations to be complete.
- The CPU, while waiting, must repeatedly check the status of the different I/O modules in some order and this process is known as Polling.
- Interrupt driven I/O is another technique which deals with I/O operations.

- The problem with Programmed I/O is that the processor has to wait for a long time for the concerned I/O module to be ready for either reception or transmission of data.
- The CPU has to issue a command to the I/O module and to proceed to do some other useful task until the device requesting the service sends an interrupt to the CPU when it is ready to exchange data
- To transfer large blocks of data at a high speed, a special control unit may be provided to allow the transfer to be direct between the main memory and the I/O device.
- During priority technique, the processor will give response to the device which sends interrupt signal having more priority.
- In daisy chaining technique the device which is electrically closest to the processor will acquire the highest priority.



7.7 FURTHER READINGS

1) Stallings, W. (2000). Computer organization and architecture: designing for performance. Pearson Education India.
2) Chaudhuri, P. P. (2008). Computer organization and design. PHI Learning Pvt. Ltd..
3) Godse, A. P., & Godse, D. A. (2008). Computer Organization and Architecture. Technical Publications.



7.8 ANSWERS TO CHECK YOUR PROGRESS

Ans to Q No 1: I/O module
Ans to Q No 2: high speed
Ans to Q No 3: command
Ans to Q No 4: programmed I/O
Ans to Q No 5: Complex
Ans to Q No 6: Multiple Interrupt Lines
Ans to Q No 7: status register
Ans to Q No 8: Hardware
Ans to Q No 9: Bus Arbitration

Ans to Q No 10: Polling

Ans to Q No 11: Direct Memory Access

Ans to Q No 12: DMA controller

Ans to Q No 13: disk controller

Ans to Q No 14: Processor is to stop the current execution, and to show response to the recent coming interrupt request.

Ans to Q No 15: INTR, RST5.5, RST6.5, RST7.5 AND TRAP. There is an interrupt acknowledgement which name is INTA. The ascending order according to their priority is as follows:

INTR< RST5.5< RST6.5< RST7.5< TRAP



7.9 MODEL QUESTIONS

Q1. Explain the different I/O techniques used in the computer.

Q2. What is Programmed I/O? What are its advantages and disadvantages?

Q3. Explain the differences between the Programmed I/O and Interrupt driven I/O.

Q4. What are the two design issues in interrupt driven I/O implementation? How can these be solved?

Q5. Explain the direct memory access.

Q6. What is the main disadvantage of polling technique? Explain the functional characteristics of polling technique.

Q7. What is Interrupt signal? Write about the concept.

Q8. Write about the functional characteristics of Interrupt driven I/O.

Q9. Differentiate between priority and daisy chaining technique. Explain in detail.

****