UNIT 2: I/O MANAGEMENT

Unit Structure

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- Concept of I/O devices
- About Bus Architecture
- Detailed regarding features of DMA controlled I/O
- Basic of Input Output Programmed
- Idea of DMA Channels

2.1 Introduction

Management of I/O devices is one of the important parts of an operating system. It is so important and varied that the entire I/O subsystems will be focussed particularly in its operation. On considering certain devices such as mouse, keyboards, disk drives, display adapters, USB devices, network connections, audio I/O, printers etc., we find that Input/Output subsystems will work on following principles:

- The focus of using devices that gets attached will help to get new developed devices for old systems
- The creation of latest devices that gets interfaced with original standard which are not easy and compatible.

Further we see that for every hardware device there is a device driver which works in support of Operating System which will handle the complete hardware.

Goals for I/O

- Users will be able to access all devices in uniform manner.
- Devices to be named in particular order.
- Operating System without the interruption of user program can able to control recoverable errors.
- Operating System should control security of the devices.
- Operating System to optimize performance of I/O system.

2.2 I/O Hardware

If we talk about I/O devices, we mean to say as storage, communications, and user-interface devices that work with computer by using signals which often can be trough wires or by air. These devices get attached with the computer through different ports which can be serial or parallel. It is seen a common set of wires that will connect many such devices is known as bus. The buses in computer architecture cover rigid protocols for certain different messages that can be sent across the bus and procedures in order to solve conflict issues.

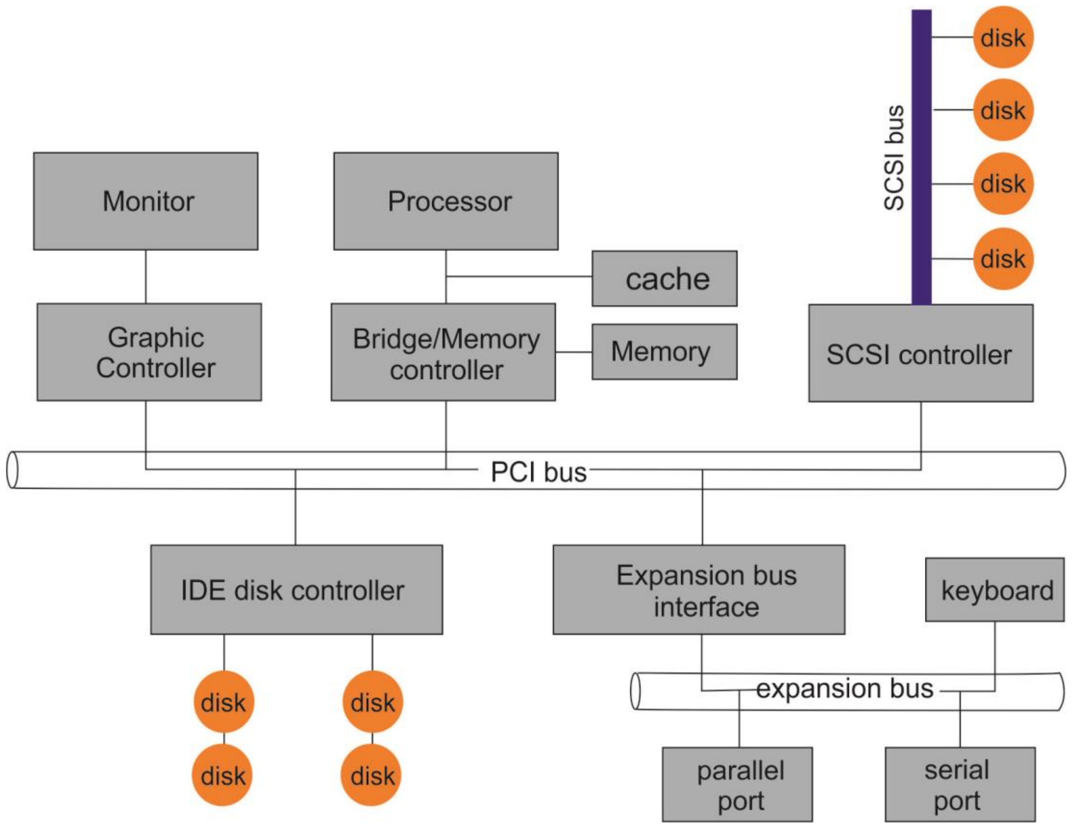


Fig 2.1 Buses

Figure 2.1 shows three of four types of buses that are usually found in modern computers:

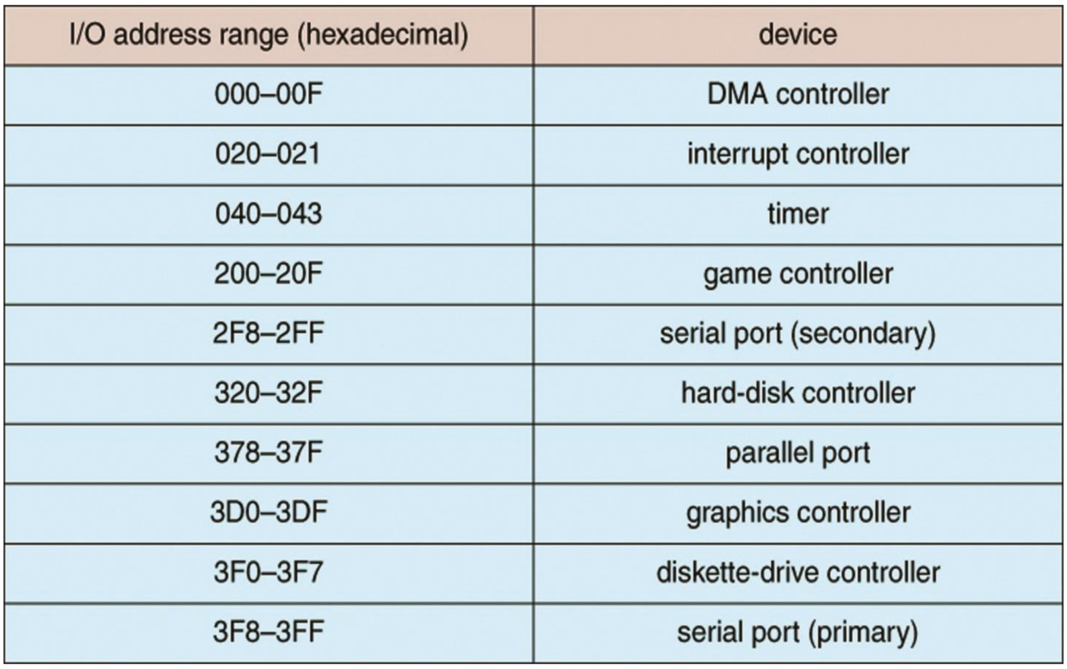
- PCI bus that connects high speed having high band width devices to the memory subsystem and the processor or CPU.
- Expansion bus will connect low band width devices that normally transmit data single character at a particular time by use of buffering.
- SCSI bus which connects many SCSI devices to particular SCSI controller.
- Daisy chain bus is a type of bus that will show when a string of devices is connected each other as beads on chain where only single device is directly connected to host computer.

It is seen that a particular way of communicating with devices is using registers that is connected with each port. Registers can be one to four bytes in size which includes a subset of:

- Data-in register which is read by host to have input from device.
- Data-out register is that which is written by host that sends an output.
- Status register having bits works by host to have a position of certain device to feel as idle, ready, busy, error, etc.

- Control register have bits applied by host which determines commands that can also alter settings of various device.

Apart from Input Output, there is memory-mapped I/O, which also communicates with computer which is shown in fig 2.2.



I/O address range (hexadecimal)	device
000–00F	DMA controller
020–021	interrupt controller
040–043	timer
200–20F	game controller
2F8–2FF	serial port (secondary)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)

Fig 2.2 Device I/O port locations on PCs

In memory mapped I/O several processor's address space parts are mapped with device where communications is done through reading as well as writing directly from the memory location.

It is beneficial for devices which travel with large quantities of data quickly. These I/O are application with additional mixture of original registers. It is seen that possibly the problem with memory-mapped I/O, is that a process is allowed to write straight to address space that are used by memory-mapped I/O device.

Check your progress 1

1. Which among the following is a category of I/O device?

a. storage	c. user-interface
b. communications	d. all

2. Buses are set of________.

a. rules c. information

b. protocols d. data

3. _____bus connects high speed bandwidth devices to memory subsystem and processor.

a. PCI bus c. SCSI bus

b. Expansion bus d. Daisy Chain

2.3 I/O Drivers

The CPU is not the single brilliant device in the system; every corporal device acquires its own hardware controller. The keyboard, mouse as well as serial ports are controlled proximate a Super IO chip, the IDE disks nearby an IDE controller, SCSI disks nearby a SCSI controller additionally so on. Each hardware controller holds its own control as well as status registers (CSRs) furthermore these contrasts between devices. The CSRs for an Adaptec 2940 SCSI controller are entirely differing from those relevantly an NCR 810 SCSI controller. The CSRs are utilized to start as well as stop the device, to begin it furthermore to diagnose several dilemmas with it. Instead of embedding code to coordinate the hardware controllers in the system into every exercise, the code continues conserved in the Linux kernel. The software that triggers or commands a hardware controller is comprehended as a device driver. The Linux kernel device drivers are, centrally, an assigned library of excepted, memory incumbent, bottom level hardware experiencing routines. It is Linux's device drivers that experience the features of the devices they are administering.

One of the elementary features of subsists that it evacuates the responding of devices. complete hardware devices observe like average files; they can be exposed, closed, read as well as written applying the identical, standard, system calls that are exercised to touch files. Every device in the system is circumscribed nearby a device definite file, for exemplary the early IDE disk in the discipline is represented by /dev/hda. It is found that for block (disk) as well as character devices, such device serves as particular files that are created by mknod command which can be describe as device that uses more and less device numbers. Network devices are shown with the help of special files since they are developed by Linux, as it locates and initialize network controllers in system. Such devices controlled by single device driver have common major device number. The minor device numbers are used to differentiate among various devices and their controllers having each partition on primary IDE disk contains different minor device number. So, we see that in /dev/hda2, the second partition of primary IDE disk contains major number 3 and minor number 2. In this, Linux maps the device special file that is passed in system calls to device driver that uses major device number and number of system tables as character device table labelled as chrdevs.

Device driver is a sort of a program which is constructed for I/O device. These derivers will make use of I/O operations on certain devices. Here the arrangement will take care of number of certain terminals that contains dissimilar terminal driver. The work of device driver is to:

- Take the request from individual software.
- Manage the request send by the device.

In order to handle a request, a device driver will take care of the request as:

- If the request appears to read block N and the driver is ease at said time, then it will process the request immediately
- If the driver is busy, then the request so appears will be placed in the queue.

Check your progress 2

1. Drivers are the________.

a. data	c. information
b. program	d. all

2.4 DMA Controlled I/O

In order to exercise an interrupt obtained from device drivers, the data is assigned from the hardware which will work correctly if the data appear is less. If the interrupt is not recognised, then number of time it will achieve in hardware device gets shoot up the interrupt with routine occurrence with result in stumpy transferring of data. In case of high speed devices such as hard disk or Ethernet, the data transfer rate will be much higher. If you see the data transfer rate in case of SCSI device, you will find that it will be above 45 Mbytes of data transfer per second.

To see such variation in data transfer, Direct Memory Access is applied will be breakthrough such type of problems. As seen, a PC's ISA DMA controller will carry 8 DMA channels out of which, 7 channels are feasible by device drivers. In order to begin with data transfer, device driver will allot DMA channel's address along with count registers in the direction of data transfer that will read or write data. During that period, the device will start the DMA when it is required.

It is found that while conducting DMA, the device drivers needs to be careful. Initially, the DMA controller will know nothing about virtual memory that has appeared singly in the system. With this effect, the memory will go through continuing DMA from required contiguous block of physical memory which explains that the DMA cannot be directly placed inside virtual address space.

It is seen that the DMA channels are restricted, as they possess only 7 channels out of 16 which are difficult while sharing among device drivers. Similarly like interrupts, the device drivers will able to judge and manage which DMA channel should be applied. Many device drivers carry standard DMA channel, which is same in case of interrupts.

Linux operating system will takes care of all DMA channels applications with the help of vector dma_chan data structures. Such DMA data structure carry pointer which shows presence of DMA channel owner and a flag that will reflect on allocation of DMA channel.

- The devices which transfer big data will be of no use of data transfer in CPU during data input to registers.
- Such type of work can be carried off by Direct Memory Access and Controller.
- Initially the command is sent to DMA controller with data storage location and data transfer with number of bytes of data to transfer.
- DMA controller is independent component of computer where different bus mastering I/O cards has their own DMA hardware.
- Handshaking exist among DMA controllers and certain devices will be done through double DMA request and DMA acknowledge wires.

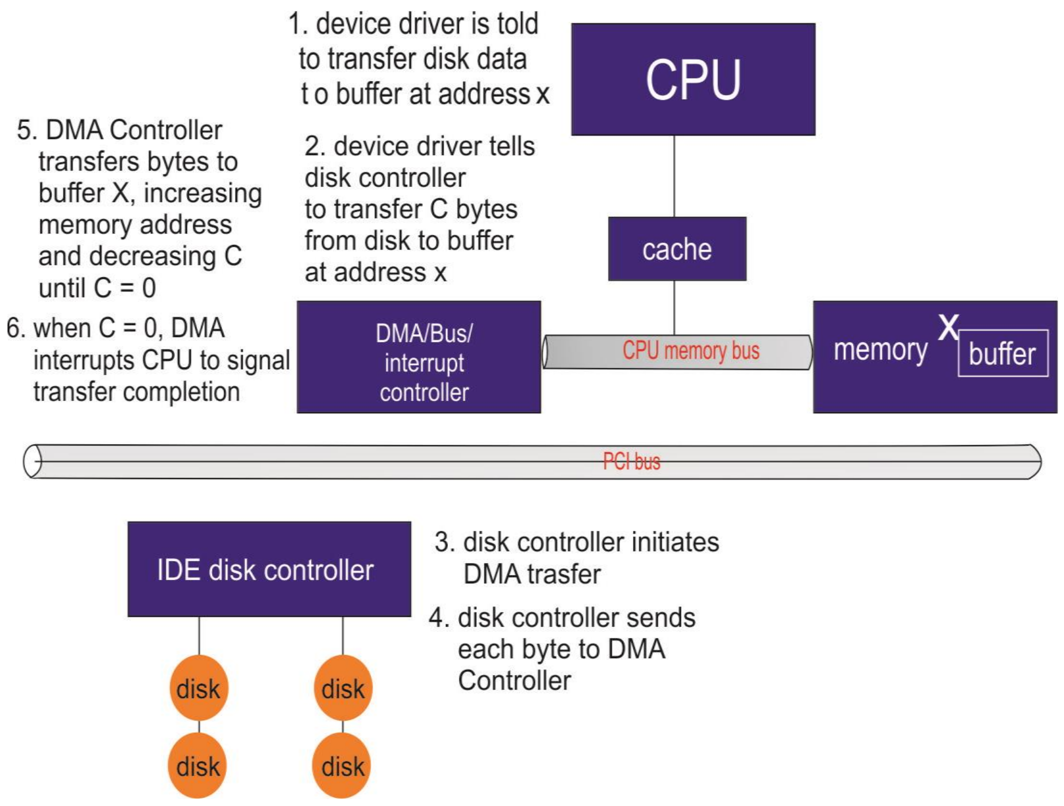

1. device driver is told to transfer disk data t o buffer at address x
CPU
5. DMA Controller transfers bytes to buffer X, increasing memory address and decreasing C until C = 0
2. device driver tells disk controller to transfer C bytes from disk to buffer at address x
cache
6. when C = 0, DMA interrupts CPU to signal transfer completion
DMA/Bus/ interrupt controller
CPU memory bus
memory
X
buffer
PCI bus
IDE disk controller
3. disk controller initiates DMA trasfer
4. disk controller sends each byte to DMA Controller
disk
disk
disk
disk

Fig 2.3 Illustrates the DMA process.

Check your progress 3

1. A PC's ISA DMA controller holds ______DMA channels.

a. 4

b. 8

c. 16

d. 32

2.5 Programmed I/O

The basic computer structure is shown in figure 2.4.

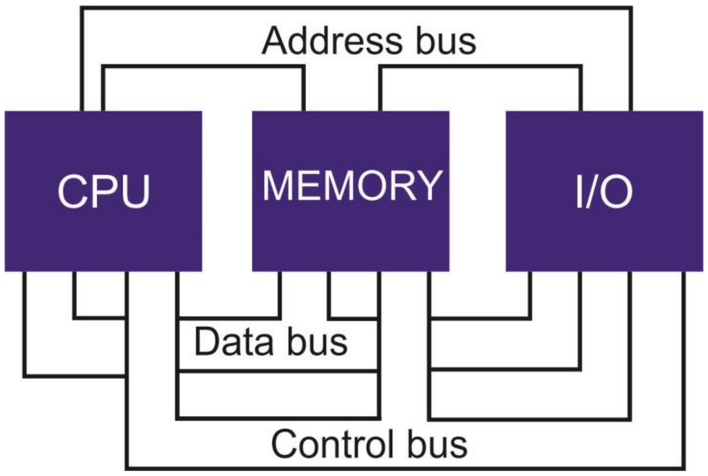


Fig 2.4 basic computer structure

Programmed I/O is the method of transferring of data done by the CPU in supervision with driver software control so as to make use of registers or memory on a certain device. Since in case of Direct Memory Access (DMA), the data is transferred through certain device in order to work with system memory. Similarly, the work of PIO is to take care of data transfer through normal memory that will be loaded and kept on others.

In case of UDI, the Programmed I/O working is done with the support of environmental service calls which is coded as function calls instead of direct memory references in certain drivers.

It is found that PIO will keep an unclear data type which will carry addressing, data translation as well as access restriction information which is needed to work device or memory address in certain address space.

In case of transaction which will work a device where PIO offset is shown will highlight the offset that carries a space which is referenced by PIO handle with which I/O operations occur.

Synchronization that exists among PIO transaction lists can be demonstrated with syntax serialization domain argument which will map PIO call. Working of PIO transaction list can be sequenced due to working of other PIO transaction also are mapped with certain device as for a particular device and serialization domain, at least one thread will be actively working corresponds to transaction list and these transaction list will work to complete before other transaction list at the start of serial domain execution.

In such cases, there will no ordering confirmation with regards to working of transactions lists having a syntax udi_pio_trans calls instead of the calls made from similar portion exist in similar serialization domain as processed in FIFO order.

Check your progress 4

1. Programmed I/O is basically data transfers by CPU under software control to access__________.

a. registers	c. files
b. data	d. all

2.6 I/O Supervisors

The operating systems will keep track of all input/output devices which are attached to a computer system.

Device drivers accompany operating systems and enable a computer system to be configured for specific hardware. Most hardware peripheral devices have their own device drivers which need to be installed for the operating system to communicate with these devices.

Check your progress 5

1. Device drives are need to__________.

a. run	c. copy
b. install	d. all

2.7 Let Us Sum Up

In this unit we have learned:

- That I/O devices are concerning with storage, communications and user-interface which will work and interface with the help of computer by using analog and digital signals which is there through wires or through air.

- The buses in computer architecture cover rigid protocols for certain different messages that can be sent across the bus and procedures in order to solve conflict issues.
- Drivers or Device driver is a program or routine that for designed for an I/O device.

2.8 Answers for Check Your Progress

Check your progress 1

Answers: (1-d), (2-b), (3-a)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-b)

Check your progress 4

Answers: (1-a)

Check your progress 5

Answers: (1-b)

2.9 Glossary

1. **I/O devices -** These can be storage, communications, user-interface devices that communicate by using the computer through signals that used to send by wires or through air.
2. **Buses -** Are protocols for certain different messages that can be sent across the bus and procedures in order to solve conflict issues.
3. **Drivers or Device driver -** Program or routine that for designed for an I/O device.

2.10 Assignment

Write short note on directory structure of Operating System.

2.11 Activities

Collect some information on I/O devices.

2.12 Case Study

Generalized the basic computer architecture and discuss.

2.13 Further Readings

1. Operating System Concept by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne.
2. Programming Be Operating System by Dan Sydow.
3. Computer Science & Application by Dr. Arvind Mohan Parashar, Chandresh Shah, Saurab Mishra.
4. An Integrated Approach to Software Engineering by Pankaj Jalote.
5. An Operating Systems by Raphael A.