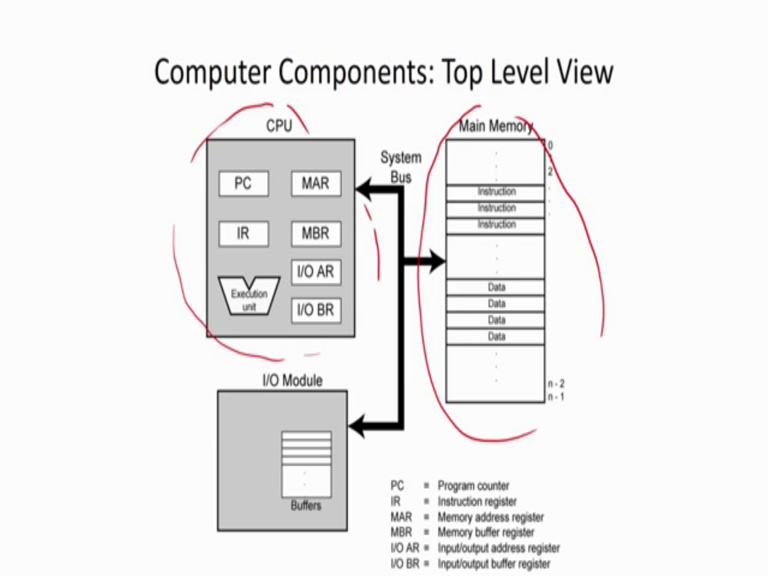
Computer Organization and Architecture: A Pedagogical Aspect

Prof. Jathindra Kr. Deka
Dr. Santosh Biswas
Dr. Arnab Sarkar
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 33
Input-Output Primitives

Hello everybody welcome back to the online course on Computer Organization and Architecture. So, in the introductory module we have discussed early.

(Refer Slide Time: 00:44)



Computer Components: Top Level View

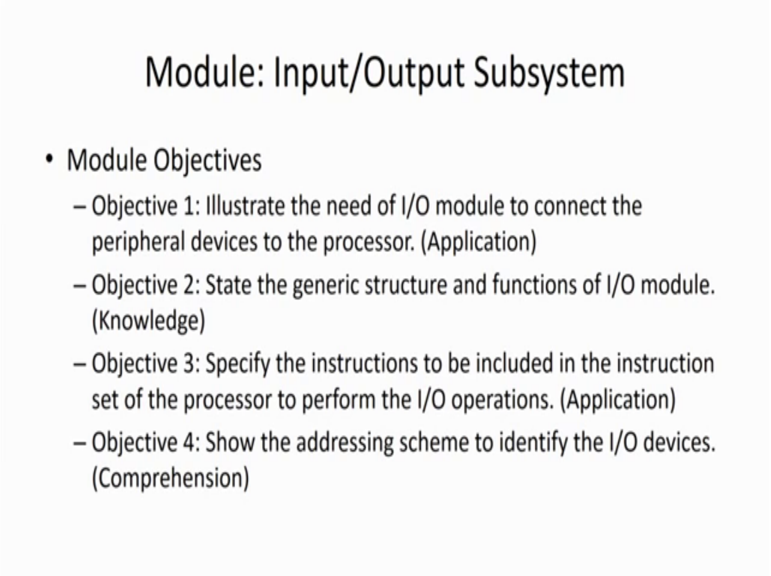
PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

I have shown you that mainly in top level you can see the computer component as follows; we are having the CPU or a central processing unit which is the main component and main processing element of the computer, computer works on Von Neumanns stored program principle. So, for that we need the memory unit ok.

So, till now I think we have discussed about the design issues of processor and we have seen how we are going to connect the main memory to the processor, what are the issues related the memory? Today we are going to look for the other component; that IO module, basically we are going to see how we are going to connect input output devices to computer. So, this module is basically input output subsystem.

So, in this particular module we are going to discuss about the issues related to input output devices and how those devices will be connected to the processor and how it works.

(Refer Slide Time: 01:40)



Module: Input/Output Subsystem

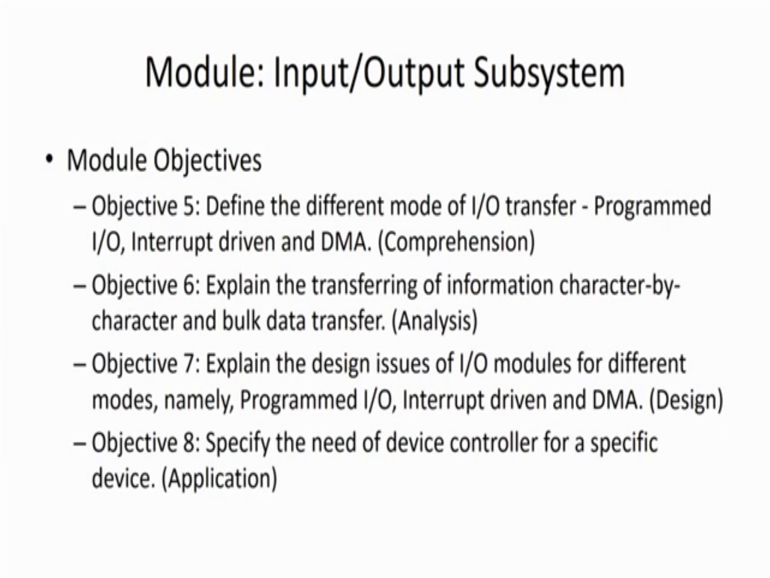
- Module Objectives
  - Objective 1: Illustrate the need of I/O module to connect the peripheral devices to the processor. (Application)
  - Objective 2: State the generic structure and functions of I/O module. (Knowledge)
  - Objective 3: Specify the instructions to be included in the instruction set of the processor to perform the I/O operations. (Application)
  - Objective 4: Show the addressing scheme to identify the I/O devices. (Comprehension)

So, as usual now we are going to see what are the objective of this particular module? So, we are going to meet some objective. So, first objective I have mentioned it like that, illustrate the need of IO module to connect the peripheral devices to the processor. So, this is the objective and we are going to touch it in application level. Objective 2, state the generic structure and function of IO module.

So, this is in the knowledge level and we will touch, we will see what is the structure of the IO module and what are the function of that particular IO module? Objective 3 specify the instruction to be included in the instruction set of the processor to perform the IO operation.

So, this is in application level, we are going to see what are the instruction that will be needed for IO operation. Objective 4; show the addressing scheme to identify the IO devices. So, they have been lot of input output devices connected to the processor. Now how we are going to identify those devices and basically what is the addressing scheme for those particular devices. So, this is in comprehension level.

(Refer Slide Time: 02:46)



Module: Input/Output Subsystem

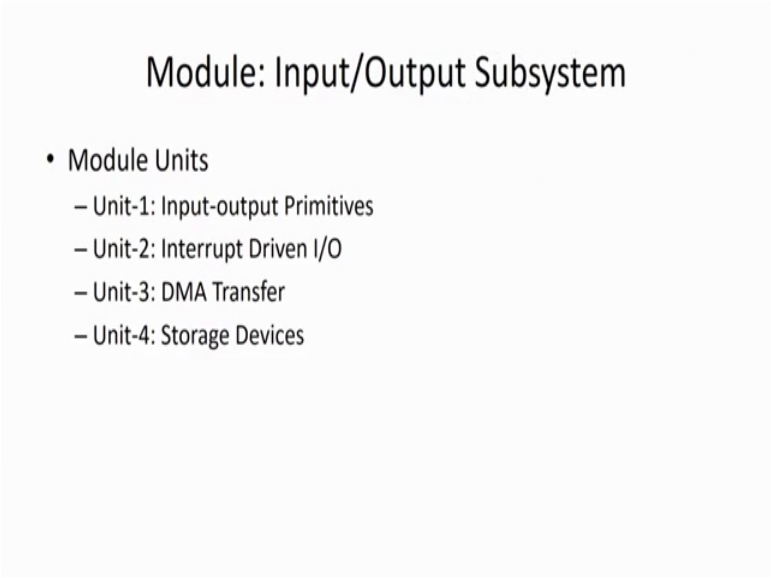
- Module Objectives
  - Objective 5: Define the different mode of I/O transfer - Programmed I/O, Interrupt driven and DMA. (Comprehension)
  - Objective 6: Explain the transferring of information character-by-character and bulk data transfer. (Analysis)
  - Objective 7: Explain the design issues of I/O modules for different modes, namely, Programmed I/O, Interrupt driven and DMA. (Design)
  - Objective 8: Specify the need of device controller for a specific device. (Application)

So, next objective is your objective 5; define the different modes of IO transfer programmed IO, interrupt driven and DMA. So, you having several ways to transfer operation and basically we have three modes. So, these are basically programmed IO interrupt driven and DMA. So, we are going to elaborate these things.

Objective 6; explain the transferring of the information character by character or bulk data transfers. So, if we are transferring information character by character like from keyboard ah, how to transfer a information in a bulk, so from one device to the other device.

Objective 7; explain the design issues of IO module for different modes; namely programmed IO, interrupt driven and DMA. So, these objective is in the design level. Now we are going to see what are the design issues for these three kind of transform mode. An objective 8 specify the need of device controller for a specific device. So, for every device you will have a device controller. So, here in application level and we will see will specify what is the need of those particular device controller. So, as usual this module input output subsystem is divided into 4 units.

(Refer Slide Time: 03:54)

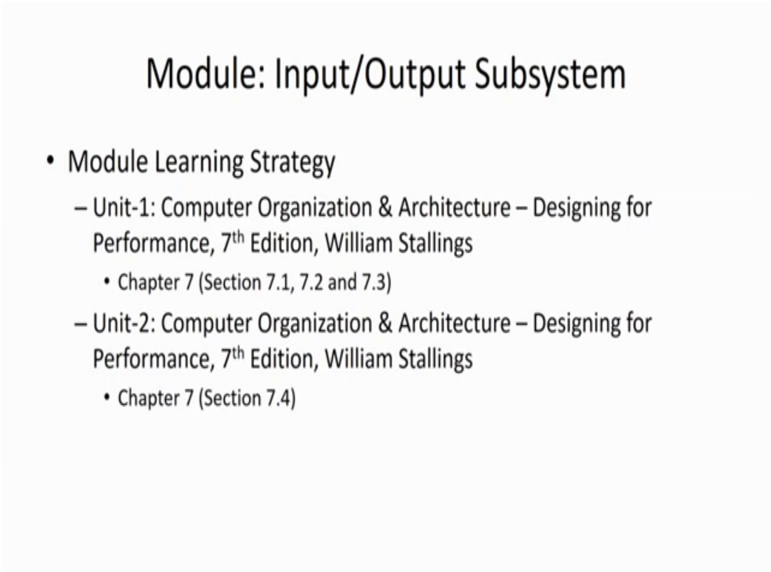


Module: Input/Output Subsystem

- Module Units
  - Unit-1: Input-output Primitives
  - Unit-2: Interrupt Driven I/O
  - Unit-3: DMA Transfer
  - Unit-4: Storage Devices

So, the module units is; so, the module units for this particular module, so unit 1 input output primitives, unit 2 interrupt driven IO, interrupt tree or unit 3 DMA transfer and unit 4 storage devices. So, basically we, having this particular 4 units.

(Refer Slide Time: 04:15)

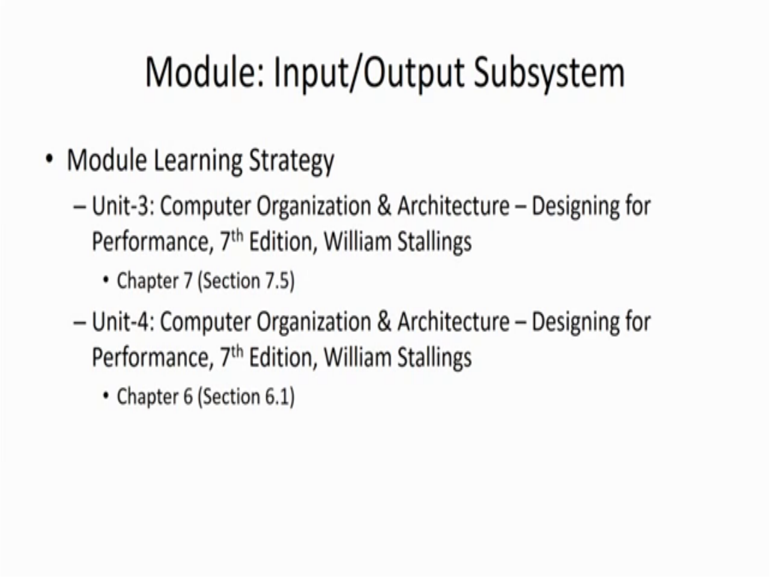


Module: Input/Output Subsystem

- Module Learning Strategy
  - Unit-1: Computer Organization & Architecture – Designing for Performance, 7th Edition, William Stallings
    - Chapter 7 (Section 7.1, 7.2 and 7.3)
  - Unit-2: Computer Organization & Architecture – Designing for Performance, 7th Edition, William Stallings
    - Chapter 7 (Section 7.4)

Now, what is a module learning strategies? basically what are the resources or what a reference material for this particular course, for this particular module. So, for unit 1 you are going to use that same book Computer Organization and Architecture design for performance by William Stallings. So, you need to look for the section 7.1, 7.2 and 7 1 3 of section chapter 7. For unit 2 you are going to use the same reference book of same chapter. So, basically you need to look for section 7.4 of chapter 7 of this particular book.

(Refer Slide Time: 04:48)



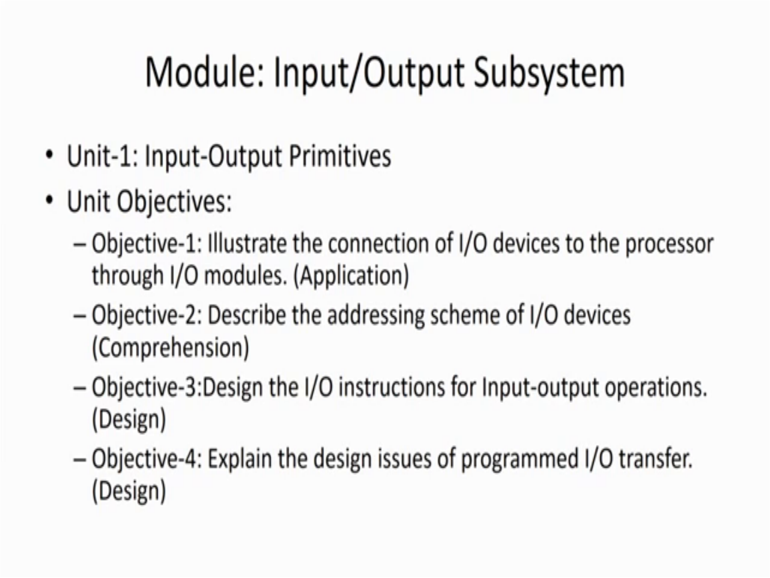
Module: Input/Output Subsystem

- Module Learning Strategy
  – Unit-3: Computer Organization & Architecture – Designing for Performance, 7th Edition, William Stallings
    - Chapter 7 (Section 7.5)
  – Unit-4: Computer Organization & Architecture – Designing for Performance, 7th Edition, William Stallings
    - Chapter 6 (Section 6.1)

For unit 3, this is the section 7.5 of chapter 7 of the same book and for unit 4 this is the same book that we are using Computer Organization and Architecture Designing for Performance by William Stallings and for that we are going to look for the section 6.1 of chapter 6. So, these are a reference material that you can look into it, you can go through it to for better understanding of this particular course.

So, for this module input output subsystem. Now we are going to start the first unit and unit 1 is your input output primitives. So, the objective that I have defined for this particular unit is as follows. So, objective 1; illustrate the connection of IO devices to the processor through IO modules.

(Refer Slide Time: 05:30)



Module: Input/Output Subsystem

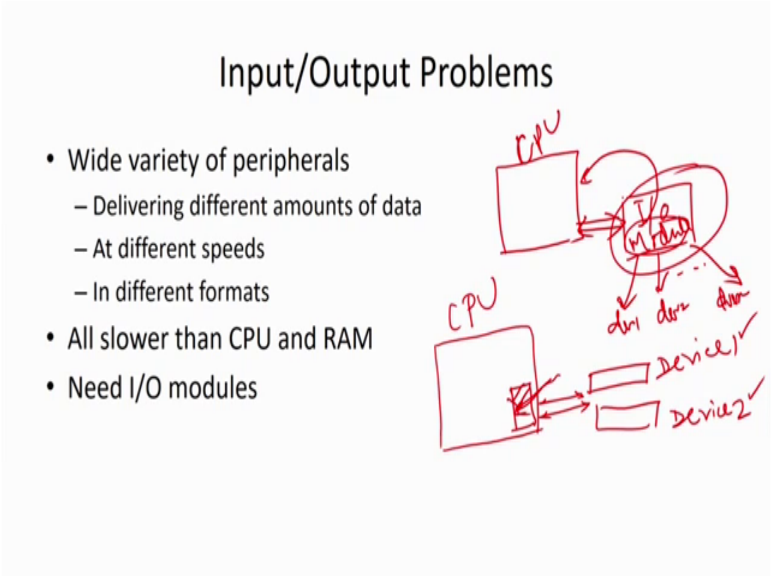
- Unit-1: Input-Output Primitives
- Unit Objectives:
  - Objective-1: Illustrate the connection of I/O devices to the processor through I/O modules. (Application)
  - Objective-2: Describe the addressing scheme of I/O devices (Comprehension)
  - Objective-3:Design the I/O instructions for Input-output operations. (Design)
  - Objective-4: Explain the design issues of programmed I/O transfer. (Design)

So, this is an application level. Now we are going to see how that IO devices will be connected to the computer through the IO modules. Objective 2 describe the addressing scheme of IO devices. So, we are going to meet this objective also. So, how to specify the address of that particular device; objective 3 design the IO instruction for input output operation.

So, basically what will happen, when we are going to design a processor in the instruction set you need to give some instruction to handle the input output devices or basically to perform the input output operation on those particular devices. So, for that what are the instruction is required. So, we are going to explain those things and we are going to give emphasis on that and object 4 explain the design issues of program IO transfer. So, this is in a design level.

So, I have mentioned that there are four ways, three ways to transfer a information. So, we are going to discuss about the program IO transform and we are going to make this particular objective..

(Refer Slide Time: 06:34)



Input/Output Problems

- Wide variety of peripherals
  - Delivering different amounts of data
  - At different speeds
  - In different formats
- All slower than CPU and RAM
- Need I/O modules

Now we are going to discuss about the input output subsystem; first of all let us see what are the problem involved in this particular input output devices? So, basically if you see that we are going to get a wide varieties of peripheral devices. So, these are different peripheral devices you are going to connect. So, you can see that we are having keyboard, we are having mouse we are having printer, monitor, hard disk CD drive all those things. So, these are having a priorities of your devices.

So, those devices are not uniform at all, they are having define data format, they are delivering the, data at different speeds. So, again the speed matters, because if we are going to connect the slow devices to the processor, then we have to synchronize that particular slower devices. If the format of the data is different and we have to bring it to an uniform format. So, all those issues are there. So, for that we are having some problems that is why you have to need to look for this particular handling IO devices.

And basically already I have talked about the speed and all are basically slower than CPU and RAM. Already we have mentioned that CPU is the faster one than the RAM, RAM is slower than CPU anyways RAM is the primary memory or the storage unit. When we come to the peripheral devices or input output devices they are much slower. It is not only electronic devices, but still they are having mechanical component also, we have to have the, we have to consider the movement of the mechanical component.

So, as for example, I think most of you have used the dot matrix printers. So, when you are using dot matrix printer we have to print the character by character and for that we have to move the head. When we are moving the printing head you just see that, it is having a mechanical movement. So, we having mechanical component. So, you have to control those mechanical movement also and when you are going to control this mechanical movement, it becomes slower.

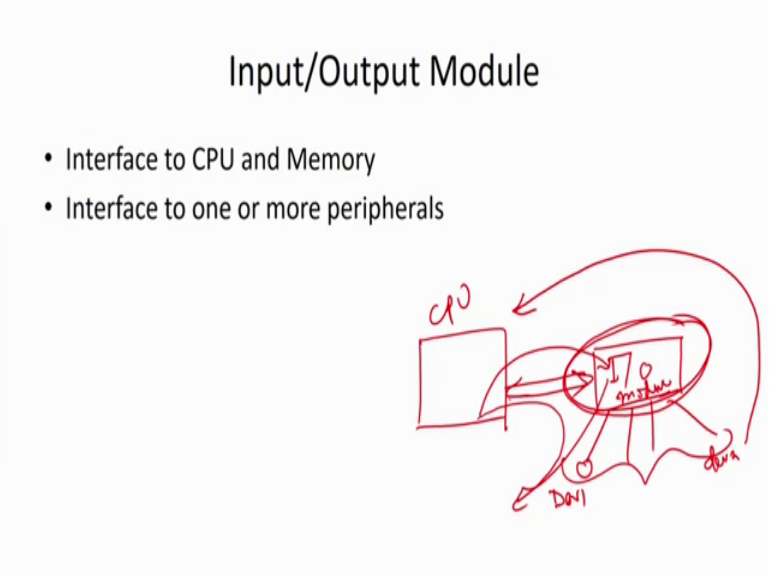
So, since we are having a varieties of devices, they transferred a information at different speed, the working principle if is different for that we need this particular IO modules input output module. So, what basically you are doing say, if you consider that this is my processor CPU; then what will happen? Directly I can connect the devices ourselves. This is device 1, so this is device 2.

So, if you follow this particular connection or for a design procedure then what will happen? To control those particular devices all the controls are to this for those devices need to be included in the processor itself. Since we are having priorities of devices, so this circuit will become more complex and it will become bigger and bigger, because you have to get to all the peripheral devices.

So, instead of pushing it inside this particular processor. So, why this, what are a design methods are those we are following. So, we are having this particular processor CPU, then we are connecting this particular IO module to system bus and all the devices we are connecting to this particular IO module. So, to perform those particular set device 1, device 2 like that device n. So, there functionalities are different, formats of data is a difference.

So, all those things will be handled by this particular IO module and finally, this IO module is going to transfer the information to the processor. So, that. So, I we need the IO module we cannot put everything inside the processor, then the design issues of processor will become very complex. So, you are simply being, bringing it out from the processor and putting it 1 unit called IO module. So, this is the need for the IO module.

(Refer Slide Time: 10:43)



Input/Output Module

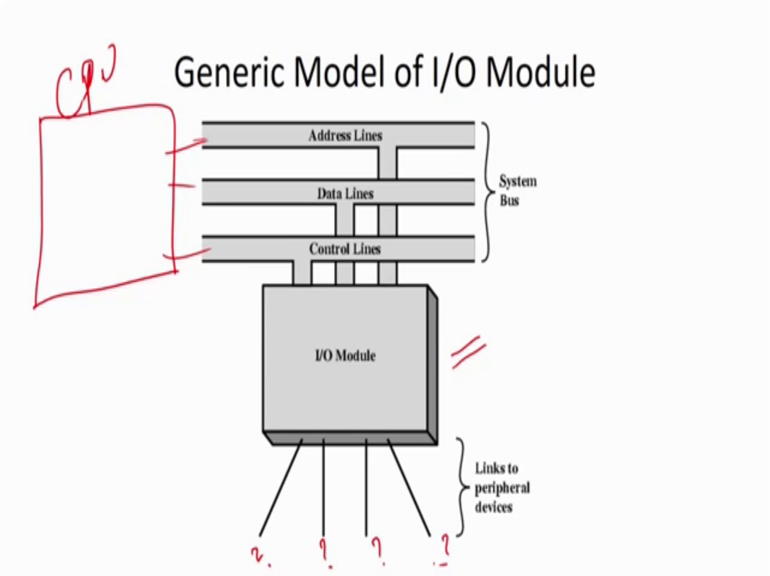
- Interface to CPU and Memory
- Interface to one or more peripherals

So, what that IO module does this is nothing, but I interfaced the CPU and memory and interface to one or more peripheral devices. So, we see the what will happen, already I have mentioned that this is the processor CPU. So, you are connecting the IO modules to the processor and to it we are connecting that devices and this is my device one, this is my device one. So, now, this IO module can now connect or it can (Refer Time: 11:19)

So, more of the, more number of devices and after that it is going to just works as an interfacing unit between your processor and devices. Well we are going to transfer information from devices to processor. Then what will happen that IO module is going to collect information from this particular devices, then IO module is transfer it from this buffer of the module to the processor.

So, when we want to transfer something to the output devices, IO module is going to take the information from the processor and finally, IO module will transfer it to the output devices. So, this is act as a interfacing between processor and IO module.

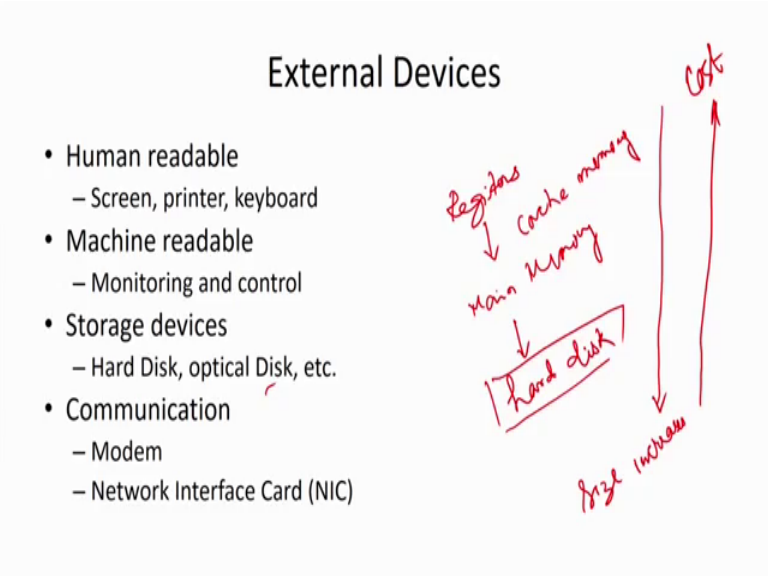
(Refer Slide Time: 12:01)



So, this is the where that you contain that this is where, you are having the system bus in system bus we having three component address bus, data bus and control bus. So, this system bus now we are having a processor. Now this CP is connected to this particular system bus and through this particular system bus we are connecting this particular IO module. And to this IO module now we are going to connect several devices.

So,. So, the IO module is going to act as an interfacing between input output devices and the processor, until we connected through this particular system bus. Now, what are the external devices that we are going to connect. So, the external devices now we concerns; one parties or one categories your human readable, so that we can at least read it or we can understand what basically happening through the computer.

(Refer Slide Time: 12:48)



External Devices

- Human readable
  - Screen, printer, keyboard
- Machine readable
  - Monitoring and control
- Storage devices
  - Hard Disk, optical Disk, etc.
- Communication
  - Modem
  - Network Interface Card (NIC)

Registers
Cache memory
Main memory
hard disk
Cost
Size increase

So, like that screen. If you are displaying something or say if you press some keys in the keyboard then that collector will be displayed in the screen. Similarly printer, so if we are storing something in our hard disk. Now we can transfer to the printer and we can print it. So, these are basically human readable devices; like screen, printer, keyboard and like that.

So, we are having some devices which are machine readable. So, this machine readable devices are basically used for monitoring and controlling purposes. So, in this particular case I can give a simple example. So, when we have using our computer at to switch on a machine or to work with a machine, sometimes we used to give password. So, when you enter a password then only you can enter into the machine and you can work with the machine.

Now, this is a password, it is a string of characters generally we use, but instead that what we can do. We can use some devices also some other features also to, unlock the computer like one simple example is your fingerprint. So; that means, what will happen we can lock our machine with the help of fingerprint. Now when we are going to unlock it at least you have to give our fingerprint to it; that means, we have to connect a biometric devices to the computer. So, this is that fingerprint recognition.

So, this is something like your machine readable devices; like that we can uh, we can get many more examples of the machine readable devices the third category we have talking about the storage devices. So, most of you know that you are having hard disk and many a time you said that the capacity of your hard disk is maybe your 500 GB or 1 terabyte.

So, what basically you are doing? You are storing our information in hard disks. So, when we have going to work with a computer, we bring the information from hard disks to the main memory and processor is going to take the information from main memory and it will be going to put into the registers that we have inside the processor.

So, in that particular case, so when you are talking about the storage or memory. So, now, you see that, we can have a hierarchy of the memory. So, first we can talk about the registers inside a processor. Then next level in is your main memory, then next level is you can say hard disk ok. So, this is the memory hierarchy and I think now, after going through this particular course, when you have gone through this memory module,then I think you have encounter with one another kind of memory which is known as your cache memory.

So, in the hierarchy cache memory will come after the register. So, top level hierarchy is your registers then cache memory, main memory and hard disk. If you see these things then what will happen. If you go from this top to bottom then what will happen in that particular case, the size increases ok. Basically, so we having at in few limited number of registers, it may be 8 16 or maybe 32, but when we are coming to the cache memory it is slightly bigger, so we can have that some Mbs of your cache memory.

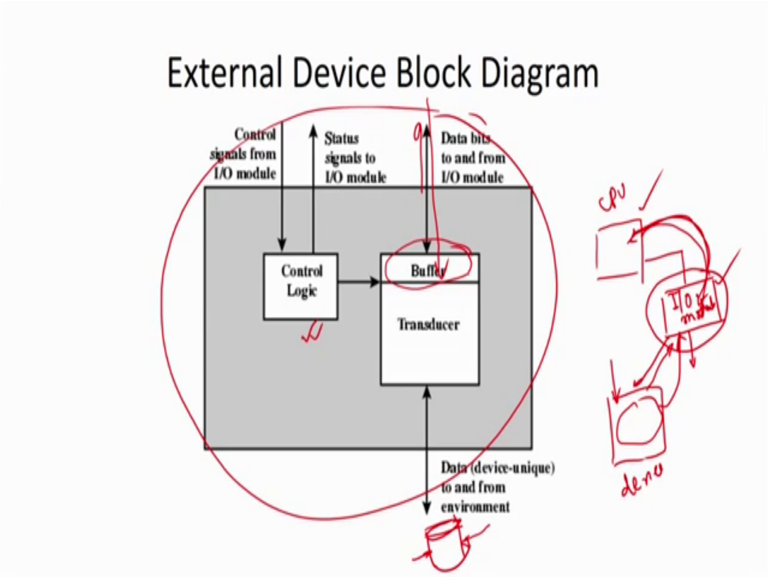
Now, when we come to the main memory, you will find that in main memory you having 2 Gb or 4 Gb many memories nowadays, but when you come to the hard disk, you will find that you are having a abundant capacity, maybe around 300 Gb 500 Gb even terabyte also. So, size increases when you go from this particular register to the external devices like hard disk. So, these are the things you. Now in the second issues when you go in this direction then cost increases. So, cost per unit memory is increases.

So, that is why we cannot give everything in main memory or we cannot keep may with give more register. If you are going to give more register than cost were increase. So, this is the awareness we can look for the awareness. So, these are external devices where we can store our information. So, like that we are having optical disks also, many a times you know about the CD compact disk ok. So, you can store, say one complete movie you can store in your CD.

Now when you are going to play the movie then what will happen. We are bringing the information form CD to the main memory and procedure is going to take the information from memory and accordingly it is going to display also. So, similarly we are having some devices for communication purpose also. So, one is your modem and another one is your network interface card. So, these are communication devices.

So, we are having a varieties of external devices, their purposes are different, the transfer rate is different, the format of data that we have storing information is different. So, to handle all those issues we are having this particular IO module. Now we are going to see what basically we are having.

(Refer Slide Time: 18:19)



External Device Block Diagram

Control signals from I/O module
Status signals to I/O module
Data bits to and from I/O module
Control Logic
Buffer
Transducer
Data (device-unique) to and from environment
CPU
I/O mod
deno

So, now, we said that this is I am having the processor CPU. So, this is connected to the IO module and different devices are connected to this particular testing. So, this is a device. So, what basically we should have in the device. So, we should have an controller who control that particular device. So, this is the external device block diagram or that controlling of the particular device.

So, what basically we are having. We are having a control logic over here. This control logic is going to receive the control signals from IO module. So, this is connected to these things, it is going to receive some control signal from the IO module depending on those particular control signals, this control logic is going to control that devices.

Now one example may be like that control, that IO module is giving an control signal like that it wants to read some information from this device, maybe you just think that this is an hard disk and we are going to reach something from the hard disk. So, IO module is giving a signal control signal to it, now control logic is going to perform the appropriate operation and it will give some signal to the IO module. Maybe one of their control signal may be the status of this particular device, whether devices ready or not like that.

So, some status information will be given to the IO module. Once IO module is receive the status information and what will happen now it knows that, now it can transfer the information. So, in that particular case now this is the device that you are connecting to this particular controller. So, it maybe that hard disks that we are connecting to it. So, we are storing the information in this particular hard disk.

Now device controller is going to read the information and we are having an transducer over here, what transducer does basically if. Basically if you look the basic definition or the generic definition of transducer, basically it change that you transfer the energy from one module to the another form. So, here also you can say that it is transferring the information from one form to the other form. So, when we are talking about the hard disk the, what is the basic principle, this is magnetic in nature.

When we you are using an CD then what is the nature, this is the optical in answer; that means, you are storing information with the help of magnetic property or we are storing information by using the principle of light. So, now, when we are going to get information then this transducer is going to convert it from that magnetic information, magnetic information to the electronic information or electrical signals.

Similarly that light in principle of light will be converted the electrical signals. So, this transducer is going to convert information from one come to the other form and finally, it will be buffered over here and it will be stored in this particular device driver itself. So, you are having a very limited space over here. Like that when you press a key in the keyboard, so it is a mechanical device, so you are pressing it.

So, what, when you press the keyboard with respect to that particular key, some information will be stored into the one buffer or one register in that particular keyboard itself and after that that will be transferred to the processor through IO module. So, here taking the information through transducer you are converting into the appropriate format; that means, you are converting or sending it to the electrical signals then we are storing into the buffer and once we are getting it then we have going to give it to the IO module.

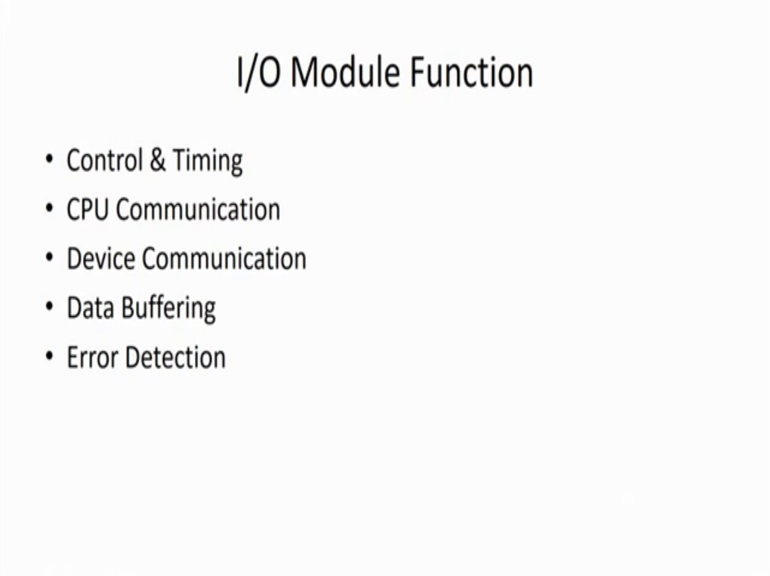
So, that data is transferred to the IO module. And once data is available in the IO module, then IO module is going to transfer this information to the processor. So, this is the way we can look into; that means, you just see that, in the hierarchy now we are having three component; one is your processor, IO module is connected to the processor and the devices are connected to the IO module and for every device we are having a device dive form ok, or you can say that this is the control for that particular device.

So, this will device, what is that device we are having or what is the electronic circuit or what is that driver that we are having that is specific to a particular device, so this is specific to this particular device. So, whatever their external device block diagram you are writing over here that will be specific to a particular device, but this IO module is generic one, we can connect any type of IO devices to the IO module and that IO module will be connected the processor.

So, these are the steps that we are going to have when we are going to transfer information from say input devices to the processor and similarly when we are going to transfer information from processor to the output devices it will follow the same thing. So, basically, now if the way I am saying that I am going to read something some information from this hard disk. Now similarly we may want to store some information to this particular hard disk, then what we are going to do.

From processor it will give to the IO module, through IO module it is going to coordinate with this particular control signals, after that from IO module it will be transfer to the device driver, it will come to the buffer and from buffer through transducer, it will going to convert it that appropriate signal. So, if there is a hard disk then from electrical signal it will convert to the magnetic signals or if you are using a CD compact device then what will happen, that electrical signal will be converted to the light signal ok. This is the way you are going to transfer information.

(Refer Slide Time: 24:07)



I/O Module Function

- Control & Timing
- CPU Communication
- Device Communication
- Data Buffering
- Error Detection

Now, what are the IO module function, already I have explained many more things. So, this is in natural you can say that. First one is your control and timing. So, this is the control circuit, control logic circuit that you are, have having. So, you have to synchronize the time, because the speed of this particular device is much slower than a processor. So, that is why you are having this particular buffering, positive buffering, then we will transfer to the IO module from IO module to this thing.

So, we have to synchronize the whole operation, so that it works in a coherent manner. So, for that we need many more control signal and turning signal. So, this is basically, it generates the timing signal and it generate the appropriate control signal to control the devices. So, in this particular module we are going to see what are the different kind of control that we may have for devices.

It is difficult to discuss you for all the devices, at least we will discuss one devices to see what are the control signals that we have for the particular devices. Second one is your CPU communication. Already I said that that IO module is connected to the processor through this particular system bus. So, communication to the CPU will be done through this particular IO modules on there. Then device communication, just see the previous slide, I am saying that through the IO modules we are connecting the device. So, there is a communication between device and devices also.

So, this is your device communication and this is your CPU communication. So, IO module is responsible for control and timing CPU communication, device communication, another one is data buffering. Already I have mentioned that devices are working in different speed and different format also. So, to transfer information from device to IO module will take more time, because device are basically slower in the processor.

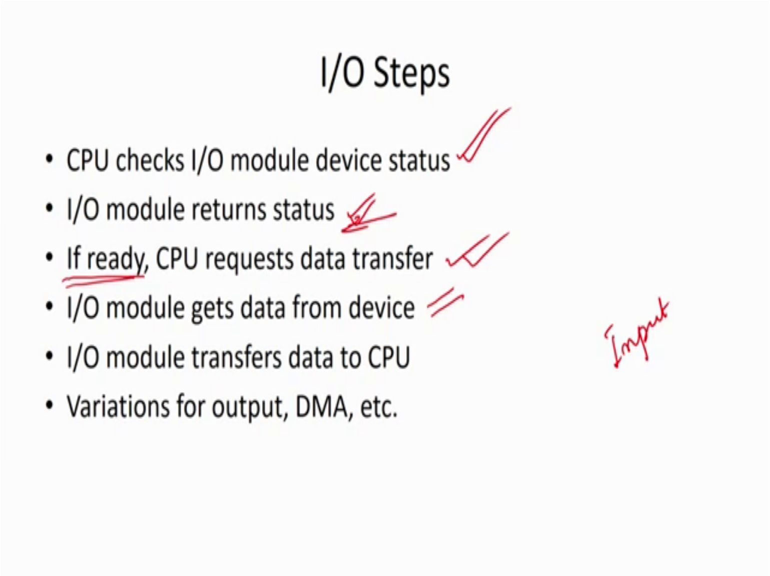
So, that is why IO model is going to buffered the information, positive, it correct the information from input devices, it will buffered over here. When that IO module is collecting sufficient information then it is going to transfer the information from IO modules buffers to that processor, because what electronics component, relatively they are having the same speed, but my devices are not properly electronics component they are having mechanical movement also, so they are slower. So, this is the buffering.

Secondly, so if you want to print some files and you are many a times used to give, now, you write a letter then you send it to the printer, you are going to get a printed copy of that particular letter. So, basically what it does, printer is a slower device. So, if processor is going to directly going to interact with the printing or printer then what will happen, many a time processor need to wait. So, for that particular case what will happen.

First, processor will transfer a information to the IO device, IO module. So, IO module is going to buffer it and after all transferring a sufficient amount of information, then IO module will transfer that information to the printer and printer is going to carry out the printing job. So, this is the data buffering; that is going to be carried out by IO module. So, this is a function of the IO module and another one is also a error detection. So, sometimes it is a transfer of information, it is in bit transfer only, we are transferring 0s and ones. So, if some 0s are converted to 1 during transfer then what will happen we are going to get an error. So, we are getting an erroneous information.

So, in some of the cases in IO modules can detect some of the errors also and it will not defer the processor that whatever the information is negative this having some error. So, that error detection is also a part of this particular IO module. So, it is a another functionalities of the IO module. So, what are the IO step.

(Refer Slide Time: 28:03)



I/O Steps

- CPU checks I/O module device status
- I/O module returns status
- If ready, CPU requests data transfer
- I/O module gets data from device
- I/O module transfers data to CPU
- Variations for output, DMA, etc.

Input

Now we just see that when I am explaining it, it is coming in a floor now. I think most of the things already I have mentioned when I discussed about this particular external device block diagram. So, now, what after all looking, having knowing the functions of the IO module, now we are going to say what are the IO step. So, now, you have, see it is very clear now CPU checks IO modules device status.

So, basically first CPU checks CPU whether the what is the device status. Now IO module returns the status. So, processor is going to, say just think that particular example that we want to print a file. Now we are giving a print command, no processor is going to carry out this particular job then what will happen. Now processor checks the status for IO modules. So, it is going to give a signals and it will say that we want to use the printer then what will happen.

Now idea module is going to check the status of the exit device, whether printer is ready or not, whether printer is switch on or not. If everything is we redone IO modules returns the status. So, when processor get the status, yes it is ready then what will happen. We can carry out the output operation, it going to send the information to the window.

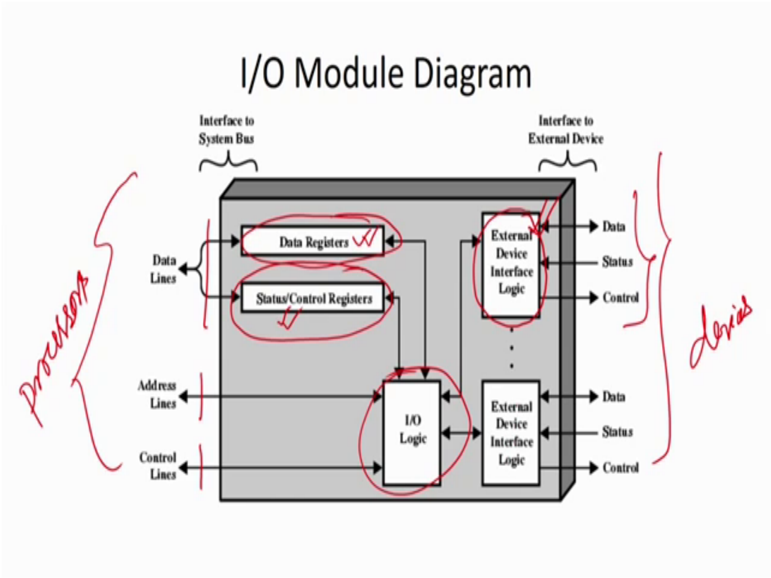
So, if it is ready. So, once that IO module is returning the status, now process is going to check, whether it is ready or not. If it is ready then CPU request data transfer. So, it may be an input transfer or output transfer. For printer it maybe an input operation a, sorry for printer it may be an output operation, but if you are reading a file then it may be an input operation. So, if ready CPU request the represents bar. Now IO modules get the data from the device.

So, here in this particular step, if you are going to look into it, basically we are going to talk about the input operation. If we are going to look for the output operation then this step would have been slightly different, it will say that IO module would get the data from the processor. So, basically then here we are going to look for it. So, IO module get data from the device. So, since device is ready, now, IO model is going to get the information from device. So, basically if you say that I am going to read a file from a hard disk.

So, now, IO module is going to collect the information from data, because it is having a buffer, so it is going to at least collect the information that may be accommodated in the buffer space, then IO modules transfer the data to the processor ok. Now once it is collects the information is storing it in the buffered and IO module will transfer it to the processor. Then this is the way that we are going to complete it.

Now here we having variation of outputs and for that we are going to discuss what are the difference, so you can do it. So, that is we are saying that there are the variations for output and variations for input also, where DMA will come in to picture, so we will discuss that thing also. Now we have seen the devices there and for that we are having a device controller we have seen these things.

(Refer Slide Time: 31:14)



I/O Module Diagram

Now, these devices will be connected to the IO module now just see that top level block diagram of the IO module. So, these are the things that we are going to connect to the processor, this side we are having the processor and these are we are having the devices. So, this is connected through our system bus. So, we are having this particular data bus, we are having that address bus and we have this control lines. So, this is the system bus and we are connecting it to the processor.

Now here we are having two basic register; one is known as your data registers and second one known is your status of control registers. So, now; So, when processor request for a device,, then this IO module is going to check for the status of this particular device and accordingly status of u, then it will set this particular status bit over here and this particular status bit will be monitored by the processor and depending on the status bit it is going to transfer the information and that transfer the function will be go through this particular registers, which is all data register.

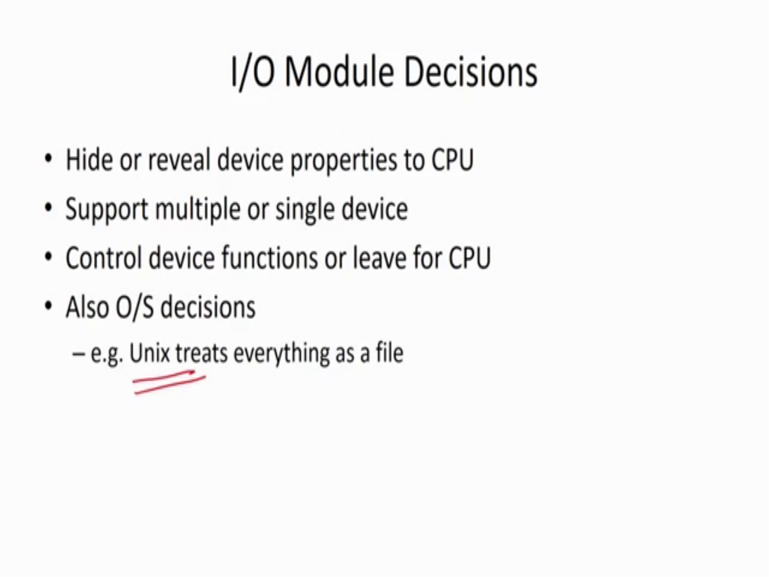
So, similarly here we are having that address lines we are giving the addresses and we are giving some controls like that, it is an input operation or whether it is an output operation all those things will come and after reaching this particular information that here we are having an logic circuitry IO logic circuitry and this logic circuitry is going to identify which device basically you are going to use, it depends on the contents of the address line.

So, we can connect different devices. So, it is going to identify one of the devices depending on the address, depending on the controllers it will going to say what is the operations that we are going to perform, whether it is read or write. So, here we are having an the external device interface. So, maybe here, we may have an space for buffer also. So, we are going to collect that information to the buffer and from that buffer through this data register we are going to transfer it to the processor.

So, this is again another interface logic circuit that we are having to handle this particular IO devices. So, one simple control signal I can think about, I can just mention here like that. So, when we are going to print a file then what will happen. We have to place the printer head at the proper position, maybe start of the line.

So, this control signal, it will send a control signal like that this particular interface logics to initiate the printings of to set the printer properly; that means, bringing the printer to the proper position. So, these are the control signal that we have to see. So, this is the way that we are going to connect. So, this is the generic view of IO modules. So, according to our recommend we have to design these things. So, these are the basic components you should have one is data register, one is that as a control register, one control logic circuitry and device interface logic circuit for different devices.

(Refer Slide Time: 34:19)



I/O Module Decisions

- Hide or reveal device properties to CPU
- Support multiple or single device
- Control device functions or leave for CPU
- Also O/S decisions
  - e.g. Unix treats everything as a file

So, what are the IO modules decision. Already we have seen the functionalities, so basically what did this, it is hide reveal device properties to the CPU, because. So, processor is going to work with the electronic signal, it will basically going to walk to it your bitch team of 0s N 1.

So, in what format we are storing in the device, in what way we are organizing the all information, this may not be relevant for the processor. So, we can hide those information and you can shift everything to the IO module and IO module is going to take care of everything. So, it may the hide or in some cases it may reveal or device properties to the processor also support multiple or single devices.

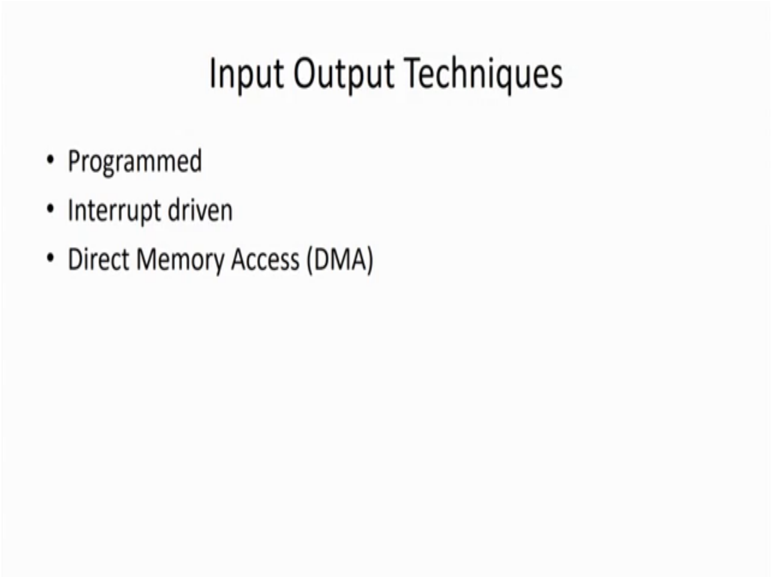
So, already here I have said that we are having a provision to support multiple devices. Generally we are having IO modules to support multiple devices, control device function or leaves for processor. In most of our cases it is going to control the device function also, basically like that one example I can say that while going to print it, I have to place the printer head in a proper position. So, that will be done by your my depth IO module.

So, this is some controlling of the devices also, some IO decision. So, depending on the instruction that we have in the instructions or of the processor, we can write the appropriate between for my operating system.

So, some OS decision, it will also be taken care by IO model they are represents something to the IO module also. So, these are said one; for example, it says that if we have that Unix operating system, say you know that we are in several operating system 1 is Unix, maybe windows, most of your custom of the windows. So, in case of Unix operating system it has different like that everything will be treated as a file. So, if we are connecting an IO devices that IO devices will be a treated as a file; that means, it will set up simply said that, it is having and file id and with the help of that file id we are going to control that particular devices.

So, some ways decision will also be transferred to the IO modules all right. Now what are the different techniques that we have in your input output, how we are going to do it.

(Refer Slide Time: 36:24)

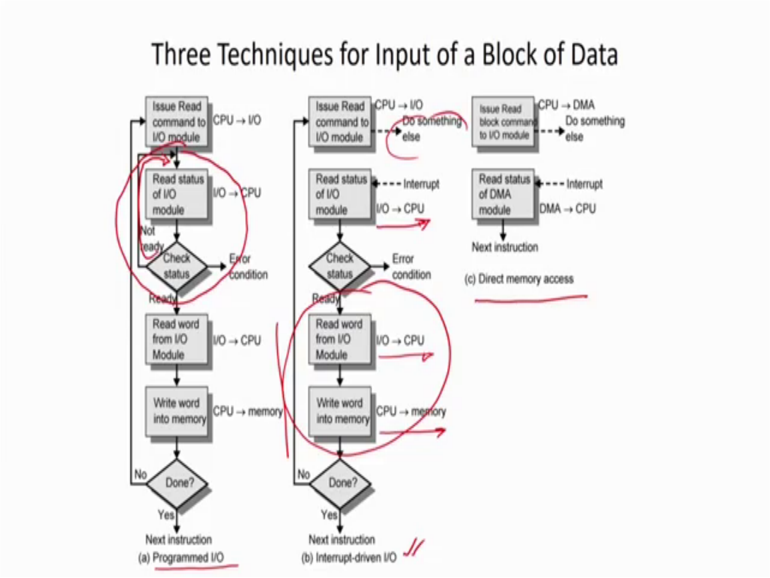


Input Output Techniques

- Programmed
- Interrupt driven
- Direct Memory Access (DMA)

So, if you are look into that way of transferring information. basically we are going to get three different way of transfer our information; one is your programmed IO, second one is your interrupt driven and third one is your direct memory access or DMA. So, here we are going to discuss about those three different techniques the way we are going to transfer our information. So, in this particular case in a nutshell I am going to just give it a brief idea, what are the differences between these three methods.

(Refer Slide Time: 36:53)



Three Techniques for Input of a Block of Data

So, this diagram it says that, we are going to perform some IO operations, so CPU initiated to the IO devices. Then reach status of the IO modules. So, processor is going to read the status whether device is ready or not. So, in that particular case if device is not ready, it will check whether not device is ready, if it is not ready then it will remain over here. So, it will keep on waiting in that particular loop.

So, basically we said this is the busy waiting and during that time processor is not doing any work ok, processor is idle just, it is busy of setting this particular status. Once the device is ready going to perform the operation, maybe write the information from the input devices, put it into the memory again after completion of the job again go to this.

Now, you just see this is the point where we have to look into. It processor want to work with an input output devices and it is going to check the status, if it is not ready then it will be in this particular loop and we said this is that busy waiting and in that particular time processor is not doing any useful work. So, this is the wastage of our processor time. So, this is called programmed IO, we are going to control the devices with the help of one program and through the program we are going to check it.

Here main drawback is your wastage of processor time. So, that is why you are thinking how that wastage of time can be minimized or can be removed. So, for that that second one is coming which is your interrupt driven io. So, this portion is same. Now what will happen? now processor is going to give a signal or giving limit to the IO module depth processor want to do some IO person, may be taking something from the input devices or going to put something in the output devices. After giving this information to the IO module.

Now, processor is going to carry out its own work. It is not going to wait for thus status, it is not going to check the status of the device, it is just simply give the information to the modules and after that processor is going to carry out its own work, because if we are exhibiting a program, processor will going to execute that program from that particular point now one IO module is getting this particular information.

Now that processor want to interact with some input output devices, then IO module is going to look for the status of this particular input devices or maybe output devices, when device is ready and everything is there, just to start of the transport operation at that particular time that IO devices is going to give the information to the processor, this is reside, this is now input output model is going to interrupt the processor, now interrupt is doing some job. Now IO module is going to interrupt it.

So, at the particular point now processor will be knowing that know that device is ready or data is ready, now processor can work with this particular input output device. So, after that it to checks status and carry out these particular transfer of information after completion of the transferring of information and again if you go to this point. So, here in this particular case you just see what we have done.

We have eliminating that particular busy waiting or wastage of time, but now come to this particular point, what we are doing, when you are going to have the read then from IO devices we are going to transfer to the processor and processor we are going to put it to the memory is that input operations or. So, that from we want to take some information from hard disk and then to put in to the memory.

So, first we are bringing in to the processor, we are bringing the processor means, we are putting into some register of the processor and from that processor register we are putting in to the memory. Similarly if I want to write some things or save some file to the hard disk then what will happen. First we are going to take it from the memory to the some register of processor and from the processor register we are going to transfer it to the IO module and from IO module it will go to the output device may be hard disk.

So, you just during this transfer, the processor is involved, processor cannot do any other work, it is simply transferring the information for maybe hard disk to the memory or maybe from memory to the hard disk. So, again we are thinking now how to eliminate this particular involvement of processor. So, that direct, we can some are aware you can simply transfer our information from hard disks to the memory directly without the intervention or without taking help of the processor, so that all need not to go through the processor.

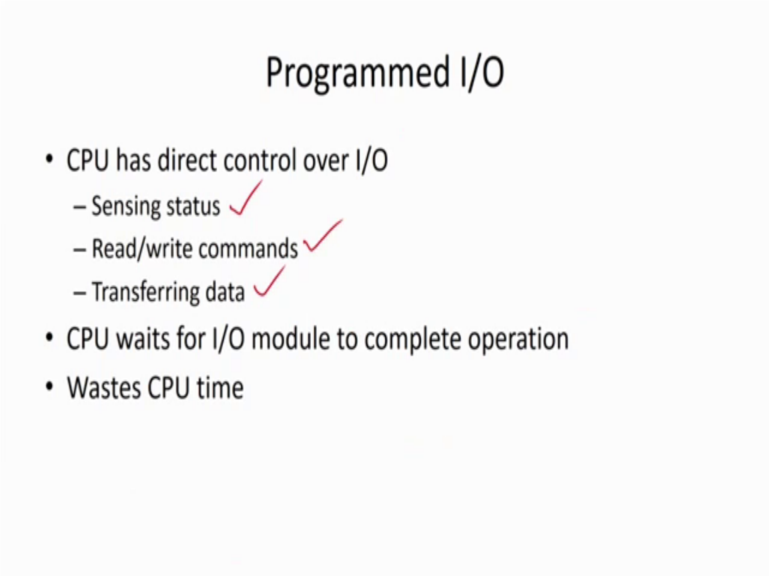
So, for that the start process is coming which is known as your DMA direct memory access. So, in that particular case what will happen? Now say when we want to read a file from the hard disk to the processor memory then what will happen? instead of sending too much processor registers straightaway we can bring it from hard disk to the memory. So, this is direct memory access. So, initially that processor involvement is there; like that to initiate the input output operation.

But at later point of time this processor involvement is may not be required, directly you can transferred from hard disk to the memory or from memory to the hard disk. So, this is basically talk about if the bulk data transfer. So, if I want to transfer a file size offset 1 Mb. So, I have to transfer a whole file. So, without taking the intervention of the process.

Processor will be initiate the transfer after that straightaway we are going to transfer it from memory to the, sorry hard disk to the memory or maybe in case of writing from memory to the hard disk. So, this is we are accessing directly. So, we said this method is your DMA; direct memory access. So, now, in this particular module.

Now, we are going to see the design issues and the walking issues or what are the implications that we have for these three kind of transfer. Now for that first one we are going to say what a program layer. It is already I have mentioned that what basically you used to do you have to check the status, continuously once it is ready we have to do it.

(Refer Slide Time: 43:23)



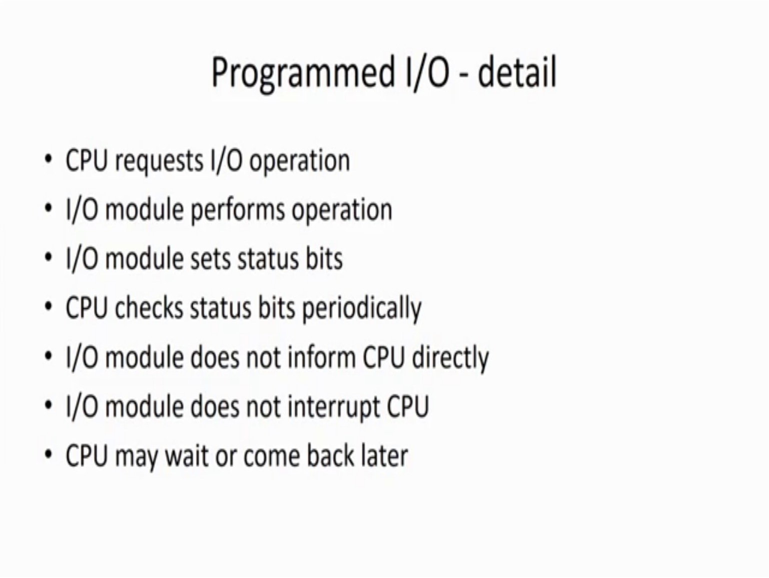
Programmed I/O

- CPU has direct control over I/O
  - Sensing status ✓
  - Read/write commands ✓
  - Transferring data ✓
- CPU waits for I/O module to complete operation
- Wastes CPU time

So, basically it is a sensing of status, then using the read that command then transferring of data; this is the way that we are going to do, but here what is the problem that we have first CPU leads to wait and P, it is checking it continuously. So, it cannot do any other work. So, there is a wasted of CPU time.

So, basically you just see that if we are going to connect something like that this processor is going to, maybe 1 bit of the status register is going to continuously monitored and if it is set to 1; that means, device is ready, then it is going to perform the transfer apart from. So, this is all programmed IO.

(Refer Slide Time: 44:01)



Programmed I/O - detail

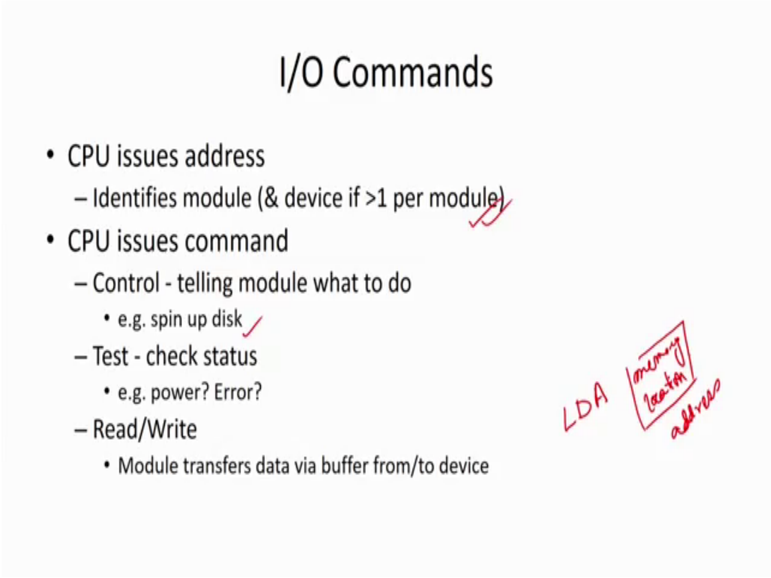
- CPU requests I/O operation
- I/O module performs operation
- I/O module sets status bits
- CPU checks status bits periodically
- I/O module does not inform CPU directly
- I/O module does not interrupt CPU
- CPU may wait or come back later

So, basically this is now programmed IO, because details now exhausted. These are a simple sets ; first one CPU request to IO operation that in processor we have shown it IO model perform operation, then by looking into the state of their devices IO model said the status bit CPU sets status bit periodically IO model does not inform CPU directly, you know user.

So, that here IO module is not going to inform the CPU directly that device is ready, but in case of interrupt driven in somewhere, in somewhere IO module is going to inform this particular things to the puzzle. So, IO module does not inform the processor directly. So, processor is going to continuously check it, then IO module does not interrupt CPU.

So, in that particular case IO model does not intercept CPU, but now CPU may with or come back level. This is unless you say are after waiting for some more times, if you may fill those. It is now going to abundant this particular IO operation, it may get you see that IO come at data point of time. So, this is the way that IO module is doing it. Now after that to perform work with IO module or input output devices what are the basic requirements.

(Refer Slide Time: 45:16)



I/O Commands

- CPU issues address
  - Identifies module (& device if >1 per module)
- CPU issues command
  - Control - telling module what to do
    - e.g. spin up disk
  - Test - check status
    - e.g. power? Error?
  - Read/Write
    - Module transfers data via buffer from/to device

LDA
memory location
address

Already I have mentioned that we need some commands IO commands. So, first issues for that IO common is like that how to identify the modules. So; that means, we have to have an device address for this, the one point how we are going to give the addresses of the IO devices like that.

When we are discussing about song a person like that reading some information from memory or writing some information to the memory; say one example I think I have mentioned something like that LDA load accumulator from some memory location; that means, if it is an accumulator based machine then we having a register called accumulator. So, we want to load something to that particular register accommodator from some memory location. So, we have to give the address of this particular memory location.

So, in the instruction we have to give the address like that, when we are going to perform the operation. First of all we have to give the address of the devices from which we are going to take information or to who we are going to give the information. So, this is the addressing scheme, we have to see and will see how we are going to give you then issue some control commands.

So, basically some control commands we have to issue just to initiate the process like that already have a explained. If we want to print something in the printer we should change some control signal to the printer to initiate it or maybe to bring the print ahead to die profit position. Similarly when we are going to read something to the, from the hard disk then what will happen.

We have to bring the appropriate location of the hard disk to the input output head, we will elaborate these things. So, basically, so spin up disk we have to rotate the disk. So, those command we have to give like that check the status, whether it is ready or not, whether power is on or not, all those things status signals we are having and read and write.

So, one is this is the mode of transfer, whether read basically we said that we have taking something from the input device and write we have saying that we have going to put some information in to the output device. So, this is the read and write. So, basically we need some instruction to perform this read write instruction also. So,. So, the command related to control the IO device can be now look into three different categories. One is your controlling, second one is your test and third one is your read and write. So, these are a commands that we have to design.

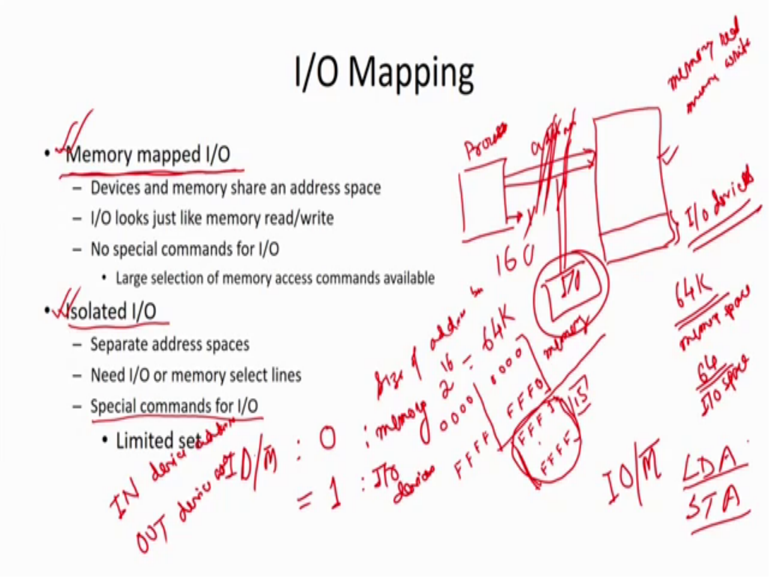
Let me see now instruction. So, we have to put some commands to carry out this particular work. Now how to address an IO devices. So, in that particular case, it is basically now, already I said that we have to give an unique identification to the processor or to the devices. So, for that we have to given address. So, again this address is nothing, but in which stream of 0s and 1, because we are going to work with the 0s and 1 only.

So, basically like that the, like memory address which is a bit stream of 0s and 1, then what will happen in the particular case. Here we assigning a particular code to the memory unit or memory location. Similarly we are going to assign a particular code to the input output devices which will be treated as an address for that particular device and that address has to be unique, because we have to identify device uniquely.

When we are going to use the printer you have to give a unique code for a printer or a unique address to the printer, when we are going to use your hard disk you have to give an unique address to the hard disk. So, basically this is the addressing scheme. So, what is the address. It is very much similar to that of a memory location which contains 0s and ones ok; that is all.

So, in binary we can say 0 than one or you can say it is having a particular number know, how we are going to give this particular address, how we are going to map this particular input output devices. So, for that we have to look IO mapping, so how we are going to map the IO devices to the processor. So, that we can identify that particular device correctly.

(Refer Slide Time: 49:27)



I/O Mapping

- Memory mapped I/O
  - Devices and memory share an address space
  - I/O looks just like memory read/write
  - No special commands for I/O
    - Large selection of memory access commands available
- Isolated I/O
  - Separate address spaces
  - Need I/O or memory select lines
  - Special commands for I/O
    - Limited set

memory read
memory write
Processor
I/O devices
16
I/O
$2^{16} = 64K$
0000
FFFF
IO/M : 0 : memory
= 1 : I/O device
IN device address
OUT device address
64K memory space
64 I/O space
IO/M
LDA
STA

So, in that particular case we are having two different ways of do it; one is your memory mapped IO and second one is your isolated IO. So, I will just simply explain it and help of small example. Now its just added, we are having the processor and I am going to say this is the address bus, we are connecting the memory.

Now, in memory mapped IO it says that same address space is paired by a, shared by my memory and input output devices. Now you consider that size of my address bus is your 16, then what is the memory capacity with my size of address bus is 16, you should know it now. So, this is your 2 to the power 16, which is your 64. So, we can have 64 k memory location.

Now address will go from 0 0 0 to all FF FF. Now out of this particular 64 k addresses what will happen. Some of addresses are reserve for my, your IO devices. So, total address space is my 64 k. Now I am not going to use 64 k memory location, we will use less memory location, but some of the memory location will be you are going to identify the IO devices. Now say now I am going to consider that I am going to look for the memory location from 0 0 to FF F 0.

So, a these are the addresses that we are going to use for memory. then what are the addresses are able in the F, this is your F F F 1 to F FFFF. So; that means, 15 addresses are remaining left. So, these 15 addresses can be used to identify my input output devices; that means, in the, if I am supporting the addresses in this particular way, I can connect 15 different IO devices to the processor.

That means, in entire memory space I am having address space 64 k out of that some of the addresses are used to connect the memory; that means, that many memory location you can use and 15 addresses are kept reserve of my IO devices and whenever I am going to give this particular addresses or addresses from this particular range we are going to identify input output devices.

So, if this is the way we are going to reserve the address of IO devices, we will say this is your memory mapped IO. Now another one is your isolated IO. So, in case of isolated IO the memory space and IO space are different. So, now, if you consider the same example then what will happen. We are having 16 a size of address bus is 16. So, we can go for 64 k. So, when I am going to have this isolated IO then we can have memory space 64 k.

Similarly, we can have 64 k IO space; that means, we can connect 64 k; that means, 64 kilo which is equal to 2 the power 16, which is equal to 65000 something you can calculate it ; that means, more than 65000 devices can be connected to the processor. So, this is the isolated IO yeah we define address space.

Now in that particular case what will happen we are giving the addresses through this particular address bus, because you are having an one address bus only which is a part of my system bus and it will be this particular system bus you are connecting the IO module also. Now how to identify whatever addresses that we have put in this particular address bus. It is an a address of a memory location or it is an address of an IO devices. So, to identify these things.

So, we are going to use one more control signal, there will several control signal will come through this particular control bus. So, processor is having an control signal and in most of the cases this control signal name is given as IO M bar; that means, input output memory. So, by looking into it what will happen if the signal of this particular signal IO M bar is 0.

So, M bar; that means, you are taking the compliment of it; that means, if the control signal available in this particular IOM by 0 it will say that basically it is going to look for a memory; that means, whatever address we have over here, it will be an address of a memory location and when this IOM bar is 1, this is your IO devices or maybe IO model.

So, when the control signal value is 1, then whatever address we have kept in this particular address bus, this will be treated as an address of an IO module or IO devices; that means, it is IO module is going to now use this particular address, but if the value of this address signal is 0 then this address will be the address of this particular memory location. Now you just see that we having two ways of mapping this particular IO devices; one is your memory mapped IO and second one is your isolated IO all right.

So, in case of isolated IO you are going to get a bigger IO space, because equal number of memory location and equal number of IO devices, but in case of memory map you are going to reserve some of the memory addresses for identifying the IO devices. So, IO space is limited over here. So, this is the way that we can look into it and for that.

Now, again I can say that. Since in case of your memory mapped IO what will happen in that particular case, this is nothing, but an address of a memory location, but we are testing to an IO devices. So, whatever memory operation that we are having; say memory read and memory write, then same operation can be used for those particular IO devices.

So, in most of the cases we are talking about your LDA load accumulator or say SPA store accumulator; one is your read command, another is write command. So, for that load accumulator we are going to load something to die accumulator. So, in that particular case what will happen. Now we are giving a address of the memory.

So, if the address happen to be an address of IO device; that means, you are going to take the information from IO device and going to put into the accumulator. So; that means, some instruction can be used for memory as well as IO devices, if we are using memory mapped IO, but if we are using isolated IO then we have to have separate instruction to control this particular IO devices. So, in most of the cases we are going to get some IO commands like depth.

So, depths are isolating with some special IO command. So, these are the IO commands that what we are having in most of the cases we can get like that; one command is your in, another command is your out in device address and out device address ok. So, in case of in whatever address device address are giving from that particular address you are going to get the information and going to bring it to the processor.

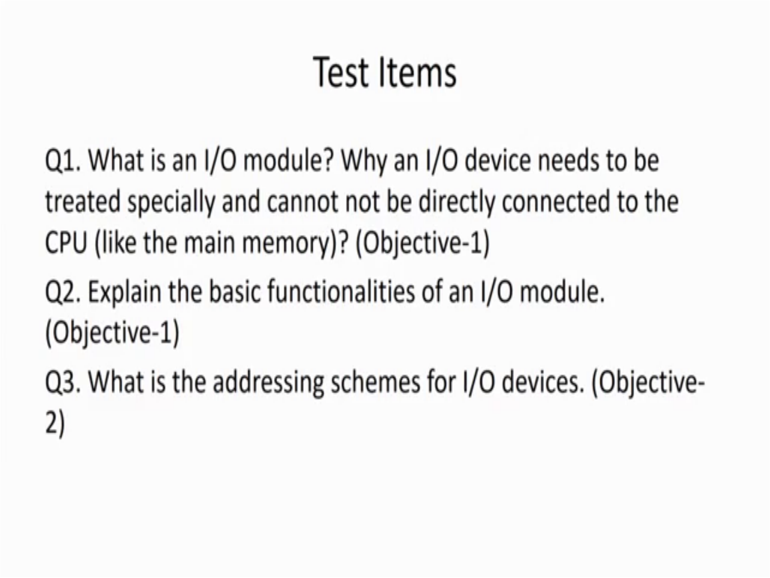
And in case with the help of all the instruction, the information of the processor is the can be change to the output devices. So, these are the issues, basically we have one is your addressing scheme, second one is what are the commands that we need to handle this particular input output devices and along with that I think we have already use those particular program IO details.

In that particular scheme, it is that in out or input and output will be controlled by a program and for this program is going to do you are going to execute this particular program, that in a loop it is going to check the status of that particular status bit; one status bit is ready, then this program is going to execute the other part; that means, going to transfer the information.

So, with the help of one simple program and we are doing it which is note and as your device service routine in, it is basically to control that particular device, we need a program. We said this is the device service routine, we are going to execute it, instead of programming cell we are going to check the status. Once the status is ready then we are going to carry out the input output operation.

So, this is the way we are going to transfer information in programmed IO and for that we need the addressing of the devices and we need that common. So, these are the things that we have discussed in this particular unit. Now, look for the some test items and here I am giving the first test item.

(Refer Slide Time: 58:43)



Test Items

Q1. What is an I/O module? Why an I/O device needs to be treated specially and cannot not be directly connected to the CPU (like the main memory)? (Objective-1)

Q2. Explain the basic functionalities of an I/O module. (Objective-1)

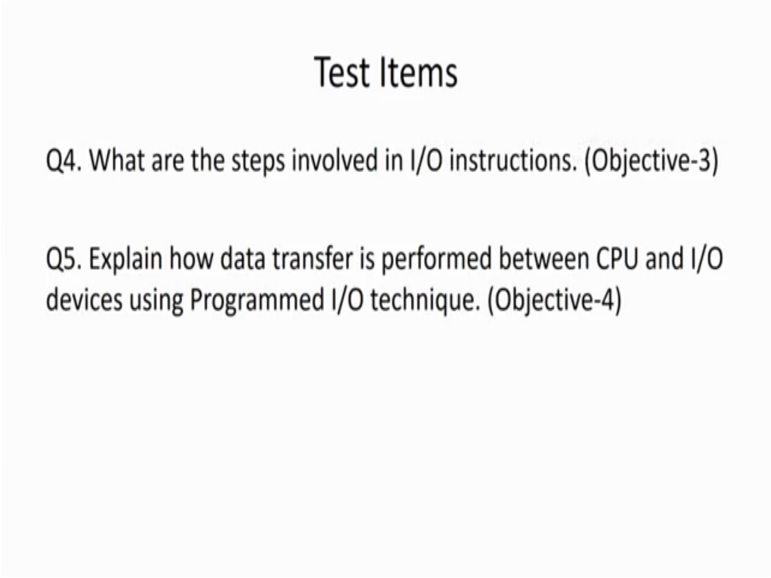
Q3. What is the addressing schemes for I/O devices. (Objective-2)

Like that question on, what is an IO module, why an IO device need to be treated specially and cannot be directly connected to the CPU. So, like memory are directly connecting to the CPU, but in IO device you are not directly connecting the CPU, because already I have mention it that integration of the processor will increase, so that is why we are transferring the responsibility to IO modules and for that we need the IO modules.

So, basically here we are guessing this particular question and we are meeting the objective 1 that we have cited for the objective of this particular unit. Second one I am saying that, explain the basic function, it is other IO module, again this is in the some knowledge level only which is the objective 1.

Again we are meeting it and I think I have explained it what are the functionalities that we are having for the IO module. What is the addressing scheme of IO devices. This is basically objected to have mention and we are met in this case and I think now you know how you are going to give the address of and IO devices, this is similar to the memory address only.

(Refer Slide Time: 59:50)



Test Items

Q4. What are the steps involved in I/O instructions. (Objective-3)

Q5. Explain how data transfer is performed between CPU and I/O devices using Programmed I/O technique. (Objective-4)

Question 4; What are the step involved in IO instruction. So, we are meeting the objective 3 and we said that already mentioned the step. This is basically processor is going to give the information to the IO modules, then I am on the check the status, it will report to the processor, then processor is going to start the transfer. So, this is how.

So, basically have to see what are the step involved for this input and output instruction. Question 5 explain how data transfer is perform between CPU and IO device using program IO techniques, and this is in the design level and that means we have just meeting the objective 4 if we can perform this particular or you can solve this particular problem.

So, already I have explained, it is nothing, but a program we are writing it and we are having a loop, it will be in that particular loop, until that device is ready, one device is ready, then it is going to execute the remaining part of the program and that program will be known as my device service routine. So, these are the things that we have discussed over here and I think that at least you can have idea now, how we are going to transfer information from the input and output devices to the processor. So, with that I am going to wind up the our lecture today

Thank you very much.