

Computer Organization and Architecture: A Pedagogical Aspect
Prof. Jatindra Kr.Deka
Dr. Santosh Biswas
Dr. Arnab Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Fundamentals of Digital Computer
Lecture - 05
Storage and I/O Interface

Hello everybody, welcome back to the online course on computer organization and architecture. We are in the module fundamentals of digital computers and this is unit number 5 and in unit number 5 we are going to discuss about storage and I/O interface.

(Refer Slide Time: 00:46)

Module: Fundamental of Digital Computer

- Unit-5: Storage and I/O Interface
- Unit Objectives:
 - Objective-1: Illustrate the use of system bus (Knowledge)
 - Objective-2: Describe the size of address bus and capacity of memory module (Analysis)
 - Objective-3: Explain the need of control bus (Analysis)

So, like earlier unit also for this particular unit we are going to mention the objective here we have stated few objective.

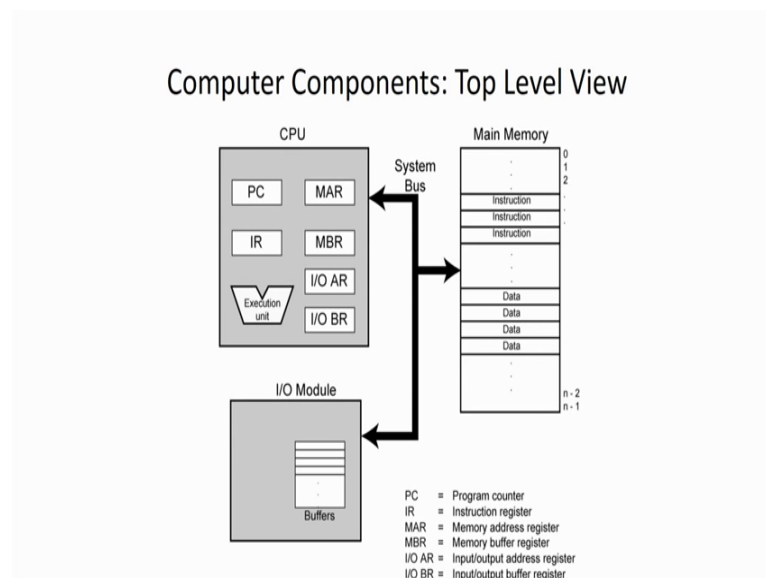
Objective-1: Illustrate the use of system bus, again we will be dealing this objective in knowledge level. Objective-2: Describe the size of address bus and capacity of memory module, so this is basically on analysis level so we will be able to analyse once we get some memory module and we can find out what is the capacity of that particular memory module and what are the other requirements. Objective-3: Explain the need of control bus this is also an analysis level we are going to see why we need a control bus?

So, we are discussing about the von Neumann stored program architecture and our computer works on this particular model already I have shown this particular model, we are having the central processing unit that I/O equipment and memory module and whatever we are going to

work everything must be present in a main memory and processor is going to interact with the main memory and is going to fetch the information from main memory.

And in the top level view of our computer system can be visualised in this particular way here we are having three main components; one is your CPU or central processing unit this is the main processing element inside a computer.

(Refer Slide Time: 02:16)



And this CPU is connected to main memory, which is the storage unit primary storage unit and along with that we are having I/O module, through I/O module we are going to connect all the input output devices like; keyboard, mouse, printer, monitor, those will be connected through this particular I/O modules and everything will be integrated through this particular system bus. So, we are talking about a bus; now, what is a bus actually? So, in this particular case this bus is nothing but a connecting wires, so we are going to connect different devices with the help of bus.

(Refer Slide Time: 02:41)

Connecting

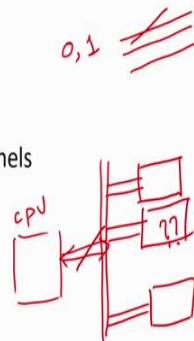
- All the units must be connected
- Different type of connection for different type of unit
 - Memory
 - Input/Output
 - CPU (internal to processor)

So, we are having different types of connection in a computer; so there may be one connection between processor and memory, so this is for memory interface. We are having another connection between your processor and input output devices.

(Refer Slide Time: 02:47)

What is a Bus?

- A communication pathway connecting two or more devices
- Usually broadcast
- Often grouped
 - A number of channels in one bus
 - e.g. 32 bit data bus is 32 separate single bit channels



And another kind of connection we are having CPU interconnection this is basically internal to the CPU, already I have mentioned in our last lecture that; all the components of processor will be connected to each other through an internal CPU bus. So, the bus is nothing but a collection of wires through which we are going to transfer information and basically it is broadcast methodology, basically what happened we can see that if I am having a device say; in this particular case, device you are going to talk about say CPU and say through this particular system bus, it is connected to different devices or different unit. So, in that particular

case whatever information we are transferring to this particular bus, this is in the broadcast mode and it will be available for all the connecting devices, but according to the need one particular connecting device is going to use this particular information.

Secondly, that collection of words are basically grouped together and we can say this is basically channels, so number of words that you having one particular bus is basically known as the number of channel and this is basically depends on the number of information that we are going to transfer.

So as for example; if I talk about a 32 bit data bus, in that particular case we are having 32 bit separate signals bit channel, so we having 32 different channel, in every channel we can transfer one bit of information, that information may be either 0 or 1. So, collecting all those particular channels or words basically we say this is a bus.

So, what is a bus? Bus, is nothing but a connecting wires through which you can transfer information from one point to the other point in that particular bus.

(Refer Slide Time: 05:05)

Data Bus

- Carries data
 - Remember that there is no difference between “data” and “instruction” at this level
- Width is a key determinant of performance
 - 8, 16, 32, 64 bit



Now, there are categories of bus; one we are talking about a data bus, through this data bus we are going to transfer data from one point to the other point, maybe we are transferring data from memory to the processor or maybe from processor to the memory, so but when we are going to talk about data or when we are going to talk about transfer of data through a bus, in that particular case or in this particular level device level there is no difference between data and instruction all will be treated as a binary bit.

.

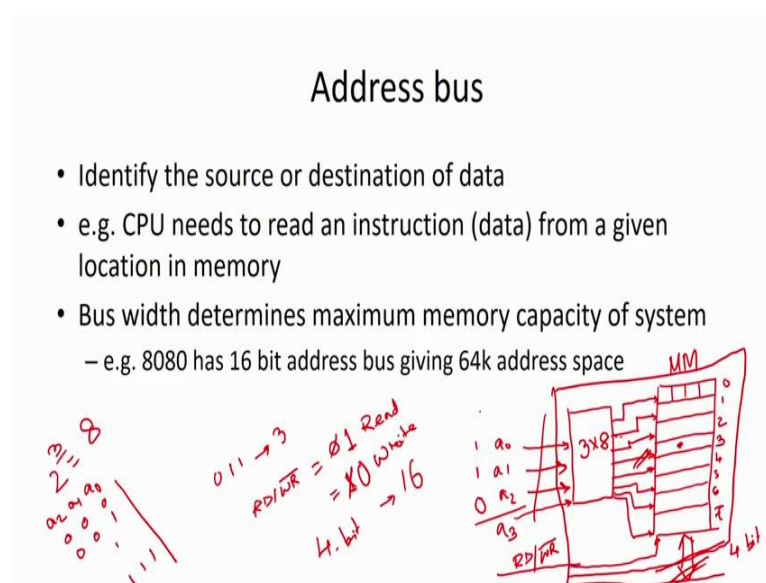
So, as for example say if I am going to consider about 8 bit numbers, so if this is an instruction then what will happen? It is going to interpret it as an instruction inside the processor, but this

can be treated as a data also that may be a data for a particular instruction. So, when we are transferring information from one point to the other point or one device to the other device at that point in the signal level we don't have any distinction between data and instruction, all are in the bit level either it is 0 or 1, in some signal line we are transferring 0 or in some signal line we are transferring 1. But when we are going to interpret it, at that time we are going to see whether we have to treat this particular information as an instruction or we have to treat this particular information as a data or an instruction.

And width is a key determinant factor for the performance, so if we are having an 8 bit data bus then what will happen? We can transfer 8 bit at a time. If we are having 64 bit data bus, we can transfer 64 bit of information at a time. So, in one go you can transfer 64 bit, so if we are having more width basically we say that we are going to achieve more performance, because in one go we can transfer a complete information. There is another bus we call this is the address bus, this address bus basically identify the source or destination of a data so basically what will happen I am saying that we are keeping our information in main memory.

Ok so this is a memory location and already I have mentioned that to go to a particular memory location, we must identify that particular memory location. And how we are going to identify it? It is with the help of an address, but what is the addressing format when we are going to work with our digital computer, just consider small example; just say that in this particular memory module itself, we are having total 8 memory locations, so I can number them as your 0, 1, 2, 3, 4, 5, 6, 7, so we are having 8 different memory location and we are going to take information from say one particular memory location.

(Refer Slide Time: 06:56)



Now, how we are going to identify that particular memory location. So, we are going to do it with the help of address. Now what is the addressing format? Now you just see that, if we are having 8 different combination to identify any one of this combination we need 3 bit of information, because we know that $2^3 = 8$ and this particular 3 bit I can say that a_0, a_1, a_2 that three information that may go have all 000, 001 like that 111, so these are 8 possible combination of this particular 3 bit a_0, a_1 and a_2 . So, since these are the unique combination, so we can use those particular combinations to identify one of the memory locations.

.

Now, how to do it? We are having 8 locations and we having 3 bit of information, so already we have talked about the decoder, we know that we are having a decoder where we are having some n input lines then maximum output lines will be 2^n depending on the input combination one of the output lines is high. So, in that particular case what we can do? We can use an 3 by 8 decoder, we are having 3 input lines, ok so this is a_0, a_1, a_2 and these output lines will be connected to those particular memory location. Ok so, depending on the information that we are putting over here it is going to enable one of the lines, so if I give the information say 110, so this is the most significant bit a_0 is the least significant bit, so 011 this is basically going to represent in decimal 3. So, that is means 0, 1, 2, 3 this particular line will be enabled.

.

So, when this line is high; that means, we are going to select this particular memory location with the help of this particular line and this is the combination. So, this combination is basically known as my address to my memory location and you just see that in every memory location we are going to store 4 bit of information. So, in that particular case what will happen? You can think that we are having a data bus which is the 4 bit wide, now along with that we may have 1 control line or 2 control lines say; if I say that, this is a line which is giving us name as you read write bar, so in read write bar what will happen? If this is 0, then we are going to perform a read operation we are going to take the information from the memory and this signal is 1 then we are going to perform a write operation.

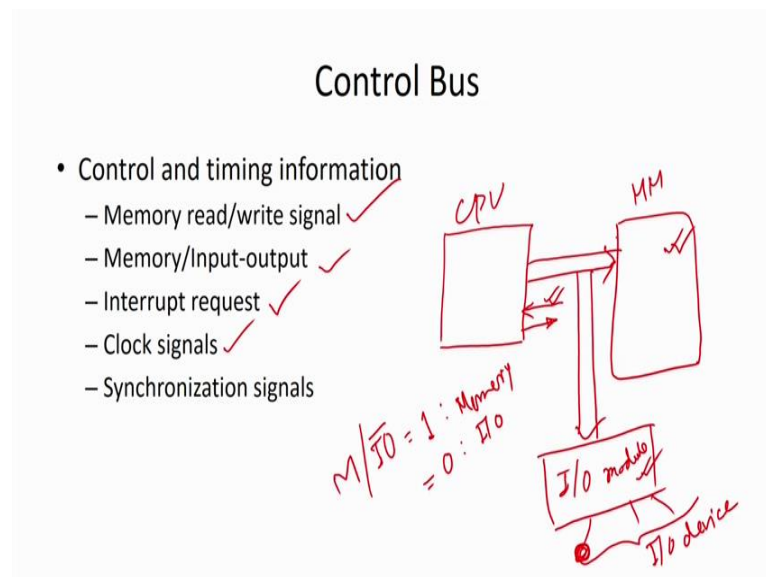
Maybe I can say like that or maybe it is the other way round since it is read no bar. So, 1 is your read and 0 is your write this is the exact definition we can say since I am defining a read write bar, so when it is 0 then it is a write operation. So, when we are going to perform read operation we will identify the memory location through address and whatever we have stored in this particular memory location that will be available in the data bus and through data bus we can take it to the another device or another component. And when it is a write operation, so

whatever information we are giving in to the data bus that will be stored in this memory location that whatever we have identified over here.

So, in that particular case what will happen? I can see the memory module, this one is a memory module, so we are having input lines called address bus, through address bus we are going to give the address to a particular memory location and we are having the data bus, through data bus we are going to read the information or we are going to write the information and along with that we are having a control line which is going to indicate whether it is a read operation or write operation. So, in this way we can visualise our memory unit, so in case of memory unit we are having address bus as well as data bus.

Again the size of data bus basically we get how many bits we can transfer in one go and size of address bus basically give the number of memory location that we have in the particular memory location. So, it is 3 bit we are going to say 8 memory locations, so if I increase it to the 4 bit then we are going to get 16 memory locations; that means, I am going to add one more bit in the address bus. So, in that particular case what will happen? With $a_3 = 0$ and I am going to get 8 combination and with $a_3 = 1$ we are going to get another 8 combination, so total 16 combinations.

(Refer Slide Time: 13:33)



So, address bus is nothing but collection of wires through which are going to give information 0's and 1's and that combination is going to identify one of the memory location. So, like that we are having a control bus, already I have said that we may have a control signal called read and write signal that we are going to either read the memory or we are going to write into the memory. We have put another control signal which is known as your memory and input output, so basically just see that this is my processor through this particular address bus we are

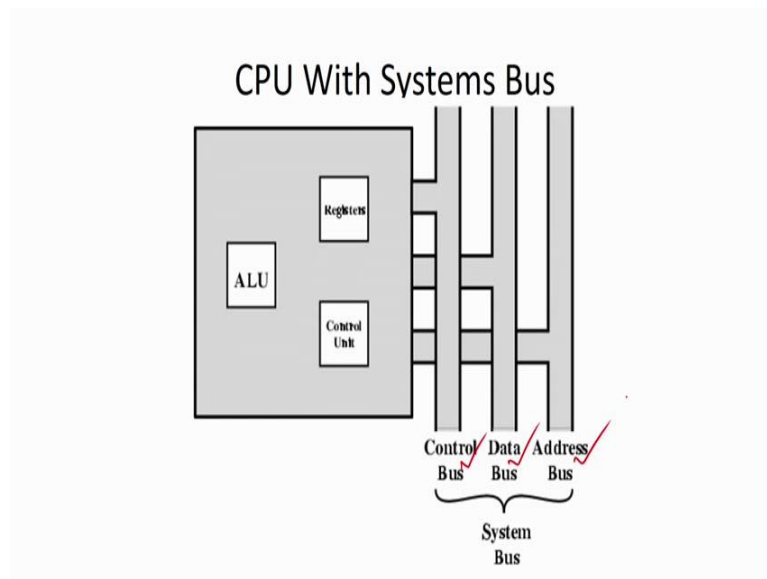
connecting to the memory location, again through this particular address bus we are connecting to input output module. Ok now whatever address we are giving over here that may be either address to a memory location or that may be an address to the I/O device, so through I/O module we are connecting several I/O devices, so it may be an address to an I/O devices.

Now, how to identify whether it is a memory address or it is an I/O address? For that we are going to have one more control signal coming out from the CPU and that control signal is identify whether it is a memory or I/O. So, if I am going to say that; memory or I/O bar, so in that particular case if this signal is your high, then we are going to said a it is a memory address and if the signal is 0 then we say that this is an I/O address, so with the help of this another control signal we can identify whether we are going to access the memory or we are going to access a I/O. Another signal this talking about the interrupt request, because again it is related to the I/O say some of the I/O device is sitting over here and say this I/O device want have to do carry out some work in the processor.

Then since during program execution that processor is interacting with the memory and taking the information from memory to a processor and carrying out its job and in between this device want to perform some work. So, in that particular case; it will give an indication through another control signal, which is known as your interrupt request. So, those things will be discussed in details while we are going to discuss about the I/O devices.

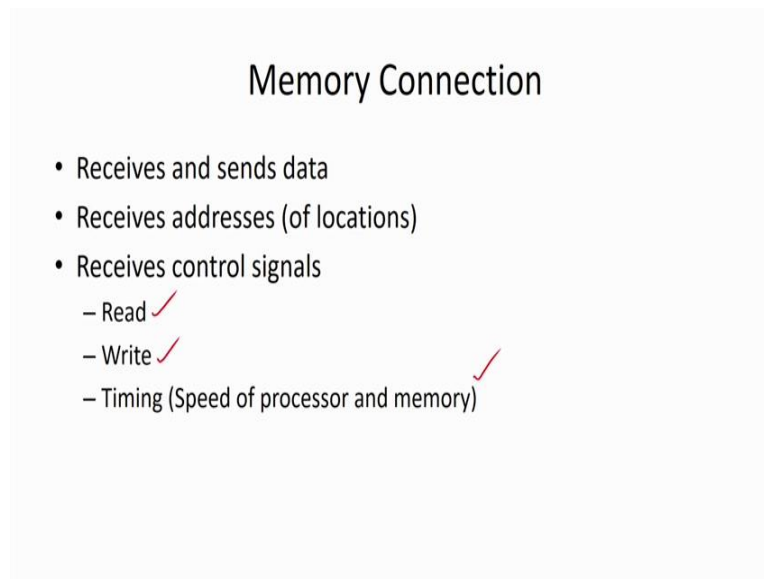
We may have another clock signal call clock signal, because it is in synchronous operation, so everything will be controlled by a continuous clock signals and that signal will also come as an input or output to that device and we may have some synchronization signal we will see what is that synchronization signal. So, finally we are having now 3 components one is your data bus, address bus and system control bus. So, combining all those 3 buses, Control bus, data bus and address bus we are going to say this is the system bus. Ok so, through the system bus we are going to connect external devices to this particular processor, those external devices basically may be some input output devices.

(Refer Slide Time: 16:16)



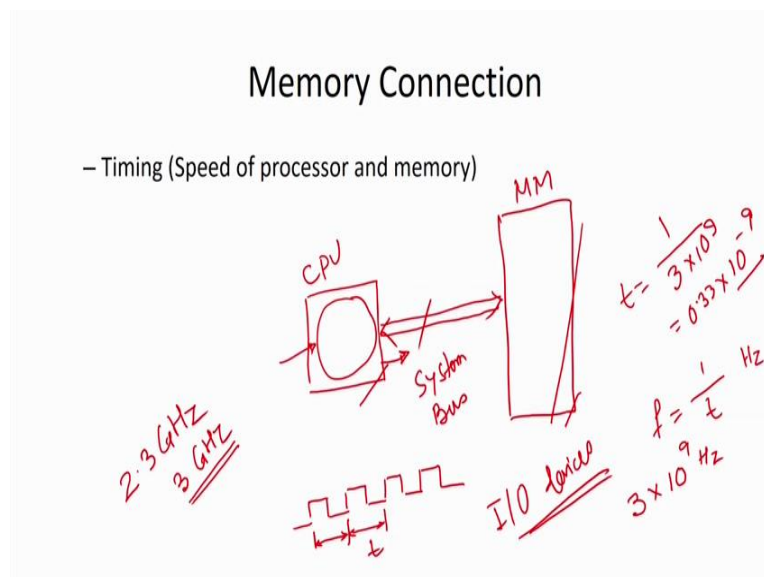
And secondly I can tell that memory can also be treated as an external devices with respect to this particular processor because memory is sitting outside the processor so; that means, through system bus we are going to connect the other devices and memory to the processor and the system bus is having 3 components control bus, data bus and address bus, now already I have mentioned all those things. So, basically we are having memory connection. So, with the help of memory connection what we are going to do receives and sends data, receives address or location because we have to identify the location also and receives some control signals. So, basically in system bus we are having 3 component with data bus we are going to receives or sends our data, we are having address bus, through address bus we are going to give the address to a memory location and through control bus we are going to receive or send some control signal, one is your read signal already said that we are going to read the information, write signal we may say that we are going to write something to the memory and some timing information also has to be sent in that particular connection.

(Refer Slide Time: 17:42)



So, here I want to mention that the speed of processor and memory is not same. Ok so, we are having a processor, CPU and we are having a memory unit and this is connected through this particular system bus.

(Refer Slide Time: 17:49)



Now most of you are working with a computer and you know that generally you used to say that I am having a computer and it works on say 2.3 Gigahertz or sometimes you may say that, we are having a processor which works in your say 3 Gigahertz.

So, what does it means? Basically this is the frequency operating frequency of the processor. So, what will happen? We are having a continuous running clock and this is the time period of this particular clock signal. So, this is during this particular clock signal this is clock pulse we

are going to perform some job or say some micro operation, then in the next clock cycle we are going to perform some of the clock cycle. So, this is basically called time period of this particular signal and when I am going to talk about the frequency that frequency is nothing but 1 upon that particular time period and its basic unit is your hertz.

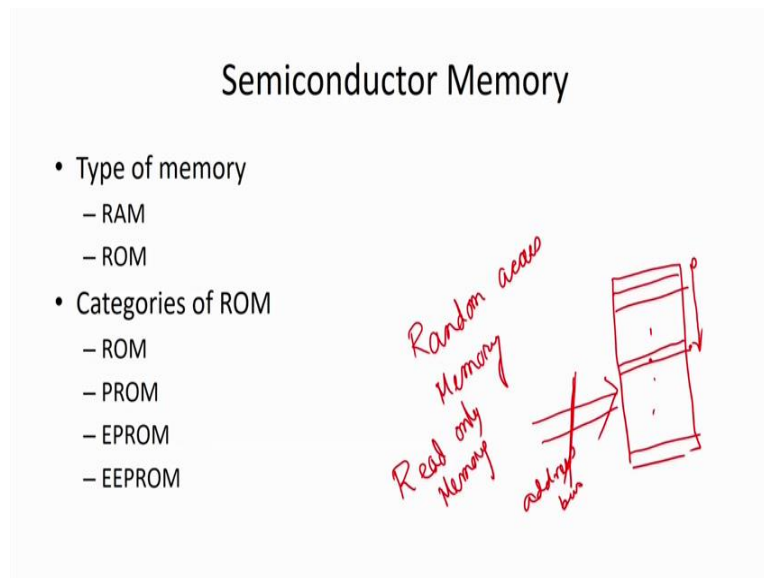
So that means, if we are going to say that it works in 3 Gigahertz; that means, 3 gigahertz is = 3×10^9 hertz, kilohertz is your 10^3 , megahertz is your 10^6 and gigahertz is your 10^9 . So, it works in this particular frequency 3×10^9 hertz; that means, in 1 second it is going to oscillate that many times 3×10^9 . So, what is the time period? We can say that this is nothing but t is equal to $1 \text{ upon } f$, so this is nothing but $1 \text{ upon } 3 \times 10^9$, which is I can say that 0.33×10^{-9} .

So, this is basically second, millisecond, microsecond, and nanosecond. So, 10^{-9} is your Nanosecond, so that many this is the time period. So, in that Nano second 0.33 Nano second I can carry out one small operation inside this processor, so this is the operating speed. So, this is that we have achieved this particular 3 Gigahertz is due to the advancement of technology and we are going to use the technology over here which is a very faster technology, generally it is a silicon technology and generally nowadays we are working with the submicron level. So, with the help of advancements of technology we are getting devices which can work in 3 Gigahertz.

So, but when we are looking for the performance the cost is also more over here, so If we are going to use the same technology to make this particular main memory, then what will happen? The cost is going to increase. So, while constructing our main memory generally we used to use a slower technology just to reduce the cost, so this is a slower technology we are using, so here we are having a faster technology, so somehow we have to synchronize these two operations, so for that we need some synchronization signal or timing signal.

So, some other control signal will come which will say that when the memory has completed these operation or not, whether data is ready for the process or not we have to give some indication, so some of the control signal will come for that particular purpose also to synchronize the other unit with the processor. Similarly, when we are connecting some I/O devices the speed of those I/O devices again still slower, again this is slower than the main memory, so again we need the synchronization signal for that, so those all the synchronization signal will come in through this particular control unit bus.

(Refer Slide Time: 22:19)



So, if I look into the memory basically that main memory we talked about this particular main memory this is basically nothing but a semiconductor memory. So, in case of semiconductor memory here we can talk about two types of memory; one is your RAM, second one is your ROM. So, RAM is basically it stands for random access memory, so RAM is random access memory. So, what is this basically? You can see that we are having a memory module we are having several location and through address bus we can identify any one of these particular memory location.

So, depending on the input or contents of the address bus we can go to any memory location. So, the accessing of memory location random is necessary by giving a particular address we can straight away go to that particular memory location. So, it is randomly we can access any memory location.

Since we are talking about the random access there may be some other access also, which is a sequential access? In case of sequential access we have to access the memory in sequence. So, if it is a sequential memory then what will happen? If I want to go to this particular point then I have to go in sequence found a first location to that particular location, but in case of random access that sequence is not required, randomly or at any point of time I can access any memory location. Just I have to give the appropriate address to the address bus, so it is talking about the random access memory.

Another one I have mentioned about ROM; this is read only memory, but again I should mention that ROM read only memory is also random access this is the same nature that we are having, but we can go to a particular location and I can read it. So, what is the difference if I talk about RAM and talk about ROM, what is the similarities over here? Both are random access. And what is the difference? In case of ROM, it is read only we can read the contents from the memory, but in case of RAM it is read and write, we can read the information that is stored in a memory also we can write some new information in some memory location. So, both read and write is possible in case of RAM If I say that it is a RAM, but if I talk about the ROM then it is only read only we can read the information from that memory. Again here I can say that these RAM and ROM can be categorised into two category; one is your volatile second one is your non-volatile or permanent.

So, when we talk about the volatile; it is basic like that, as long as we are working the system; that means, as long as we are giving the power to the system then we are going to have the content in the memory and we are going to work with those particular contents of the memory, as soon as I switch off the machine if I remove the power then what will happen? The contents of this particular memory will go away, so that's why we have to say it is volatile, so RAM is volatile in nature. So as long as power is there contents is there and we can work with this particular contents, but once you switch off then all the contents will vanish if next time if I we are going to switch on the machine again we are not going to get those particular contents, so it is volatile in nature, but ROM is read only memory it is a non-volatile memory and it is permanent in nature.

So, during the manufacturing time itself we can write the contents in the ROM and while working with the ROM we can simply read it we can take the information and we can work with the information. So, it is a permanent in nature, so it is non-volatile memory and when we are going to give the put the information in the ROM if we say simply ROM, then during the manufacturing time only we are going to keep all information. Again we are having several types of ROM and these are basically one is known as ROM that I have already mentioned this is read only memory. So, during the manufacturing time itself we are going to put the information and that information will be used in our devices, so this is ROM, but we are having the enhancement we are having another type of ROM which is known as your PROM programmable read only memory, so it is a ROM but it is programmable.

So, while manufacturing when it will come from the company, then the contents are null basically it is a null contents or it is a blank memory. Now with the help of programmer we can program this ROM or we can say that we can write some new information in to that particular ROM. So, once I write it then the contents become permanent, we cannot erase it and it is permanent in a nature now we are going to use those particular information, so this is called PROM; programmable ROM, it can program only once, once you program it once you write a contents it will remain permanent and we can use it, but after programming if we find that there is some mistake then what will happen? I have to use another PROM only I cannot use that one. So, for that we are having next level of ROM which is known as your EPROM. EPROM is your erasable programmable read only memory; that means, it is a ROM read only memory, it is programmable, but again it is erasable say once you program it we have written contents into the memory, but if you find that there is some mistake in some data or some information, then what we can do? We can erase it and we can reprogram it.

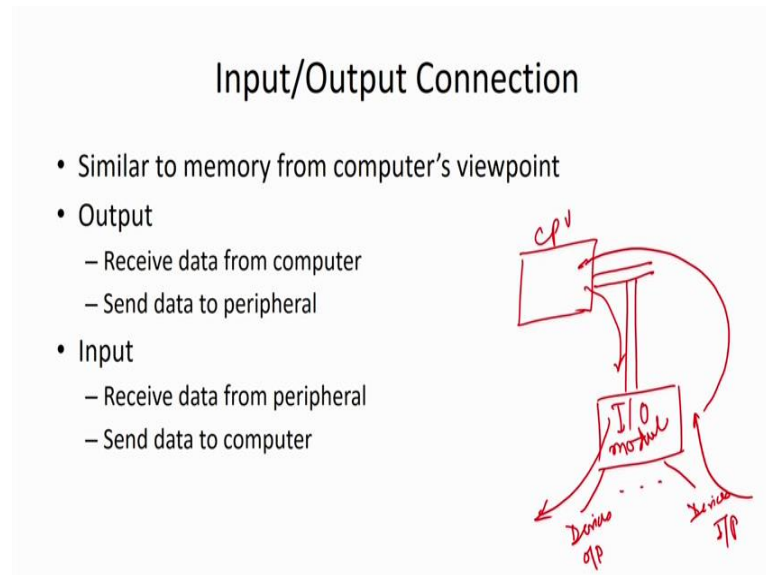
So, here erasing means? Erasing the contents of the entire memory, if there is a mistake in only one bit still you have to erase the entire memory, so erasing can be done by exposing that particular rom chip to ultra violet rays and once it becomes blank again we can reprogram it. So, in that particular case what will happen? If there is a mistake in one particular memory locations still we have to erase the entire ROM and reprogram it.

So, for that we are having a next level which is known as your EEPROM electrically erasable programmable read only memory. So, this is programmable, it is erasable but erasing is not like that we have to erase the entire chip; here, it is called electrically erasable, so in that particular case what will happen? If you find a there is a mistake in this particular memory location, then we can erase this particular memory location and we can rewrite it and need not to rewrite the entire things over here we need not to rewrite the entire memory, so this is your EEPROM.

So, all are semiconductor memory and those semiconductor memory; we are going to connect to the processor to build our computers, which will work as a primary memory. So, like memory connection we are having input output connection also, so through input output connection we can connect input output devices. So, basically what will happen? That input output devices will be connected to the processor through I/O modules. So, what will happen in case of output? It will receive the data from the computers. So, basically what will happen?

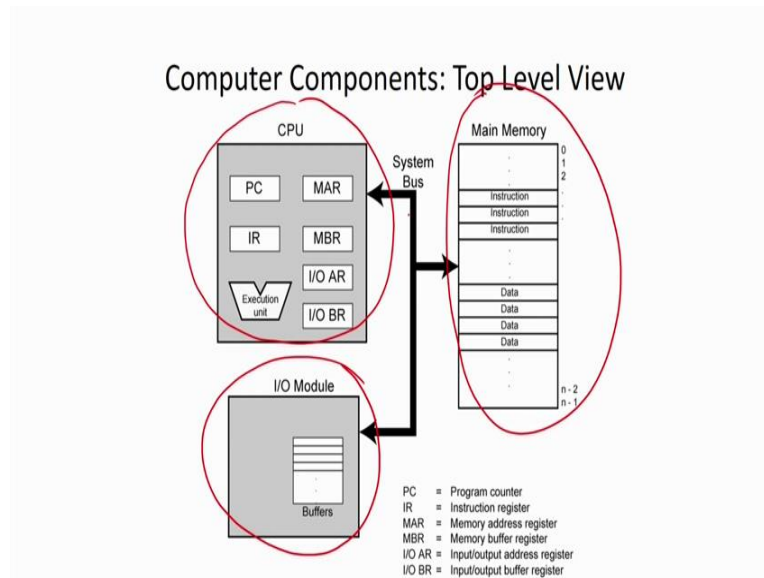
I can say that this is the processors, CPU this is the system bus, through system bus we are connecting the I/O module and I/O devices will be connected to this module.

(Refer Slide Time: 29:44)



So, what we can do? In case of output, it will receive data from the computer and these module will transfer it to the device. So, in case of input it will receive the input from the device and that module will transfer it to the processor. So, this is basically output device and in this is case this is your input device. So, these are the things that we are going to do with the help of input output connection, again those I/O modules will be connected to the processor through the system bus.

(Refer Slide Time: 30:36)



So, this is the top level connection already I have seen this slide you have seen this slide for several times this is the CPU, the main memory unit and this is I/O module. Through I/O module we are going to connect this particular input output devices and all those things are connected to this particular system bus, again I am I would like to mention that system bus is having 3 component; address bus, data bus and control bus.

Now what is an address bus and data bus? And what we can Store? And what are the limitations? Just we are going to give one example. So, we are talking about the address bus, already while talking about the memory module what I said that? If I am having 8 memory location then I need 3 bit of address, ok because $2^3 = 8$ and we are having 8 different combination.