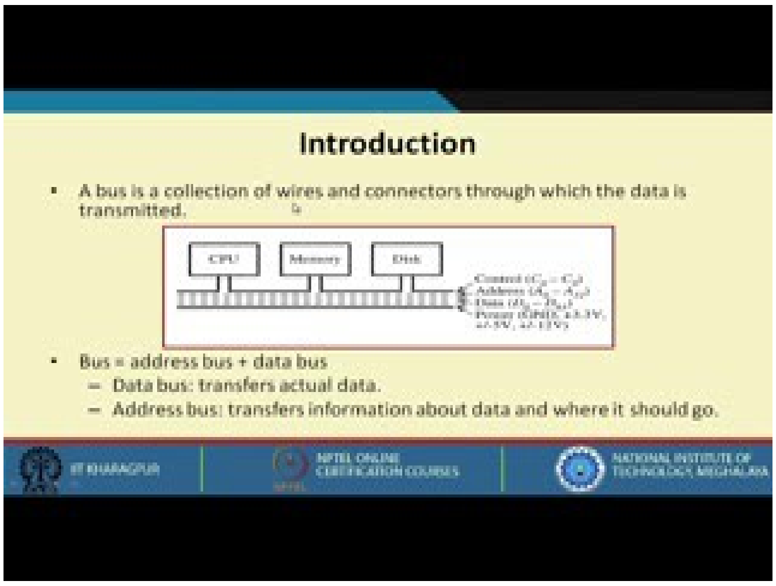
Computer Architecture and Organization
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 51
Bus Standards

In this lecture we shall be discussing on some of the bus standards and protocols that are used inside a computer system. Not necessarily only for interfacing the peripheral devices, but also the bus that exists between the processor and the memory. We will be looking at the overall picture first, and then we shall be looking at one very popularly used bus standard, namely USB.

(Refer Slide Time: 00:52)



Introduction

- A bus is a collection of wires and connectors through which the data is transmitted.
- Bus = address bus + data bus
  - Data bus: transfers actual data.
  - Address bus: transfers information about data and where it should go.

CPU
Memory
Disk
Control
Address
Data
Power
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

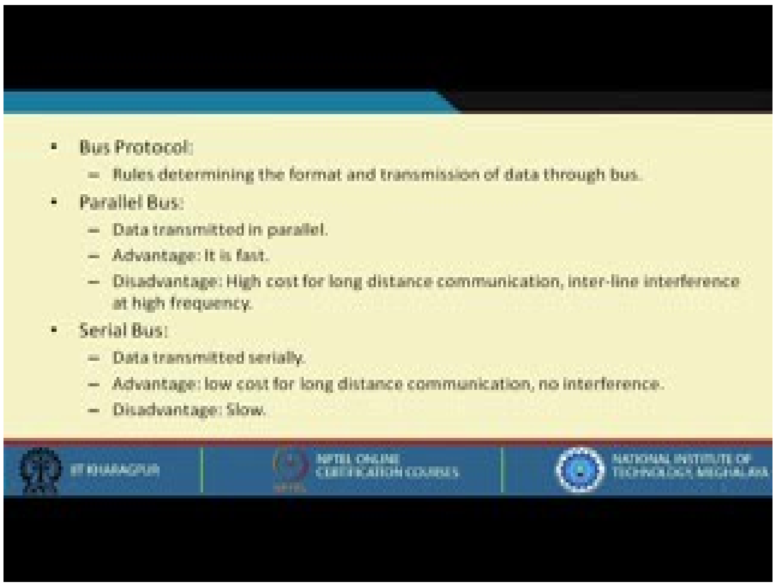
Let us start by defining a bus. A bus roughly refers to a common shared path. In a city when you board a bus, you see the bus takes a group of people from one point to the other. It is like a common shared path. Many people are using the same communication facility, unlike a private car that is dedicated to a person. So, a bus means a common shared path that can be used by more than one entities for communicating between two end points.

Basically the bus is a collection of wires. As this diagram shows, you can have your CPU, your memory and some secondary IO device like disk. This is a simplified diagram and this is your bus. This will be a collection of wires and connectors; through this bus data will flow.

Now, in this bus what are the basic things that should be there? Well, of course, there should be the address lines. When CPU is sending some data, CPU will also mention that to whom that data is meant because there can be multiple devices on the bus. Then of course the actual data. How many data bits can be sent at a time depends on the width of the bus. How many wires are there, then there can be some control signals, which will tell what kind of operation CPU is trying to do. Whether CPU is trying to read the status of the device, or wants to send some data or receive some data and so on. In addition, there can be a set of power supply lines, various voltages, so that some of the devices that are connected can directly draw power from those lines.

So, the most important lines in the bus are address and data, because address will specify the destination. And data bus will carry the actual data.

(Refer Slide Time: 03:24)



- Bus Protocol:
  - Rules determining the format and transmission of data through bus.
- Parallel Bus:
  - Data transmitted in parallel.
  - Advantage: It is fast.
  - Disadvantage: High cost for long distance communication, inter-line interference at high frequency.
- Serial Bus:
  - Data transmitted serially.
  - Advantage: low cost for long distance communication, no interference.
  - Disadvantage: Slow.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Let us look at some of the terminologies. When you say bus protocol, what does it mean? The dictionary meaning of protocol means a set of rules or conventions that both the end systems should comply, so that faithful communication can take place. Similarly, in a bus or for any communication system, there has to be a protocol. Both the end systems should be following the same set of rules, so that data communication can take place without any problem.

So, bus protocol essentially constitutes the rules that determine the format and transmission of data through the bus. Now, the bus can be either parallel or serial. For a parallel bus, the idea is that data are transmitted in parallel, meaning I can have 8 or 16 or 32 parallel lines in the bus through which data can be transmitted per cycle.

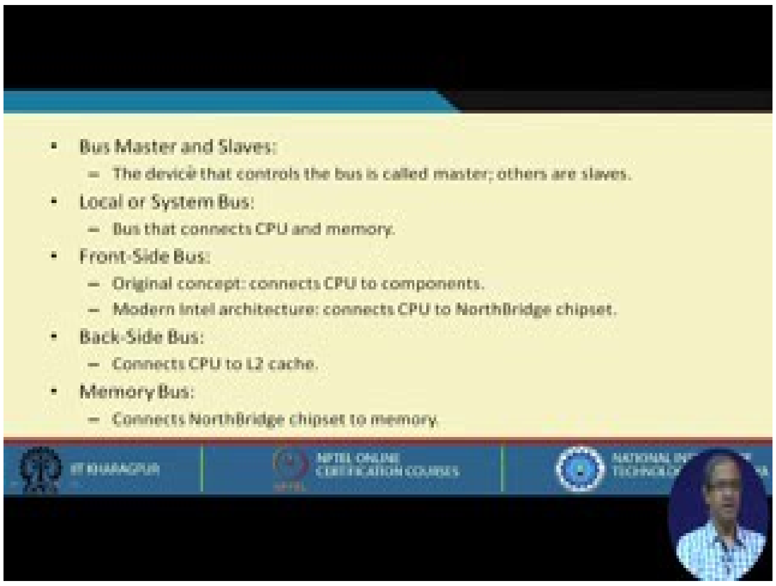
The advantage here is obvious because there are so many lines with which data can flow. It is fast, but the disadvantage is that your cable or the bus will be very thick. There will be so many wires. So, for a long distance communication, the length of the cable will be large, the cost of the cable will also be higher and because there are large number of lines that are stacked in the same cable, there can be inter-line interference, particularly when data communication is taking place at higher frequencies. Because of this, parallel bus cannot be used for longer distances.

In fact, most of the communication that we carry out today is based on some kind of serial bus. In our earlier systems like that printer port I talked about, some parallel bus standards were there, but today you will find very few.

In contrast the serial bus has a single line for sending and a single line for receiving. When you have a single line for sending, then you can have that serial communication protocol I talked about earlier using start bits and stop bits to synchronize between the sender and receiver. That kind of a format can be used. Well, there are other ways also. Some synchronous kind of communication can also be done, where the sender and receiver knows the exact speed of data transfer. Those are also possible.

Because the number of lines are very few, the cost of the cable will be very low, so for long distance communication, it will be very suitable, and because there are very few lines, interference will be very less. The disadvantage is that because you are sending the data serially bit by bit, it will be relatively slow as compared to the parallel bus.

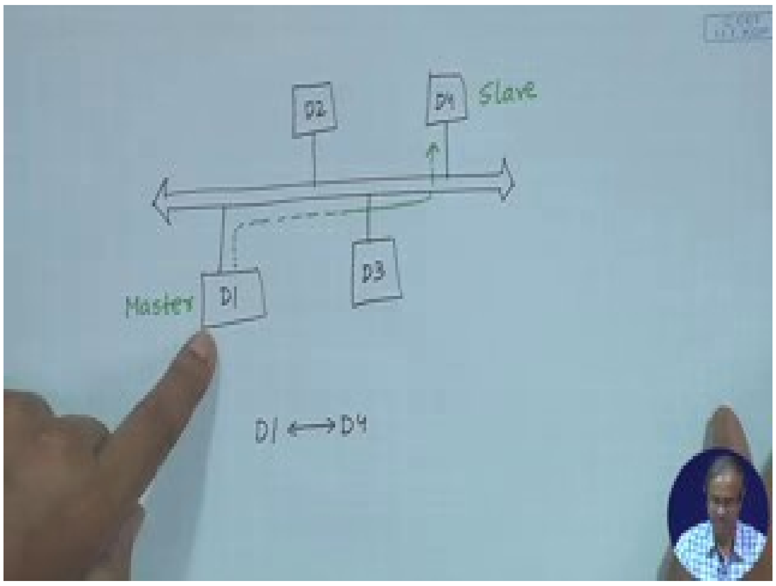
(Refer Slide Time: 06:53)



- Bus Master and Slaves:
  - The device that controls the bus is called master; others are slaves.
- Local or System Bus:
  - Bus that connects CPU and memory.
- Front-Side Bus:
  - Original concept: connects CPU to components.
  - Modern Intel architecture: connects CPU to NorthBridge chipset.
- Back-Side Bus:
  - Connects CPU to L2 cache.
- Memory Bus:
  - Connects NorthBridge chipset to memory.

In a bus as it is said there can be many nodes or devices connected to it.

(Refer Slide Time: 07:03)



Let say here we have a bus and we denote the bus like this. There can be several devices that are connected to the bus. The devices may not all be of the same type, they can be different. One can be more powerful, one can be very simple. I am calling them D1, D2, D3 and D4.

Now, by calling the bus master and slave what I mean to say is that suppose D1 and D4 wants to communicate, now depending on the bus protocol and the capabilities of the device interfaces, there can be a scenario like this where this device D1 will be designated as the master and D4 will be designated as the slave. What does this mean? D1 being the master will be able to initiate the data transmission, but D4 on its own because it is the slave cannot initiate. The master is always responsible for imitating every data transfer over the bus.

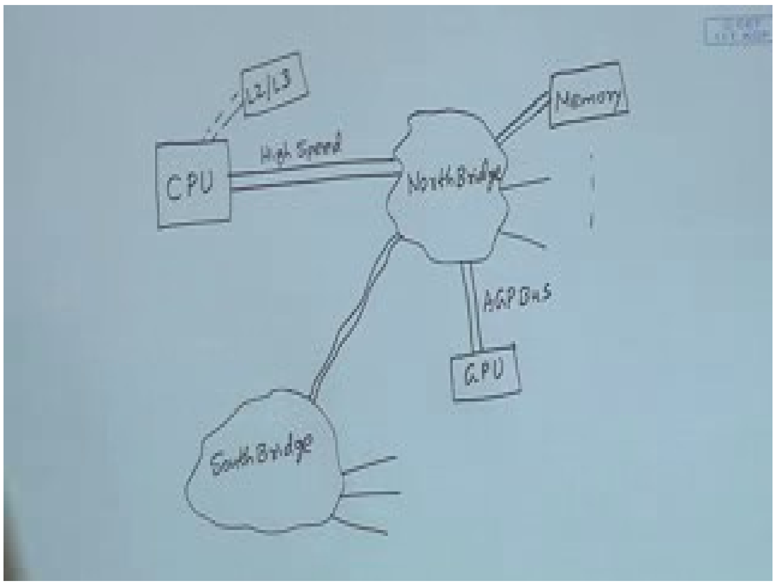
There are many protocols where there is a master-salve relationship between the various devices and it is the master which takes care or initiates the communication with the other devices that are slaves.

Now, we are talking about a bus or the buses that are there inside a computer system. If you just open your desktop or a laptop, you will see a lot circuitry inside, but schematically or architecturally how does the devices get connected inside, say on the mother board. Here we are talking about that.

Local or system bus is a term we used to refer to a bus that connects CPU and memory. You know that in a computer system, the CPU and the memory are the fastest two components and the communication between them is the most crucial in determining the overall performance of the system. So, this bus that connects the CPU and the memory is referred to as the local bus or the system bus.

Following the nomenclature of Intel, we can distinguish the buses as front side or back side. Front side bus means the bus that connects CPU to the other component. This was the original concept, but if you look at the modern motherboards, front side bus refers to a bus that connects CPU to the north bridge chipset.

(Refer Slide Time: 10:30)

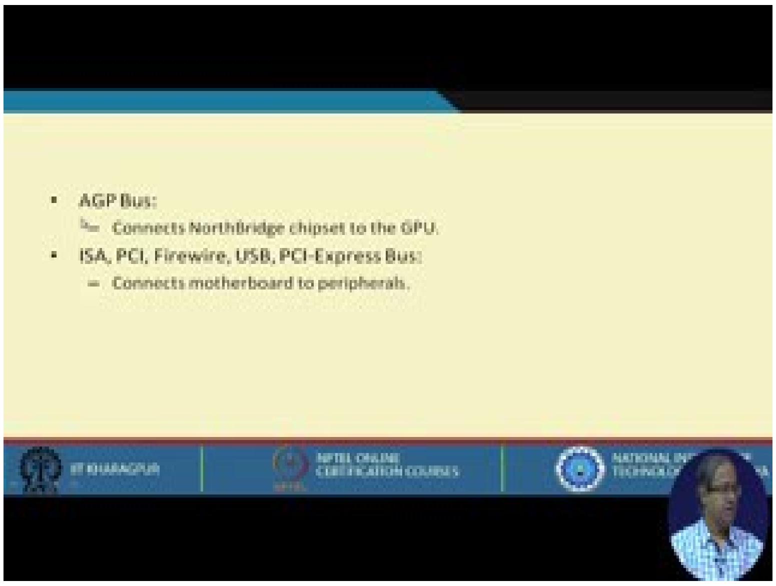


So, here we are talking about a connection like this. There is CPU, there is something that is called North Bridge and the CPU is connected to the north bridge by a bus that is a high speed bus. North Bridge may be connected to various high speed devices like memory, GPU, etc.

There can also be a backside bus that can connect CPU to the L2 cache. If it is outside the processor or level 3 cache, there can be a memory bus which connects north bridge chipset to the memory.

There are different kinds of devices that are connected there. There can be a backside bus also where the cpu is connected to L2 or L3 cache.

(Refer Slide Time: 11:58)

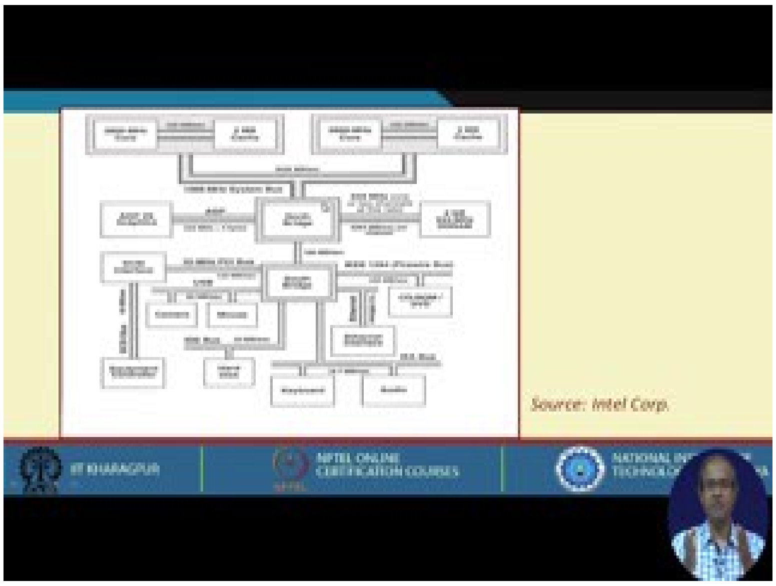


- AGP Bus:
  - Connects NorthBridge chipset to the GPU.
- ISA, PCI, Firewire, USB, PCI-Express Bus:
  - Connects motherboard to peripherals.

You have AGP bus. It refers to the bus through which the graphics cards are connected. Here the north bridge chipset is connected to GPU. These buses are referred to as AGP bus, and in addition there are several other buses inside CPU. There is another device which is called the South Bridge. The North Bridge is connected to the South Bridge and the South Bridge is relatively much slower in speed. South Bridge typically connects devices that are not that high speed.

Architecture wise you can see there are so many buses inside the system, and the devices are connected to one or more of these buses. Now, talking about some of the standards that evolved over the years, these are standards through which you can connect some peripherals to a computer system. It can be motherboard, it can be some connectors that are connected to the mother board. Well, ISA, PCI, Firewire, PCI express and today you have USB, these are all examples of bus standards that connects peripherals to the motherboards. In some way, they vary in speeds and other capabilities.

(Refer Slide Time: 13:51)



Now, let us look at a typical architecture of a motherboard inside a modern day PC. At the top I am assuming that this is a dual-core processor. There are two cores that are shown here. You see there is 3800 MHz clock. The core is running at 3.8 GHz, there is another core; and inside the core, there is 2 megabyte cache. This is the L1 cache and here you can see is the north bridge.

When you buy a computer system or when you look at the specification of PC, you will see that it does not only talk about the processor and the clock, it also uses a code name for a chipset.

Chipset actually refers to the north bridge and the south bridge combination, what are its capabilities, what are its speeds and so on. A chipset will determine that. So, this north bridge and south bridge taken together, this is the chipset you can refer to.

You see the north bridge, how it is connected. On one side, it is connected to the core, the processors. This is the high-speed bus. As you can see this runs at 1066 MHz. So, it is about 1 GHz speed and the data bandwidth is about 8528 MBps. So, it is pretty fast. North Bridge connects to main memory via the memory bus. In this example memory bus is running at 533 MHz and it can transfer data 4264 MBps. This is DDR RAM at 533 MHz.

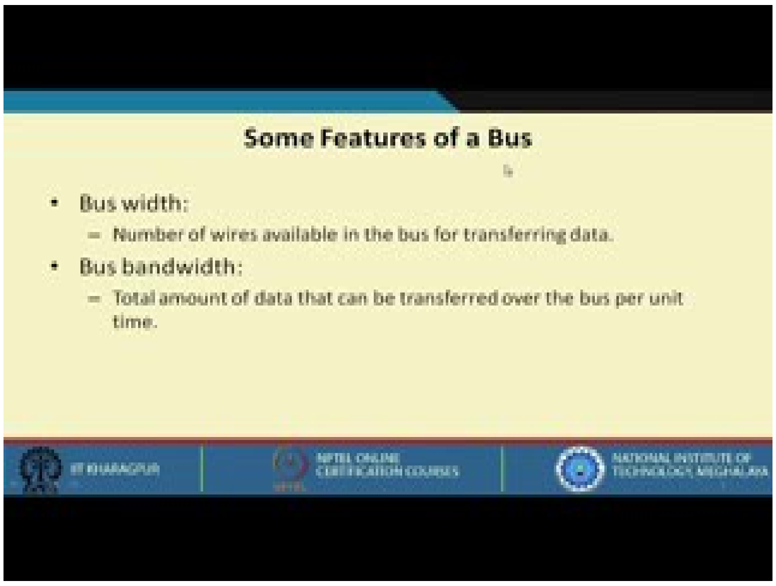
From the other side of the north bridge through AGP bus, you have the graphics card. North Bridge is connected to the south bridge through a relatively lower speed link. This is 100 MBps.

Now, south bridge connects to the other devices like it can have IEEE 1394 interface that is sometimes called firewire. Of course, now firewire has become obsolete. In modern computers, you do not see firewire any more. You can connect devices like CDROM, DVD through this. You can have a network adapter card, which also can be connected through it. You can have the slow systems like keyboard and audio. This can have a much slower bandwidth. You can have a hard disk connected through either IDE interface or SCSI, or here you can have USB interfaces, where you can connect various sort of devices.

The point to note is that in modern systems as the standards evolve, USB standard has become more widely used and faster. So, there are USB versions which instead of being connected to the south bridge, they are connected directly to the north bridge because of their much higher speeds.

So, architecturally this is what that exists inside CPU, motherboard and you can see there are so many different buses that exist there and each of this bus will have their protocols, their rules. We are not going into the detail of all the buses, rather some of the basic concepts I will try to highlight that will be common to most of these buses.

(Refer Slide Time: 18:12)



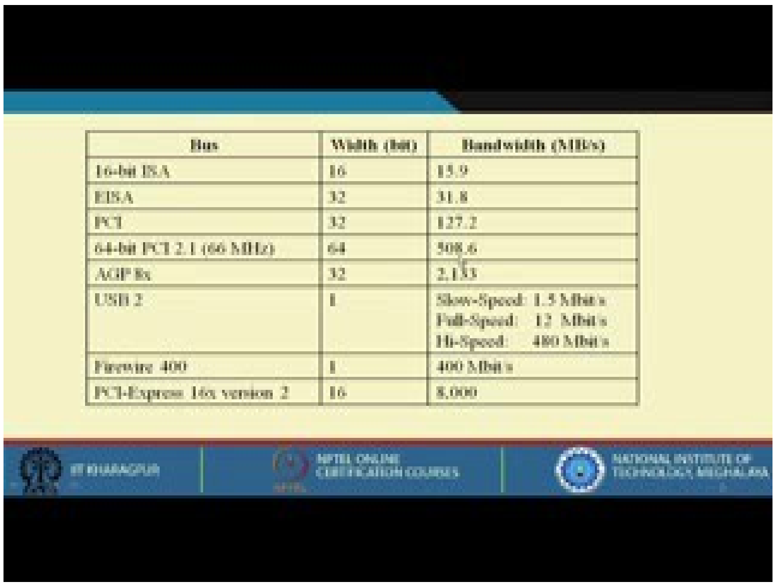
Some Features of a Bus

- Bus width:
  - Number of wires available in the bus for transferring data.
- Bus bandwidth:
  - Total amount of data that can be transferred over the bus per unit time.

Talking about a bus, some of the characteristic features are bus width, meaning how many data lines are available, how many bits of data you can transfer in every cycle, and the speed of the bus or the bandwidth. It is total amount of data in bits per second or bytes per second that can be transferred over the bus.

Let us look at these features, bus width and bandwidth for some of the commonly used bus standards.

(Refer Slide Time: 18:44)



Bus	Width (bit)	Bandwidth (MB/s)
16-bit ISA	16	15.9
EISA	32	31.8
PCI	32	127.2
64-bit PCI 2.1 (66 MHz)	64	508.6
AGP 8x	32	2,133
USB 2	1	Slow-Speed: 1.5 Mbit/s Full-Speed: 12 Mbit/s Hi-Speed: 480 Mbit/s
Firewire 400	1	400 Mbit/s
PCI-Express 16x version 2	16	8,000

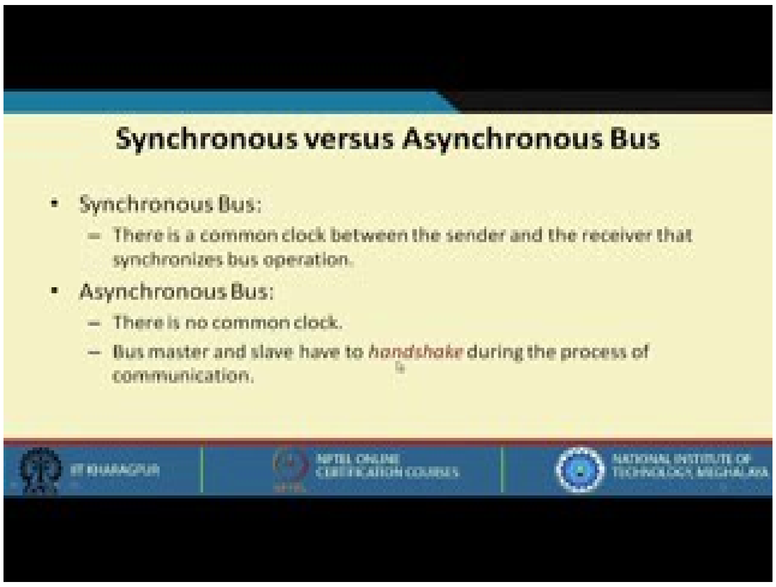
Some of these you cannot see anymore; they have become obsolete. You see the ISA bus was 16-bit bus that carried 16 wires and the bandwidth was 15.9 MBps. This was extended to EISA that is 32 bits at 31.8 MBps. PCI was also 32 bits, but the speed was about four times. Then, 64 bit PCI came version 2.1 where the bus width was 64. So, the speed was again increased significantly.

So, you see that the parallel buses exist more inside a system within the mother board, but today when you connect an external peripheral with the computer because of the cost of the cabling, the data communication is mostly serial in nature. But inside the mother board as you can see, these buses they are all carrying 16, 32, or 64 bits in parallel.

The graphics AGP bus is 32-bit, but it runs much faster at 2133 MBps. Various versions of USB are there.

Firewire was supposed to be one of the fast standards. This is also serial. This can go up to 400 Mbps, and PCI express is one of the fastest buses available. It carries 16 wires and the speed is 8000 MBps.

(Refer Slide Time: 21:05)



Synchronous versus Asynchronous Bus

- Synchronous Bus:
  - There is a common clock between the sender and the receiver that synchronizes bus operation.
- Asynchronous Bus:
  - There is no common clock.
  - Bus master and slave have to *handshake* during the process of communication.

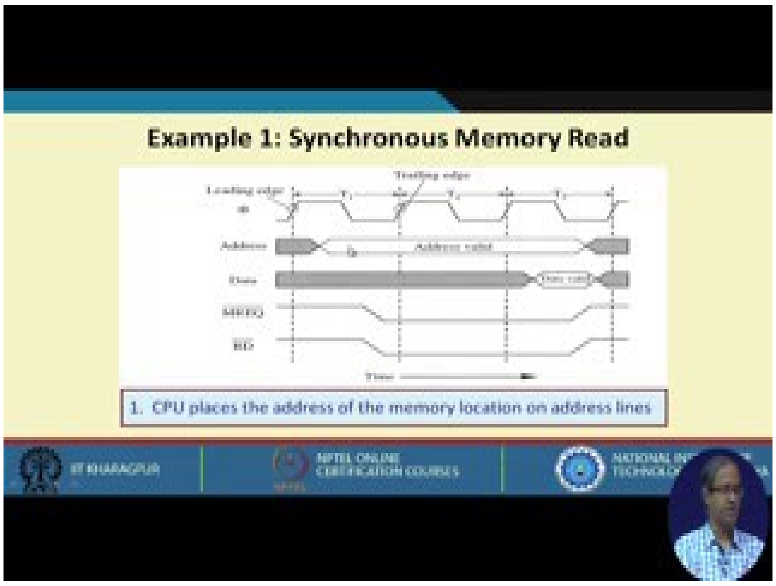
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Buses can be synchronous and asynchronous. I will take an example to illustrate the difference between synchronous and asynchronous. With respect to IO transfer, you have already seen. The concepts are very similar for a synchronous bus. There will be some kind of a common clock between the sender and the receiver that will synchronize all data transfer operation over the bus.

In contrast an asynchronous bus does not have any common clock and just like in asynchronous IO we had to use handshaking, here also there will be a number of handshaking signals the master and slave will be sending each other, so that the data communication can be completed and both sides will be knowing about it.

So, let us take an example.

(Refer Slide Time: 21:57)

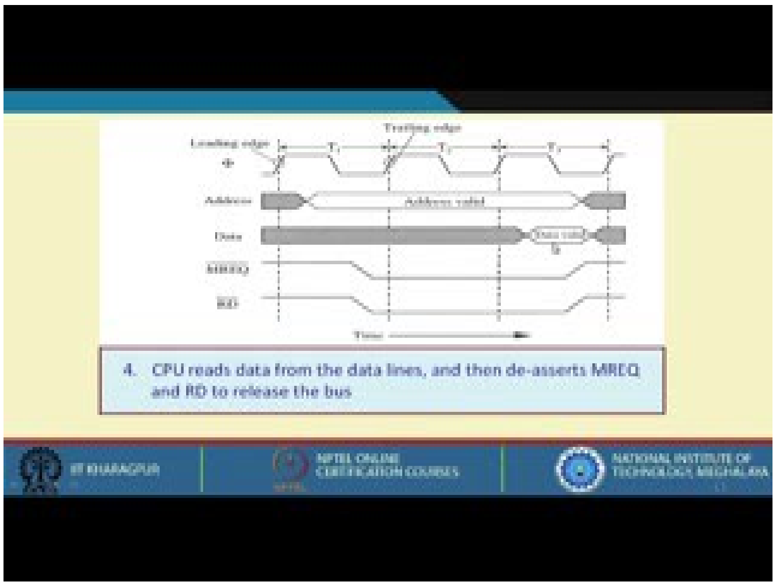

Example 1: Synchronous Memory Read

1. CPU places the address of the memory location on address lines

This is an example of a bus that connects CPU and the memory. The first example we take is that of a synchronous memory read. We are showing the signal timing diagram. First one shows the clock and these are the clock states T1 T2 T3. This is the address bus, data bus. These are two control signals, memory request and read.

In the first step CPU places the address of the memory location on the address lines. It is here in the first clock. After the clock rises high, there will be some delay. After that delay here, this part, the CPU puts the address of the memory location in the address line. During this time, none of the memory request and read lines are active. The bar means they are active low. They will be active when they are put to low or 0. So, first step is to put the address.

(Refer Slide Time: 23:06)



4. CPU reads data from the data lines, and then de-asserts MREQ and RD to release the bus

After the address is put, second step is CPU will assert the memory request, and read lines. Memory request line will be set to low and read line will also be set to low, so that the memory system will now know that it is a memory request, and this CPU is requesting read. At the end of T1, these two signals have been asserted. This address is already valid.

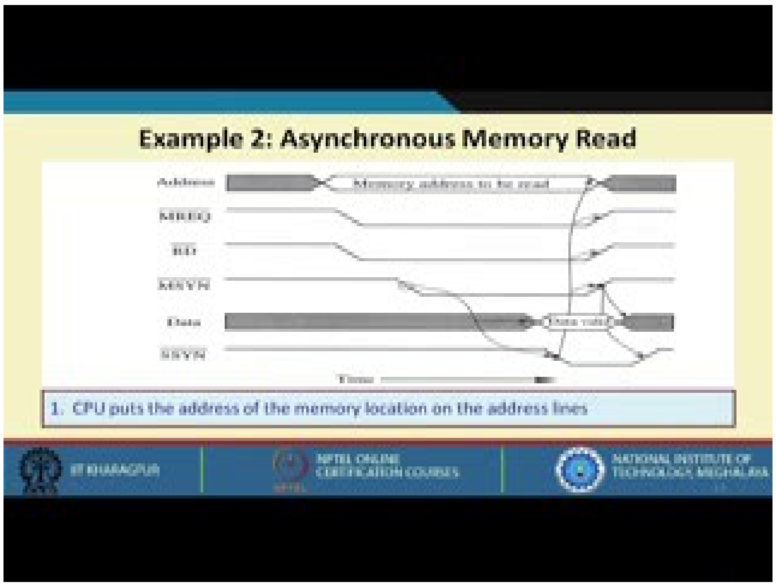
In the third step, memory controller will be accessing memory location and load the data on the data lines. Now, memory will be having some access time. Let say the read signal has activated here, and it will take so much time and it is here when the valid data will come on the data bus. So, you are giving sufficient time for the memory to access the contents.

The data will be loaded on the data line here, somewhere in T3. When the data has already arrived, sorry the last step will be CPU will be reading the data because it is already on the data line and it will deassert memory request and read lines because it is already done. So, memory request will again be set to high, read will again be set to high.

This is synchronous because CPU knows exactly how much time memory will take to read the data. It is waiting for exactly that much time. It will expect the data is already there. It will read from the data bus. So, this is what is meant by synchronous interfacing.

CPU puts the address on the address bus. It activates read and memory request signal because it is synchronous, and it knows exactly how much time memory will take. Just after that much time, it reads the data from the data bus and again de-asserts memory request and read. But for asynchronous interface, CPU may not know in advance exactly how much time memory is taking. So, there has to be some signal back from the memory indicating that data transfer or memory access is complete.

(Refer Slide Time: 25:36)



Example 2: Asynchronous Memory Read

1. CPU puts the address of the memory location on the address lines

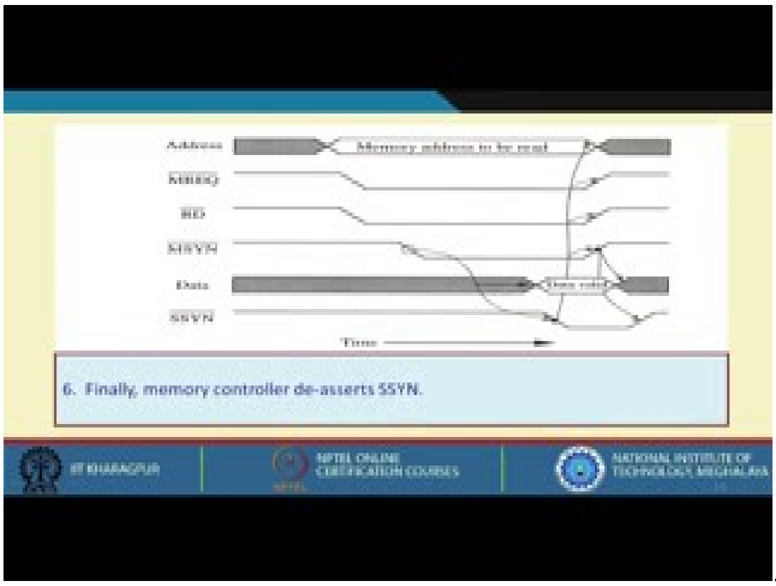
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY MEGHALAYA

This example is that of asynchronous memory read. I am not showing the clock signal because earlier everything was happening in synchronism with the clock. We are assuming both CPU and the memory system were having access to the same clock, but for asynchronous system here, we are assuming that there is no clock. There are other handshaking signals using which CPU and the memory, both will know exactly what is going on.

There are address lines, data lines and there are some other signals. You see memory request read ,this MSYN and SSYN. These are synchronization signals.

The first step as usual will be for CPU to put the address of memory location on the address lines. So, a valid address is available on the address lines.

(Refer Slide Time: 27:01)


Address
Memory address to be read
MREQ
RD
MSYN
Data
Data valid
SSYN
Time
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

6. Finally, memory controller de-asserts SSYN.

.

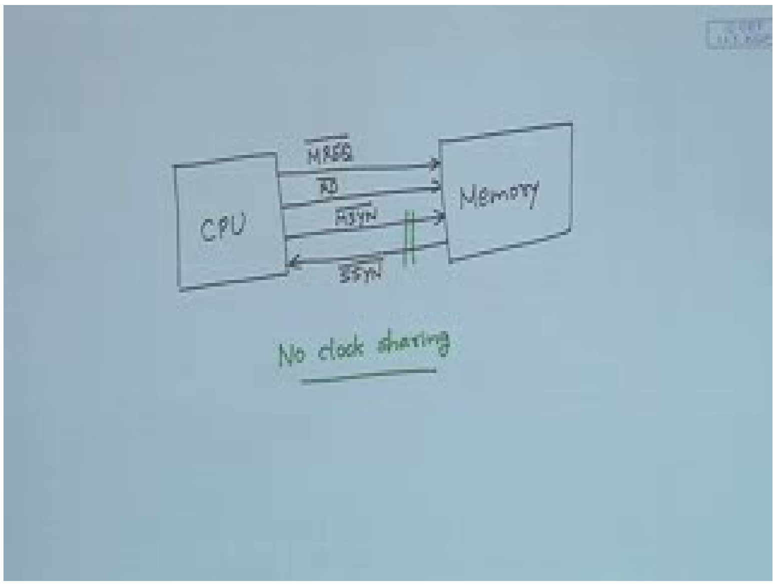
After the address has become stable, this symbol means it has become stable. It is here stable, CPU will assert memory request and read lines. These are two signals that CPU is sending to the memory system. So, memory request line is activated and read line is activated. This will be after a little delay, after the address lines have been stabilized. Once these two have been activated, now memory read process starts.

In the third step, what will happen is CPU will assert MSYN line. After memory request and read, now MSYN; this is the handshaking signal. MSYN will tell the memory system that CPU is now expecting some data from memory. This MSYN is the signal generated by CPU. It is asserted after memory request and read are asserting.

Then, memory controller will take some delay depending on the access time. This can less and that can be more. So it is not fixed; load the data from the data line. After some delay, the data will come on the data line and once it is valid, the memory controller will activate SSYN. The signal that comes from the memory controller to the CPU, when it sees that SSYN is activated, it will know that data is now available on the data bus.

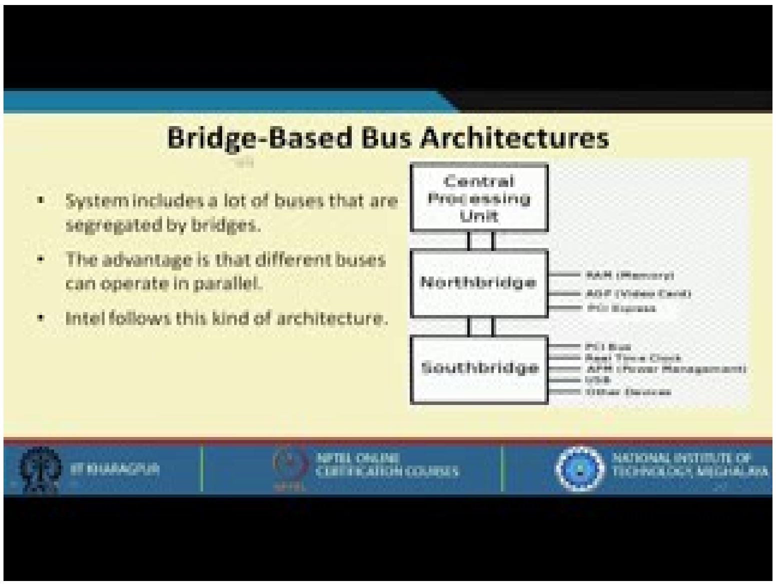
Now what CPU will do is, it will read the data from the data bus and will de-assert memory request, read and MSYN. And lastly, memory controller will de-assert a SSYN after all these lines have done indicating that the operation is over.

(Refer Slide Time: 29:08)



Essentially in this method if you see, you have CPU and you have the memory system. Actually the memory controller is interacting. You have the signals that are used for memory request read. Then, there are two handshaking signals, CPU sending to memory the MSYN signal and memory sending back to CPU the ASYN signal. Using these handshaking signals, data transfer can take place even without CPU and memory hearing a clock. This is the point to be noted.

(Refer Slide Time: 30:13)



Bridge-Based Bus Architectures

- System includes a lot of buses that are segregated by bridges.
- The advantage is that different buses can operate in parallel.
- Intel follows this kind of architecture.

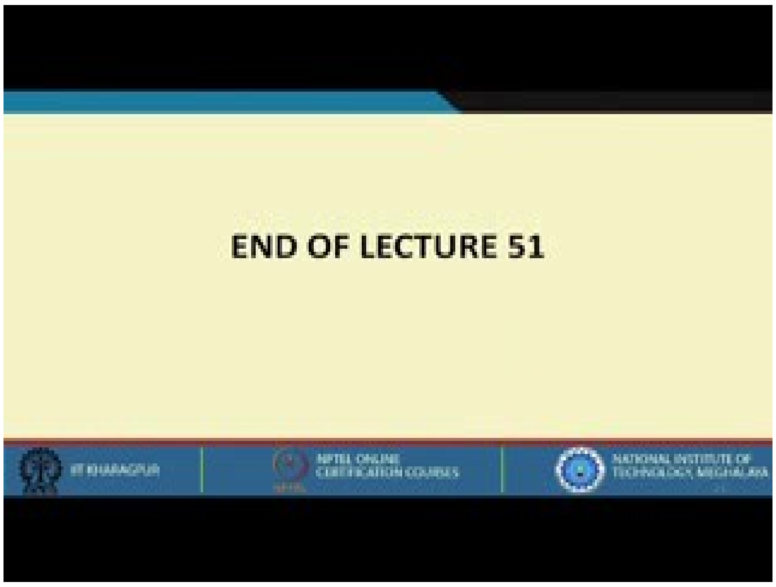
Central Processing Unit
Northbridge
RAM (Memory)
AGP (Video Card)
PCI Express
Southbridge
PCI Bus
Real Time Clock
APM (Power Management)
USB
Other Devices
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

As I have given the example of Intel motherboard, bus architectures are typically based on the bridges, this bridge concept which Intel had proposed. Many of the other manufacturers are also going by that. Here system includes a lot of buses that are not all connected together. They are segregated by bridges like you recall I mentioned the north bridge will be connecting RAM, memory, video card, AGP, PCI express, the highest speed buses.

The south bridge will typically connect the PCU that is the lower speed, PCI clock, USB and other devices like keyboard, disk etc. So, Intel and also other companies have started following this kind of bridge based bus architecture.

So, this is the diagram that we showed earlier. This is an example of a bridge based bus architecture, north bridge south bridge. This is a very commonly used architecture that is available in the desktops and laptops that we mostly see today around us.

(Refer Slide Time: 31:22)



END OF LECTURE 51

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

With this we come to the end of this lecture. In this lecture, we have talked about the need of buses inside a computer system, why we have so many different buses. Because each bus has a specific requirement, their speeds can be varying widely. So, instead of all devices connected to the same bus, it is always better to segregate the buses and by isolating them using bridges, we can have a better management.

In the next lecture, we shall be looking at one specific bus standard in some detail which is really becoming universal in today's context, that is Universal Serial Bus or USB.

Thank you.