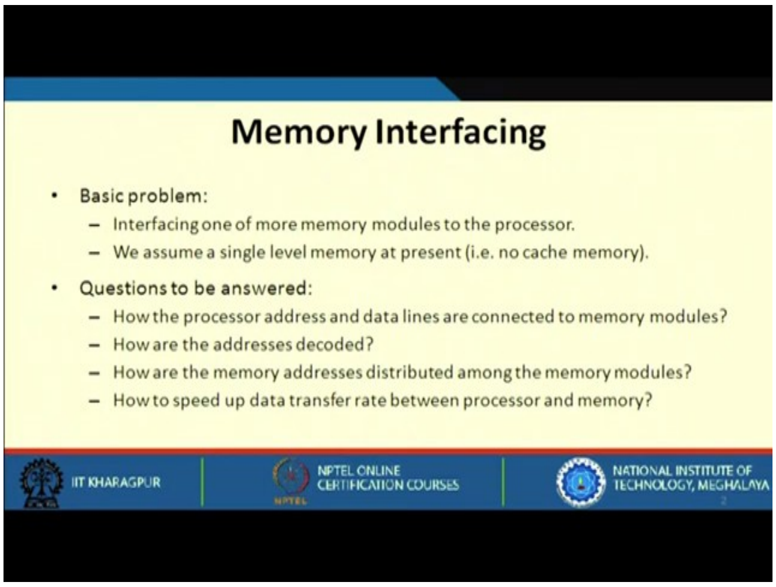
Computer Architecture and Organization
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture - 27
Memory Interfacing And Addressing

Welcome to lecture 27 on Memory Interfacing and Addressing. Till now we came to know how semiconductor memory is built what are the technologies used. In this lecture we will see that we have a memory chip available to us, but we want to built a larger memory system, how we can do that. This is one thing we will be seeing in this lecture. Another thing that we will be looking into is that how we can further increase the speed of this memory.

(Refer Slide Time: 01:06)



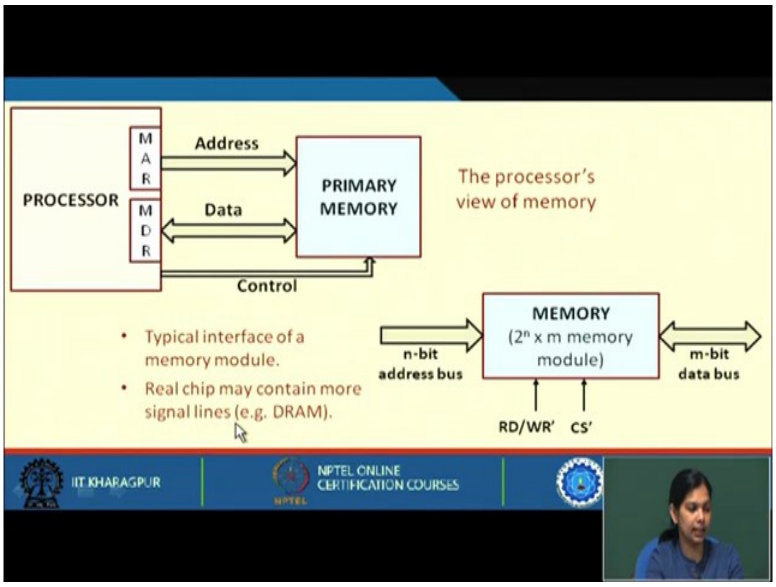
Memory Interfacing

- Basic problem:
  - Interfacing one of more memory modules to the processor.
  - We assume a single level memory at present (i.e. no cache memory).
- Questions to be answered:
  - How the processor address and data lines are connected to memory modules?
  - How are the addresses decoded?
  - How are the memory addresses distributed among the memory modules?
  - How to speed up data transfer rate between processor and memory?

Memory interfacing is the basic problem of how we can interface one or more memory modules to the processor. Here we assume that we have a single level of memory at present, so no cache memory for the time being. The question that is to be answered here is how the processor address and data lines are connected to the memory modules. Because now we are saying that we will not be having one single memory chip, rather multiple memory chips to make that memory, how the address and data lines from the processor will be connected to this. How are the addresses decoded, how the decoding of the address will take place? Because here you have to select and module and then select an address within that module. How are the memory addresses distributed among the memory modules that also we will be looking into, and how to speed up data transfer read between processor and memory?

So, these are the four things that we will be seeing in this lecture.

(Refer Slide Time: 02:27)

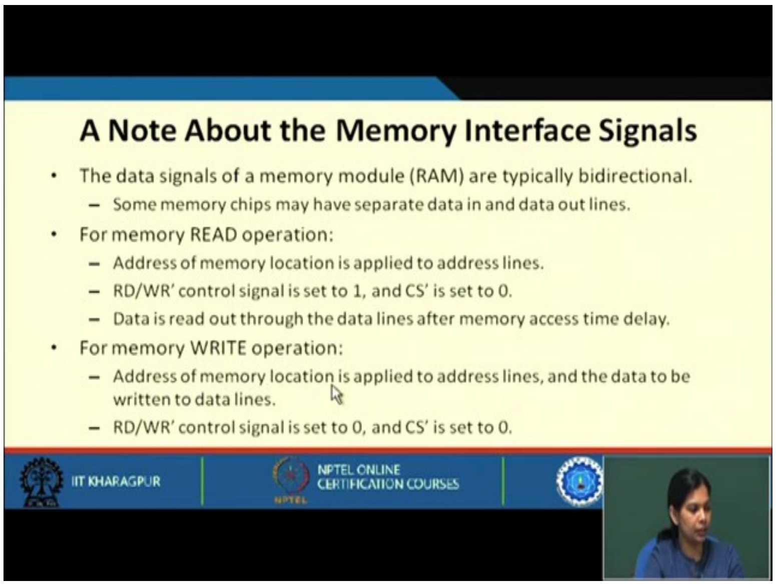


- Typical interface of a memory module.
- Real chip may contain more signal lines (e.g. DRAM).

As we know that for an n-bit address bus we can have 2^n memory locations, and m data bus we have read write and control signal. And now this memory address register and memory buffer register connected through address and data lines to the primary memory. So, this is the processor's view of the memory, and of course, we have some control lines.

Now we will be seeing that now this primary memory or whatever memory we are talking about will be connected with this address line and this data line. So, typical interface of the memory module real chip may contain more signal lines, so that is what we need to see.

(Refer Slide Time: 03:25)



A Note About the Memory Interface Signals

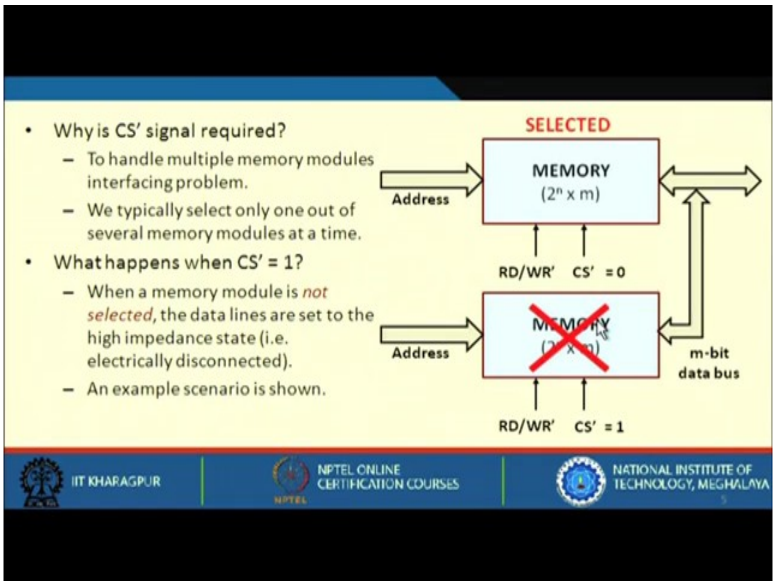
- The data signals of a memory module (RAM) are typically bidirectional.
  - Some memory chips may have separate data in and data out lines.
- For memory READ operation:
  - Address of memory location is applied to address lines.
  - RD/WR' control signal is set to 1, and CS' is set to 0.
  - Data is read out through the data lines after memory access time delay.
- For memory WRITE operation:
  - Address of memory location is applied to address lines, and the data to be written to data lines.
  - RD/WR' control signal is set to 0, and CS' is set to 0.

A note about this memory interface signal is that the data signals of the memory module are typically bidirectional, we have seen that the data bus is bidirectional because from memory also, some data can be passed to the data bus and written into memory, and from memory we read a data that comes to the data bus. So, it is bidirectional.

Some of the memory chips may have separate data in and data out lines. For a memory read operation what happens? The address of the memory location is applied to the address lines. Then for reading the signal this read/write signal will be set to 1 and the chip select is set to 0. Data is read out through the data lines after memory access time delay.

Similarly for a write operation, the address of the memory location is applied to the address line. The data to be written has to be written to put on the data lines. And then this read/write control signal is set to 0, and chip select is set to 0 again.

(Refer Slide Time: 05:17)



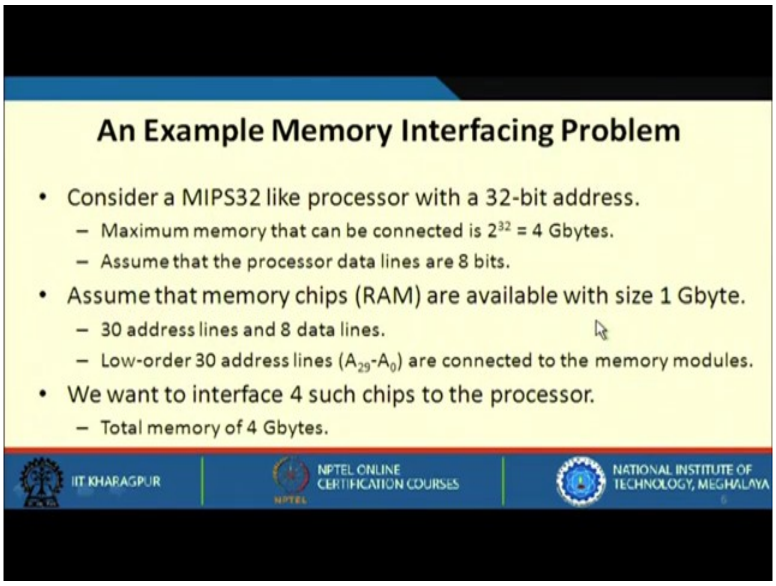
- Why is CS' signal required?
  - To handle multiple memory modules interfacing problem.
  - We typically select only one out of several memory modules at a time.
- What happens when CS' = 1?
  - When a memory module is *not selected*, the data lines are set to the high impedance state (i.e. electrically disconnected).
  - An example scenario is shown.

SELECTED
MEMORY
(2^n x m)
Address
RD/WR'
CS' = 0
Address
RD/WR'
CS' = 1
m-bit data bus
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So, why is this chip select signal required? It is not required if you have a single module. It will be required only when you have multiple modules and you have to select one particular module from multiple modules. What happens when chip select is 1?

So, when the chip select line is 1, then the memory module is not selected and the data lines are set to high impedance state; that is electrically disconnected. Let us see an example scenario. Here chip select is 0 so memory module is selected, and here this chip select is 1; so this memory module is not selected. Let us take an interfacing problem where we consider that.

(Refer Slide Time: 06:18)



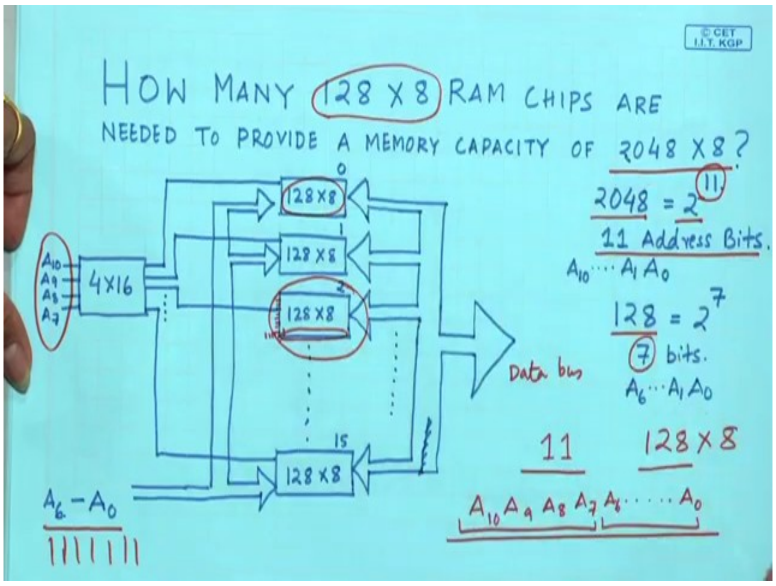
An Example Memory Interfacing Problem

- Consider a MIPS32 like processor with a 32-bit address.
  - Maximum memory that can be connected is 2^{32} = 4 Gbytes.
  - Assume that the processor data lines are 8 bits.
- Assume that memory chips (RAM) are available with size 1 Gbyte.
  - 30 address lines and 8 data lines.
  - Low-order 30 address lines (A_{29}-A_0) are connected to the memory modules.
- We want to interface 4 such chips to the processor.
  - Total memory of 4 Gbytes.

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

So, before taking an interfacing problem I will take a small example to show how a larger memory is built from smaller memory.

(Refer Slide Time: 06:38)



Let us take this example. In this example we have to find out how many 128 x 8 memory RAM chips are needed to provide a memory capacity of 2048 x 8. Basically all you need to do is that you will divide this by this and you will get the number of memory chips required to build that; that is coming to (2048 x 8) / (128 x 8) = 16. So, we require 16 chips to built up 2048 x 8 memory.

Now, 2048 is your total memory that is 2^{11}, so we need to have 11 bits total in the memory address. So, total address bit is A0 to A10. Now each module you have a smaller chip which is 128 x 8. So, this module is 128, that is 2^{7}. So, here you require 7 bits to select a particular row. So, once a particular row is selected then this set of 8 bits can be transferred, because one chip is 128 x 8. So, the total address bit is now 11.

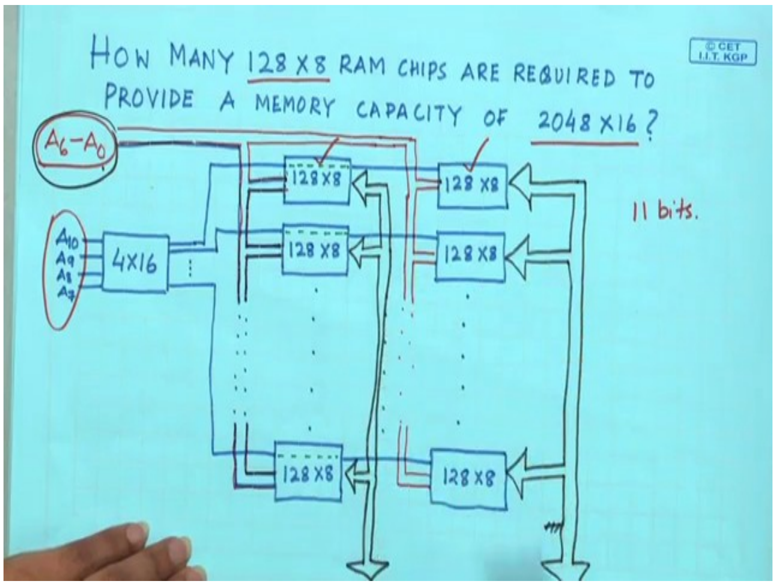
How this total address bit will be divided? We have already seen that, total address bit now here it is 11 we need to select a particular row first. In this case how many rows are there, how many chips are there in total? We have 16 total chips. So, we require a 4 x 16 decoder to select a particular chip. So, the high order 4 bits that is A10 to A7. So, the total address bit will be A10, A9 to A0 this is the total address bit 11. The high order 4 bits A10 to A7 will be connected to this 4 x 16 decoder that will select one of these 16 chips that is available. So, we have total 16 128 x 8, starting from 0 to 15. Now the output of this decoder is connected to all these chips.

Now see in the individual chip how it is organized it is organized as 128 x 8. So, the lower order 7 bits that is A0 to A6 will be connected to all the chips. So, we will first apply high order 4 bits of the total addresses to the decoder, it will decode and it will select any one of the chips, let us say this particular chip is getting selected.

Once this chip gets selected then we have to select one location from this chip, one particular row from this particular chip that will be A0 to A6. If it is the last location then the value will be 1111111. So, it will be this particular chip and the last location that is the location is 1111111.

So, this is the data bus connected here. So, this is how we can actually make a larger memory chip from a smaller memory module.

(Refer Slide Time: 12:14)


HOW MANY 128 X 8 RAM CHIPS ARE REQUIRED TO PROVIDE A MEMORY CAPACITY OF 2048 X 16 ?
© CET I.I.T. KGP
A_6-A_0
A_{10}
A_9
A_8
A_7
4X16
128 X8
11 bits.

We will take another example where we will see that we need a capacity of 2048 x 16 and we have a chip which is 128 x 8. So, we have a same 128 x 8 and we need a memory capacity of this much. So, if you divide this by this how many chips do you require? You will be requiring 32 chips. So, we require 32 128 x 8 chip to built a memory capacity of 2048 x 16. So, let us now see again there are 11 bits in the address, how those 11 bits will be organized. In the same way high order 4 bits will be connected to the decoder to select any one of the 16 chips, but we have a total of 32 chips.

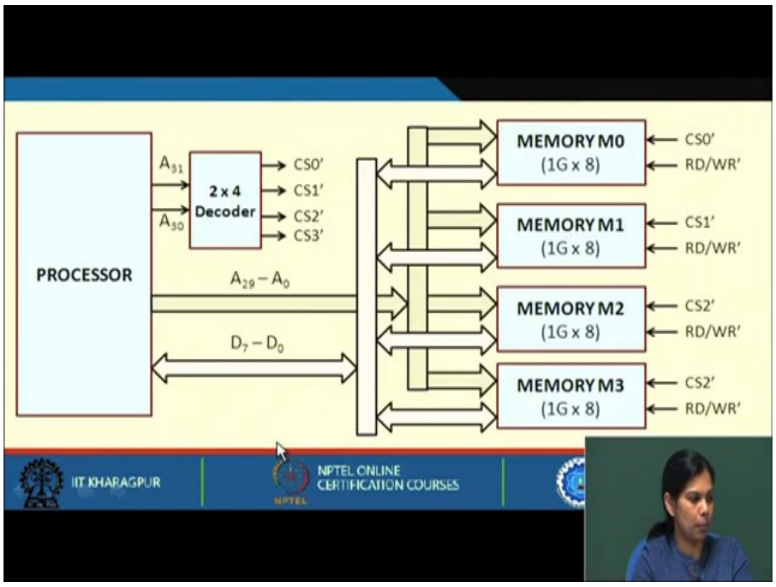
So, this decoder is now see it is connected to both the chips in the same row. So, this is a row in this particular row when I give 0 0 0 0 then this chip gets selected and also this chip gets selected simultaneously because both are connected. Now once we select this chip now I will apply the lower order bits, that is A0 to A6. So, whatever address we put in the lower order address bit it goes to this particular chip and it goes to this particular chip as well. So, from here a group of 8 bits will come out, and from here a group of 8 bits will come out.

So, this is how we can organize even larger memory from a smaller memory module. Now let us take this a little bigger example. So, here we will consider a MIPS like processor with 32-bit address; maximum memory that can be connected is 2^{32} that is 4 GB. And the assumption is that the processor data lines are 8 bit.

So remember this, the processor data line is 8 bit. Assume that the memory chips are available with size 1 GB and we need a maximum memory, we need to build a memory that is 4 GB. So, roughly how many chips will be required? We will be requiring four chips. For 1 GB, 30 address lines and 8 data lines are required. So, lower order 30 address line A0 to A29 are connected to the memory modules.

And now we want to interface for such chips to the processor. Let us see how we can do this.

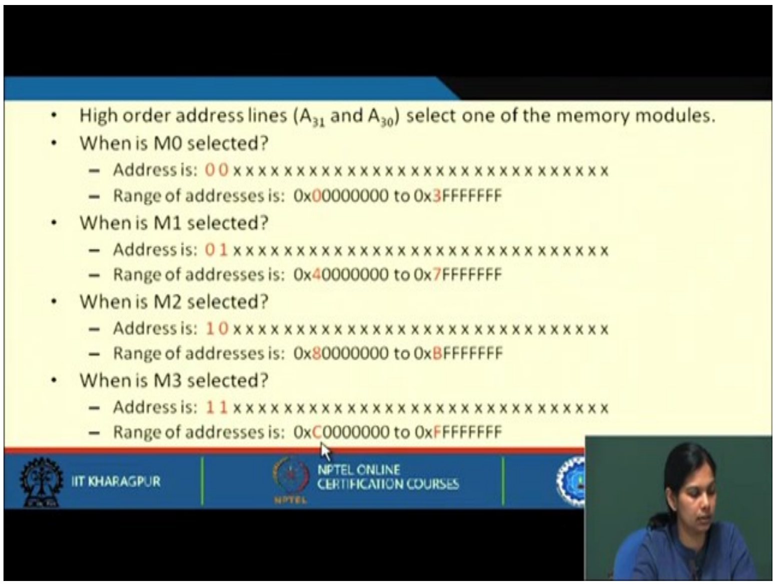
(Refer Slide Time: 16:44)



So, total memory of 4 GB we need to make. Let us see how we are doing here. So, we require 4 such chips. These are the 4 chips and then the 30-bit address A0 to A29 is connected to all the memory modules.

We need two more extra bits that is A30 and A31. So, a 2 x 4 decoder is required depending on these two values any one of the four chips will get selected. So, if you want to select chip 0, then it has to be 0 0, if you want to select chip 1, it has to be 0 1, and so on. So, this is how it is organized. And at a time when we select a particular memory module, then a group of 8 bits will be transferred through this data bus to the processor.

(Refer Slide Time: 18:11)



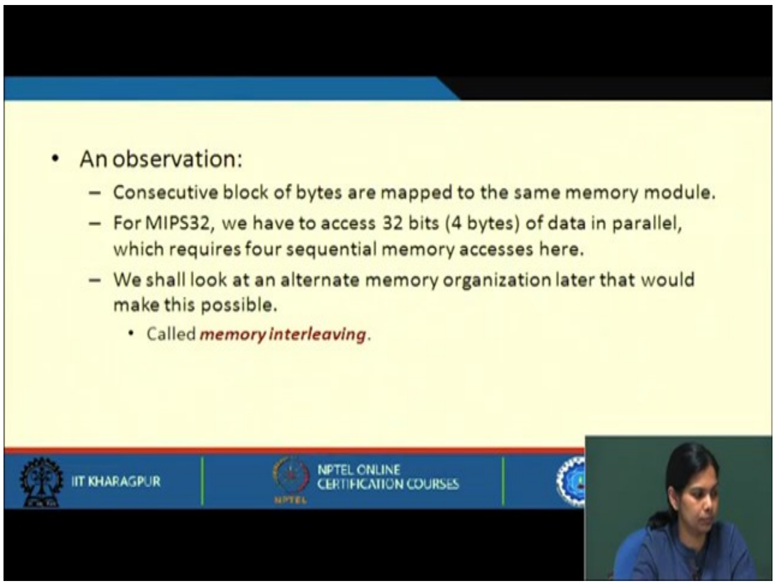
- High order address lines (A_{31} and A_{30}) select one of the memory modules.
- When is M0 selected?
  - Address is: 0 0 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0x00000000 to 0x3FFFFFFF
- When is M1 selected?
  - Address is: 0 1 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0x40000000 to 0x7FFFFFFF
- When is M2 selected?
  - Address is: 1 0 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0x80000000 to 0xBFFFFFFF
- When is M3 selected?
  - Address is: 1 1 x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x
  - Range of addresses is: 0xC0000000 to 0xFFFFFFFF

So, as I said higher order address line A30 and A31 selects one of the memory modules. So, what happens when M0 get selected? So, when M0 get selected the higher order address line will be 0 0, when this higher order address line is 0 0 then only memory module 0 is selected. So, what will be the range of address? So, the first address this will be 0 0 it is a 32 bit address these all are 30 bits and these two are 0 0. So, total 32 we can group it into hexadecimal digits.

So, what will be the range of address? So, as we know that this will be 0 0, first two bits will be 0 for all and the address range will actually start from here. So, till this, this will be 0 0 0 0 0 0 and the last address will be this last bit will be this will be 0 0 because this is the 0 th module and this will be 1 1 1 1.

So, what will be the range? The range will be 000000000 to 3FFFFFFF. This will be the range for module 0. Similarly for module 1 this will be 0 1, if this is 0 1 all will be 0 1s and all will be 1. So, if all is 0 then this will be 0 1 0 0 that is 4 first this set of 4 bits will be 4 so it will be 40000000 to 7FFFFFFF. Similarly when M2 is selected it will be 80000000 to BFFFFFFF. And similarly when M3 is selected it will be like this. So, the range of address will be like this.

(Refer Slide Time: 20:45)

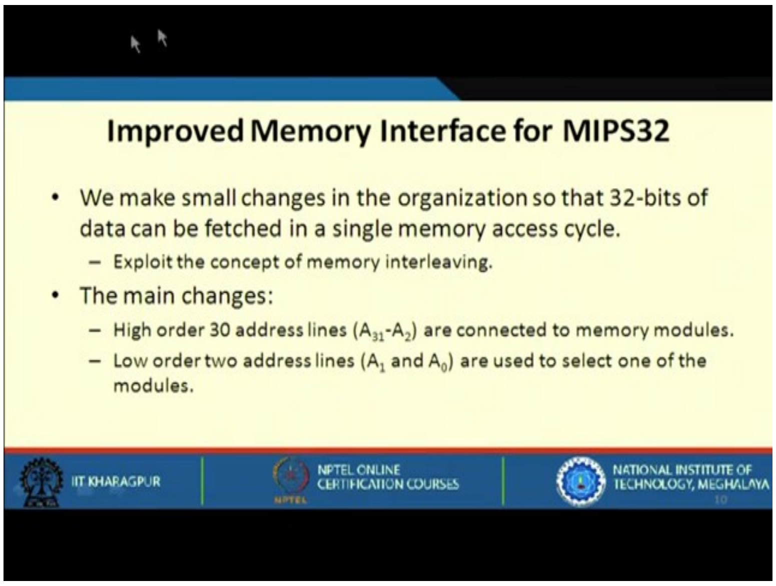


- An observation:
  - Consecutive block of bytes are mapped to the same memory module.
  - For MIPS32, we have to access 32 bits (4 bytes) of data in parallel, which requires four sequential memory accesses here.
  - We shall look at an alternate memory organization later that would make this possible.
    - Called *memory interleaving*.

We can observe one thing that the consecutive block of bytes is mapped to the same memory modules. For MIPS we have to access 32 bits, that is 4 bytes, of data in parallel. So to do that we require 4 sequential memory accesses, because it is 8 bit, so one-by-one-by-one we have to access it.

Now we shall look into an alternative memory organization if there is a way that we can do something such that all the modules can be selected at once and then the data can be transferred. This is called memory interleaving.

(Refer Slide Time: 21:45)



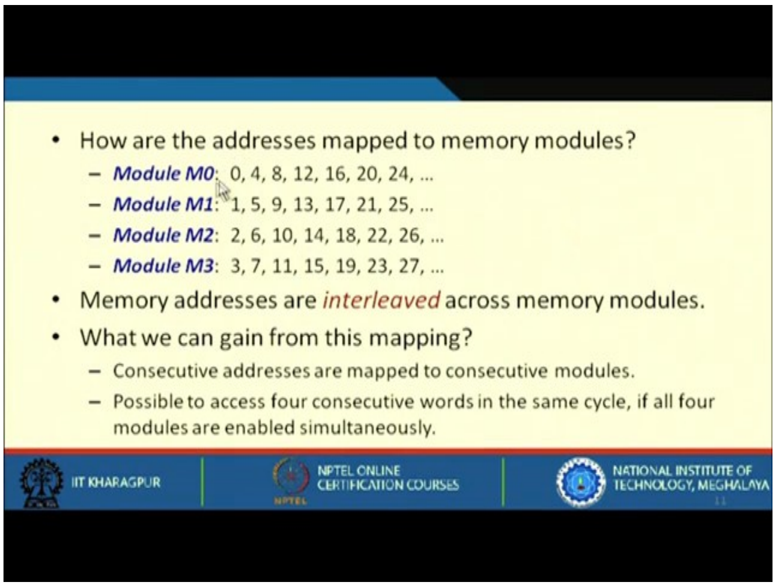
Improved Memory Interface for MIPS32

- We make small changes in the organization so that 32-bits of data can be fetched in a single memory access cycle.
  - Exploit the concept of memory interleaving.
- The main changes:
  - High order 30 address lines (A_{31}-A_2) are connected to memory modules.
  - Low order two address lines (A_1 and A_0) are used to select one of the modules.

We make a very small change here. So, we organize these 32 bits of data that can be fetched in a single memory access, we exploit this concept of memory interleaving. The main change that we do here is that high order 30 address lines that is A2 to A31 are connected to the memory modules. And the lower order to address lines that is A0 and A1 are used to select one of the modules.

Now earlier the higher order bits were used to select the module. Now we are using the lower order bits to use the module. So, what will happen how the addresses will be mapped?

(Refer Slide Time: 22:45)



- How are the addresses mapped to memory modules?
  - *Module M0*: 0, 4, 8, 12, 16, 20, 24, ...
  - *Module M1*: 1, 5, 9, 13, 17, 21, 25, ...
  - *Module M2*: 2, 6, 10, 14, 18, 22, 26, ...
  - *Module M3*: 3, 7, 11, 15, 19, 23, 27, ...
- Memory addresses are *interleaved* across memory modules.
- What we can gain from this mapping?
  - Consecutive addresses are mapped to consecutive modules.
  - Possible to access four consecutive words in the same cycle, if all four modules are enabled simultaneously.

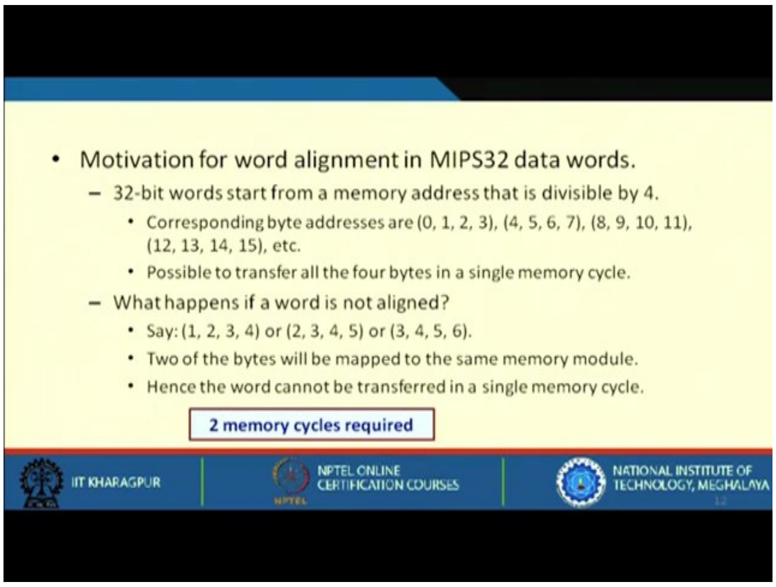
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So, in the module 0 it will be 0, 4, 8, 12, 16, 20, and 24. Module 1 it will be 1, 5, 9; so in a block of four words consecutive addresses are now mapped into separate modules. Now in module 0 earlier we were having 0, 1, 2, 3, 4 like this.

Here, we are having the consecutive words in consecutive modules. So, what advantage can we get if we have consecutive words in consecutive modules? If there is a way to select all the modules, and can transfer the data from all these modules simultaneously, then we can get 32-bit data at a time. First address of the memory is in module 0, the next address in next module, next address in next module, next address in next module. So, this is the way to represent; we call it memory addresses are interleaved across the memory modules.

So, what we can gain from this mapping? As I said if the consecutive addresses are mapped in consecutive modules then possibly we can access the four consecutive words in the same cycle. If all four modules are enabled simultaneously, let us say all four modules are enabled simultaneously. So, from this address you can access the word from next address, from next address, from next address. So, all four words which are there in these four different modules can be accessed together.

(Refer Slide Time: 25:02)



- Motivation for word alignment in MIPS32 data words.
  - 32-bit words start from a memory address that is divisible by 4.
    - Corresponding byte addresses are (0, 1, 2, 3), (4, 5, 6, 7), (8, 9, 10, 11), (12, 13, 14, 15), etc.
    - Possible to transfer all the four bytes in a single memory cycle.
  - What happens if a word is not aligned?
    - Say: (1, 2, 3, 4) or (2, 3, 4, 5) or (3, 4, 5, 6).
    - Two of the bytes will be mapped to the same memory module.
    - Hence the word cannot be transferred in a single memory cycle.

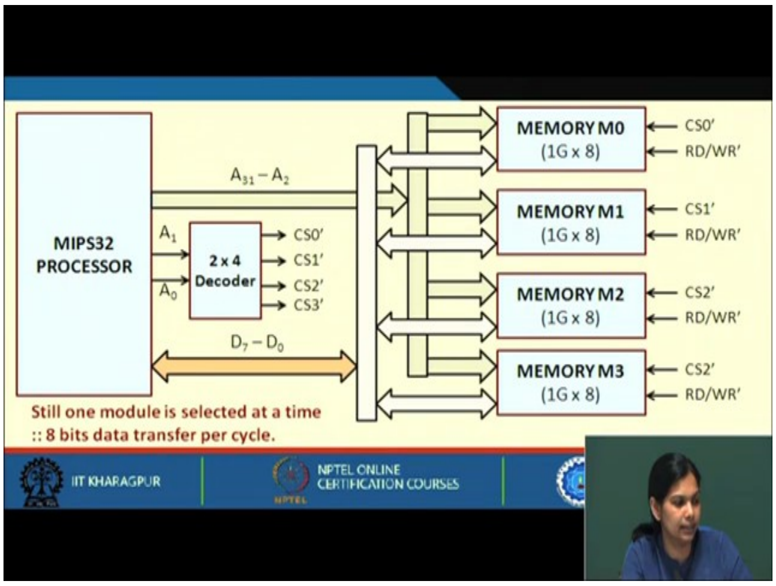
2 memory cycles required

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

So, now we can see the motivation for this word alignment in MIPS data word. If the words are not aligned what can happen. See the 32 bit words starts from a memory address that is divisible by 4. So, corresponding byte addresses are (0, 1, 2, 3); (4, 5, 6, 7); all these. So, these are in different modules, and it is possible to transfer all the four bytes in a single memory access. What happens if the words are not aligned? Let us say some words are starting from 1 some words are starting from 2 then what will happen? My words are in consecutive modules. So, (1, 2, 3, 4) I require; this will not help us, this is not aligned because it is not starting from 0.

So, in that case I need 0 as well, so to do that in one memory access I can get this, and in another memory access I also need to get this 0-th location. Hence, the word cannot be transferred in single memory access. It will be accessed in a single memory cycle only when it is aligned; if it is not aligned it will require more number of memory access, in this case two memory cycles will be required.

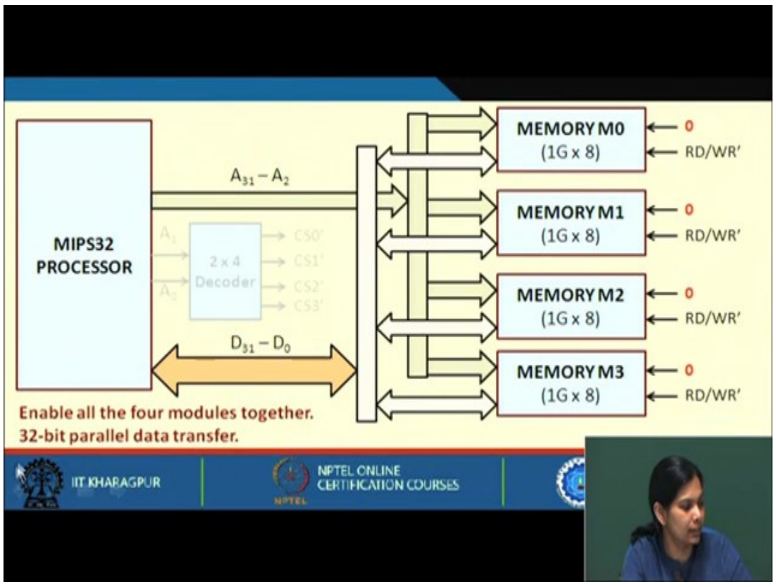
(Refer Slide Time: 26:35)



Still one module is selected at a time
:: 8 bits data transfer per cycle.

Now, let us see what we are saying. So, the higher order bits are connected to the address modules and the lower order bit is connected to the decoder; it is still connected to the decoder and through this decoder at a time one of the memory modules gets selected. Once one of the memory modules get selected then 8 bit of data is transferred. So, a bottleneck still exists that we have a data bus of 8 bit. Still one module is selected at a time and 8 bits data transfer per cycle is taking place.

(Refer Slide Time: 27:18)



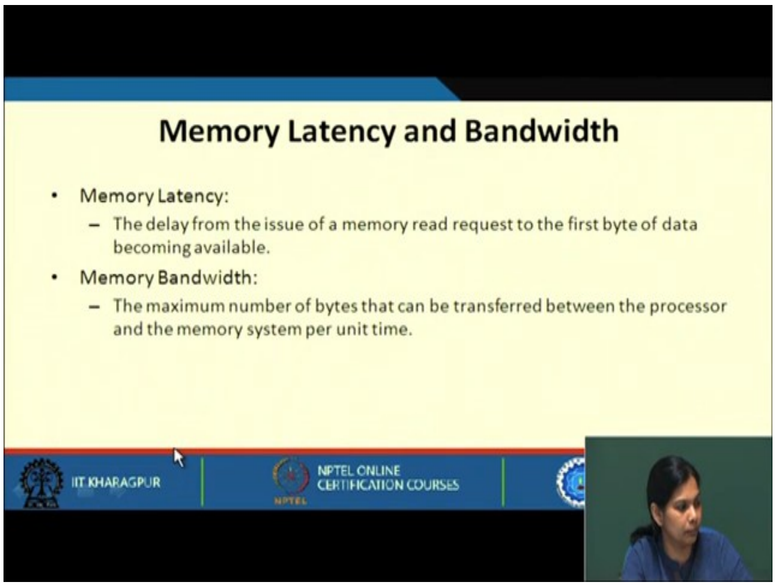
Enable all the four modules together.
32-bit parallel data transfer.

Now, let us say all these all modules are selected; all modules are selected at all times. And one more change we have done we have made this data bus as 32 bit. And the high order address is connected to all the memory modules. In all the memory modules the high order address is connected.

So, in this case what will happen? As all the modules are selected whenever we give that address first address then the data from all the four modules get selected and it will be transferred to this 32-bit data bus. So, it from this module 8 bit will come from this, this, this, and then it will be transferred to this 32-bit data bus to the processor. So, the advantage we get here is that all the modules are selected. And we apply a same address to get different data from different modules. It would not have been possible if the addresses are mapped in a single module. It is only possible because the addresses are interleaved across the modules.

So, it enables all four modules together, so 32 bit parallel data transfer is possible.

(Refer Slide Time: 28:55)

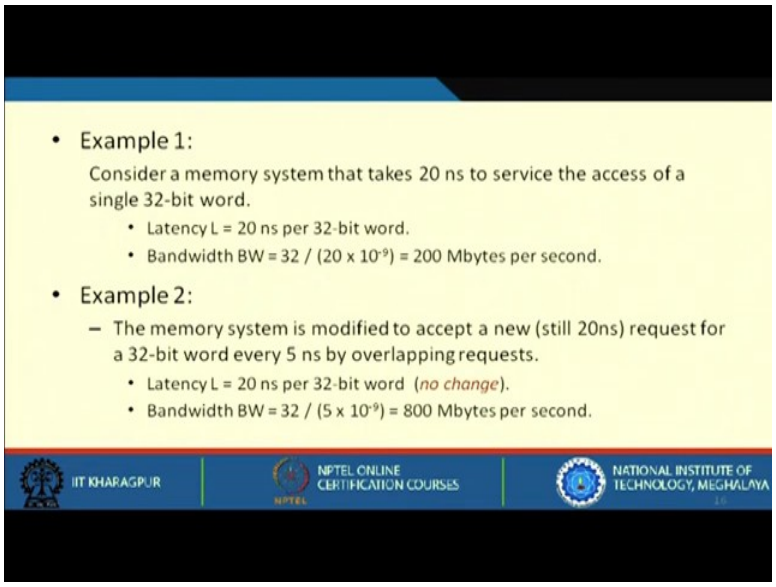

Memory Latency and Bandwidth

- Memory Latency:
  - The delay from the issue of a memory read request to the first byte of data becoming available.
- Memory Bandwidth:
  - The maximum number of bytes that can be transferred between the processor and the memory system per unit time.

We already discussed about this regarding memory latency and bandwidth, so we will take a small example to explain this. What is latency? Latency is the delay from the issue of the memory read request to the first byte of data becoming available. So, it is the time required to access the first data and then consecutive data can be accessed in a much quicker time.

What is memory bandwidth? The memory bandwidth is the maximum number of bytes that can be transferred between the processor and the memory system per unit time.

(Refer Slide Time: 29:58)

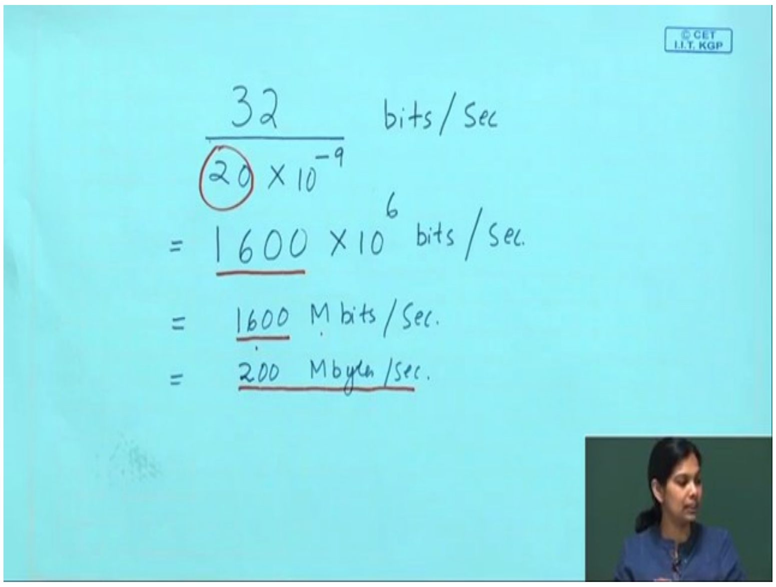


- Example 1:
  Consider a memory system that takes 20 ns to service the access of a single 32-bit word.
  - Latency L = 20 ns per 32-bit word.
  - Bandwidth BW = 32 / (20 x 10^{-9}) = 200 Mbytes per second.
- Example 2:
  - The memory system is modified to accept a new (still 20ns) request for a 32-bit word every 5 ns by overlapping requests.
    - Latency L = 20 ns per 32-bit word (*no change*).
    - Bandwidth BW = 32 / (5 x 10^{-9}) = 800 Mbytes per second.

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

Let us consider this example. So, here a memory system takes 20 nanosecond to service the access of a single 32-bit word. That means, 20 nanosecond is the latency; in 20 nanosecond it is transferring a single 32-bit word. What will be the bandwidth? Bandwidth will be 32 / (20 x 10-^{9}).

(Refer Slide Time: 30:36)



$$\frac{32}{20 \times 10^{-9}} \text{ bits/Sec}$$
$$= \underline{1600} \times 10^{6} \text{ bits/Sec.}$$
$$= \underline{1600} \text{ M bits/Sec.}$$
$$= \underline{200 \text{ Mbytes/Sec.}}$$

So, you can just see this that latency is 20 nanosecond. This is the bandwidth so you can just do a simple calculation which is coming down to 1600 megabits per second. Finally, if you divide it by 8 you will get 200 megabytes per second. So, the bandwidth is 200 megabytes per second.

Let us take another example.

Here the memory system is little bit modified to accept a new, still this 20 nanosecond request, for a 32 bit word every 5 nanosecond by overlapping the request. Now the next word, next word, next word is overlapped; that means every 5 nanosecond the next word is available. So, how we can change this? So, the latency will be still 20 nanosecond per word so there is no change there, but now the bandwidth can be increased, because after every 5 nanosecond the next, next, next word is available. So, it is 32 divided by 5 into 10 to the power minus 9 which is coming down to 800 megabytes per second.

So, we have seen some of the examples, and what is latency and bandwidth, and how are they important in the context of memory system design. So now, we came to the end of lecture 27. In the next lecture we will be looking into how we can further make the memory faster. So, we will be moving on with memory hierarchy, cache memory, virtual memory, etc.

Thank you.