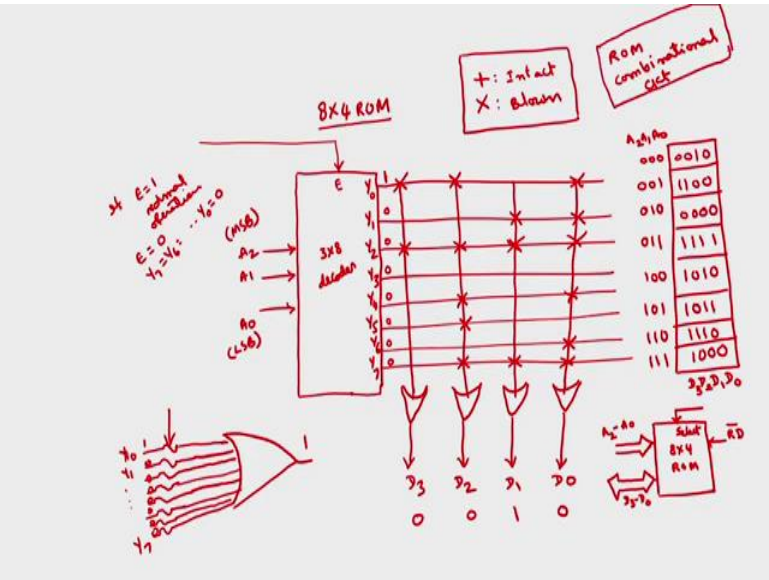
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture – 15
ROM, RAM

In the last class, we are discussing about this read only memory. So, if I take the architecture of 8 by 4 ROM, Read Only Memory, so we did 8 will be the output of the decoder. So, we need a 3 to 8 decoder. So, here this is A_2 A_1A_0. A_0 is LSB, A_2 is MSB.

(Refer Slide Time: 00:36)



There are 7 outputs Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7. And 4 means we require four 8 input OR gates. D_3, D_2, D_1, D_0. There are 8 inputs. Depending upon the data which is to be stored in each location, we will keep the fuse intact or will be blown. Suppose, if you want to store in 8 locations, whose address is these are 8 locations. So, in the location 000, A_2, A_1, A_0. So, these bits are here I am going to store D_3, D_2, D_1, D_0 4 bits 001, 010, 011, 100, 101, 110 and 111. So, I want to take store some data you can randomly take say some 0010, 1100, 0000, 1111, 1010, 1011, 1110, 1000, suppose I want to store this data in corresponding locations, ok.

So, in 000 I want to store 0010, ok. So, D_3 should be 0, D_2 should be 0, D_1 should be 1, D_0 should be 0, this is due to a decoder. So, if I give A_2, A_1, A_0 as 000, then Y_0 will be 1 remaining all will be 0s. Then if I keep all the connections intact, this will represent

intact, this is the notation I am going to follow and this cross represents blown; means fuse is broken.

If I keep all the connections intact, then what will be the output? Because these are all 8 inputs I have shown as a single line, but all these 4 OR gates are 8 input OR gates. The inputs are connected to fuses, this is 8 input AND gate, OR gate, this is Y_0 connected through fuse, Y_1 connected through fuse, Y_2 through fuse, Y_3 through fuse, Y_4, Y_5, Y_6 and Y_7. This is Y_0, Y_1, so on up to Y_7.

So, this fuse can be programmed, this can be keep intact, if I keep intact, if this is 1 0 0 0 0 0 0 0, then output will be 1, for OR gate if 1 input is 1, output is 1, ok. So, if I keep all the connections intact then the output will be 1111. But I want to store 0010. Only here I want 1, so remaining three I want this as 000; because I want to store 0010 in first location whose address is A_2, A_1, A_0 is equal to 000, ok.

For that all the connections of this D_1, whose output is D_1 has to be intact; whereas, for this D_3 the OR gate you have to break this connection. If I do not break this connection what happens; output will be 1, but I want output as 0. Anyhow the remaining inputs are 0s, even if I keep intact, that is not going to I mean produce the output as 1, the only the input with 1 has to be broken, ok. You have to break these.

Similarly, D_2 also I want 0, so you have to break this connection and D_1 we will keep as it is. D_0 I have to break because I want 0. So, whenever if I use 000 here the output data will be 0010. Now, if I give the second address 001, the data that I want to store in this 001 is 1100. So, Y_1 will be 1, this A_2, A_1, A_0 is equal to 001, Y_1 is 1. If I keep this Y_1 connection to all the 4 OR gates output will be 1111, but I want the output as 1100, so you have to break the connections of D_1 and D_0 OR gates so that D_1 and D_0 become 00, so these two has to be broken. Remaining connections are all intact. If I give this address as 001 data output will be 1100.

Similarly, for third address 010 I want 0000, so all the connections has to be broken. And for the next address 011 A_2, A_1, A_0 equal to 011. I want the output as 1111, so I have to keep all the connections intact. Then for the next address 100, I want 1010. So, this Y_4 output has to be broken for this D_2 and D_0, D_2 and D_0. Similarly, if I give address lines A_2, A_1, A_0 as 101, 101, I want the data as 1011, so the only the second D_2 OR gate connection has to be broken, Y_5 this connection.

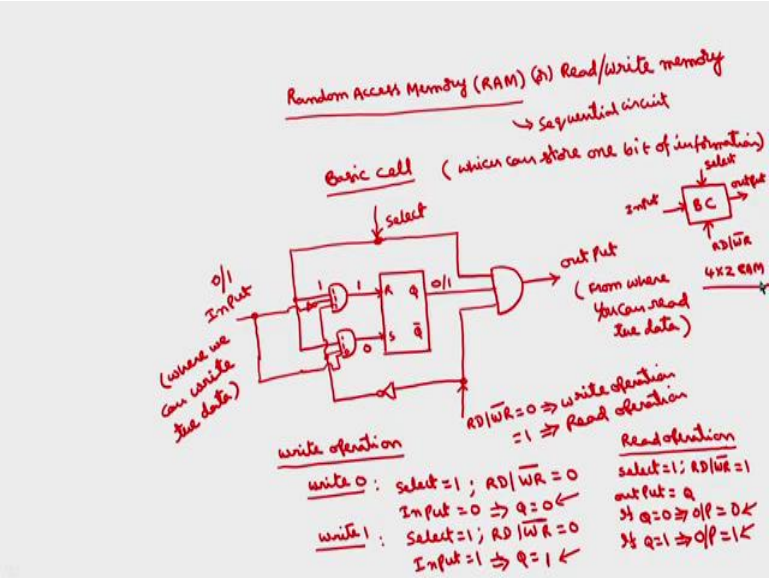
Then 110, I want to store 1110. So, the only the last OR gate connection has to be broken. And for 111 the last address I want to store 1000, so only the first connection has to be intact, the remaining three has to be broken. So, this is about the storing of the data into the read only memory. This will be done at the time of manufacturing itself. So, this fuse has to be keep intact or blown that depends upon the data that I want to store in the each location.

Here if I do not want to select this IC at all, so for that I can use a enable line for the decoder, that is enable line E. If E is equal to 1, normal operation, if E is equal to 0 all the outputs Y_7 is equal to Y_6, so on up to Y_0 is equal to 0 means. All the outputs will be 0s, so output of the OR gate becomes 0000 nothing can be stored. So, I can show this is I mean 8 by 4 ROM, accordingly you can draw the circuit diagram of any size of the ROM.

So, I can represent this with a simple block diagram. 8 by 4 ROM means, 8 by 4 ROM; what are the various signals available for 8 by 4 ROM? We have three address lines; address lines will be represented by unidirectional. This is the representation of the bus, A_2 to A_0, three address lines and we have this 4 represents number of data lines. Data bus is bi-directional, so we have D_3 to D_0.

And then one select signal which enable, that you can call as select signal one select signal is available to select this particular IC. If select signal is 0 we cannot perform any operation in this 8 by 4 ROM. So, the primary requirement is you have to first select this ROM, then you can perform any operation, then we will be having read bar signal, you can read only memory. The name implies there is a signal called read bar signal. These are the various signals available for the 8 by 4 ROM.

(Refer Slide Time: 11:59)



Next we will discuss about the Random Access Memory, RAM. Remember that read only memory. So, what are the circuitry in this read only memory? Only an array of OR gates, an array of AND gates. Inside this decoder we will be having array of AND gates, AND gates and OR gates. There is no memory storage element, so that is why read only memory is a combinational circuit. ROM is combinational circuit. And there is no feedback also. Whereas, random access memory can be built by using flip flops. So, this is a sequential circuit.

So, basic cell, first we are going to discuss about a basic cell which can store 1 bit of the information, basic memory cell which can store 1 bit of the information. So, this random access memory can also be called as read write memory. Another name for this one is read write memory. We can perform read operation as well as write operation, ok.

So, how can you write 1 bit of the information? How can you read the 1 bit of the information? First I will discuss the basic cell. After that using the basic cells, I can construct any big memory sizes, ok. So, 1 bit memory basic cell will be consisting of basically SR flip flop. This is R, S, Q, Q bar. And there will be one select signal to select this particular select and there will be some output. This is read by write bar, you can perform either read operation or write operation, and this is 0 because write bar is complement this is write operation. If this is equal to 1, which implies read operation.

Similarly, you should have some input from where you can give the input data and there will be some output from where you can read the data. This is the output where you can read the data, and there will be some input, where you can give the data. So, basically I have 3 AND gates, 3 input, 2 AND gates. So, this select signal will be common to all the gates, similarly read by write bar also will be given to all the inputs. This is read by write bar signal and there will be input signal, which will be connected through inverter to this AND gate and directly connected to the other AND gate. This is input, where you can write the data. This is the complete 1 bit basic memory cell. We have one input signal, one output signal, one select signal, one read by write bar signal, ok.

Now, the operation is if we want to perform write operation first, I will discuss about the write operation. You can write either 0 or 1, say write 0. So, write 0, so what are the different signal? The basic first thing is select should be 1, otherwise this cell will not be selected, select should be 1. Then what about read by write bar? Because write operation read by write bar must be 0. And to write 0, we have to make input is equal to 0.

Now, what happens if input is 0, select is 1, read by write bar is 0? So, what happens to output Q? So, for this AND gate, what are the 3 inputs? For the first AND gate the 3 inputs are select is 1, input is 0, but after the inverter this will be 1; so, this will be 1, this will be 1 and this is read by write bar is 0 after the inverter this is also 1, 3 inputs are 1. And what about this? Input is 0, so input this is 0 and select is read by write bar is 0, but through NOT gate, this will be 1, this will be 1. So, the output of this AND gate will be because 3 inputs are 1, this is 1, this is 0. For S is equal to 0, R is equal to 1. What will be the Q? Reset input is 1, so output will be reset. This is 0. Implies Q is equal to 0. So, Q is equal to 0 means you are writing 0.

So, whatever the value of the Q, that is the information that you have written into the cell. Similarly, if you want to write 1, select each 1, read by write bar should be 0, but input will be 1. If you make input is equal to 0, a 0 is written, if input is 1, a 1 is written. Now, if I evaluate Q becomes because now this will be reverse. So, the output of the first AND gate becomes 0, second AND gate becomes 1, so Q becomes 1. This is how we can write 1 bit of the information. If we want to write a 0 you take a 0 here, if you want to write a 1 you take a 1 here. So, automatically Q becomes 0 or 1. This is how we can write.

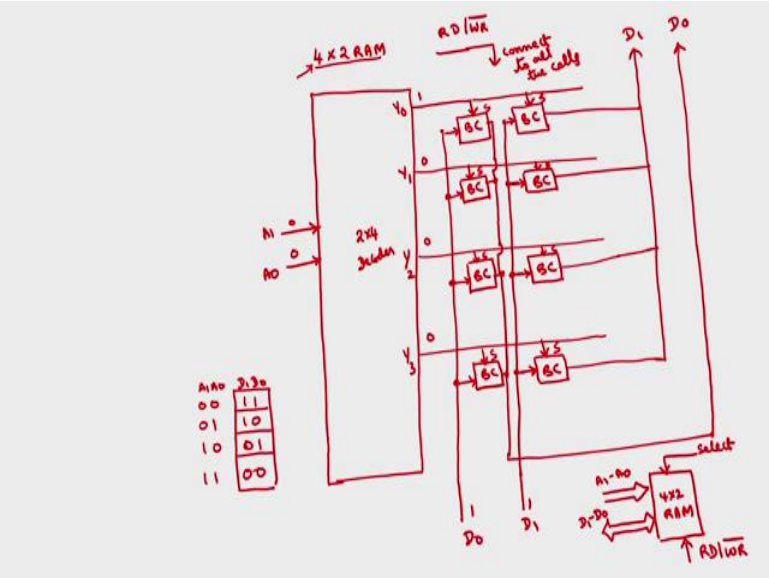
So, what about the read operation? For read operation also, so you have to read from this point output. Select should be 1, regardless of whether read operation or write operation select must be 1. Now, read by write bar should be 1; because this is a read operation. So, now, what happens to output of the AND gate? If Q is equal to 0 already 0 is written, I want to read that 0 information, ok. So, if Q is whatever this Q, I want to read that Q information, Q can be a 0 or 1, ok. So, for select is equal to 1 and read by write bar is equal to 1 what will be the output of the last AND gate output? Output is simply ending of select which is 1, read by write bar that is also 1 and then Q, this is simply Q.

So, if Q is equal to 0, output is 0. If Q is equal to 1, output is 1. So, you are reading 0 information, if Q is equal to 0 you are reading, one information if Q is equal to 1, ok. This is how we can read and write 1 bit of the information in the basic cell.

Now, what are the various signals available for the basic cell? So, this is a basic cell which we can use to construct the larger memories. Now, I will take the block diagram of these instead of drawing this circuit every time, I will call this one as basics cell BC. So, what are the different signals here? There will be one input signal through which we will give the data which is to be written, there will be output signal from where we will read the data, there will be one select signal to perform any operation first of all this cell has to be selected, then one read by write bar signal which decides the mode of operation, read or write.

Now, using this basic cell, we can construct any big memories. Suppose, if I want 4 by 2 RAM, 4 by 2 random access memory. So, at 4 locations each location is capable of carrying 2 bits. So, total how many bits will be there? 4 into 2, 8. So, we require total 8 basic cells, ok. How these 8 basic cells has to be connected? I will show in the next slide. So, basically here also regardless of read only memory or random access memory the first block is a decoder.

(Refer Slide Time: 23:39)



So, if I want 4 by 2 RAM random access memory, the only difference is the way in which the data is going to be stored. In case of read only memory, the data is going to be stored by programming the fuses of the OR gate, whereas here we are going to store in a basic cell which consist of flip flop. So, anyhow this 4 locations means you have to use 2 by 4 decoder.

This is 2 by 4 decoder, two address lines A_1, A_0; A_1 is MSB, A_0 is LSB and 4 outputs Y_0, Y_1, Y_2, Y_3 and total we will be having 8 basic cells; because 4 into 2 is total 8 bits, so we need 8 basic cells. Here 2 basic cells, this is one basic cell, we have 4 signals, here 2 basic cells, here 2 and here 2.

Now, we know that there are 4 signals, one is select signal. So, select signal of this has to be connected to the outputs. Here this Y_0 has to be connected to select of this BC as well as this BC, ok. This is select signal. I will call select as S. Similarly, this has to be connected to select of this BC this has to connected to select of this BC. Here this select, select. Here also this will select this BC, this BC. Then we have input; input will has to be there here, here also this input.

So, there will be one common signal, two input signals. They are all connected together. This is one input signal, this is another input signal. There are two input signals. And there are two output signals connected together, this is one output signal and this is another output signal. Those are all that has (Refer Time: 28:31) to be connected

together. You can connect here, otherwise you can connect here itself. So, no need of this 7 now. I will connect here itself this.

This is output of the second cell, this I will call as D_1, this I will call as D_0; because data bus is bidirectional and this is for the first cell. So, I will call as this one as input signals, so these two are the input signals. This is D_1, D_0. This is bidirectional data bus. Then we will be having input output select signal read by write bar signal. Read by write bar is common to all the cells. There is one read by write bar signal connect to, I am not showing that connection. Read by write signal is common to all the 8 cells, ok.

Now, what will be the block diagram of this 4 by 2 RAM? 4 by 2 RAM, 2 address lines which are unidirectional A_1, A_0; 2 data lines bidirectional D_1, D_0, 1 select signal corresponding to each of these memory cells select signal then read by write bar signal. This is a D_0, D_1, this will be to give the data writing of the data, so this will be this direction and for reading the data this will be this direction, ok.

Now, suppose if I want to store in 4 by 2 locations, 4 locations are there. This is the address A_1, A_0, inside this we have D_1, D_0 data lines. So, in address lines 00, so I want to store 11 any data, ok. In 01 I want to store say 10, in 10 I want to store 01, in 11 I want to store 00, ok. So, if I want to store 11, like in case of read only memory that I have explained, in the earlier slide. So, if A_1 and A_0 are both are 00, A_1 and A_0 are both are 00, so I want to store 11. So, if this is 00, what happened? Y_0 is equal to 1 remaining all will be 0s. So, these three cells will not be selected at all, these three cells will not be selected at all. So, only these two cells will be selected, these two cells will be selected. So, and I have to give 11, you simply give 11 here. So, that this 1 will be written into this, this 1 will written into this this.

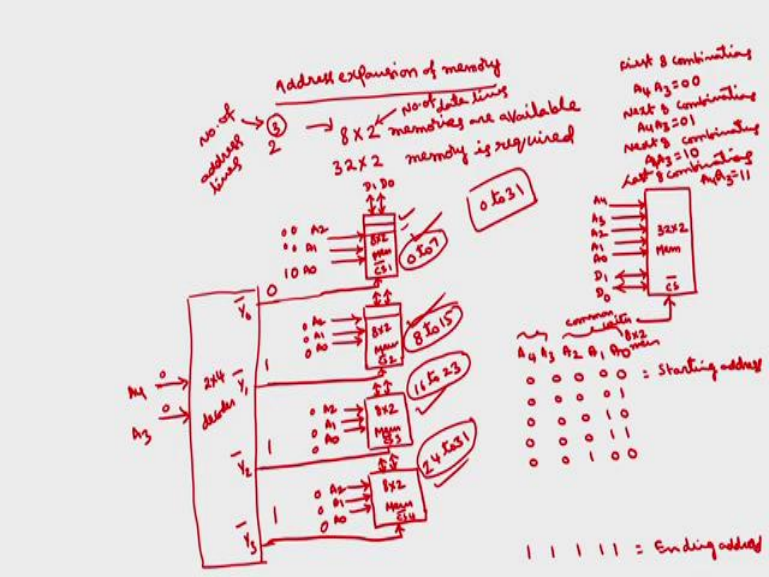
Now, for 01, Y_1 is selected, ok. So, you write here 10, at D_0 and D_1 you have to write 10. And similarly, if I give 10 address and you also, write this data as 01 at D_0 and D_1. Similarly, for 11 you write the data at D_0 D_1 as 00, so that that will be stored. And if you want to read, you can read from that output of that particular each cell, ok. This is how you can read and write the information into 4 by 2 cell. This is about the random access memory.

So, with this we will go for the expansion of the memories. So, we have discussed about the basic structures of random access memory and read only memory, now we will see

regardless of whether its read only memory or random access memory how to extend the size, because if I take 8086, 1 megabyte of the memory we can connect to the 8086 that is the maximum memory capacity. But if 1 megabyte of the memory IC is not available I have only 1 kilobytes of the memory ICs, then we can connect several such 1 kilobytes of the memories to make the total memory as 1 megabyte, ok.

So, suppose for example, if I have only 1 kilobyte of the memory, so I want some 4 kilobytes of the memory. So, I can use 4 such chips, ok. So, that is what is called address expansion. So, you can expand the data also. Suppose, if I have only 1024 by 2 memories means 1024 location each location is capable of storing only 2 bits of the data, but I want 1024 by 8. So, I want the data as 8 bits. Then you have to expand the data or you can expand both address as well as data, ok. First I will discuss about the address expansion of the memories.

(Refer Slide Time: 34:40)



Means, it can be any memory read only memory or read write memory. Suppose, you will be given 8 by 2 memory are available and I want 32 by 2 memory, how many such 8 by 2 memories are required is first question. And second question is, how to interconnect this 8 by 2 memories to make 32 by 2 memory.

First you draw the block diagram of both, ok. So, I have 8 by 2 memory will be; so, this is direct that. So, we have 8 locations here, I want 32 locations, ok. So, basically here this will be having the number of address lines of 8 by 2 memory will be, it can be read only

memory or random access they have same logic. So, 8 by 2 how many address lines will be there? The number of address lines is equal to you have to express this 8 as 2 raised to the power of something that something represents number of address lines.

Normally, we will have the number of locations power of 2, and these directly represent number of data lines. So, this will be having three address lines. So, this is A_2, A_1, A_0; A_2, A_1, A_0. I am writing individually instead of bus notation; A_2, A_1, A_0 and if I want 32 by 2 memory and this will be having two data lines D_1, D_0 one select signal.

Normally, chip select signal will be available as active low signal, I will call as chip select bar, ok. So, in the previous I mean examples we have discussed about the select as high, but normally this will be active low signal; because inside the decoder instead of using the AND gates normally you will be using NAND gates. If you use the NAND gate select signal should be active low signal.

So, reason for using the NAND gates is because they are universal gates. So, this will be having some chip select signal. And read by write bar that depends upon whether this memory is read or write memory, I am not showing that, the read by write memory signals will be available. So, 32 by 2 will be having how many address lines? We have 5 address lines A_4, A_3, A_2, A_1 and A_0, and two data lines bidirectional; D_1 D_0 and one chip select signal available as active low signal this is 8 by 2 memory, ok.

So, here we have 5 address lines, here we have 3 address lines, ok. So, how many such 8 by 2 memories are required to construct this? Ok. So, we have two more signals extra here, compared with this 8 by 2 memory, we have additional address lines are A_3 and A_4, ok. So, using A_3 and A_4 we can have 4 combinations. So, we require 4 such 8 by 2 memories so I will draw 4 such 8 by 2 memories, same A_2, A_1, A_0 will be there, D_1, D_0; A_2, A_1, A_0; D_1, D_0 and each one will be having its own select signal, chip select signal.

This is chip select signal of this, this is chip select signal of this, this is chip select signal of the last memory and assuming all are active low signals. So, this all the address lines of all the 4 has to be connected together, this A_2 you have to connect all the A_2s together, all A_1s together, all A_0s together. Similarly, D_1s and D_0s you have to connect together. So, this D_1, D_0 also you have to connect together.

Now, only thing is you have to have different chip selects. We can call this as cheap select 1, chip select 2, chip select 3, chip select 4. Now, if I take this 32 by 2 memory we will be having 5 address lines A_4, A_3, A_2, A_1, A_0. So, what is the starting address? Starting address will be all 0s. Then ending address will be all 1s. So, if I take this first 4 locations what happens here, this is 00001, 00010, 00011, 00100 and so on. If I take first 8 locations here, what will be this A_3, A_2? You have these extra signals, these are all common to all the 8 by 2 decoder, 8 by 2 I mean memory. These are common with 8 by 2 memory, these store the extra signals.

If I draw this total 32 combinations here, you can observe that in the first 8 combinations A_4, A_3 is equal to 00. So, in 32 we have four 8s. In next 8 combinations, A_4, A_3 is 01. We can draw this 32 combinations and you can check these. In the next 8 combinations, A_4, A_3, A_4, A_3 is equal to 10. Next, last I mean 8 combinations A_4, A_3 is equal to 11. Means, so I want to store the first 8 location in the first this 8 by 2 memory; in the next 8 location in next memory and the next 8 location in the last memory, ok. So, for that I require again another decoder.

So, I will take a decoder here I will take active low signal Y_0 bar, Y_1 bar, Y_2 bar, Y_3 bar. This is 2 by 4 decoder, 2 inputs; I will give the 2 inputs as A_4 and A_3 in between these two, A_4 is the MSB, A_3 is LSB, ok. If I use 00 here, so Y_0 will be selected, means selected means 0, not selected means 1. This is negative logic. If you have this I mean bubbles inside this if you have the NAND gate then this logic is called negative logic, selected means 0 not selected means 1s. If I use the AND gate, then you have to use a positive logic selected means 1, not selected means 0, ok.

So, here if I use 00 here, A_4, A_3 is equal to 00 Y_0 bar is selected thereby for the first 8 combinations of this total 32 combinations A_4, A_3 is 0; so, the first this one will be selected for the first 8 combinations. So remaining 3 will be disabled. So, if I give A_4, A_3 is equal to 01, what happens? The second A_2 two memory will be selected and for 10 the next one is selected, for 11 next is selected, ok. This is how we can select only one of these 8 by 2 memories for each combination of A_4 and A_3.

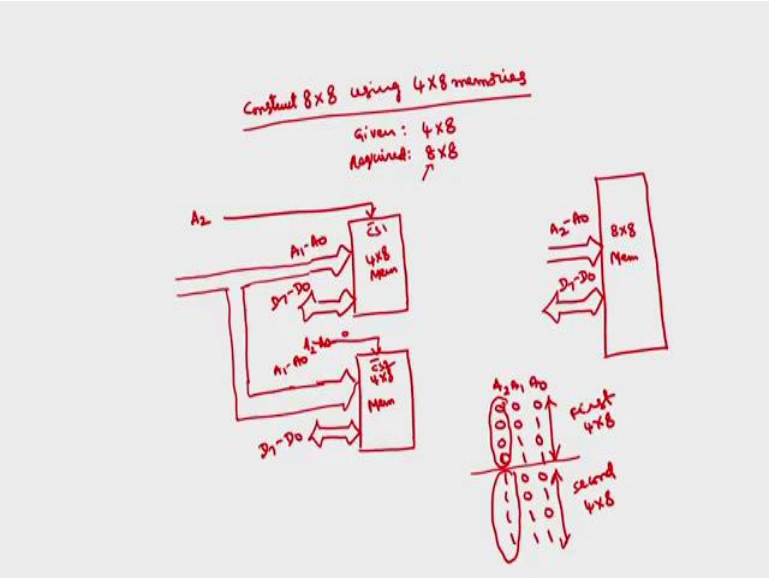
Now, after selecting this particular each 8 by 2 memory, how to select the particular location in that? For that we have A_2, A_1, A_0. If I take A_2, A_1, A_0 as 000 and if I select this memory the 0th location of this one will be selected. If I take 001 the first location of

this will be selected, but at any time only one of this 8 by 2 memories will be selected, but A_2, A_1, A_0 is common. Even if I give this 000, these 000 is common to this also 000s, this also 000, this also 000. But the chip select of the lower three, 8 by 2 memory is 1. So, they are they will not be selected. Even if I give this A_2, A_1, A_0 as 000, that 0th location of those memories will not be selected, they will be disabled. Only the 0th location of the first 8 by 2 memory will be selected because further chip select signal is 0. For the remaining chip select bar is 1 means chip is not at all selected, ok.

Similarly, for A_2, A_1, A_0 as 1 only the first location of this one will be selected, then second location, third location. After this, first 8 locations, the 9th location will be selected here. From 9 to here this will select from 0 to 7, if I start with 0 to 7. And this will select from 8 to 15, total 0 to 31 is required; because for 32 by 2 memory we have total 0 to 31 locations, total 32 locations. So, out of this 0 to 31, 0 to 7 will be selected by the first ROM, 8 to 15 will be selected by second memory and 16 to 23 will be selected by the third memory, 24 to 31 is selected by last memory. This is how we can expand the address of the memories.

So, what is that you have to do is, you have to connect the address lines of all these four 8 by 2 memories together, data lines also together, then chip selects are different, read by write bar also you have to connect together. So, this read by write bar you have to give to all these signals, depends upon the read operation or write operation you have to set that read by write bar signal. So, in chip select will be selected through the additional address lines of this 32 by 2 which is A_3, A_4. If you have 00 you can select the upper one, then 01 second one, 10 third one, 11 the last one. So, like that we can expand the address.

(Refer Slide Time: 49:04)



If I take another example, suppose if I want to only construct 8 by 8 using 4 by 8 memory. So, given is 4 by 8, required is 8 by 8. So, 8 by 8 you have to draw the block diagram, so 8 by 8 will be having 8 locations, each location is capable of storing 8 bits of the information. So, how many address line will be there 8 by 8? Three address lines, A_2 to A_0 and 8 data lines D_7 to D_0. And what about 4 by 8? 4 by 8 will be having two address lines; simply A_1, A_0 and 8 data lines, D_7 to D_0, ok. But, how many such memories are required? This 8 is nothing but 4 plus 4. So, we required two such 4 by 8 memories. Another 4 by 8 memory, this same A_1, A_0, D_7 to D_0, A_1 and A_0.

So, this has to be connected here, this has to be connected together. Only single bus will be available. In fact, these two are still connected to this. Similarly, data bus will be connected together. The only difference is we have chip select signal different. We call this as chip select 2, chip select 1.

Now, this is having only one extra address lines, which is A_2. If I take all the 8 combinations A_2, A_1, A_0, 000, 001, 010, 011, 100, 101, 110, 111. So, in the first combination 4 combinations A_2 is 0, in the next A_2 is 1. You know this varies from 00 to 11, this varies from 00 to 11, ok. So, for the first 4 combinations A_2 is 0. So, I want to select this, I will connect this directly to the A_2, ok. And the next combination I want to select the lower one, so I will connect these to A_2 bar, A_2 I will connect through

complement. A_2 you pass through the NOT gate and to connect to chip select of the last one.

So, that when A_2 equal to 0, the first 4 by 8 memory will be selected, when A_2 is equal to 1, A_2 is equal to 1 after the NOT gate, this A_2 becomes 0, so thereby the second 4 by 8 memory will be selected. So, after selecting this, this is first 4 by 8 memory, second 4 by 8. So, within these first 4 by 8 memory, how to select the 4 different locations? Again A_1, A_2 we have to change. 00 means first location, 01 means second location, 10 means third location, 11 means fourth location.

Similarly, in the second 4 by 8 if I want to select the first location, A_1, A_0 should be 00; A_1, A_0 if it is 01 the first location, 10 second location, 11 the last location. So, like that you can extend the memories. So, constructing this is called address expansion, we are expanding the address and we are fixing the data, ok.

So, in the next class we will discuss about how to expand the data, and then after that how to expand both simultaneously address and data. That we will discuss in the next class.

Thank you.