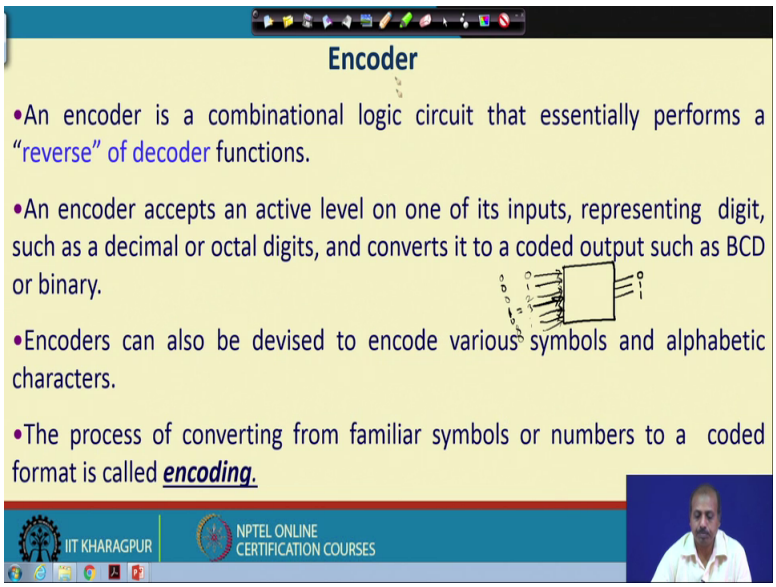
Digital Circuits
Prof. Santanu Chattopadhyay
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 24
Decoders, Multiplexers, PLA (Contd.)

(Refer Slide Time: 00:20)



So encoders, they are used for one purpose which is for which you say it is a reverse of this decoder functionality. So, it is a combinational logic circuit that essentially performs the reverse of the decoding function and an encoder, it will accepts active level on one of its inputs representing the digit such as a decimal or octal digit and convert it into a coded output such as a BCD or binary.

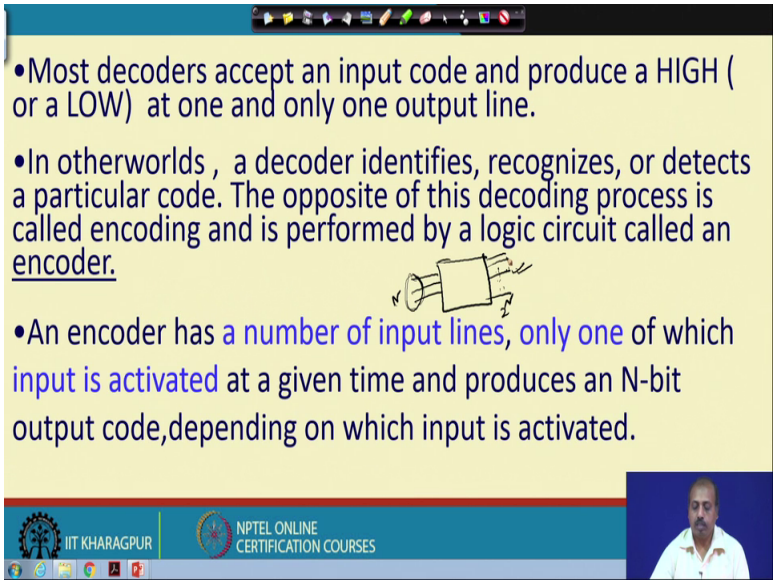
So, what you mean is something like this. So, it is a box ok. If this is the logic circuit for the encoder, so it will be accepting a number of lines as input. So, these are inputs. And at any point of time only one of the inputs will be will be active. So, other should be in deactive. So, it does not allow by definition more than one input to be high simultaneously. So, it will allow, so it is accepted that whoever is operating this encoder circuitry will make only one of the inputs active at a time and accordingly you have got this output. So, this output will tell the number of the input which has been made active. What I mean is suppose this side, I have got say 8 inputs. So, I have got say 8 inputs and then say input number, so these are the input numbers; so 0, 1, 2, 3 etcetera.

Now, suppose this input number 3 has been made active, so this is equal to 1 and rest are all 0's as per as the value is concerned. So, this is made equal to 1 and rest are all equal to 0. Then on the output side, what we will happen? So, this will output the bit pattern 0 1 1 telling that the third input had been the input 3 has been made high. So, this is actually encoding the number, the input number which is active now ok. So, that is how it is the encoder.

So, instead of telling the number directly, so you can you can send a 3 bit pattern like if you want to tell at this point or time which input is high. So, if you do not have this encoder circuitry, then you have to give this 8 bit pattern and in that 8 bit pattern whichever input is 1, so that we will tell us that that input is active. So, that will require 8 bit of information to be communicated. However, if you use this encoder circuitry, so you can transmit a 3 bit information ok; so, that has got the number directly. So, that is the purpose of the encoder.

So, encoders can also be devised to encode various symbols and alphabetic characters as I was telling. The process of converting familiar symbols or numbers to a coded format is known as encoding. So, encoder performs encoding operation

(Refer Slide Time: 03:28)

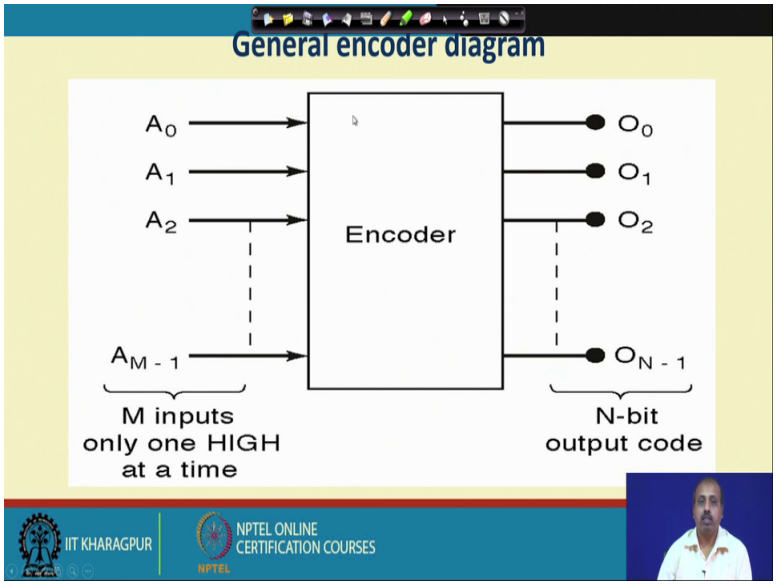


So, most of the decoders accepts an input code and produce a high or low at one and only one output line. So, why is it told that encoder is the reverse of decoder? So, this is the thing. So, in a decoder what happens is that if this is a decoder, so this decoder; it accepts

some input say n inputs and then it has got 2 power n outputs. It has got 2 power n outputs and depending upon the input combination 1 of the output is made high or made active, others are made low.

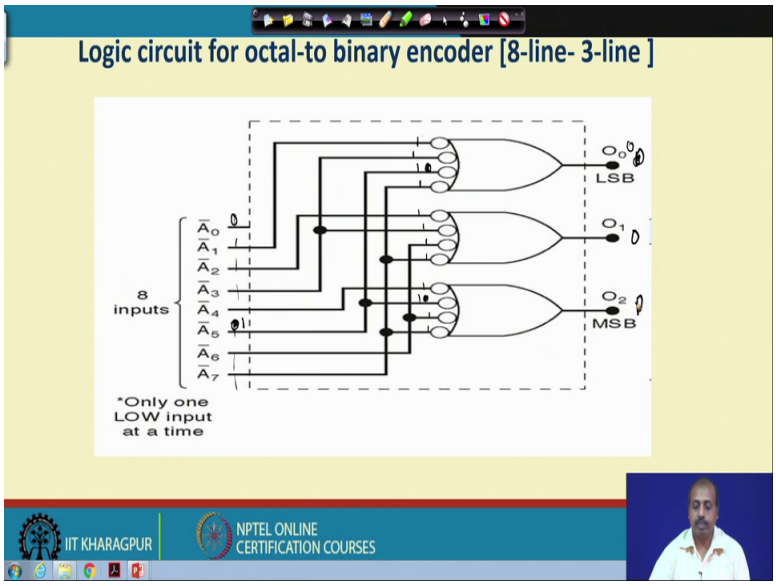
So, decoder we can say it identifies recognizes or detects a particular code. So, it can tell you which pattern you have applied here accordingly. So, from the code, so it can identify the pattern that you want to apply. Then opposite of the decoding process is the encoding and it is performed the logic circuit that does it is an encoder. Encoder has a number of input lines only one of which will be active at a given time and produces n bit output code depending upon which input has been activated. So, we will see how to design this circuit and all

(Refer Slide Time: 04:50)



So, this is the block diagram. So, we have got this A 0 A 1 A 2 up to A M minus 1 as input lines and then we have got so there are M input lines out of which only 1 is high at a time and then at the output side, we have got O 0 to O N minus 1. So, N output lines. So, that will be identifying which M which of these M inputs is high. So, that is the encoder functionality.

(Refer Slide Time: 05:20)



Now, how do you realize the encoder by means of basic logic gates? So, this is octal to binary encoder. So, it has got 1 octal input and then it is say, if the octal input A 0 bar A 0 A 1 up to A 7. So, whichever value you want to output. So, 0 or 3 or 4 or 5 or 7 accordingly, the corresponding input line will be made equal to 0.

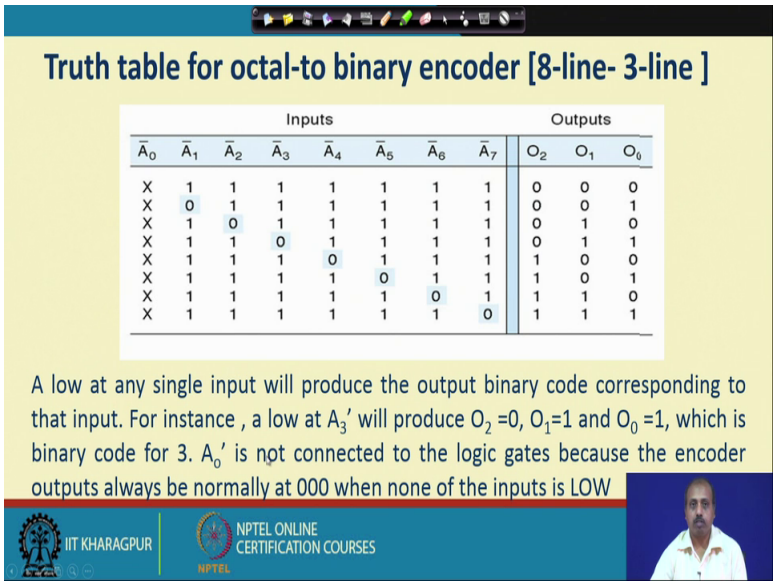
So, for 5 for example, the A 5 line will be made equal to 0 and all other lines will be made equal to 1. So, this output, output will be coming in terms of it will identify which input has been made active. So, accordingly it will be output in the bit pattern. So, if the 5'th input has been chosen here, so if I as an input; So, if I give an A 5 equal to 0 and rest are all 1, then it is expected that here I will get the bit pattern 1 0 1.

So, how does it happen? So, if all of them are 1 and this A 5 is 0, you see that all these bits are so, wherever this A 5 is going. So, this line is 0. You are getting A 0 here and here it is getting a 0 and on all other places I have got a 1, at all other places I have got 1 in this circuit. So, what is what will happen? So, it is inverted and then OR so, this is basically the NAND gate and NAND gate all inputs being equal to 1 output is 0 and if any of the input is equal to 0 output is equal to 1. So, you will get the patterns 1 0 1 at the output.

So, this A 5 being equal to 0, output is 1 0 1. If A 0 is equal to 0 and rest are all equal to 1, then what happens? If A 0 equal to 0, then and all other so, A 0 line is not connected to anybody. So, all these inputs are that we are getting are 1. So as a result, the NAND

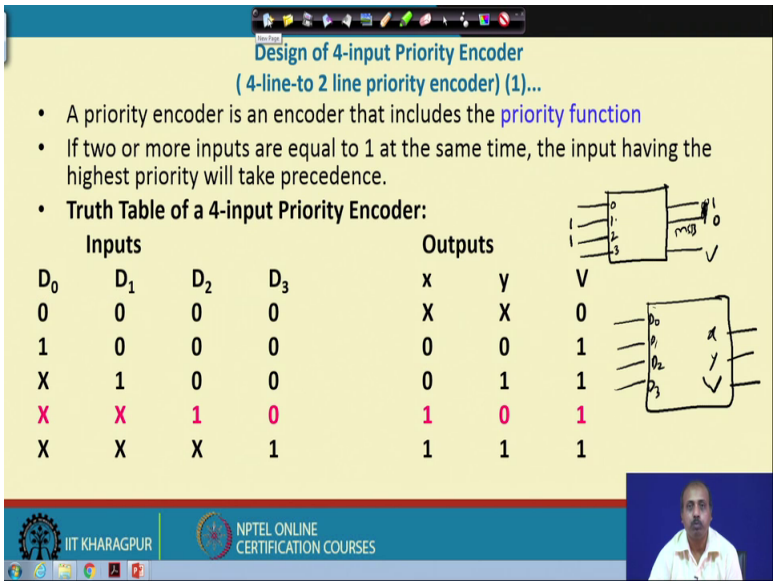
function, so this will give all 0 here. This also sorry this will be this will be 0 and this will also be 0. So, the 0 0 0 will be the output which definitely identifies the A 0 input. So, this way I can use, I can design this encoder circuitry by using NAND gates ok, the simply connecting some NAND gates. So, we can design this circuit.

(Refer Slide Time: 08:08)



So, a low at any single input will produce the output binary code corresponding to that input. So, if I have a low at A 3 bar, so that is A 3 bar here. So, if I have got A 3 bar, so it produces O 2 equal to 0, 1 equal to 1 and o 0 equal to 1 which is the code for 3. So, this way it is done. So, we have already explained how this circuit is operating. So, we can get it like this.

(Refer Slide Time: 08:37)



Now, one important constraint that we have put on the operation of the encoder is that at any point of time only 1 of its inputs can be equal to 1. So, many a times, it is a difficult to ensure that in the practical operation. So, it may so happen that the inputs are coming from different sources and as a result, it is simultaneously more than one input is becoming active. But even if they are active, so we may want to give priority to some of them. What I mean is that so, suppose I have suppose I have got a block like this. I have got a block like this and it has got say 4 inputs, so 4 inputs. Now it has got 2 outputs, this is an encoder.

So, these are line 0 1 2 3. So, if the line if the line 1 is active and 0 2 3 are inactive then the output should be 0 1 that will identify, if I take this as the MSB, if I take the lower one as MSB, so this will be 0 1 that will identifying the for the input which is equal to 1.

Now if this 1 and 2 are simultaneously equal to 1, then also we may say that I will give priority to 1 and this 2 the 2 input also be equal to 1 will just be ignored as long as 1 the input 1 is having equal to 1. So, that way I this input 1 has got higher priority than input 2. So, priority encoder is a function is an encoder that includes the priority function. So, if 2 are more inputs are equal to 1 at the same time, the input having the highest priority. We will take precedence. So, it may be like this that see here. So, this if all the inputs are 0 then, so this truth table. So, apart from this x y, so there is another output valid ok. So, there is another output valid. So, you can say that it is a 4 bit input. So, you have got this

D 0, D 1, D 2, D 3. So, this is D 0, D 1, D 2, D 3 plus there is so, there are two outputs x and y and there is another output which is valid v.

So, for the operation, so we have to look into the valid bit. So, if the valid bit is not equal to 1; that means, whatever output is produced as at x y. So, that is not a valid combination. So, when this D 0, D 1, D 2, D 3 so, they are all equal to 0. So, you say that x and y they can assume any value, but this valid bit should be equal to 0; so, as in invalid bit. Then if it is 1 0 0 0 that is only D 0 is equal to 1, so then there is no problem; so only one of the inputs are become equal to 1. So, we can say that output should show that input 0 was 1. So, it is the x and y is 0 0 and this valid bit is equal to 1 telling that it is it is a it is valid ok. The pattern output it is valid.

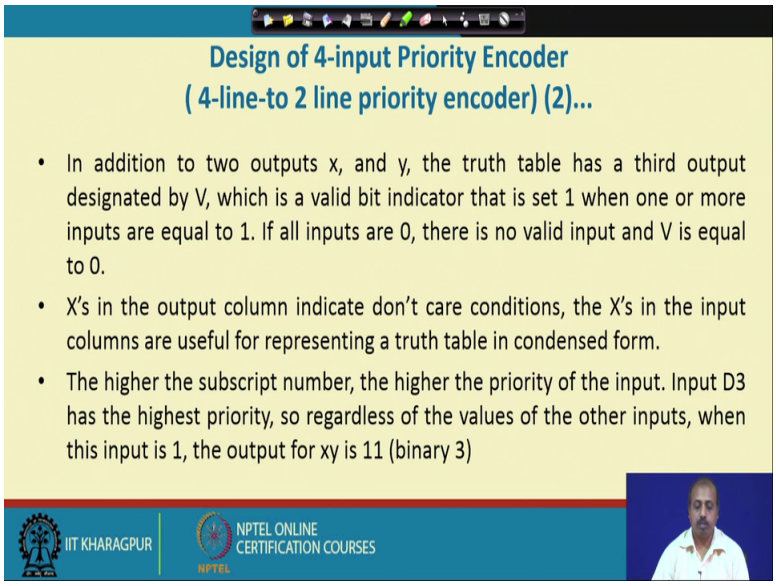
Now, if D 1 equal to 1, then whatever be the so if D 1 equal to 1, D 2 equal to 0 and D 3 equal to 0, then without looking into the value of D 0, So, it will output 0 1. So, if this is the condition that D 2 D 3 are equal to 0, D 2 D 3 are equal to 0, but D 1 equal to 1. So, it will not consider what is the value of D 0 will straight way output 0 1. So, that is it will tell that now input 1 is equal to 1.

Similarly, if it is if input if the input D 2 is equal to 1, then irrespective of the values that we have in D 0 and D 1. If D 3 if D 3 remains equal to 0, so then, this x y should be 1 0 and the valid bit should be 1 telling that input 2 D 2 is 1 and finally, if D 3 is equal to 1, it does not consider whatever the values we have in D 0 D 1 and D 2. It produces output as 1 1 telling that D 3 is high.

So, you see that this line D 3, it has got the highest priority. So, if D 3 is equal to 1, we do not look into other values of values of D 0, D 1 and D 2 and we are straight way telling if the output is 3. On the other hand, if D 3 remains 0 and D 2 equal to 1 we are do not considering D 1 and D 0. D 3, D 2 both being equal to 0 and D 1 equal to 1, we are not considering D 0 and we are considering D 0 only when this D 1, D 2, D 3 all of them are equal to 0.

So, I can say the D 3 has the highest priority followed by D 2 followed by D 1 followed by D 0 for getting identified at the output. So, this is the function of this priority encoder. So, as if we have got 4 inputs and there are priorities. So, priority of D 3 is the maximum followed by D 2 followed by D 1 followed by D 0.

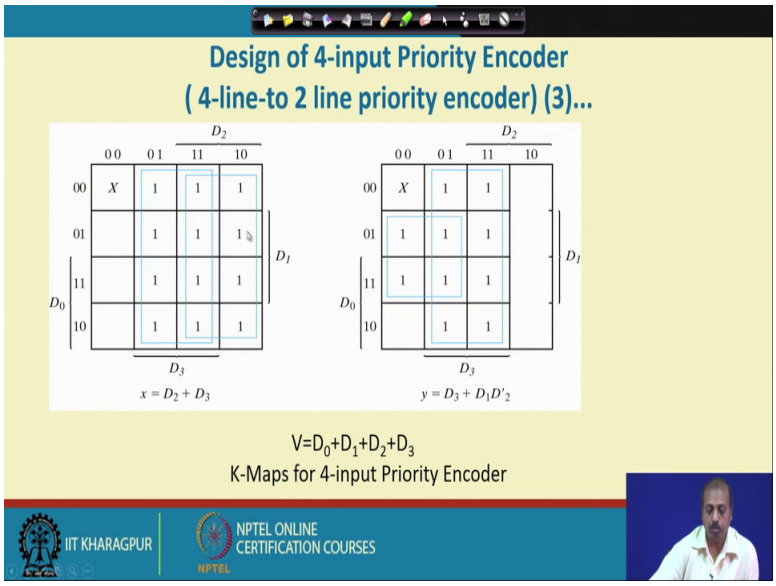
(Refer Slide Time: 14:00)



So, we can in addition to two outputs x and y, the truth table has a third output V which is which is a valid bit valid bit indicator and is set to 1, when 1 or more inputs are equal to 1. If all inputs are 0, there is no valid input and V is equal to 0.

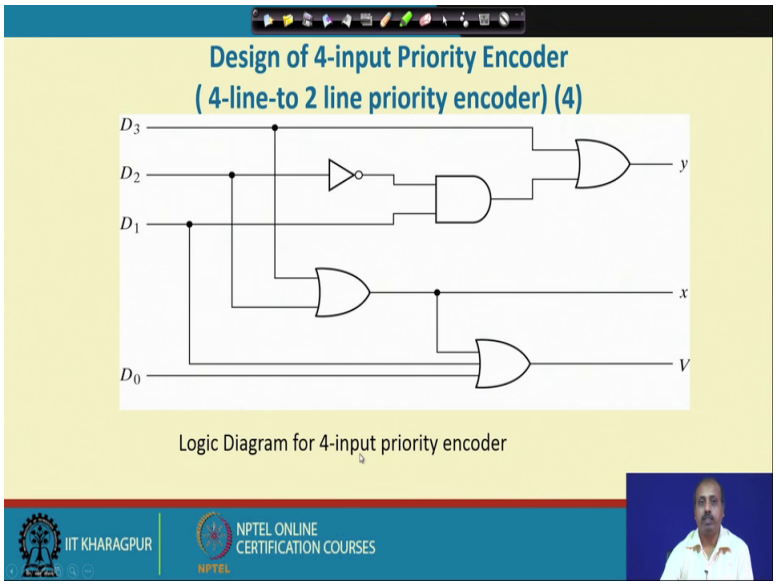
Now, don't cares that we have in the output column indicate that this input columns, similarly we have got don't cares in the input column also. So, that is that actually makes the truth table more condense and the priority that we have is like this. The higher the subscript number, higher the priority of the input. So, input D 3 has the highest priority. So, regardless of the values of the other inputs, when this input is equal to 1, output x y is equal to 1 is equal to 3; 1 1. So, this way we can have this priority function.

(Refer Slide Time: 15:00)



So, if we are looking into this design. So, we have got this x, if we draw the truth table then it will be like this. So, x equal to D 2 OR D 3, then this y equal to D 3 OR D 1, D 2 bar ok.

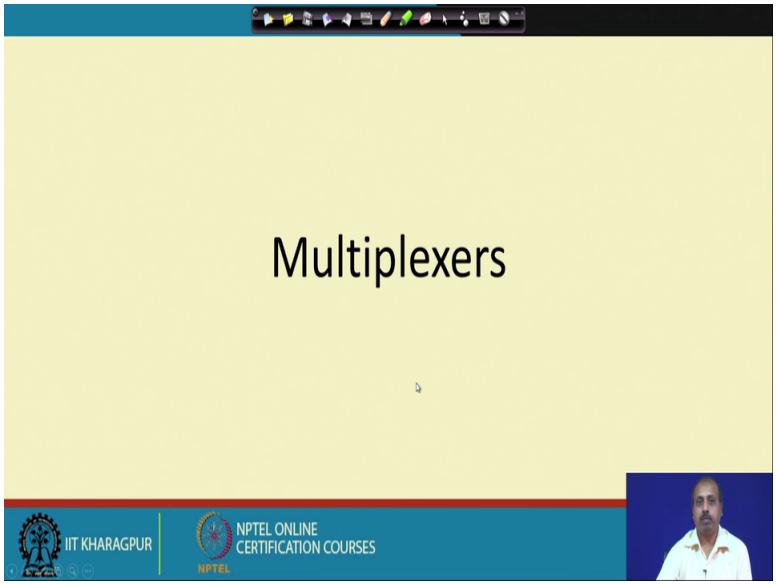
(Refer Slide Time: 15:18)



So, similarly from this you can x and y you have got and this valid bit is equal to 1, if all of them are not equal to 0; at least 1 of them is equal to 1. So, this here there is a slight modification in the circuit. So, instead of ORing D 0 D 1 D 2 D 3 separately, so we have take we have already done an OR of D 2 D 3 here for getting the value of x. So, that has

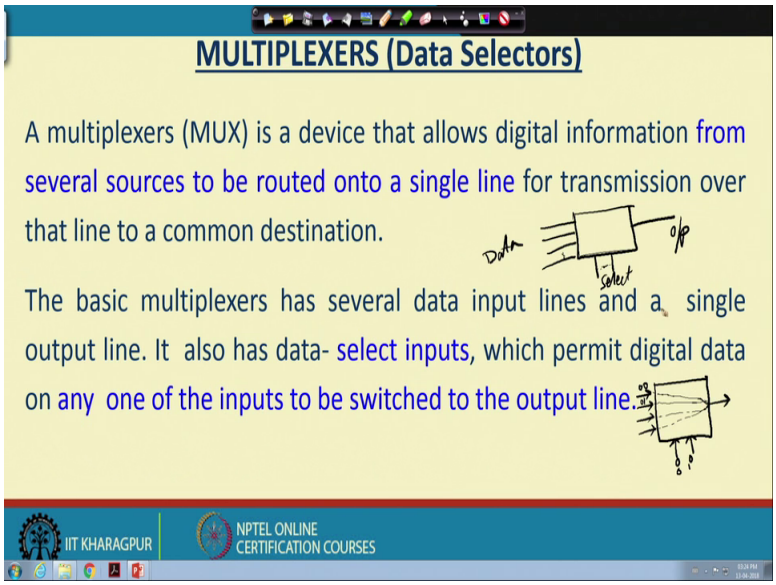
been utilized here for getting the value of v. So, in this way we can design a very simple logic circuit for realizing this 4 input priority encoder.

(Refer Slide Time: 16:00)



Next we look into another fundamental circuit element which is known as multiplexer. So, multiplexers; so they are actually they are also called data selector. So, it is like this that multiplexer, it is a device that allows digital information from several sources to be routed onto a single line for transmission or over that line to a common destination.

(Refer Slide Time: 16:22)



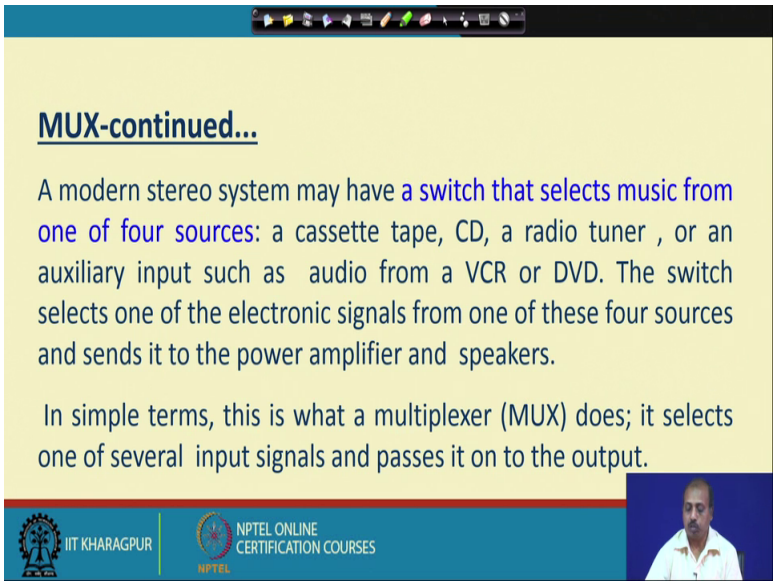
So, multiplexing is very common like in say when we are say a telephone line, it is carrying signals from several people and it is multiplexing between the that people that are talking so that way so there is a multiplexer.

So, it is actually sometimes, it is selecting one data item; some other time, it is selecting some other data item to be transmitted through the channel. So, the basic multiplexers has several data input lines and a single output line and it also has data select inputs which permit digital data on any of its inputs to be switched to the output line. So, a basic multiplexer is block diagram wise. So, you can say that it will look something like this that we have got this has multiplexer and then we have got some data lines. So, it has got two types of lines, some of the lines are called Data lines and some of the lines are called Select lines. So, these are called select lines.

Now, depending upon the values of this and the output this is the output and output is only 1 bit. So, we have got data lines. So, which can be many we have got select lines which can also be many and then depending upon the values that we are put into the select lines, one of these inputs will be selected and that will go to the output. For example, if I have got say 4 data lines, 4 data lines and 2 select lines like this; 4 data lines and 2 select lines, then depending upon the value, so suppose it can it may say that if the value is 0 0, then these output will be routed to the, this input will be routed to the output. So, this corresponds to the select lines 0 0.

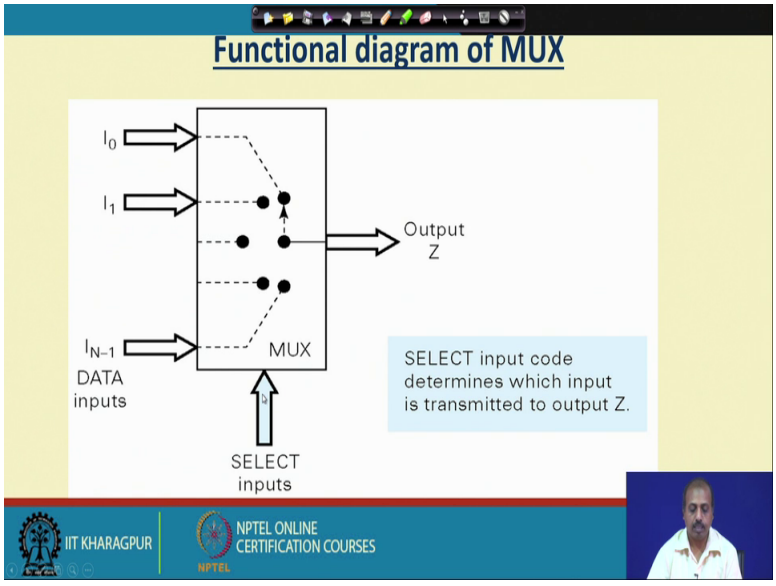
Similarly, if the value is equal to 0 1, then these input will be routed to the output. So, this is corresponding to the pattern 0 1. This is for 1 0 and this for 1 1; so depending upon input select line combination that I have one of this inputs that will be routed to the output. So, this is very useful in many circuit design problem, we will see that this multiplexers; they make it very simple for realizing functions and that will be used in designing many systems.

(Refer Slide Time: 19:08)



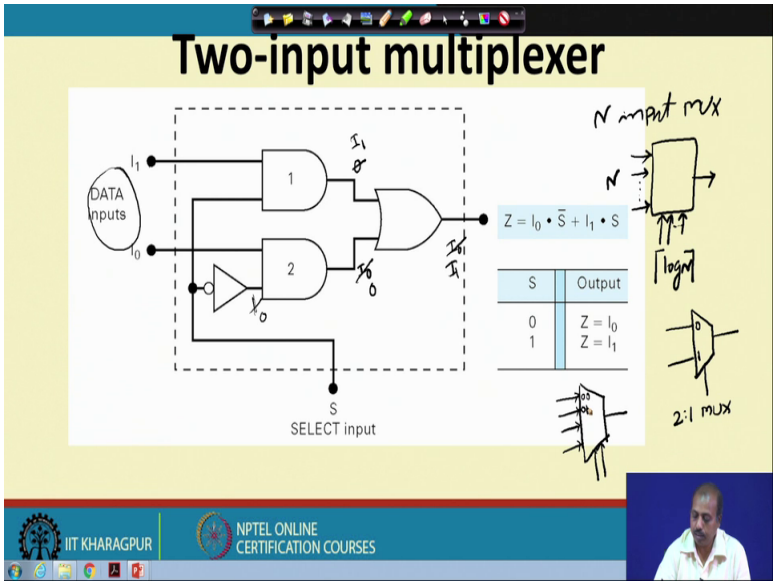
So, a modern stereo system, a typical example; so it may have a switch that selects music from one or one of four sources, when we may be it has got a cassette, tape, CD, radio tuner or an auxiliary input from VCR or DVD or may be some other sources also. Then switch selects one of the electronics signals from one of these from one of these four sources and sends to the power amplifier and speakers. So, that is nothing but some sort of multiplexers.

(Refer Slide Time: 19:49)



So, a multiplexer does, what it does is that it selects one of several input signals and passes it on to the output. So, that is the basic function of multiplexer. So, this is the SELECT line. So, SELECT inputs are applied here. So, depending upon the SELECT input as if logically you can say that this if this switch change the changes its position to from between these points. And accordingly one of the DATA inputs, we will get selected to be sent to the output. So, SELECT input code, the SELECT input code, so this will determine which input is transmitted to the output.

(Refer Slide Time: 20:20)



So, start with a two Two-input multiplexer. So, Two-input multiplexer is how to realize? So, it Two-input multiplexer means so, when I say two-input, so these two is corresponding to the DATA inputs ok. So, two-input it immediately tells that number of SELECT lines that I will have is only 1. So, it is like this. So, if there are if it an N input multiplexer N input multiplexer, then I has got N input lines. It has got N input lines. So, number of SELECT lines that I should have here is log of N. So, this is N inputs. So, at the here I should have log of N number of SELECT lines that will be an output here. So, because otherwise I cannot select all the N inputs by applying the input to the select lines; so we cannot applying SELECT.

Internally how will it look like is something like this. So, this is when say S is equal to 0; when S is equal to 0, then you see that these AND gate output is 0. So, whatever is the value of I 1, so you will get a 0 at this point. On the other hand, when s equal to 0; so

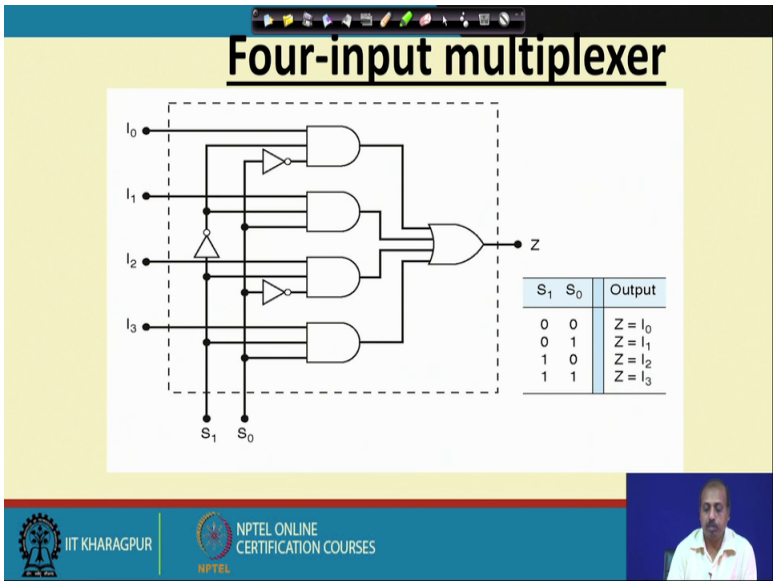
here this input is 1. So as a result whatever is coming at I 0, so that will be outputted here. So, this point a I am getting a 0, this point I am getting I 0 and this is OR of these two. So, here I will get I 0. So, when s equal to 0, I am getting I 0 and when s equal to 1 then what happens? So here, I will get I 1, 1 will come here. So, this input is 0. So, as a result I will get a 0 at this point. So, they will be ORed and I will be getting I 1.

So, when S equal to 0, then this I 0 is transferred from the input at I 0 is transferred to this output Z and when I S equal to 1, then input I 1 is transferred to output Z. So, I can say that the functionality that I am realizing is Z equal to I 0 S bar plus I 1 S ok. So, this is the basic multiplexer functionality.

So, these multiplexers are often represented by this type of notation. There are several notations for multiplexer. So, this is the most common on. So, we have got this data inputs and this is the SELECT input. So, the it 0 is written here and a 1 is written here meaning that if the input if the control signal is or the select input is 0, then these input will be selected to go to the output and if this select input is 1, then these input will be selected to go to the output. So, this is a 2 to 1 multiplexer or 2 is to1 multiplexer. There are several names or 2 is to 1 multiplexer.

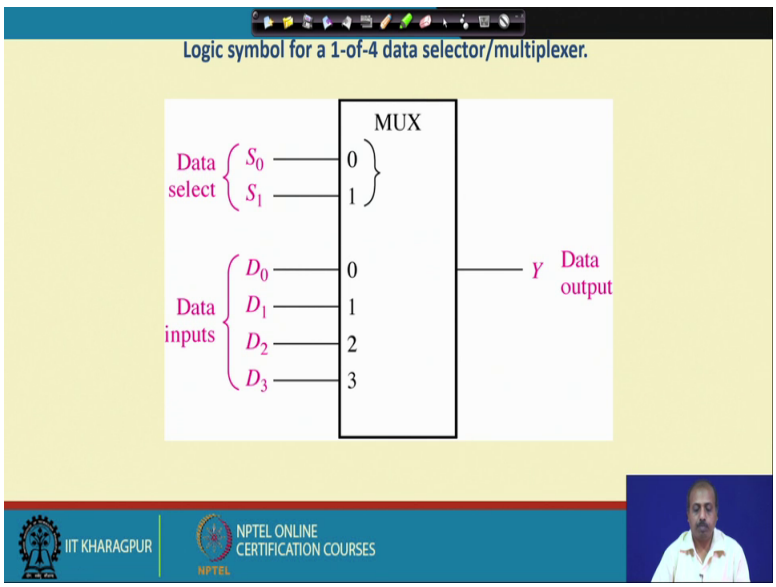
Now, if it is a 4 is to 1 multiplexer, then the corresponding symbol is something like this ok. So, it will have 4 lines, 4 DATA input lines and 2 SELECT input lines and 1 output ok. So, it will have 1 output and again the same thing. So, if it is 0 0, then this input is selected. So, write a 0 0 here. If SELECT lines are 0 1, then this input is selected. If it is 1 0, then these input is selected and if it is 1 1, this input is selected. So, this way we represent 4 input multiplexers in this form ok. So, we will see them in detail. So, this is the 4 input multiplexer.

(Refer Slide Time: 24 is to 19)



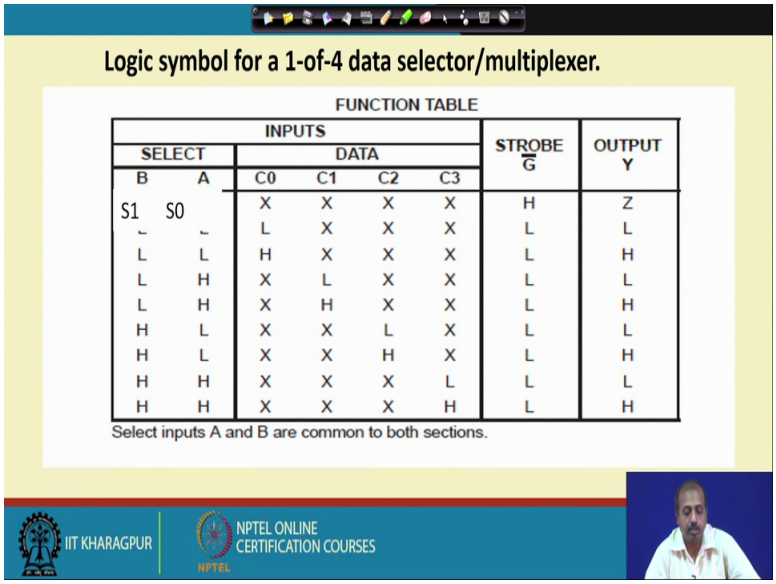
So, you see that when S 0 and S 1, these values are 0 0, then what I want that I 0 should get transfer to Z. So, we can quickly check that it is happening like when S 0 S 1 both are 0; then this input is equal so, they when both are equal to 0, then this input is equal to 1 and this input is also equal to 1. So, whatever we are getting at I 0 is transferred to this point, whereas for this second and gate, so this S this input is 1, but this input is 0. As a result, whatever be the value of I 1, you are getting a 0 at this point. In this way you can find that for all the cases. So, you were getting a 0 here. So, this OR gate; so, one input is I 0 and rest are all 0. So at Z, I am getting equal to I 0 at Z I am getting equal to I 0.

(Refer Slide Time: 25:30)



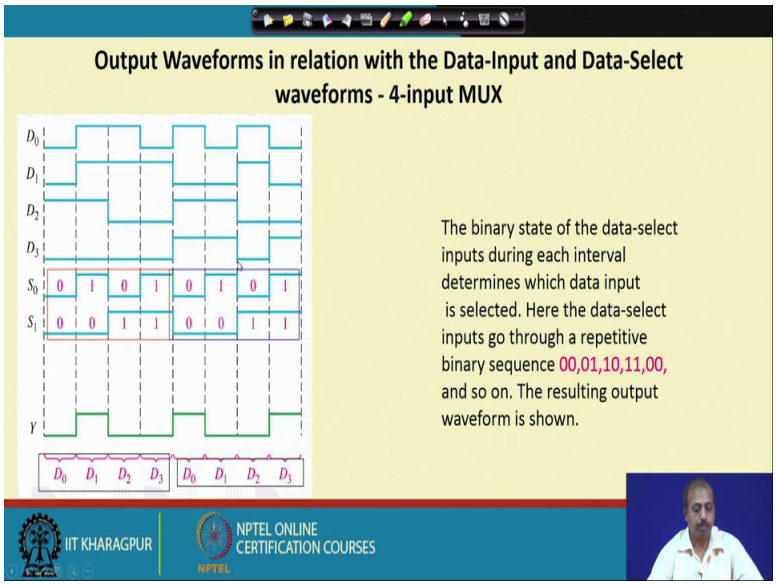
So, other combinations also you can check. So, this is the functionality of a four input multiplexer or Four to one multiplexer whatever you call it. So, this is logic symbol for one of four Data selector or multiplexer. So, it is also represented like this. So, we have showed you another representation where it is a trapezium like structure. So, you can also represent it by means of a rectangular symbol like this ok. So, normally the data inputs are marked as D 0 D 1 D 2 D 3 or we write down the corresponding decimal values that corresponding to and in the SELECT input are normally written as a S 0 S 1 and they are also we tell which one is LSB and which one is MSB where putting 0 and 1 this numbering actually.

(Refer Slide Time: 26:16)



So, this S 0, logic symbol for a 4 is to 1, 1-of-4 data selector or 4 is to 1 multiplexer. So, it can be when the SELECT inputs S 0 S 1, so they are if they are going to be low; if a both are low, so this if both are low, then this c 0 will be selected. In fact, I think there is a actually this timing diagram explains it better.

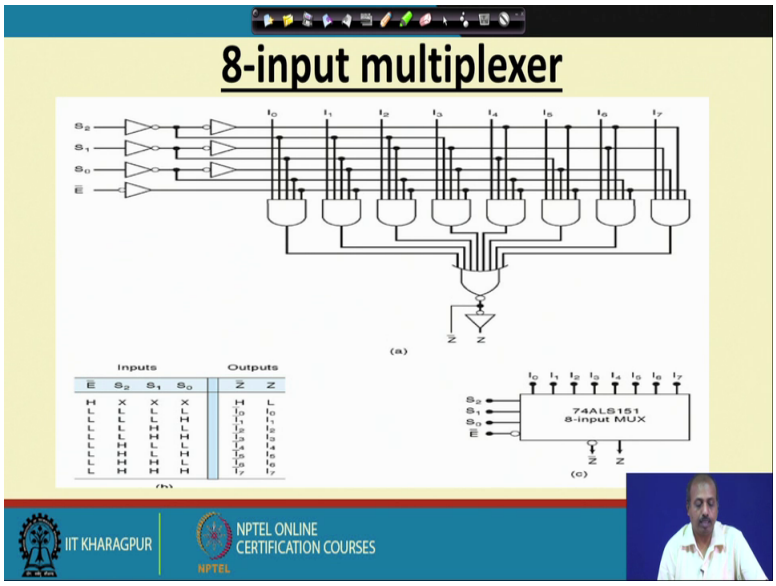
(Refer Slide Time: 26:46)



So, this D 0, D 1, D 2, D 3; suppose the values are going like this. Now the SELECT line when it is both of both the inputs are 0, then these D 0 line is selected at the output. When the this S 1 S 0 is having 0 1, then this D 1 line is selected. So, for this much time D 1 is put at the output. After that S 0 S 1 S 0 has become 1 0 and it is 1 0. So, D 2 output will be put at the, D 2 input will be put at the output. So, it is done here. Then it is 1. So, D 3 input will be put. So, it has come here. After that S 0 S 1 again has become 0 0. So, in that case again D 0 will be copy to the output then it is 0 1. So, we will get say this D 1 copied to the output. So, this way it goes ok.

So, by looking so if a varying bit varying signal is applied to this input symbol input D 0 D 1 D 2 D 3 and this S 0 S 1 SELECT lines so they also are having some bearing time bearing signal. Then you can draw the corresponding timing diagram like this. So, the binary state of the data select inputs during each interval determine which data input is selected. So, this S 0 S 1 values it will determine which input is selected. Here the data in this particular example, so they are going by a repetitive sequence. So, 0 0 0 1 1 0 1 that is it is first selecting D 0 then D 1 then D 2 then D 3; again it is selecting D 0 D 1 D 2 D 3 etcetera. So, it is going in the sequence. So, as a result it produces this type of waveform of course, it is specific for these examples.

(Refer Slide Time: 28:38)



So, for 8-input multiplexers, so we can have it like this. So, there should be, so we assume that here we have got one enable line as well. So, this is the truth table that we are followings. So, when this enable line is E bar. So, when this E bar is equal to 0, then only this multiplexer is enabled otherwise it is not. So, E bar equal to 1, so this multiplexer is disabled. So, this Z bar so, Z is Z bar is high and Z is low. So, it is not selecting any of the input. So, it is output it just outputting a low here.

Now, if this E bar is low that is it is the multiplexer is active. Now it depends on the values of S 0 S 1 and S 2. If all of them are low, then this I 0 line it will be coming to Z as a result I 0 bar will coming to Z bar. So, this is one particular chip 74ALS 74151. So, it has got 8 input. It has got 8 DATA input, 3 SELECT inputs, 1 Enable line and it has got both outputs and its complemented version. So, Z and Z bar both are available.

So, in this way internally we can think that the circuit is consisting of something like this, which produces the proper output depending upon the values of S 0 S 1 S 2 and the enabled line E.