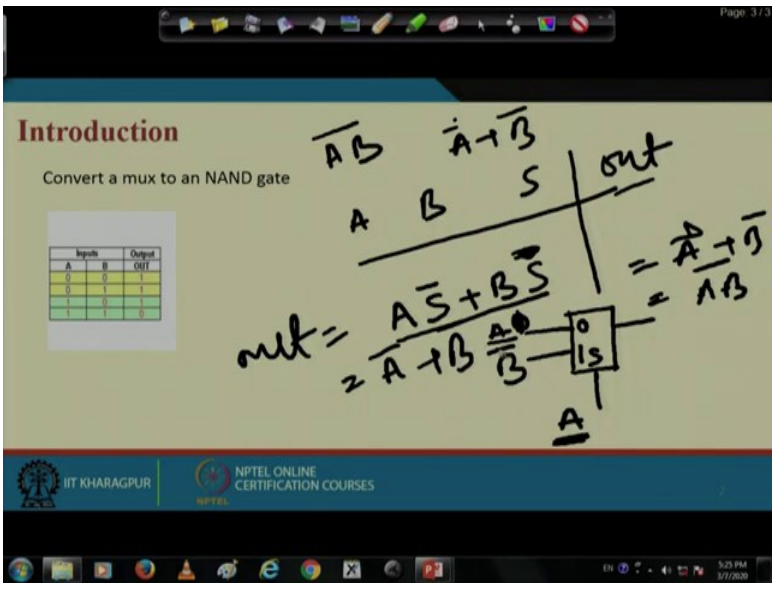
Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture - 58
Design Tips for Basic Circuits Design

Hello everyone, welcome to the course on Architectural Design of ICs. So, till now we have seen different kind of architecture of the various operations like this multiplier then divider then adder. So, all these structures, how we can design efficient architecture in corresponds to the speed power area as a major constant we have seen the architectures till now.

Now, we will see some of the very basic circuits, how to actually if they have been asked in the that is means interview or they have been asked you to design that circuit, so, at that time how to start with the corresponding circuit. First focus on what is the problem and then from that particular point we will start doing with by step by step how to solve the problem. So, we will see very basic design tips and tricks for the very basic circuits, very basic digital circuits by which we can implement any complex circuits.

(Refer Slide Time: 01:36)



So, the very first thing is that; suppose, the one of the major that means; circuits which is being moistly used in digital circuits that is the multiplexers. So, using multiplexers I can implement any circuit. So, how any circuit? Basically, using multiplexer I can design

that AND gate, NAND gate, OR gate or NOR gate inverter. So, any gates or I even I can implement XOR gate I can implement this flip flops using multiplexer. So, we will see that how to design this let us first consider this case which is how to convert one multiplexer into a NAND gate.

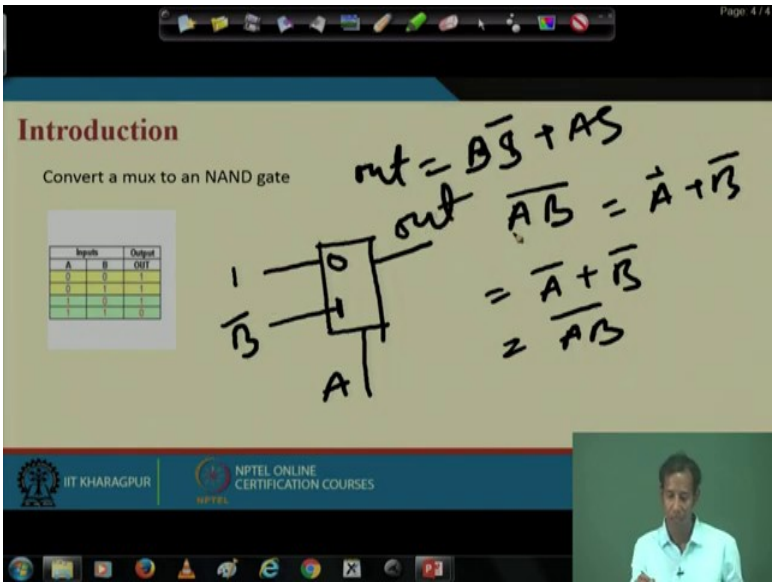
So, to start with that at the very beginning what we have to do; that means, the NAND gates the truth table for NAND gate is that when all these A, the actually is suppose A and B both are 0 0, 0 1, 1 0 for all these case the output should be 1 and for 1 1, they should be 0.

So, if we consider the particular logical expression for 1 2 is to 1 multiplexer. So, at that time if I consider the two input as A and B and the select line as the S, at that time I can write the output expression will be out equals to A S plus B S bar ok. So, now, what I need is that I need this particular expression should be work as a NAND gate ok. So, actually to perform this particular task into one NAND gate so, what we have to do is that at that time this. Now, if I just draw the corresponding two is to 1 multiplexer. So, we will be having this.

So, this is my select line this is A and this is B ok. So, now, what I need is that I need AB this bar ok. So, whenever I need this AB bar. So, at that time what I can write is that here suppose at these particular at these particular terminal if I provide if I provide these as or AB bar I can write that as A bar plus B bar ok. So, if I just write this particular terminal as so, here let us say this is A this is 0 and this is B bar. So, at that time what will happen? Now this, a signal is now that is sorry not this one this one is will be 1. So, now, instead of this A, if I provide here as 1, so, suppose not this one this will be AS bar and BS.

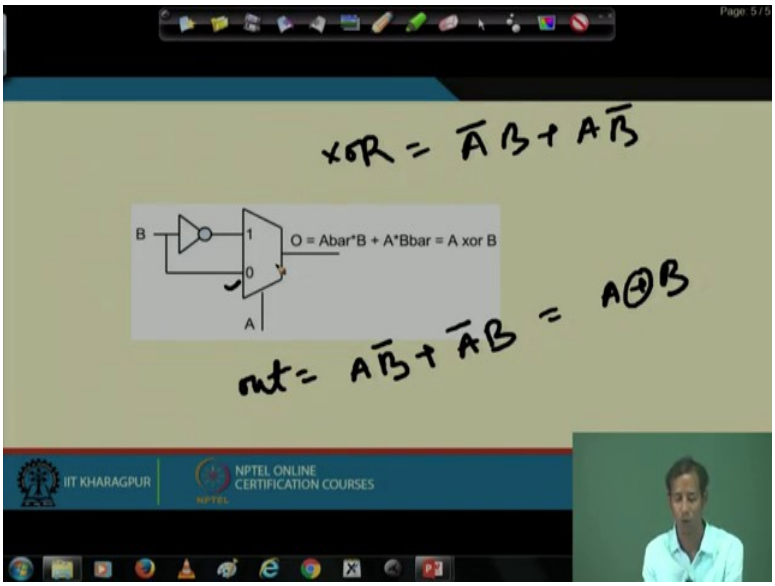
So, if I provide these particular terminal as 1 and the select line as a. So, at that time what it will be? It will be A bar plus and here what is that here actually the as the input we provide as B bar and A is already this is for 0 this for 1. So, A is already 1 so, it will be 1. So, this is A bar plus sorry A bar plus B bar that is nothing but, AB whole bar. So, that means; this expression now I can write as the NAND gate ok. So, in this manner, if I want to implement this particular function to any other gate, according to these particular logic; now you have to put what value will be at 0, what value will be at 1 and what value will be at select line ok.

(Refer Slide Time: 06:19)



So, now I can see or I can actually I can say that ok, now depending on the corresponding logic which is this now the output will be; that means, if I freshly if I draw. So, at that time it will be at 0 terminal it will be just at 1 terminal it will be if whenever this is A, A means; this will be just A B. I just want to implement this so, as this is A bar plus B bar. So, and according to the logic it is A S, AS bar sorry, BS bar plus AS. So, in the A, what I will do? I will provide B bar and here I will provide 1. So, it will give me A bar plus B bar which I can write as AB whole bar ok. Now, this particular multiplexer will be act as a NAND gate at that time.

(Refer Slide Time: 07:37)



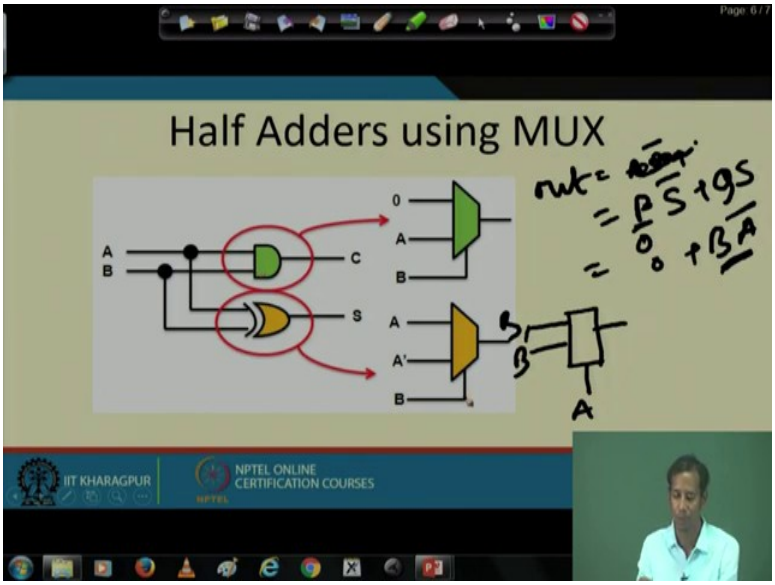
Now, let us consider another example ok. So, where I have to design these particular multiplexer should be work as a XOR gate ok. So, in XOR gate the corresponding logic for XOR gate is what? This XOR function says that it will be A bar B plus AB bar ok.

Now, what I have to do; this multiplexer I need to be work as this XOR gate. So, according to the multiplex; that means, multiplex and logic what is that? That is A actually if I select if I mention that select is my A. So, at that time this will be A B bar plus A bar and here it is comes under B. So, that means; at 0 signal I provide the B, where A bar B and for this 1 terminal it is providing B bar so, A B bar.

Now, this is nothing but, A XOR B ok. So, that means, the same multiplexers just by changing or by putting different values to its input and to the select line, I can make these multiplexer to be work as a XOR gate. So, in the previous case in the previous case whenever we have just applied 1 and B bar along with the select line as A.

So, at that time the same multiplexer is working as a NAND gate here with this particular configuration the same multiplexer now it is working as a XOR gate. So, it is not that it is only confined to that you can implement only XOR gate and NAND gate it is not like that using multiplexer with different configuration at this particular inputs and the select line you can make this XOR gates to be work as or gate you can make this XOR gate to be work as a NAND gate, any gates ok. So, all the gates you can implement using this multiplexers ok.

(Refer Slide Time: 10:13)



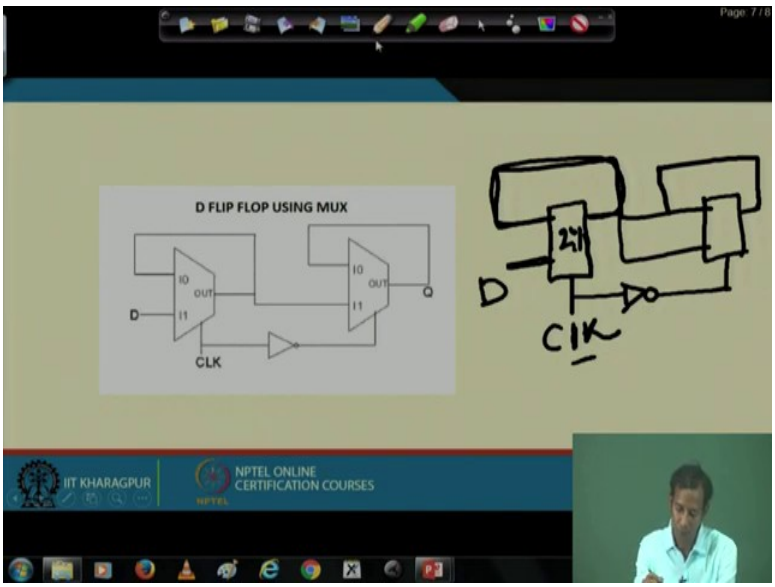
Now, we will see how to implement the half adders using multiplexers. So, according to the logic of half adder, what we need is that; we need to implement the carry we need one AND gate and to implement the sum we need one XOR gate.

So, how to implement the XOR gate? The XOR gate can be implemented by using this A and A bar at the inputs and B as the select line. You can just do the reverse B and B bar here with select line as the A. Then how to implement these particular function as AND gate. So, if I provide this as 0 and the other input as A and B; that means, when that will be act as a AND gate. The AND gate, when it will be act as an and gate for out logic this is A S bar plus let us say not A consider a different value suppose P S bar plus Q S ok.

So, now, for this particular case when this S is 0, at that time at that time if you make this as 0 so, it will be 0 plus Q as B and S as A. So, it will be act as a AND gate at that time. So, that thing has been implemented using this multiplexers and the XOR gate can be done either you can use B and B were over here and A here the same it will be act as a multiplex sorry XOR gate.

If you provide A A bar at this input line and B at the select line again it will work as a XOR gate. So, that means; now using two multiplexer 2 is to 1 multiplexer with this type of configuration, now I can implement this as a which will be work as a half adder ok.

(Refer Slide Time: 12:41)



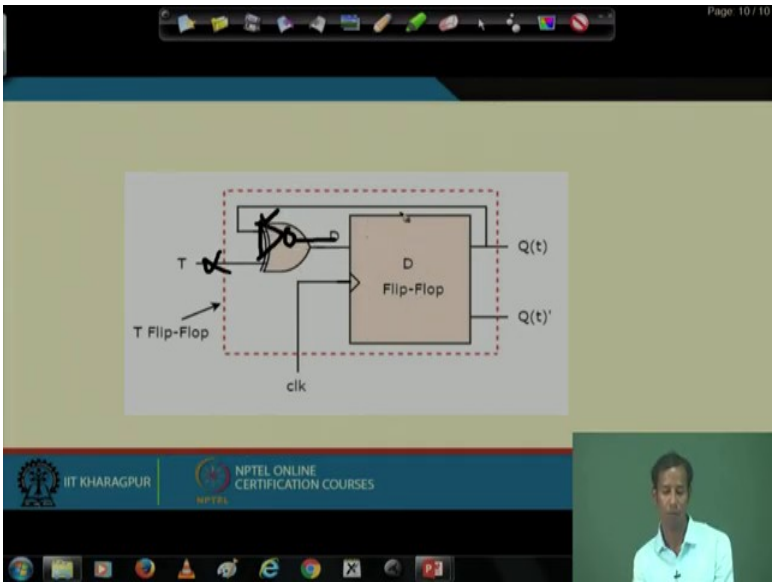
So, now let us consider another example how we can make this multiplexers to be work as D flip flop ok. So, D flip flop means actually it has to store it is value. So, now, whenever we are having this particular actually if we do not consider, if I consider only this or if I just want to; that means, if I just ask you to implement this multiplexer as the latch so, at that time what it will be means; the output again it will be connected to this and the select line will be your corresponding clock signals and this is the D input.

So, this is nothing but, this is nothing but 1 latch example so; that means, whenever this clock signal is 1 so, at that time the D will pass whenever this clock signal is 0 so, at that time these output will again it will come to it is; that means, to it is input. So, now, in flip flop we are having this actually we are this is the example of latch using 2 is to 1 multiplexer. Now whenever we are having this that flip flop we have to design.

So, at that time we can connect that another 2 is to 1 multiplexers where this input is; that means, sorry, this input is connected here and this is also another latch which is connected and here the clock is just inverted one; that means, this is nothing but the master slave configurations what we did using; that means, this using a satellite whatever we develop the; that means, this that means, master slave J K flip flop.

That means; the D flip flop the same configuration here we have used to implement using 2 latch which has been generated or which has been designed using 2 is to 1 multiplexer. So, that configuration we have used to implement the D flip flop using multiplexers ok.

(Refer Slide Time: 15:28)



Now, then again if the question has been asked to make these as T flip flop. So, the D flip flop we already made ok. So, then again actually T flip flop means we have to the D flip flop, the output from the D flip flop that will be just inverted and it will go to the D input of the corresponding flip flop. So, D flip flop using this 2 is to 1 multiplexer we have already designed.

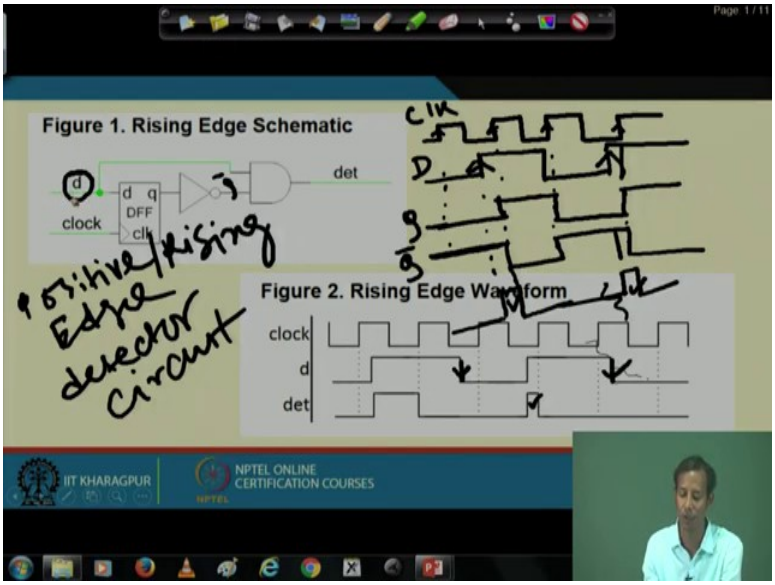
Now, we can put if we are having this control signal. So, at that time whenever this T is 1 so, at that time it will be just act as a inverter of operation which will be at that time this particular flip flop will be just be work act as a T flip flop ok. So, instead of XOR gate if you can just make this as only one inverter without this T input, then, this will be the T flip flop made by 2 is to 1 multiplexer.

Now, if you have if you do not have in the library again if you do not have this inverter operation. So, at that time the multiplexer itself you can make that as a inverter and you use that particular; that means, multiplexer over here to make this D flip flop to be work as a T flip flop ok.

So, these are the very basics circuits which are; that means, using multiplexer only you can implement. And in actually in FPGA we know that FPGA the all the logics are basically implemented through multiplexers only. So, how these multiplexers as we have discussed or as we have already seen that multiplexers can implement any function; that

means, any of these logic gates, then, definitely I can use these particular fundamentals while I am designing 1 particular FPGA based circuit design ok.

(Refer Slide Time: 17:54)



So, then we will see another example which is very basic circuit for this edge detector; that means, whenever, what is this edge detector? Whenever this clock edge occurs so, at that time it will detect that there is one edge of the clock which is happening. So, how to detects? Actually, there will be this rising edge, there will be falling edge of the clock and we can also detect that rising and falling both altogether.

So, the fast circuit is basically related to the rising edge detector circuit. So, what is this is that so, we are having one D flip flop where the input is basically ANDed with the inverted of it is q output. So, then if I just make these particular circuit; that means, make this configuration, so, at that time what will happen? Suppose this is the clock signals or the clock waveforms which is running like this and the corresponding d input is let us say is something like this.

So, actually this rising edge when it will detect is that suppose at this particular edge as this is D flip flop at this particular edge it will find that what is the value of d. Now the value of d is already 0, then up to this we have to make what will be the corresponding q output ok. So, if we just draw that this is my clock signal and the d is basically let us say up to this and then we are having up to this and then we are having the end.

So, this is my clock, this is my D input. Now what will be the corresponding value at Q. So, the Q signals will be so, at this edge, at this edge, at this edge and at this edge. So, at this particular edge as these value is D value is already 0. So, it will continue to 0 and again it will continue to 0 up to this and here it finds that this Q value is sorry D value is 1.

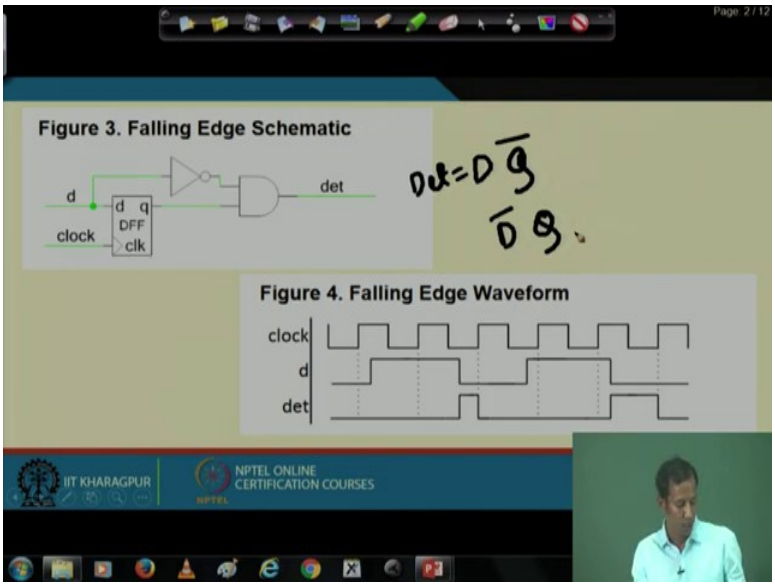
So, again it will start 1 it will start 1 and here at this edge it finds that the d value has becomes to 0 then again it will continue then again here it will find 1 and it will continue like this.

So, this is nothing but the Q. So, when this Q, then this is basically inverted of Q. So, inverted of q means this is now 1, these particular is 0 this is again 1, this is 0. Now if I ask say that this is the Q bar so, this is Q bar. Now depending on this D input and the Q bar if I just as they are ANDed. So, at that particular instant, at these particular instant both are 1 so, there will be 1 1, rest of the case it is 0.

Then again it is 0 and here whenever this; that means, for this particular time period both are 1 so; that means, again I will get 1 one pulse over here ok. So, that means; it finds that whenever there is this rising edge of this and rising edge of this, it is creating 1 pulse this is creating 1 pulse which says that that this d input is of rising edge ok.

So, that means, this is the edge detector circuit which edge positive edge sorry, not positive or I can say that rising edge ok. So, now, in the next; if I want to detect the falling edge; that means, when these particular data is falling. So, at that time what will be it is corresponding architecture or what will be the circuit which can detects the falling edge of the data input or the signals which is appearing over here.

(Refer Slide Time: 23:45)

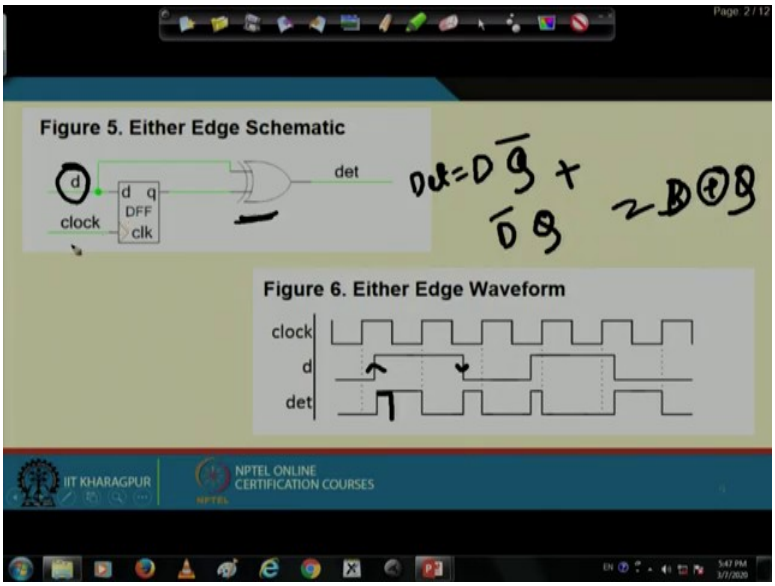


So, at that time, in falling edge what we have to do? In earlier case what was there the inverter was placed over here the data input was coming there, but here what I did, I just placed I just interchanged this position of in inverter in the upper path. So, at that time what will happen? At this particular edge; this negative edge it will give me one pulse and again at these particular edge it will give me another pulse.

If I just draw again if I draw the q and this will be the q bar if I get. So, in earlier case this was creating the pulse, this was creating the pulse when there was positive edge as I make inverting of this d so; that means, at that time what will happen the 1 will become 0, 0 will become 1. So, at this time this falling edge will be detected and it will give you the pulse at this detector output.

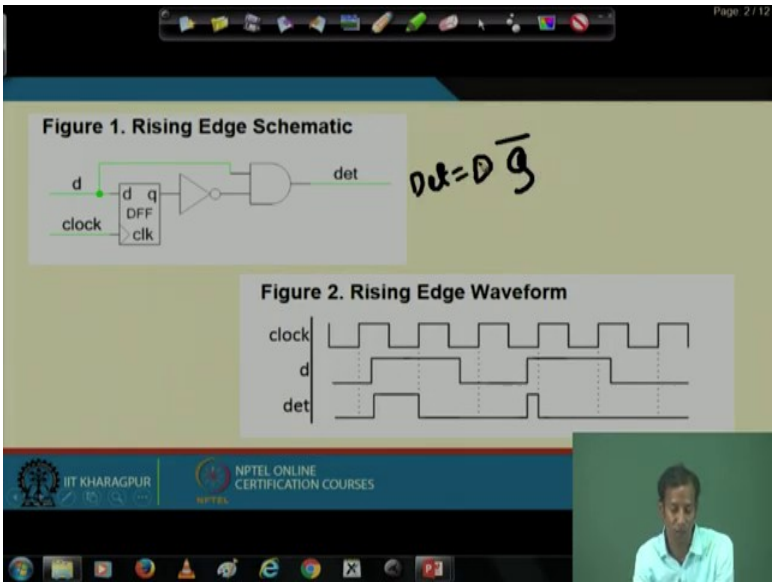
So, using this circuit now you can detect whether there is any falling edge of your signal. So, this signal is these particular circuit is falling edge detector circuit.

(Refer Slide Time: 25:07)



Now I need; that means, both the edge should be detected at the output ok. So, then how to do that? Then that can be done using if I use instead of this here what we are doing? Here we are we are doing if I say that this is what this is D into Q bar ok.

(Refer Slide Time: 25:33)



So, this is used for positive edge sorry, falling sorry, rising edge and for this what is this in this case this is D bar Q. So, when I need to combine both together in the particular circuits so, at that time so, at that time what I need? I just need to add them. So, if I add them DQ bar and D bar Q is nothing but D sorry D XOR Q. So, that has been done over

here. So, these particular circuits now it can detect both the edges; that means, this either of this rising edge or this falling edge ok.

So, if you just write the very log code of this and then if you can run then you can say that if there is a change in this whether there is change means whether there is 0 to 1 or 1 to 0 you always get one pulse depending on the corresponding structure what we have already design over here ok.

So, this particular circuit is used suppose for one such application you need to detect this edge; that means, whether there is any event at the corresponding input signals, so, that you can just detect using this kind of circuit, one for the rising edge, another for the falling edge and this particular circuit for rising and falling both the things. So, thank you for today again we will discuss some other basic building blocks, how to design them in an efficient way.

Thank you for today.