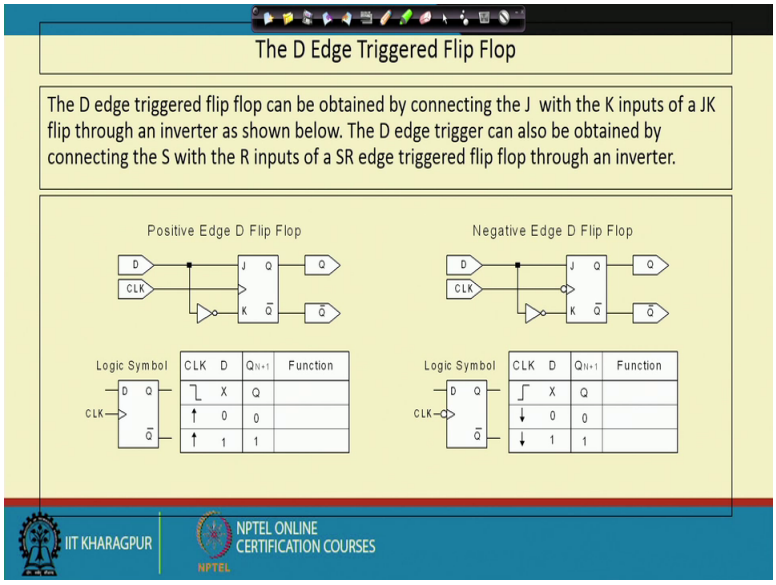
Digital Circuits
Prof. Santanu Chattopadhyay
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture – 30
Sequential Circuits
(Contd.)

Next, we will look into D edge triggered flip flop. So, it is similar to the JK edge triggered flip flop.

(Refer Slide Time: 00:21)



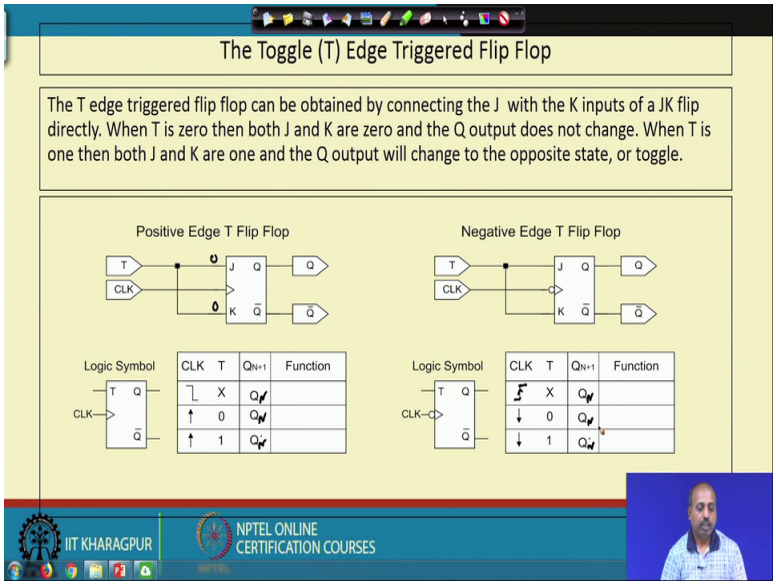
But, this D D edge triggered flip flop can be obtained by connecting the J and K inputs or JK flip flop using an inverter. So, just like that instead of JK, it can be SR also.

But it does not matter, but this if we connect an inverter between J and K, then it will behave as a D flip flop. So, it can also be obtained by connecting inverter S and R inputs through an inverter and the behavior is already behavior is whatever we have said so far that as long as the clock signal is a high or low, all it makes a high to low transition. The Q Q value does not change and so we write it as Q N plus 1. Sometimes, we write it as Q N plus 1. So, to mean that at the N plus 1-th time instant, what will happen? So, if we are we are currently at the N-th time instance.

So, that is Q N and then at the N plus 1-th time instant what will happen? So, to be precise; so, you can write it as Q N ok, so to be more precise instead of writing Q; so, you can write Q N. So, otherwise when the clock signal is making a transition from low to high. So, depending upon the value of D, it will either become 0 or it will become 1.

So, this way we can have this D type edge triggered flip flop.

(Refer Slide Time: 01:45)



Another flip flop that we get from this JK flip flop is known as the toggle flip flop or T flip flop. So, this T flip flop is simply the JK flip flop when this J and K inputs are tied together. So, they behave like this since this is JK flip flop internally. So, you see, whenever I make this T value equal to 1, then this J and K, both are they both of them are getting 1 as a result whatever we was the previous value of Q. So, that will be complemented.

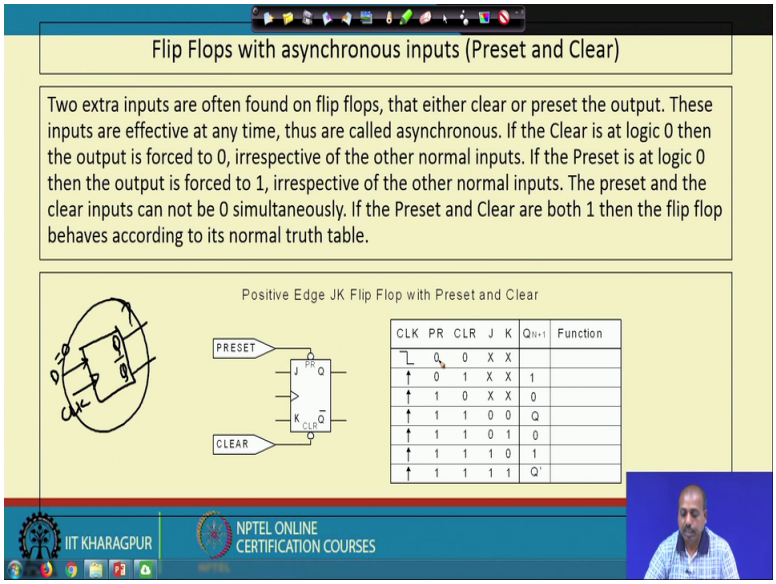
So, you see here that when this T value is equal to 1, then at the rising edge of the clock this Q N plus 1 will be equal to Q dash N to be more precise. So, you can write it as Q dash N, you can write it as Q dash N whereas, if the T value is 0; that means, this J and K both are both of them are getting 0. So, as a result, the value will not change. So, it will remain as Q N and if the clock is not active that is it is high low or making a high to low transition, then also it remains at the previous value.

So, this way, we can have this T flip flop made, it is a positive edge triggered T flip flop because the triggering is occurring at the positive edges of the clock, we can for all the flip flops that we have discussed. So, there can be negative edge triggered version also where at the falling edge so have seen previously, how to detect the falling edge of the clock signal.

So, when the following edge is coming, then the signal the value of the Q will change. So, here when the clock signal is low or high or making a low to high transition in those cases Q N plus 1 remains equal to Q N. So, the value does not change where as if it is making a transition from high to low, then if the T value is 0, then this Q N plus 1 will may maintain the previous value of Q that is Q N and if it is equal to 1, then it will toggle.

So, it will change. It will become equal to Q N bar. So, this T flip flop is also a special case of JK flip flop. So, you see that often this JK or SR it is actually the main flip flop that we design and from there, we can derive this remaining type of flip flops. So, from SR flip flop. So, you can come to D or JK flip flop or you can from the JK you can come to D or T flip flop. So, like that you can get one flip flop from the other.

(Refer Slide Time: 04:28)



So, some flip flops they come up with some asynchronous inputs which are known as preset or clear. Now one question like whenever I am suppose I give you 1 flip flop.

Suppose, I give you 1 flip flop and it is a some types say D type, ok. Now suppose this is say D and this is say Q and Q bar. And if I say; if I give D equal to 0, if I give equal to 0, there is a clock, then what is the value of Q? Ok the answer that we have got so far; it says that the value of Q will be same as its previous value, but what is the previous value of Q.

So, that is again dependent on the previous the time before that. So, that way how long we go back. So, that is we cannot go back infinitely. So, what is done is that there must be a starting point ok. So, there must be like if this circuit, if I make a circuit like this and switch on the circuit then what will happen is that this Q and Q bar. So, they will pick up some arbitrary values and from that point onwards depending on the value. So, it will go on operating.

But the system behavior will become unpredictable for SR latch, we have seen that we should not apply the pattern 1 1, but we do not have any control over that ok. So, we somehow want to the Q and Q bar should always be complementary that is also said, now we do not have any control, if we do not have any; if we just switch on the circuit and this Q and Q bar can pick up any value.

Now, how can we control it ok? So, for that purpose, most of this flip flops that we have studied, they come up with two extra controls which are asynchronous in nature one is known as preset another is known as clear. So, preset and clear as long as this, they are be 0 0. So, if it is in this particular case in this particular case. So, this is a rising edge triggered JK flip flop. So, if it is for these cases, if it is falling as long as they are falling edge falling edges or the high or low, this preset and clear both are 0 0, then nothing happens.

So, this remains it; it continues to the previous value, on the other hand, if this if there is a rising edge like this, if there is a rising edge here from low to high, then this preset and clear this clear is preset is equal to 0 and clear is equal to 1, then it is this Q N plus 1 will be equal to 1. So, that is we can we reset this flip flop value to 1; so, irrespective of the value of J and k. So, if this preset value is preset, control is made equal to 0, then this Q output will become equal to 1.

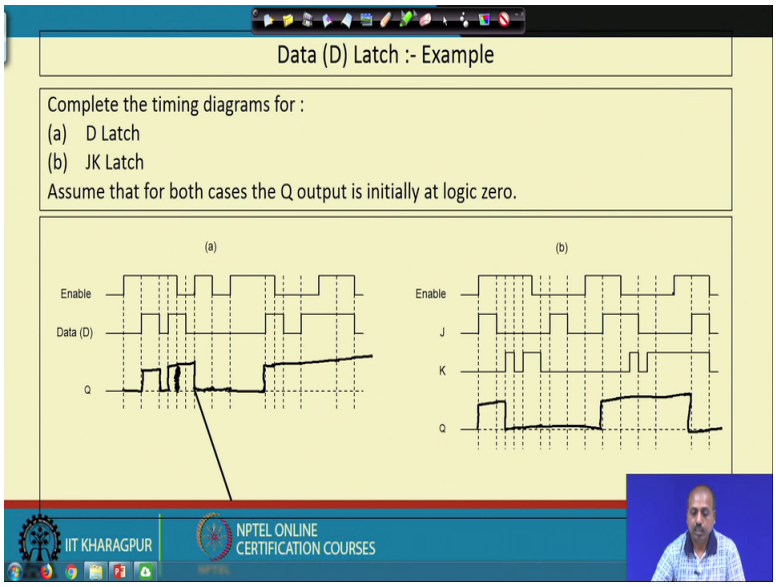
Similarly, if this depending upon this preset and clear, if this clear is made 0; so, irrespective of the value of this J and K, this Q N plus 1 will become equal to 0; the flip

flop content will be cleared. So, that is why the name preset and clear. So, irrespective of the values of J and K, if this clock signal comes the rising edge of the clock signal comes, then this preset and clear will be presetting the value on the other hand, for the proper operation of the flip flop, this preset and clear both of them should be made equal to 1. So, they are always made equal to 1. So, then you can just when they are made equal to 1. So, then this flip flop operates as a normal JK flip flop ok.

So, if we read this particular text, it says that 2 extra inputs are often found on flip flops that either clear or preset the output these inputs are effective at any time and that is why they are called asynchronous. If the clear is that logic 0, then the output is force to 0 irrespective of other normal inputs if the preset is at logic 0, then the output is force to 1 irrespective of normal input the preset at, and the clear inputs cannot be 0 simultaneously. And so, it is a design constant that is they cannot be made 0 simultaneously.

So, that is why it is the designer has to do the user ha to ensure that I am not making this preset and clear 0 simultaneously and if both of them are 1, then the flip flop will behave as a normal flip flop ok, otherwise, it will be behaving as per this preset and clear. So, this if the preset signal is 0, then this then this circuit will be preset and if the preset signal is if the clear signal is 0, then the circuit the flip flop will be cleared. So, that is why it behaves like this. So, we have got another exercise.

(Refer Slide Time: 09:30)



So, we have got say this D latch, how does it work? So, it is it says that a complete the timing diagram for D latch assume that both cases; both cases the Q output is initially 0. So, you see that this is D input is high at this point. So, till this much, it continues like this because enable is high, but D is also 0, then it is high, then it will become low, then it is it will follow that D as long as enable line. Sorry, this is not correct as long as the enable line is active. So, it will follow that D line, then enable becomes deactivate this point.

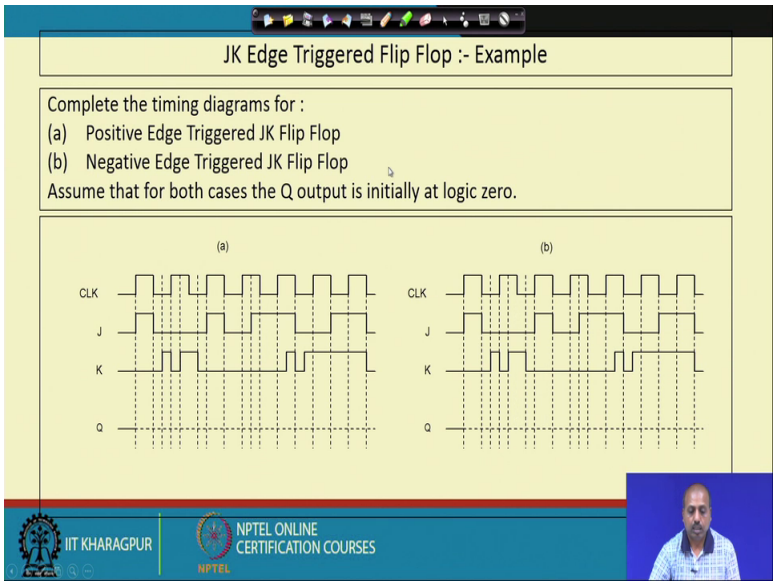
So, it will continue like this, then enable becomes active at this point again. So, this D value is 0. So, it comes down, then it then it goes like this enable is low enable is high. So, it continues like this D value is also low, then enable is low. So, it continuous like this enable becomes 1, but the D value is 0. So, it goes like this enable is 1 and D is 1. So, it becomes high, it goes like this, then enable becomes low, but D is D was high.

So, it continuous to hold the value like this next change can occur only at this point ok. So, D is already 1. So, nothing happens and then at this point enable is going low and D is also going low. So, it will continue like this. So, you can draw the timing diagram for the JK accordingly. So, here this Q is 0. Now this enable is high and J is 1. So, as a result, it will become high, it will continue like this, then at this point K is high. So, it will come low and then it will be K is again becoming high J is continuing to be 0. So, after this enable has become 0. So, enable becomes 1 at this point, but at this point J and K both are 0. So, as a results; it will continue with the previous value of Q and then at this point J is equal to 1.

So, it will rise like this and it will be then enable becomes low. So, as a result, nothing happens in this part of time, then enables becomes high at this point and then we have got J and K both are high at this point ok. So, it will be making a transition like this and it will continue like this.

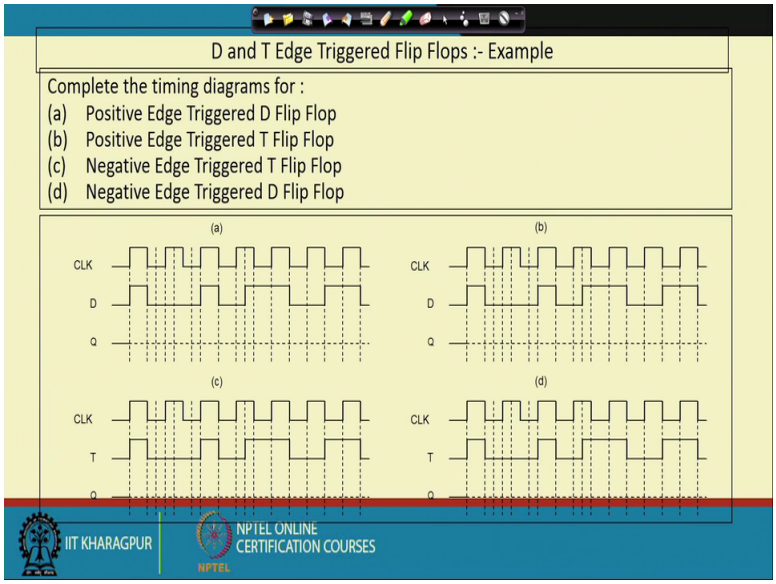
So, this way, we can make the timing diagrams for this D latch and JK latch, next we will look into some circuit changes.

(Refer Slide Time: 12:25)



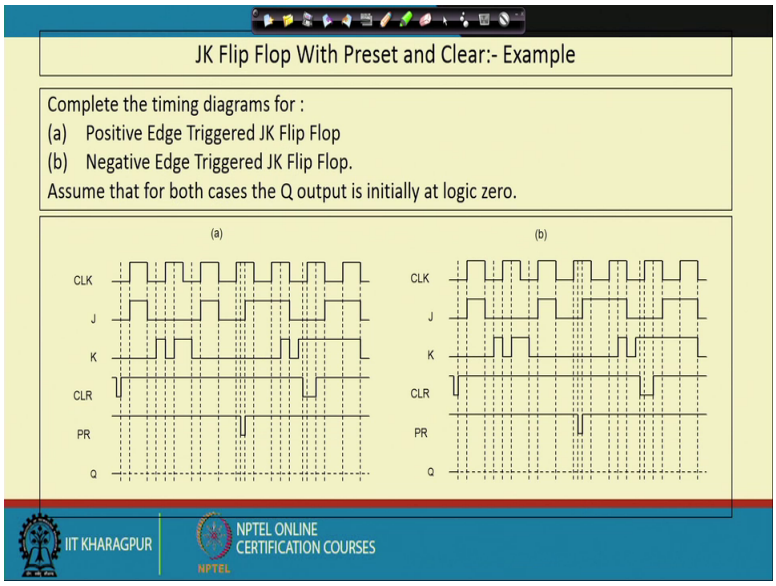
So, this diagram also, you can continue, I am not doing it. So, you can just draw the timing diagram fill up this Q Q bar and Q output in a similar fashion.

(Refer Slide Time: 12:40)

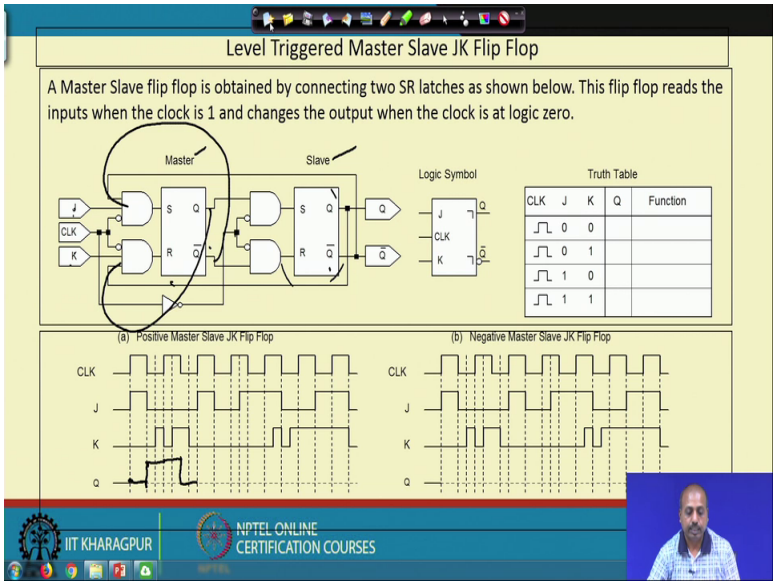


This is another example. So, you can I am just putting this slide so that you can just make a few practice ok.

(Refer Slide Time: 12:45)



(Refer Slide Time: 12:48)



Now if we try to there is another type of JK flip flop which is known as master slave JK flip flop. So, the problem with edge triggering was that. So, we were requiring some extra edge detection circuit for making it to operate properly. So, if we instead of that what we can do? We can use a use a 2 such flip flops 2 such JK flip flops and the ones 1 is called the masters flip flop.

So, this is the master and this is the master flip flop and this is the slave flip flop. So, master flip flop operates at the point when the clock signal is high and the slave operates

when the clock signal is low because there is an inverter connected like this. So, you see that when the clock signal is high then this J and K. So, depending upon this value of J and K this Q and Q bar will come. Now during this, then the then the clock signal will become low when the clock signal becomes low, then what ever values of the whatever the values of this J and K. So, they cannot change this SR flip flop.

So, the previous value will be remaining latched here. Now this slave flip flop will be following the JK flip the master flip flop. So, whatever master have done previously slave will do now. So, this way we can. So, this 2 SR latches. So, the flip flop the reads the value of inputs and the clock is 1 and changes the output when the clock is at logic 0. So, here for example, this J and so, when the clock signal is high this Q output does not change ok. So, this Q output will change when the clock signal is low. So, when the clock signal was high this J and J was 1 and K was 1. So, it evaluates to 1 and when the clock signal becomes low at that time the signal makes a transition, it goes, it becomes high like this.

Now, again at the next time instant; so, by that time when the clock signal was now when the clock signal becomes high depending upon this J and K. So, J value is low here K value is also low here after sometime K value becomes active K value become. So, so it becomes to 0. So, it this master flip flop becomes equal to 0 at the end of this clock.

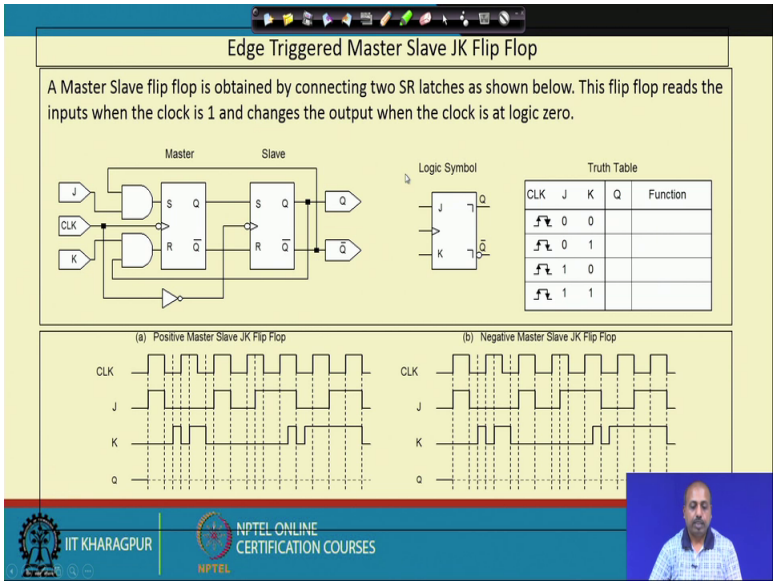
So, at the end of at this clock boundary; at this clock boundary when the clock signal becomes low; so, it will be making a transition to low and it will continue like this. So, this way depending upon this clock signal boundaries, so, whatever the master flip flop has made a transition to the slave flip flop will make the transition in the when the clock signal becomes low. So, this is the master slave JK flip flop that. So, you see that we need a both rising and though both level high and level low situation of the clock for the output to get affected ok.

So, this way so, the advantage is that here also, we do not have that continuous toggling problem, because for this the slave flip flop will change only when once only during when the clock signal is low, and then only those values will be available to the J and K, this the first 2 AND gates. So, these 2 AND gates are values will be available only in the next clock pulse. So, in between previously, what was happening is that the feedback values were coming feedbacks are coming from this points ok, but now the feedbacks are

coming from the slave. So, so slave will not change immediately when the master changes. So, as a result the master will see the change only in the next clock. So, that continuous toggling problem that we had in the JK flip flop or JK latch that will not happen in the master slave organization.

So, this also solves that master toggling problem of in JK in JK latch.

(Refer Slide Time: 16:59)

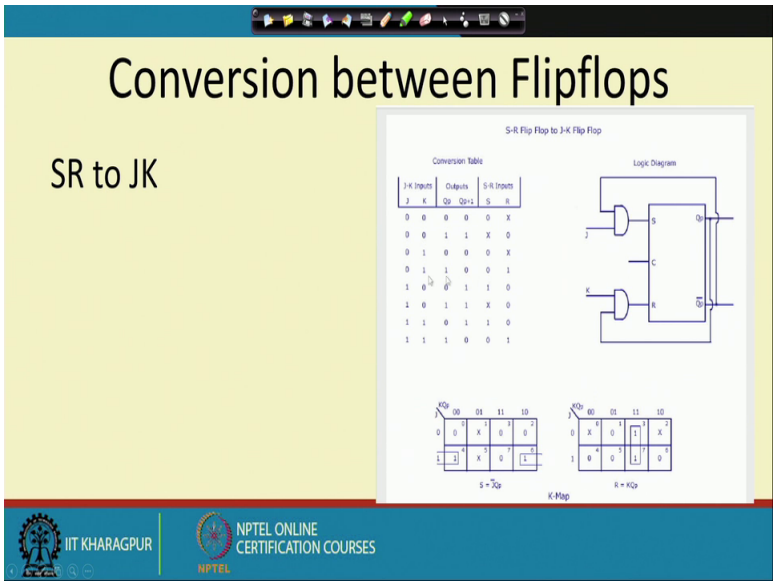


So, we can have this master slave JK flip flop also, the same thing that is you have got this 2 flip flops, and then a 1 flip flop operator on positive edge triggered 1 edge of the clock and the other. So, this works at the negative edge of the clock and this works at the positive edge of the clock, then slave will operated operate at the positive edge of the clock and accordingly, the master will read the values the values coming on the J and K.

So, they will be read by the master when the clock signal is. In fact, when the clock signal is 0, then the value will be coming here and when the clock signal is one then the value will be outputted by slave. So, here it is written as when the clock is 1. So, in that case this bubble should not be there and in that it is written that it is evaluated on the clock is 0, in that case, this bubble should not be there.

So, there is a problem with this diagram and this right up anyway, it conveys the basic idea.

(Refer Slide Time: 18:02)



So, next we will be looking into some conversion like how can you convert 1 flip flop to another type of flip flop that is very common like; it may be while doing the operation. So, we are looking for some type of flip flop, but what we are getting in our laboratory is of a different type. So, we need to control do some conversion between the flip flop the first conversion will look into is SR to JK flip flop. So, basically we have got an SR flip flop and we want to make a JK flip flop out of that. So, for doing this thing, this type of exercises first we have to make a conversion table. So, in the in the conversion table, we write down the values of this JK and the current value of Q and what is the expected; next value of Q.

So, these if I take these three as input JK and Q P; so, this is 3 as input and it see; what is the value expected value of output. So, by from the JK flip flop, we know if it is 0 0 0 Q P plus 1 is also 0 if it is 0 0 1. So, it maintains the previous value. So, this Q P plus 1 equal to 1 that is equal to Q P it is 0 1 0, then irrespective of the value of Q P the output is 0. So, Q P plus 1 is 0; so 1 0 0.

So, irrespective of if the J input is 1 irrespective of the previous and K input is 0 irrespective of the previous value of Q P, the values are one and for if J and K both are equal to 1, then if Q P was 0. Now it will become one if Q P was 1. Now it will become 0. So, if we now we try to mimic this behavior by means of an SR flip flop that is in SR

flip flop, we will try to put the values in S and R such that we get this type of behavior that is in a flip flop I have got the present state as 0.

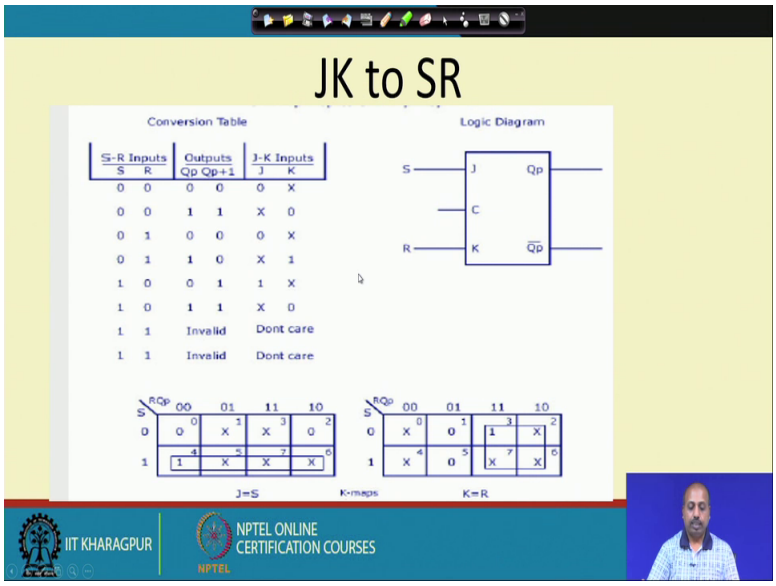
Next state we also want as 0. So, what are the values that we should feed to a S and R and you know that only if I in SR flip flop if I set this S input to 1, then only output will become equal to 1. So, if I maintain this S input as 0 and irrespective of the R input, it will maintaining the maintain the value 0 0. So, if R input is 0, then also it will remain 0 if R input is 1 that also will be said the flip flop and take it to 0. Similarly, if you have if the current value is 1 and the next value expected is also 1, then what is to be done. So, you can ignore the value of S. So, you can feed it to a; we can feed 1 or 0 2 S, but you have to feed a 0 to; so, as a result if you if you set R to 1, then this will change to 0.

So, what is restricted is that this R should, must be equal to 0. So, should get 1 1 here, similarly, we should if we are if the current state is 0 and the next here also it is same thing; so here also 0 X 1 0. So, if we want that the current value is one and the next value, we want as 0, then the R input should be made equal to 1 and S input should be made equal to 0.

Similarly, 0 1 a S input should be made equal to 1 and R input should be made equal to 0. So, in this way we make we write down the values to be applied on S and R inputs of the flip flop now we make corner map that has got the inputs JK and Q P and the outputs as S and R. So, far the S, we make the corner map here and for R, we make the corner map here and after simplification it turns out that we should make it as J, there is there is a it is J Q P bar ok.

So, S equal to J Q P bar and R equal to K Q P. So, if you simplify. So, you will get it like this. So, you see that we are what we have done. So, you have connected 2 and gate and this Q P bar line is taken here and connected here and Q P line is taken here and connected here. So, as a result, now this structure it behaves as a JK flip flop. So, in this way if we have got one type of flip flop; so, you can very easily convert to another type by making by putting some additional logic into it.

(Refer Slide Time: 22:24)



How to do the other side that is from JK to SR for JK to SR?

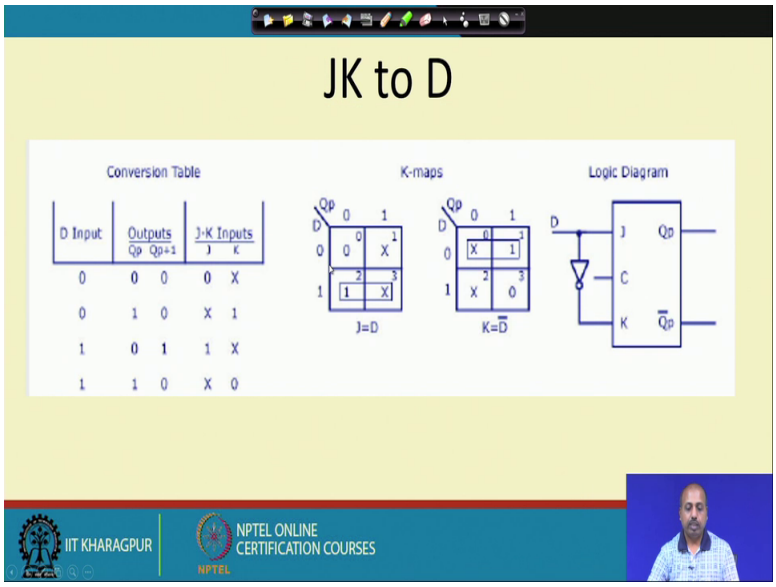
So, ultimate behavior expected of the S R flip flop is like this that it is 0 0, if it is 0 0, then the if the if the S and R are 0 and Q P is also 0, then Q P plus 1 should be equal to 0 if it is 0 0 and 1, then Q P plus 1 should be 1 that is it is holding the previous value if it is 0 1, then this Q P whatever be the value Q P plus 1 will be equal to 0. Similarly, 1 0 whatever be the value of Q P output be equal to one and for SR flip flop or S R latch, we do not apply this 1 1. So, these 2 are not put; so for J K.

So, we can simply consider them as do not cares. Now if we want to get this behavior that is what value should be applied to J and K such that we get this behavior that is Q P is 0 Q P plus 1 is also 0. So, here also this is if you put J equal to 0 if J equal to 1, then this Q P plus 1 will be equal become equal to 1. So, we have to make J equal to 0 K is do not care similarly 1 1. So, K should be equal to 0 and J is a J is do not care, similarly 0 0.

Again the same thing J is 0 and K is do not care. So, in this way, you can fill up this J and K part and then you can do a truth table corner map and try to simplify this corner map and then what you get is J equal to S and K equal to 1. So, you do not need any additional gate to convert 1 JK flip flop to SR flip flop and that is quite expected because that is the we have seen previously also the behavior of JK and SR, and we have seen that they work exactly like that JK and SR only thing is that that in SR, this 1 1 input pattern will come that is not there in the JK.

So, we are not bothered about the 1 1 input pattern for JK.

(Refer Slide Time: 24:28)



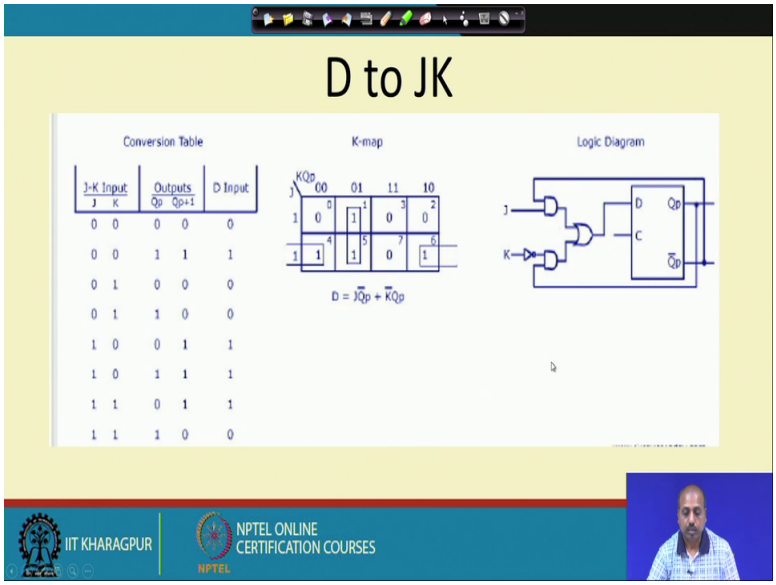
Next, we will be looking into the technique how to convert a JK flip flop to a D flip flop take the strategy is similar. So, for the first, we write down the table for the D flip flop in the D flip flop D input is 0 and Q P is 0. So, we want that Q P plus 1 should be equal to 0 if it is D is 0 and Q P is 1, then also it is 0 D is 1 and Q P is 0. So, this Q P plus 1 should be equal to. So, this is the D flip flop behavior. Now if we want to realize this behavior by means of an SR JK flip flop, then what we have to do is. So, I have to apply a 0 at J because to get a this K input can be do not care, then it can be since the Q P is 0 it will always come to 0.

Similarly, so, if we are looking for if the previous value of Q P was equal to 1 and we are trying to get a 0 here, then this K value must be set equal to 1, then it will be resetting it to 0. Similarly, if the previous value of Q P was 0 and the next value is 1. So, we should we have to set this J value to be equal to 1. So, that it makes a transition from 0 to 1 and from 1 to 0 again, this x and this K value.

Since this is the D input is 1. So, this is a 1 to this sorry this is D input is 0 previous value of Q P is also is 1. So, I have to make this a. So, then if we do a simplification of this corresponding K map; so, it will turn out that between J and K. So, J is equal to D and K equal to D bar. So, previously you I have said that while making the D flip flop that you have to connect an inverter between J and K inputs. So, these particular derivations

shows that you can do that really that is really the case that we have to connect an inverter between J and K.

(Refer Slide Time: 26:35)



So, can we convert a D flip flop to JK flip flop ok? So, how can we do this now you see that this is a bit cumbersome because this JK. So, we have got this truth table like JK 0 0 0, then the expected. So, this is a expected behavior. Now we try to note down what should be the D value like from 0, I want to go to 0. So, D input should be 0 1 to 1. So, that should be 1 0 to 0 0 1 to 0 0 0 to 1. So, I need to put a value 1 so that way I write down this D input color.

Now, we make a truth table out of JK and Q P for the D and it if you do a simplification, then you can find that it turns out to be a function D equal to J Q P bar plus Q bar Q P. So, if you are given this add this from this J and K we have to derive this two this function for D ok. So, J Q P bar. So, Q P bar is taken it is ANDed and K bar Q P the Q P is taken K is inverted and ANDed and they are ORed here and then put to D; so this logic. So, this will be converting 1 D flip flop into a JK flip flop. So, internally it remains a JK flip flop; but sorry internally it remains a D flip flop, but you we can realize a JK flip flop out of that.