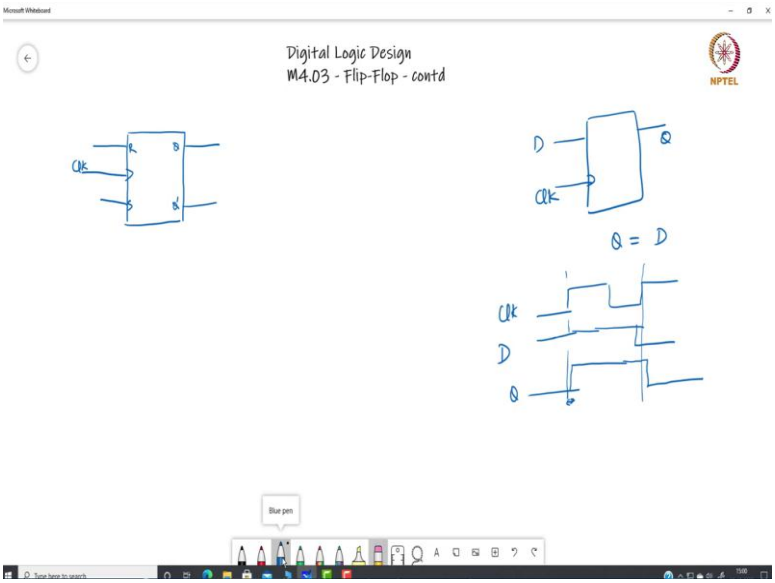
Digital System Design
Professor Neeraj Goel
Department of Computer Science Engineering
Indian Institute of Technology Ropar
Lecture 43
More Flip-Flop

(Refer Slide Time: 0:15)



Hello everybody, today we are going to discuss further on the flip flops. In our previous lecture, we have understood a flip flop is an edge triggered memory device that at the time of whenever clock transition is there, an active clock transition is there, then the output can change and how output will change that depends on the type of flip flops. So, for example, in our previous lecture we have seen we have seen D flip flops.

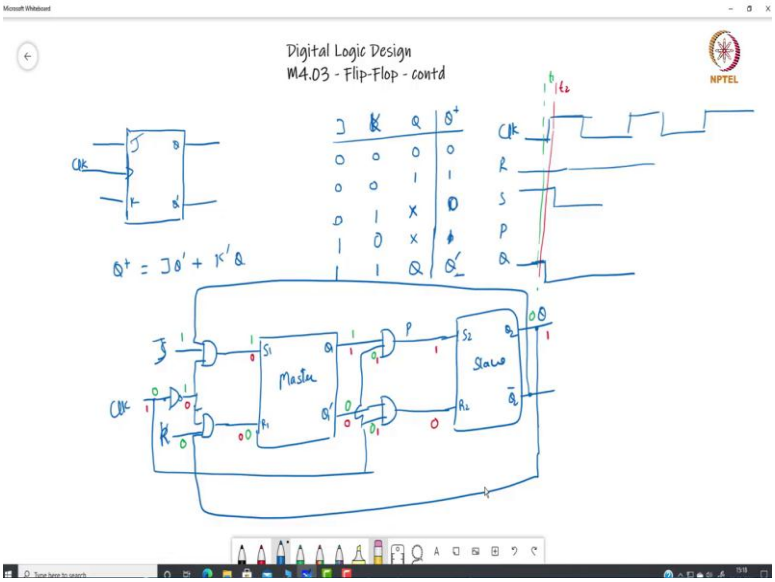
In case of D flip flop the output, so output will change as whatever is the input at the positive edge. So, let us say this is my positive edge of the clock and D value this is my clock and D value is 1 here. So, whenever this positive edge will come, the output Q would be reflected at some, after some this propagation delay. So, after this propagation delay whatever is the value of d that would be reflected in the output. So, this is one this is the characteristic equation or this is the output function for D flip flop.

Similarly, there could be other different type of flip flops, where output will change whenever there is a positive transition or negative transition or the active transition of the clock at the active transition of the clock, the output Q or QR can change, rest of the time the output will have no impact at all. So, the output will not change until next positive edge will come.

So, at this point it will edge again we will see what is the value of D if the D value is 0 here. If D is 0, then the output will become 0, otherwise output will not change during one clock period. So, similar to D flip flop there are other types of D flip flop which we are going to discuss today. One of them is RS flip flop.

So, this RS flip flop is actually the fundamental flip flop, because similar to RS latch, this flip flop is also is kind of a very basic flip flop and this is also fundamental because all other flip flops can be designed using RS flip flop. When we see in the production basically the chips which are produced or flip flops which are used in different circuits, usually RS flip flop is the one which is rarely used because of some constraint that will discuss now. But otherwise, this is fundamental because it is most easy to design and all other can be derived using this.

(Refer Slide Time: 3:36)



So, let us quickly see what is the behaviour of this RS flip flop, the behaviour of the RS flip flop can be mentioned using this truth table. So, if the value of RS and Q is this, then will write what is the value of Q plus. So, if it is 0 and 0, if the Q was 0, the output is also next state is also going to be 0. If Q is 1, then next state is also going to be 1. So, it depends on the previous state what is going to be next state. However, if R is 0, S is 1 then output it does not depend on the output, but Q plus is always going to be 1.

Similarly, if R is 1 S is 0, it does not depend on the previous state, but output is always going to be in 0. If both of them are 1, then this combination is a invalid combination. So, that is the input we should not give and essentially because of this particular reason that R and S both cannot be 1 that combination does not is not valid in case of RS flip flop that is the reason that these are not used in many of the circuits.

Now, let us see one of the implementation of RS flip flop. So, RS flip flop can be designed using master slave combination. This is called master and this is called slave. This is called master because my original inputs are connected to master and whatever is the output of master is connected to slave.

So, because it is edge triggered and we have a clock signal here, so we give the inverted clock as a input to this, inverted clock is used as a gate to the master flip flop. So, this is sorry this is S and this is R. Now, the output of master flip flop master latch is given as an input to the slave latch and it is gated with the original clock without inverted. Now, let us say how it works. So, let us consider this as a intermediate input P and this is my final output Q.

So, let us consider that this is my clock signal and this is value of R this is S. So, initially let us say R is 0 and S is 1 and then we can see the value of P as well as Q. So, let us consider the at some point t, which is slightly before the positive edge. So, this is a time T1. So, at time T1 the clock value is 0.

So, inverted clock value is going to be 1 and at that time my R is 0 and S is 1 correct. So, at this time, because S is 1 and inverted clock is 1, this will become 1 and this will become 0. Because this is 1 this is 0 set is 1 reset is 0, so Q will become 1 and Q dash will become 0, clock is 0 at this time, this is 0 at this time, so my output is going to be the previous state.

And let us assume the previous state is 0. So, let us assume the previous state is 0. So, we do not know. This is 0 so that means this is going to be 0. And we do not know what is the value here. So, let us consider it as P was also 0. So, now, let us see what would happen at some time after this positive edge at very, very small difference of time after positive edge, let us consider what would happen let us consider this as T2.

So, at T2 my clock has become 1 and because it has become 1 the inverted clock has become 0, this became 0, this became 0 and now P value will be the previous value. So, this Q 1 will still remain 1, Q 1 dash will still remain 0. Now, the clock has become 1 this has become 1. So, the value which is given as the input to slave set is 1 and slave reset is 0 because of this 0.

Now, because of that my Q is going to be 1, but when it will be 1 only after clock has been there and when this 1 will reach here this 1 can be only after clock has become 1. So, after some small duration after clock has become 1 my output will become 1. So otherwise, because of this gate, because of this gate, this master latch will not change its value when clock is 0.

And because of this gate, the slave flip flop will not change its value when clock is 1. So, the output will change only at the positive edge of the clock. And that is how this master slave combination will work like a RS flip flop. Now because of this combination, when R and S both are 1 the input combination is not allowed. This input combination is not allowed because both of these J and K flip flop. So, both of these master flip flop, master latch and slave latch, they will not work correctly.

So, because of that, my master slave RS flip flop will also not work correctly. And that is why we need to find some alternative method which could serve as a better flip flop. That is why JK flip flop is used. In case of JK flip flop, let us do it within this diagram only. So, in case of JK flip flop, what do we do?

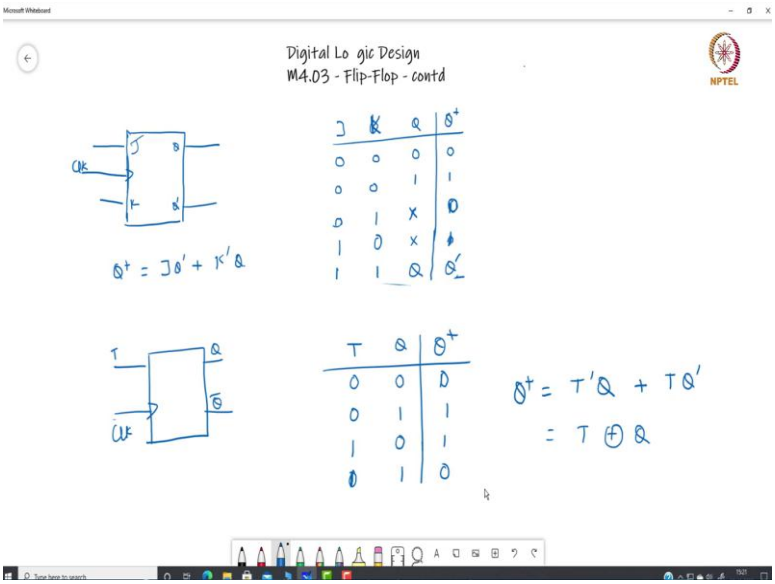
JK flip flop is very similar to RS flip flop, the only thing is the input a J and K you have Q and the characteristic table is also somewhat different in case of J and K if a J and K both the inputs are 0 Q plus the next state is same as whatever is the Q, if J is 0, K is 1 then the output is; if J is 0 and K is 1 then the output is 0 and when J is 1, K is 0.

So, this is K when J is 1, K 0 the output is 1, then both of them are 1 then the output is the inverted output of the previous state. So, if we write the characteristic equation for the characteristic equation, it will become something like this that next state depends on J Q dash plus K dash Q and as far as implementation is concerned, so, the implementation is very similar to my J my RS flip flop.

The only difference is that, I will now call this particular input as J and this I will call as K. And similarly, I will use as RS master latch and RS slave latch to make sure that this invalid combination does not occur, what would be done that whatever is this input, the output final output is given as a feedback loop to this input and whatever is the final output here that is given as a feedback loop to my master latch, this feedback loop make sure that this 1 1 combination will result in the previous or the reverse of Q.

So, you see that J is given input as a Q dash and K is given as input as Q. So, because of that your output will be the inverted 1. Now, there is another flip flop which is also quite commonly used, that is called T flip flop. So, in case of a T flip flop, we use this particular idea of J and K. So, let me erase this part.

(Refer Slide Time: 13:55)



In case of a T flip flop like D there is only 1 input T and we have a clock as a input here this is Q and this is Q bar. Now, in case of a T flip flop, if T is 0 and then whatever is the previous value of Q this will Q next state is going to be the if T is 0, then it is it is going to be same. If T is 0, the output is going to be Q or yeah, I can write it like this.

But if T is 1, the output is going to be the inverted value of the previous state sorry this is 1 and this is 1 then this will be 0. So, essentially my Q plus is always going to be if T is 0 then it is going to be equal to Q and if T is 1 then it is going to be inverted of Q dash. So, this can be written as the characteristic equation of my Q dash I can also write in this T XOR Q.

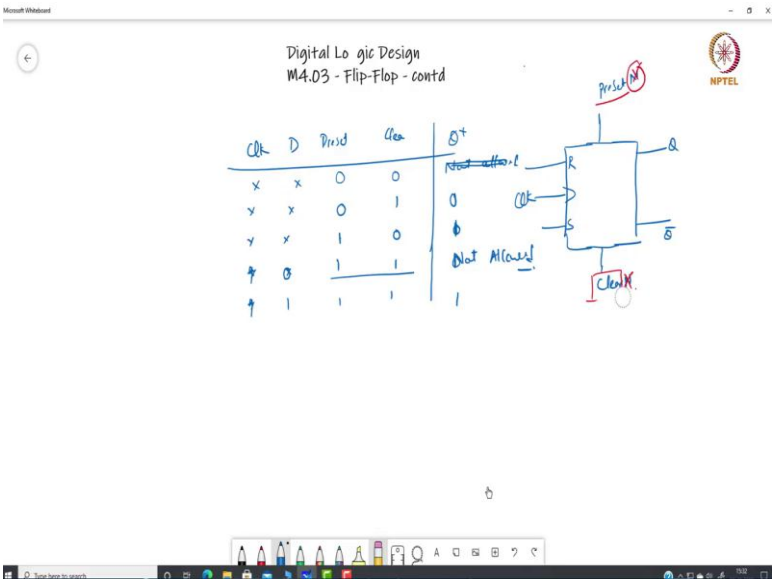
The implementation of this can be like, if I connect J and K as the inverted ones, then also we can design the T flip flops. So, these T flip flops, we will see that they would be used in counters and we will see them in some of the next lectures. So, these are the different type of flip flops, most of the time we use D flip flop because they are more simple one and most effective to store the memory.

But some time for example, for counters or for storing certain kinds of combinations T flip flops are also used and JK flip flops are also used in some cases. Now, in addition to this general category like different type of flip flop, there are also couple of additional inputs, which we should see that which could be applied in these flip flops.

So, one of the important one you see that when we initialize these flip flop how do we know that what should we the initial value, we always say that your flip flop if your input for

example, if it is RS flip flop, if R and S both are 0 the output is whatever is the previous state and what is the previous state we do not know.

(Refer Slide Time: 17:30)



So, because of that, an additional input to these flip flop has been designed, which is called clear or reset. So, let us say this is my R and S flip flop. So, this is true for all other D flip flop also. Now, there are two additional inputs preset and clear and this is Q this is Q bar. Now, these preset and clear are usually asynchronous inputs.

So, that means they does not depend on clock so when the positive edge or negative edge of clock will come it does not matter on them, but this preset and clear can initialize the inputs initialize the outputs. So, usually you will see that most of your hardware, they have a small reset pin.

So, in any of the hardware you will see some at least one reset pin which reset pin will ensure that the starting operation is clear. So, which will initialize all the memory elements to correct values. So, and this reset pin has to be independent of clock because after the clock will come into picture clock will your next value will depend only on the, only at the certain point of the synchronization points.

So, that is why these preset and reset are usually asynchronous inputs and whenever they are given their value would be affected. So, in another way this reset and clear, preset and clear can change the functionality of my flip flop. So, that is why it has to be the care has to be taken that preset and clear values are only given at a certain point usually on the in the initial ones.

So, and it is also some time there that these this is the inverted values so that if it is inverted then we can call it preset N or clear N. So, what is going to be the behaviour. Whenever preset, whenever clear is 0 it will make Q as 0, and whenever preset is 0 then it will make Q as 1. So, this is the, let us see it using a truth table, let us write clock D preset N, clear N and this is value of Q plus. So, independent of this, if the value is actually both of them if it is inverted value both of them would be 1 that is not allowed, if clear is 1.

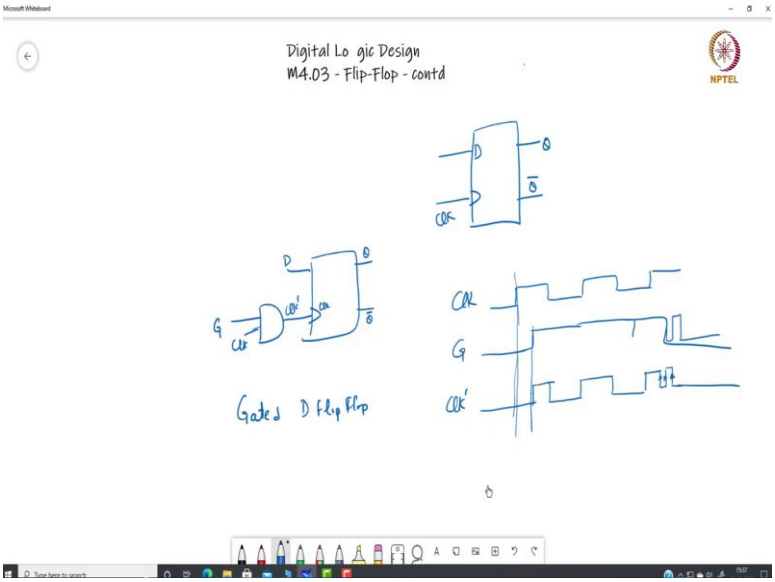
So, when preset is 0, so that because it is active low so, when preset is 0, the output is going to be 1 and similarly, when a clear is, clear is 0, that means clear is active and then output is going to be 0. Similarly, if both of them are 1 then circuit will behave according to clock and T. So, if D is 0 then output is going to be 0 and so, this only happens at the positive edge. So, at the positive edge if D is 1, the output is going to be 1.

So, we must understand that this circle means active low if the circle is not there, so we have to consider that let us say if this this active low signal was not there, in that case, we can say this is only preset and N is not there and there is only clear N is not there and then the circles are also not there.

So, we can reconstruct this. So, we can say this is circle is not there, and this is preset and clear. So, in case of preset and clear, which is non-negative edge. So, that means, in that case, if it is 0 0 then, if it is 0 0 then circuit will behave like a regular circuit, but if clear is 1 then output is going to be 0 if preset is 1 then output is going to be 1 when both of them are 1, it means it is a conflicting situation, this situation is not allowed only one of the preset or clear would be there as the as the input.

So, this would be a condition of not allowed. So, this preset and clear are one of the additional input circuits which are also commonly used in some of the flip flops. Along with that now let us consider one more scenario before finishing this lecture. So, let us consider one more one more scenario.

(Refer Slide Time: 23:31)



Now, let us consider that we have this D flip flop, in case of a D flip flop you know that whatever is the value of D, whatever is the value of D that would be reflected in the output. Let us say we do not want to do that, we want to say that D value should not be reflected with every change in the clock. So, what should we do? This particular question is also coming because as the transitions are there, we each transition consumed product power.

Now, to avoid that, can we do something like if we want only during certain time certain period of clocks, then we want this D to be transparent otherwise we do not want. So, then we can also gate this clock. So, that means we can so this is my regular D flip flop, I can create a gated D flip flop. So, one of the idea of gated D flip flop could be something similar like let us say I have this value of gate and I use this clock and then I use this as an input to my regular clock.

So, this is a straightforward circuit what this will ensure that if I do not want to use clock or I do not want my flip flop to change its value every time when clock is changing, then what I can do is I can use a gate I can, I can. So, in other words, what I am doing is I am redefining my clock, I am saying that I am using a new clock, which is a gated clock.

So, this gated clock is a very powerful mechanism to remove certain activity in the flip flops whenever it is undesirable. Now, this has certain disadvantages also, this particular implementation has certain disadvantages also because what you are doing now, clock is otherwise used as a synchronization mechanism.

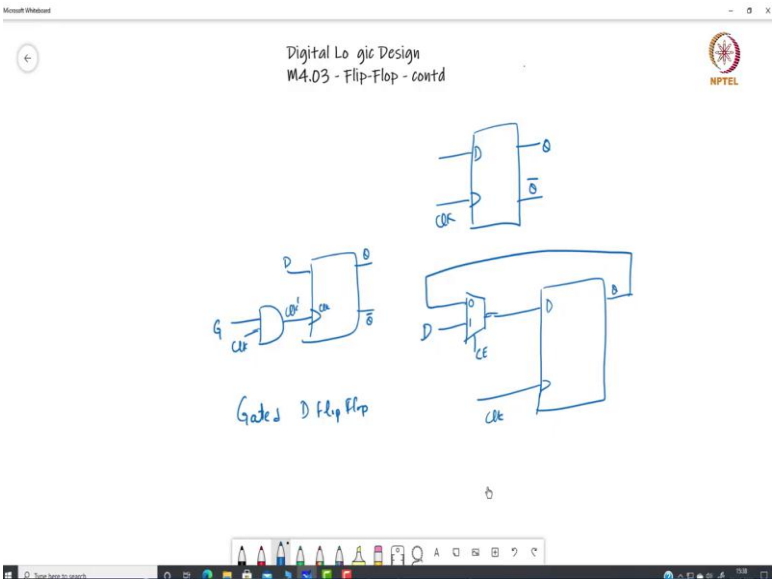
Now, you are redefining the clock because you are redefining the clock this new clock, which is a gated clock is shifted by one gate delay. So, that means now because it is shifted by one gate delay it is not does not remain a global synchronization point, for different circuits, there would be different synchronization point.

The other thing is let us say this gate behaviour is this gate behaviour is not synchronized with clock. So, let us say I am doing this this is my clock signal and this is my gate signal. So, gate signal is let us say want to avoid. So, I would like to do something like this or there is also possibility that it does something like this. Now, what would happen, this is my clock dash or my new clock in the new clock what is happening now, first of all, it is not synchronized.

You see because of this. Now, even after this, this there would be a certain delay of this AND gate. And because of that my clock dash is easy when shifted. The other thing is, let us say because of certain reason the clock has a flicker, this gate signal has a flicker, and because of that there will be multiple there would be a, yeah so there will be multiple positive edge which is created.

And each positive edge can create some side effect. So, that is why this particular mechanism where we get the clock, clock being the signal, which is controlling everything, it is not a good idea. So, what could we have better on or an alternative arrangement, the alternative arrangement could be something like instead of getting the clock, what we can do is we can we can do some sort of a chip enable signal.

(Refer Slide Time: 28:09)



So, this is some, like we can create a D flip flop here. And let us say this was my original input D. Yeah, let us put it on the below side. So, let us say this was the original input D. And now we have some other signal like let us call it CE chip enable. So, this chip enable, if it is 0, then what it does is whatever was the Q value, it takes that as input, here we are not changing the behaviour of the clock, but if chip enable is 0, then whatever was the previous state, it will remain in that previous state if chip enable is 1 then whatever is the value of D that will go in the Q.

So, this could be some time of better implementation which can save some of the, which can control that when D has to be applied to the 2 over D flip flop. So, with this, let us close today's lecture and we will hope to do some more discussions or some more circuits based on these flip flops in forthcoming lectures. Thank you very much.