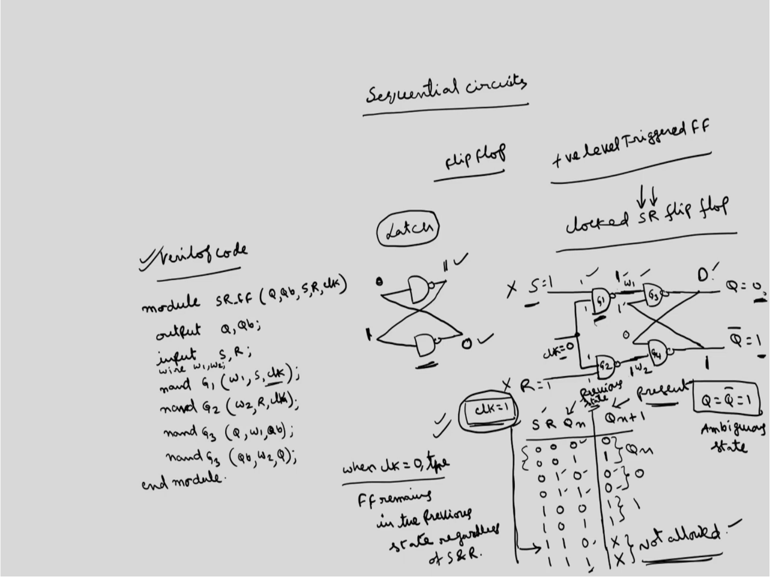
System Design through VERILOG
Prof. Shaik Rafi Ahmed
Department of Electrical and Electronics Engineering
Indian Institute of Technology, Guwahati

Gate level modeling - II
Lecture - 09
Review of flip-flops

In the last lecture, we were discussing about the combinational circuits and we wrote the Verilog codes for the various combinational circuits.

(Refer Slide Time: 00:51)



Sequential circuits

flip flop

+ve level Triggered FF

clocked SR flip flop

Latch

✓Verilog code

```
module SR_FF (Q,Qb,S,R,clk)
  output Q,Qb;
  input S,R;
  wire w1,w2;
  nand G1 (w1,S,clk);
  nand G2 (w2,R,clk);
  nand G3 (Q,w1,Qb);
  nand G3 (Qb,w2,Q);
end module.
```

clk=1

when clk = 0, the FF remains in the previous state regardless of S&R.

S	R	Q_n	Q_{n+1}	
0	0	0	0	Q_n
0	0	1	1	
0	1	0	0	0
0	1	1	0	
1	0	0	1	1
1	0	1	1	
1	1	0	X	not allowed
1	1	1	X	

$Q = \bar{Q} = 1$

Ambiguous state

So, the next type of digital circuits are Sequential Circuits. So, the basic fundamental sequential circuit is called Flip Flop. If I take a simple circuit which can store 1 bit of information which is called as a Latch, if I take two NAND gates and if I connect, this is single input, so I have shorted both the inputs, if I connect one output to other input, if I say feed logic 0 here.

So, this will be logic 1 this is not gate, this 1 will be sent back to this input of the second gate this will be 0. So, this is 0, this is 1, this is 1, this is 0. As long as the power supply is there for these NAND gates. So, this 1 will be stored here, 0 will be stored here. So, in order to store 1 bit of the information you can use a simple latch where we will use two NAND gates connected back to back.

But in many applications, so we need to apply these bits in synchronism with some inputs and also in sometimes we need to store the bits along with some clock signals. So, if I take, first, the flip-flop which is called clocked SR flip flop. So, S stand for set, R stands for reset; set reset flip-flop.

So, this can be designed by using the NAND gates here we have clock. This is S input, this is R input and we will be having two outputs; one is Q, another is Q complement. So, in order to understand the operation of the circuit, if I take S R, if I call this Qn as previous state and Qn plus 1 as present state.

So, one point that you have to note here is in sequential circuits the output not only depends upon the present input, but also on the previous outputs that is why the present output not only depends upon the present inputs, but also on the previous output.

So, there are four possibilities for this S and R, 0 0, but when S is 0, R is equal to 0, Qn can be either 0 or Qn can be 1; when S and R both are 0's previous output, Qn is 0 what will be the present state. When S is equal to 0, R is equal to 0, if previous state is 1, what will be the present state.

This can be easily analyzed from this circuit diagram; if I take this as 0 0 and of course, clock should be 1, I will discuss what happens if clock is equal to 0. So, I will assume that clock is equal to 1, I am assuming that the previous state is Q is equal to 0 means Q bar is equal to 1.

So, this is 0, this is 1. So, if 1 input for the NAND gate is 0 regardless of the other input output is equal to 1. So, both the outputs of these first two NAND gates are 1. So, this gate will be having both the inputs as 1. So, output is 0, this is 0 is tied here, 0 1 output is 1.

So, this is 1 1, output is 0. So, 0 1, output is 1. So, if the previous state is 0, present state is also 0. Similarly, you can verify this table in a similar manner I am just drawing the table; for previous state as 1, present state also we'll get as 1.

So, if 0 1 0, output will be 0; 0 1 1, output is 0. This is when reset input is equal to 1 regardless of the previous state, present state will be 0. Similarly, when set input is 1 regardless of the previous state, output will set, Q is equal to 1. Whereas, the drawback of this SR flip flop is 1 1 is not allowed. 1 1 0 or 1 1 1 is not allowed. Why this 1 1 combination is not allowed?

If I take this example. So, we will take this previous state as 0. So, this previous state is 0, this is 1. So, this 0 is tied here, 1 is tied here, but this S is equal to 1, R is equal to 1. So, for this NAND gate, both the inputs are 1, so output is 0. For this NAND gate, both the inputs are 1, output is 1.

If 1 input is 0 regardless of the previous state, output is equal to 1. So, you are getting Q is equal to Q bar is equal to 1 which is not allowed Q and Q bar should be complement to one another. So, this will give ambiguous state, even if the previous state is 1 or 0. So, this will give ambiguous state. This is not allowed.

So, this is the truth table of SR flip flop when both are 0, this remain set Qn. If reset input is 1 regardless of the previous state, 0; if set input is 1 regardless of the previous state, set 1 output is equal to 1 and 1 1 is not allowed; this is the case when clock is equal to 0. So, what happens if clock is equal to 0?

So, we know that if 1 input is 0 for the NAND gate regardless of the other inputs, output will become 1. So, this is 1 1, this is independent of S and R. Whatever this S and R; if clock is 0, then the output of these two NAND gates will be 1.

And if I assume that the previous state is 0, then this is 0, this is 1. So, this 0 is tied here, this 1 is tied here. So, this 1 1, the output becomes 0 and 0 1 becomes 1. So, it remains in the same state.

Similarly, when previous state if I assume that this is 1, it will remain in the 1 only. So, when clock is equal to 0, the flip flop remains in the previous state and this table is enabled only when clock is equal to 1.

So, we can also call this flip flop as now positive level triggered flip flop. Why? Because this table is valid only when clock is equal to 1; when clock is equal to 0, the flip flop simply remains in the previous state regardless of the inputs, you can also call as regardless of inputs.

So, but the drawback of this 1 is, this 1 1 is not allowed. So, in order to avoid this difficulty with SR flip flop, we have another flip flop, JK flip flop. So, if I take the Verilog code corresponding to this clocked SR flip flop. So, we can write the Verilog code corresponding to this SR flip flop. So, what are the inputs and outputs of this? We have module SR flip flop, outputs are Q and Q bar; I will call this one as Q and Qb, Qb stands for Q bar.

Then the inputs are S R and clock; then the output is Q and Q bar, you can write small q also instead of this. Inputs are S and R normally we will use lowercase letter, but here I have written uppercase letter, you can write using lowercase letter also.

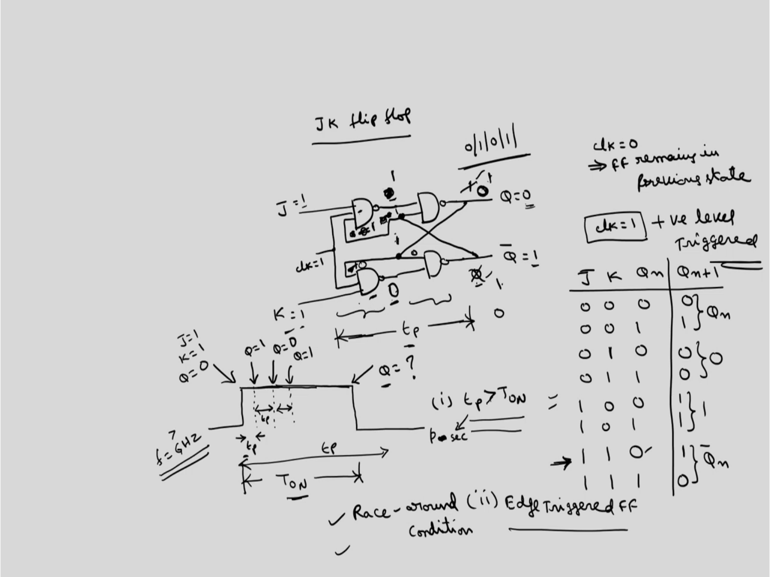
Then there are four nand gates; if I call this one as gate 1, gate 2, gate 3, gate 4; nand G1, the output of this nand gate is we have to define this point is not accessible. So, we have to define this as a wire; we call this one as wire 1, you call this one as wire 2, you can define wire w1 and w2.

So, the output of the nand gate G1 is w1, inputs are S and clock. And the second nand gate G2, output is w2, inputs are R and clock. Then for the third nand gate G3, output is Q, inputs are Qb and w1 and fourth nand gate output is Qb, inputs are w2 and Q.

Later we are going to discuss about the test bench. There I will write about when this output will change. So, you have to use the always statement, always at positive edge or positive level of clock, the output will change.

Before going for the details of Verilog, I will first discuss about the different types of the flip flops. So, I will review these flip flops after that I will go back to the Verilog codes. Now, the problem with SR flip flop is 1 1 condition is not allowed to avoid this, we will use JK flip flop.

(Refer Slide Time: 13:50)


JK flip flop

clk = 0
⇒ ff remains in previous state

clk = 1 + ve level Triggered

J	K	Q_n	Q_{n+1}
0	0	0	0 } Q_n
0	0	1	1
0	1	0	0 } 0
0	1	1	0
1	0	0	1 } 1
1	0	1	1
→ 1	1	0	1 } $\bar{Q}_n$
1	1	1	0

(i) $t_p > T_{ON}$

✓ Race-around Condition

(ii) Edge Triggered FF

So, in JK flip flop what we are going to do is. This is SR flip flop only with slight modification, this was S and this was R, this was clock in the earlier circuit that we have discussed and this will be feedback from this output, so this will be Q, Q bar this was simple SR circuit.

Now, to get JK flip flop what you will do is. So, we will add one more connection to this. Now, this becomes JK. By using this the problem with 1 1 can be avoided. So, this also you can verify that when clock is equal to 0, the flip flop remains in the previous state. When clock is equal to 1, you can check that J,K, Qn, Qn plus 1. So, 0 0 0 will get 0 itself, 0 0 1 will get 1 itself; this is simply Qn; 0 1 0, 0 itself; 0 1 1 also 0, this is simply 0; 1 0 0, 1 0 1 we will get 1.

So, the problem with 1 1. Now, I will take 1 1 0. Now, what will be the output? In case of SR flip flop; this condition is not allowed whereas, in JK flip flop. So, this condition is allowed; now what will be the output? Clock is equal to 1, J is equal to 1, K is equal to 1 and previous state Q; I am assuming as 0.

So, this Q bar is equal to 1. So, this is 0, this is 0; because the same 0 is this is 1 1 and this is 0 0 0 0 and these two are 1 1, this is also 1. Now, if I take this NAND gate, 3 inputs are 1. So, output is 0 if I take this NAND gate 1 input is 0. So, regardless of the other input, output is 1;

if one input is 0, output is 1. So, this 1 is fed here. So, 1 1 becomes 0. So, this 0 is fed back here, 0 0 becomes 1 and 1 1 becomes 0. So, when the previous state is 0, present state becomes 1.

Similarly, you can see that 1 1 1 the present state becomes 0 this is complement of the previous state.

Now, if you clearly observe here, here also there is some problem. Of course, it will allow this J is equal to K is equal to 1 condition. It is not giving any ambiguous state, but still there is some problem if you clearly observe this circuit.

So, you see here if I start with J is equal to 1, K is equal to 1. I can explain this with the help of a clock diagram, this is the clock where clock is equal to 1 because this truth table is valid only when clock is equal to 1.

So, at the starting of this clock what we have made J is equal to 1, K is equal to 1, Q is equal to 0; this was the situation. So, what happens is. So, this Q becomes 1 and 0, but after how much time? The time taken to propagate these signals from this gate and this gate if I call as this total time propagation delay of this 1 is say some tp.

And if this ON time of the clock is T ON. So, normally this t p is much much less than T ON. So, t p will be somewhere here. So, at this point what happens is, now Q becomes 1. Now, if you observe here this Q becomes 1. So, this 1 is tied here also here. So, this previous 0 will be now changed to logic 1 and this 0 is tied here, here also this previous 1 becomes now 0.

Now, what happens the situation will be different. Now, for this gate all the three inputs are 1. So, output is 0, for this gate 1 input is 0, output becomes 1, then if 1 input is 0 output is equal to 1. If this 1 is tied here, so this will be 1, 1 1 becomes 0. So, what happens here after another t p seconds, this Q becomes 0.

Now, again situation is the same, we will repeat this 0 is tied here. So, this 1 again becomes 0. So, this 1 is tied here this 0 becomes 1 and situation will reverse there by you will get here 1 and here 0. So, what happens is this output will change between 0 and 1, 0 and 1, 0 and 1 always for every t p seconds.

So, after another t p seconds, Q becomes 1 now if I want to know what is value of Q at the end of this clock signal this is indefinite. So, we do not know whether it is a 0 or 1 to exactly find out whether this is 0 or 1. So, we have to find out how many t ps are present in T ON, but that becomes difficult to estimate the values of t p. So, this Q value at the output of this at the end of this clock signal is again ambiguous.

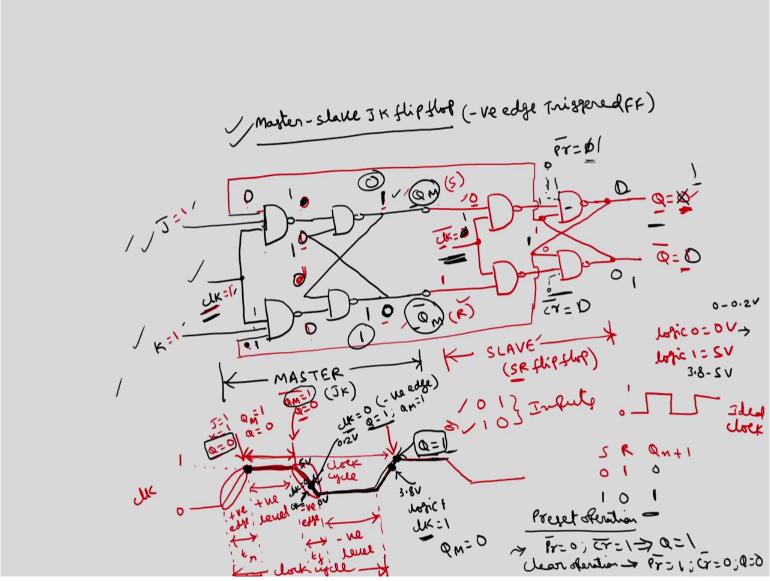
So, this particular problem is called race around condition. So, JK flip flop it will allow this J is equal to 1, K is equal to 1 combination, but still there is a problem called race around condition. The output will change from 0 to 1, 0 to 1 at every t p seconds. As I have told, this is called positive level triggered.

So, this is the problem with level triggered flip flops. There are two solutions to overcome this race around condition. So, one is if I design this circuit such that if this t p value is greater than T ON then how many times the output will change? Because this t p is now more than this, this is the t p. So, the output will change only once, the problem will be solved.

But this is only theoretical solution because t p is of the order of picoseconds even less than that if you want to design t p greater than this means. So, the clock frequency should be in the order of giga Hertz or tera Hertz, you have to design this clock with the frequency of giga Hertz or even more, several giga Hertz. So, this becomes difficult to design these high frequencies in the IC form.

Nowadays you have a lot of technologies. But that time this problem was there. So, this is only theoretical solution. The second one is you can use edge triggered flip flop instead of using level triggered flip flop, if I use edge triggered flip flop then the problem with the JK flip flop which is race-around condition can be avoided.

(Refer Slide Time: 23:29)



So, what is this edge triggered flip flop and how this race around condition is avoided? So, this edge triggered flip flop, the famous name for this one is called master slave flip flop. As the name implies, here we will use two flip flops; one is called master, another is called slave. So, the Master flip flop is JK flip flop.

So, we have clock here and here we have J input, K input and there is a third input that we are going to connect later. This we will call as some intermediate output QM, QM bar this particular flip flop is called Master flip flop and this is JK flip flop.

Then, I will draw the slave flip flop. These are final outputs Q and Q bar and clock for this 1 is, this clock is inverted and then applied as a clock bar. The same clock, there is a not gate. I am not showing the not gate directly, I am writing clock bar. Then this third input of this 1 is connected to this. If I connect this to here then there is a problem of race around there, I will connect this here. Similarly, this point we are going to connect as third input to this master flip flop.

Now, from here to here, this is SLAVE, this is S R flip flop because the problem with S R flip flop is only S is equal to 1, R is equal to 1 is not allowed, but here you can see that the inputs for these flip flops, they can be this is Q M and Q M bar. So, the inputs for this 1 is 0 1 or 1 0, inputs are applied for this flip flop.

So, for 0 1 or 1 0, there is no problem with SR flip flop. So, we will use the slave flip flop as SR flip flop. So, this is a circuit diagram of the master slave JK flip flop this will avoid the race around condition. So, how to verify that this will avoid the race around condition.

So, we take the same case of J is equal to 1, K is equal to 1 case. But the clock that I have discussed in the last slide, this is the ideal clock this I have taken this as clock 1, 0 to 1 it will take as 0 time, this is called ideal clock, but if you take the practical clock, it will take some time to change from 0 to 1 as well as 1 to 0; this is practical clock.

This is logic 0, this is logic 1; this much is the time taken to change from logic 0 to logic 1, but this is very very small even less than t p seconds. This you can call as rise time of course, this is 10 percent to 90 percent is called as rise time, but I am roughly calling as t r and this is t f fall time, this is also normal definition of the fall time is 90 percent to 10 percent fall.

So, this particular portion is called as positive edge or leading edge; and this particular portion is called negative edge or falling edge and this particular portion is called positive level and this particular portion is called negative level. Now, from here to here total 1 clock cycle.

So, what is required here in this JK flip flop is over this entire one clock cycle I have to change the output by only 1 when J is equal to K is equal to 1. If I start from this point then up to here this will be 1 clock cycle. So, I will consider this clock cycle from here to here, this is also 1 clock cycle you start with one point you have to end with the same point; this is also another clock cycle.

So, I want to demonstrate here that within this 1 clock period the output will changes only once. So, I will start with here J is equal to 1 K is equal to 1 and Q is equal to 0. So, what will be the situation. Now J is equal to 1, K is equal to 1, clock is equal to 1, this is starting of clock is equal to 1 1.

So, this clock bar will be 0 and this Q is 0, Q bar is equal to 1. So, we are getting Q is equal to 0, Q bar is equal to 1 means for SR flip flop, we know that to get a Q is equal to 0 for SR, Q n plus 1 what will be SR regardless of the Q n. So, to get reset input is 1. So, 0 1, this is S and R this will act as S input, this will act as R input.

So, we are getting the outputs as 0 1 means what does it mean. So, this is reset means reset input is 1. So, this is 1 this is 0. So, this is the situation of this circuit, the signals at various points at this point where this J is equal to 1 K is equal to 1 Q is equal to 0.

So, now, what happens because this clock is equal to 1. So, this master flip flop will be enabled, enabled in the sense it follows the truth table. So, what will be the outputs of this. So, for these this input is 1. So, this third input is 1. So, this input is this is 0. So, this is 0. So, for this NAND gate all the three inputs are 1, this is 0, this is 1, input is 0, output is 1.

And, what about this? This is 0 and this is 1. So, this 1 input is 0 means output is 1. So, this becomes 1 of course, in between you will get some transient state, but we want the steady state. So, this 1 1 becomes 0. So, this becomes 0. So, 0 0 becomes 1 and 1 1 becomes 0. So, these two states are called the steady states.

So, what happens is here when J is equal to 1, K is equal to 1, Q is equal to 0. So, this Q M becomes 1, Q M was previously 0. Now, Q M becomes 1, this is intermediate output.

Now, what happens to the final output because clock is equal to 0. So, this slave will be disabled. So, for a clock is equal to 0; we have verified in the previous slides that the flip flop remains in the previous state regardless of whether it is a JK flip flop or SR flip flop.

So, this Q becomes 0 only. This will happen throughout this period; here also what happens at the end of this positive level this is the positive level where clock is equal to 1. So, at this end also Q M becomes 1, Q becomes 0 means starting with Q is equal to 0 throughout this level Q remains 0 only. It has not changed the output.

Now, coming for the next portion where the clock is changing from logic 1 to logic 0. Ideally, we will take logic 0 as 0 volts, logic 1 as 5 volts, but practically there will be a range of voltages which can be taken as logic 0 and range of voltages which can be taken as logic 1.

So, the clock is changing from 1 to 0. Suppose, at this particular point, it will change to 0. This is the voltage corresponding to this is 5 volts, this is 0 volts, in between some voltage which is corresponding to logic 0, there is a range of voltages. So, this is the starting value where this signal can be treated as logic 0.

So, at this point we can call this as clock is equal to 0. So, at this point what happens? this master will be now disabled. So, this is the exact point where we are assuming that here this clock is equal to 0, this is in now negative edge. So, when clock is equal to 0 at this particular point.

So, what will be the status of the signals at different points? So, this clock 0. So, that this will be disabled; disabled in the sense the outputs of this master remains in the previous state. So, this master is Q M is equal to 1, this Q M is equal to 1 remains same.

So, Q M is equal to 1 means this Q M bar is equal to 0. So, these two values, this 1 and this 0 will remain when clock is equal to 0. Now, this master slave will be enabled because clock is equal to 0, clock bar becomes one. So, this will be enabled. So, for an enabled flip flop, the input is S is equal to 1, R is equal to 0. When S is equal to 1, R is equal to 0, what is Q n plus 1? Its 1.

So, this becomes now 1, this becomes 0 means output is changing. So, where exactly the output is changing at this particular point now what happens Q becomes 1 whereas Q M remains at 1 only. So, after that this will goes to 0, this is also logic 0 only; this is the starting point of logic 0, suppose if this is logic 0 is taken as 0 to 0.2 volts. So, this is a 0.2 volts, this point is corresponding to 0.2 volts.

Then if it decreases it will go to 0; throughout this, this will be treated as logic 0 only throughout this during the negative level also logic 0 only. So, throughout this even now after this logic 0, it will change to logic 1. Suppose, if I assume a point where. So, this logic 1 range says a logic 1 range is say from 3.8 volts to 5 volts this varies from the logic family to logic family.

Suppose, if it reaches 3.8 volts where we can take this as a logic 1, this is clock. So, till this logic 1 point Q remains in 1, but where this Q has changed? At this point. So, after this point, clock becomes 1. So, that master is enabled slave is disabled. So, the master is enabled means Q M will change at this point what will be this Q M.

Now, this 1 is tied here, 1 is tied here, this becomes 1, this J and Ks are ones, 1 only, clock is 1 and this will be 0. So, what happens this 1 this is 0 mean this becomes 1 this is all three ones are means 0.

So, this 0 is tied here; this becomes all the three inputs for this 1 are ones means this is 0 1 input is 0 mean this is 1 ok. So, this 1 was previously applied here this is 0 mean this is 1 this is 1 1 1 becomes 0 0 0 becomes 1. So, Q M becomes now 0. Previously, Q M was 1; now Q M becomes 0. So, Q M bar equal to 1. So, at this point Q M becomes 0, Q M bar is equal to 1.

But what happens to the final output? Because the clock of slave is disabled, this remains at the previous state. What is that previous state? At this point Q has changed to 1. So, this will continue until this Q M equal to 1. So, at this point what will be this Q is equal to 1.

So, I have discussed the total 1 complete clock cycle starting from this point to this point is 1 complete clock cycle. So, here we have assumed that Q equal to 0 whereas, here Q becomes 1 and how many times this Q has changed only 1 time and the exact position where this Q has changed is this because this point lies in negative edge this particular flip flop is called negative edge triggered flip flop.

Now, race around is not there because throughout this 1 clock cycle from here to here output has changed only once at this point. So, in that way we can avoid the race around condition. Now, in addition to this we require two more signals called as preset and clear signals. Here, we are going to apply preset signal and here we are going to apply clear bar signal, this is preset bar.

So, regardless of J and K if preset bar is equal to 0 clear bar is equal to 1; what happens for the final in this NAND gate; if 1 input is 0, output is 1 this 1 is tied here.

So, this clear is 1, 1 1 becomes 0. So, Q becomes 1; without applying the clock, without applying the inputs, if I want to establish the initial state, we will use the preset and clear operations. So, what is preset operation is without clock. So, we will make this preset bar is equal to 0, clear bar is equal to 1 implies you will get Q is equal to 1 this is called preset operation.

So, these signals are applied without clock and without inputs. Similarly, clear operation is reverse or clear operation we will make preset is equal to 1, clear bar is equal to 0. So, that you will get Q is equal to 0. If this is 0, output becomes 1, preset bar is equal to 1. So, this is also 1, this is also 1, output becomes 0. So, Q is equal to 0.

So, this preset and clear are called as asynchronous inputs which are used to establish the initial state of the flip flop. Initially, if we want to set the flip flop, we will use preset operation; initially we want to reset the flip flop, we will use the clear operation. So, this preset and clear will be useful in the designing of the counters. So, this is about the master slave flip flop.

So, we can write the Verilog code corresponding to this master slave flip flop and using this master slave flip flop, we can design the circuits such as the counters and shift registers of course, you can use the D flip flop. Basically counters uses this master slave JK flip flops.

So, we will discuss the Verilog code corresponding to master slave flip flop and two more types of the flip flops called as D and T flip flops we will discuss in the next lecture.

Thank you.