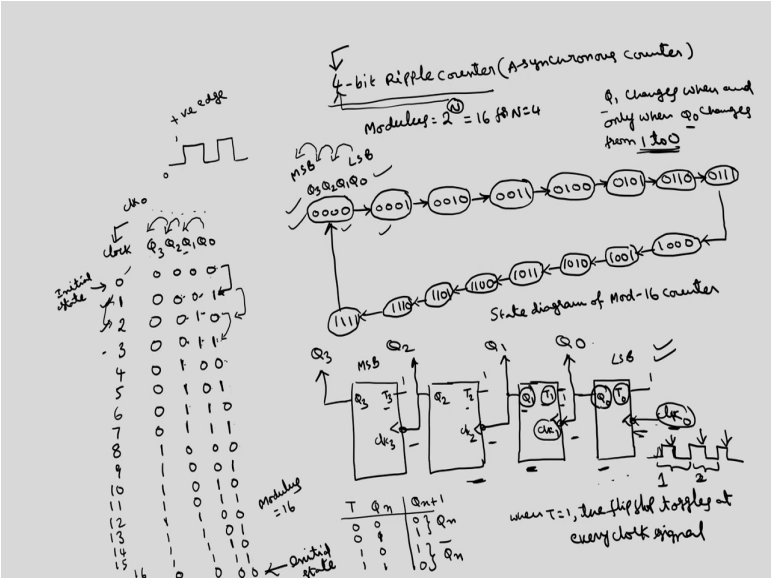
System Design Through VERILOG
Porf. Shaik Rafi Ahmed
Department of Electrical and Electronics Engineering
Indian Institute of Technology, Guwahati

Behavioral modelling of sequential circuits
Lecture - 18
Verilog modelling of counters

In the last lecture we are discussing about the Verilog code for the D flip flop and then we derived the T flip flop using D flip flop and we have instantiated this D flip flop code to write the code of T flip flop. Now the further extension is using T flip flop we can construct ripple counter.

(Refer Slide Time: 01:01)



If you consider 4-bit ripple counter. It will start with some state after some clock cycles, it will reach the same state. So, the number of clock cycles required to reach the same initial state is called as the modulus. So, if I take this 4-bit ripple counter the modulus of this one is 16 moduli will normally 2 raised to the power of N where N is the number of bits in the counter.

So, here N = 4, 2 raised to the power of N is equal to 16 for N is equal to 4. The counter will goes through the states you can write the starting state is 0 0 0 0 then it will goes to 0 0 0 1, 0

0 1 0, 0 0 1 1, 0011, 0 1 0 0, 0 1 0 1, 0 1 1 0, 0 1 1 1, 1 0 0 0, 1 0 0 1, 1 0 1 0, 1 0 1 1, 1 1 0 0, 1 1 0 1, 1 1 1 0, 1 1 1 1 after that it will goes to the initial state.

So, this is each state you can represent in the box or a circle, then this particular diagram is called as a state diagram. This is the state diagram of mod 16 counter. Now how to realize this? So, normally this ripple counter is realized using T flip flops. So, each bit we require one T flip flop. So, for 4-bit ripple counter we require 4 T flip flops.

I will start with the T 0 because these outputs will call as Q 3 Q 2 Q 1 Q0 this is MSB this is LSB. So, the output of this one is Q 0, this will require the clock, now to apply the external clock this is T1, Q1 is also required. The clock of this one is the output of the LSB flip flop.

Similarly, the clock of T 2 is Q 1, clock of T 3 is Q 2. So, these are the points where we will take the outputs. There is Q 3 Q 2 Q 1 Q 0 we will get this sequence. So, here normally in most of the books T 0 will be taken here T 1, but I am following the same order in which the MSB to LSB is given. So, depends upon the LSB bit, this bit will affect this bit will be affected by Q 1, this will be affected by Q 2 and so on that is why I am writing in the same order.

This is LSB, this is MSB and here one important thing is each T you have to make as 1. So, we know that for the T flip flop the truth table is T, Qn, Qn+1. If T is 0 whatever the previous state will be retained these becomes Qn only. If T is 1 if previous state is 0 present state will become 1, previous state is one present state is becomes 0, this is compliment of Qn.

So, what happens is, when T is equal to 1 the flip flop toggles at every clock signal. We have this clock in whether positive edge triggered or negative edge triggered here I have not given the bubbles so; that means, its positive edge triggered. So, if I take the count sequence of these if I assume that the initial state is Q 3 Q 2 Q 1 Q 0. 0 0 0 0 is the initial state before application of the clock signal, this is 0th clock signal.

How to establish these initial states? We have discussed using preset and clear signals so at first clock signal what happens? So, this external clock is applied, this is clk which is applied here whereas, the clock for this T 1 is Q 0, clock for T 2 is Q 1 and so on. These are the clocks of this you can call as clock 3, clock 2, clock 1 of course, this you can call as clock 0.

So, this is clock 0 here. So, because clock is 0, 0 to 1 transition. So, 0 to 1 transition if I take this clock signal, this is logic 0 to 1 transition is positive edge. See here I am assuming that these are all positive edges because I have not taken the bubble set positive edge what happens is, this will change Q 0, 0 to 1.

So, in order to change this Q 1, because Q 1 is always tied to 1, Q 1 changes when and only when Q 0 changes from 0 to 1 because positive edge 0 to 1, if it is negative edge 1 to 0. Let us assume that otherwise the negative edge for this like otherwise you have to change the circuitry from negative edge means 1 to 0, these are all I am calling as negative edges.

So, this is one complete clock. So, this Q 0 will changes always at 2nd clock signal this becomes 0, at 3rd clock signal this becomes 1 because this clock external clock that we are going to apply is this entire thing is called as clock 1, this entire thing is called as clock 2, this clock 2. So, over this entire thing the negative edge also will comes positive edge will also comes.

But because these are all negatives edge trigger at this particular point, this Q 0 will changes from 0 to 1, again at this particular point Q 0 changes from 1 to 0, again at this particular point Q 0 will change from 1 to 0 and so on whereas, the condition for the Q 1 2 change is Q naught also change from 1 to 0.

So, here from here to here Q 0 has changed from 0 to 1. So, it will remain in the same state. Similarly, condition for the Q 2 change is Q 1 has to change from 1 to 0 because it has not change this will remains in the same state. So, condition for the Q 3 to change is Q 2 has to change from 1 to 0, but Q 2 remains in 0. So, Q 3 will remains at 0 only.

So, after the first clock signal the next state is 0 0 0 1. So, at the second clock signal Q 0t always changes because external clock, if I apply on complete clock signal it will contains positive edge, negative edge, positive level, negative level. So, the output will change. Now, from 1 to 2. So, this Q0 is changing from 1 to 0. So, Q 1 also will change the condition is 1 to 0.

So, the condition is if Q 0 changes from 1 to 0, Q 1 changes. So, Q 1 changes. So, the condition for Q 2 to change is, Q 1 has to change from 1 to 0, but here its changing from 0 to

1. So, no change here Q 2 remains at 0 no change. Say again I set the 3rd clock signal these changes from 0 to 1, 0 to 1 transition means no change, 1 to 1 means no change, 0 to 0 means no change.

So, 4th clock signal this always will changes 1 to 0 means change 1 to 0. 1 to 0 means, there is a change. So, this 0 to 1 means no change at Q 3. 5, 0 this is 1 So, this is 0 to 1. So, 0 to 1 transition means no change, 0 to 0 means no change, 1 to 1 means no change similarly 6, 1 to 0 1 to 0 means change, 0 to 1; 0 to 1 means no change, 1 to 1 means no change.

7 this is 0 to 1. So, 0 to 1 means no change, 1 to 1 means no change, 1 to 1 means no change. 8 clock signal this is 0, 1 to 0 means change, 1 to 0 means change, 1 to 0 means change, 0 to 1 no change, 0 to 0 no change, 0 to 0 no change. So, this is 1, 1 to 0 change, 0 to 1 no change, 0 to 0 no change, 0 to 1 no change, 1 to 1 no change, 0 to 0 no change.

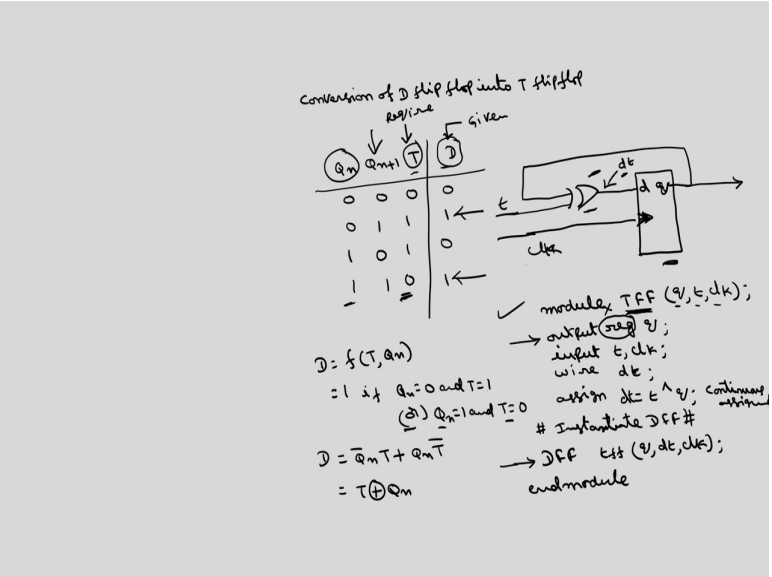
1 to 0 change, 1 to 0 change, 0 to 1 no change, 0 to 1 no change, 0 to 0 no change, 1 to 1 no change, 1 to 0 change, 0 to 1 no change, 1 to 1 no change, 0 to 1 no change, 1 to 1 no change, 1 to 1 no change. At 16th clock signal this is 1 to 0. So, 1 to 0 means change, 1 to 0 means change, 1 to 0 means change.

This is your initial state. So, 16 clock signals are required to get the same initial state. So, the modulus is 16. So, this is the 4-bit ripple counter. So, the name ripple counter is because this bit ripples through this Q 0 change will reflect in Q 1, Q 1 change will reflect in Q 2, Q 2 change will reflect in Q 3. So, this change sees in the form of ripple. So, this is name ripple counter also this function is called as asynchronous counter.

So, what is the name asynchronous is, the clock is different for the different flip flops. So, on the other hand we have the synchronous counters also where the clock is common to all the flip flops, but the complexity of this synchronous counter is more than the complexity of the asynchronous or ripple counters.

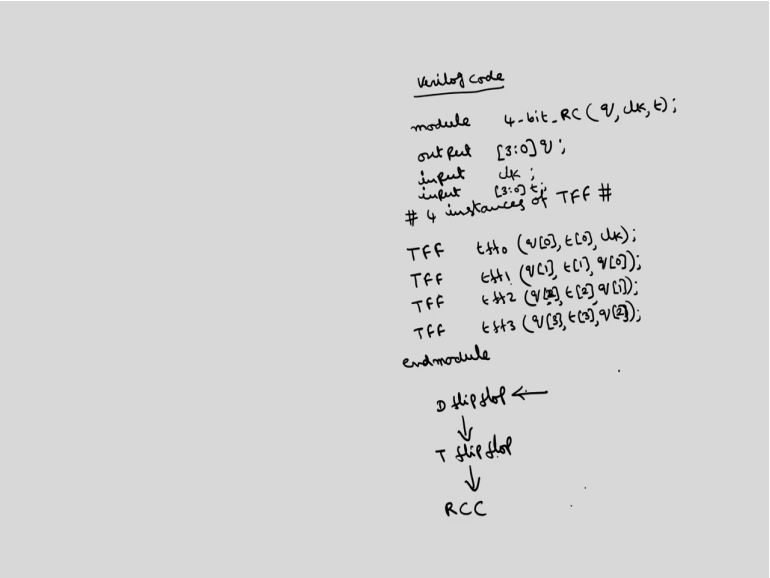
So, normally in applications where complexity is important means area and power we prefer ripple or asynchronous counters the applications where speed is important we will prefer synchronous counters. Now this is more 4-bit ripple counter what is the Verilog code corresponding to this one?

(Refer Slide Time: 15:54)



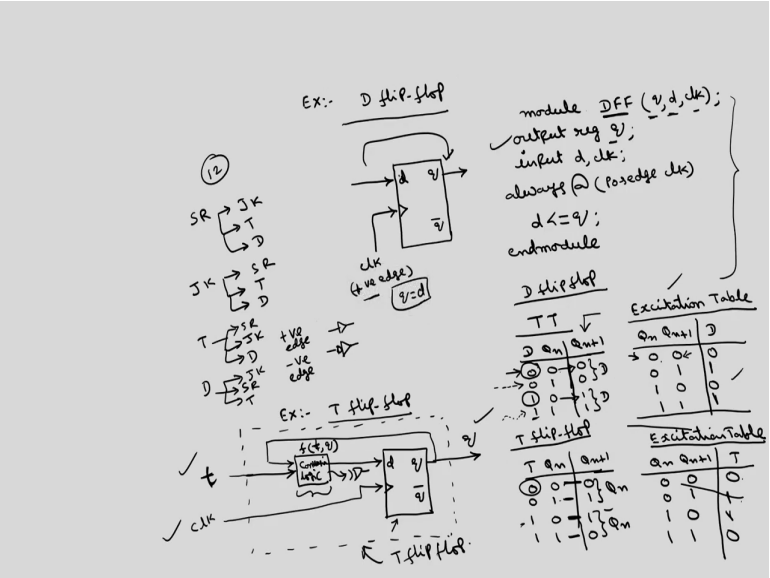
So, we have written here for this T flip flop you have instantiate this 4 times. So, this T flip flop name given is TFF. So, you have to call TFF, output is Q input is T and another input is clock.

(Refer Slide Time: 16:13)



Overall, the module name is 4-bit ripple counter. So, what are the overall inputs and outputs of this ripple counter? Clock is the input, outputs are q 0, q is output, later we will define as q 0 q 1 q 2 q 3. q and clock, output [3:0] q.

(Refer Slide Time: 17:10)



So, this register no need to declare because here also we are not going to use this register because we have defined this registor in the main module, the flip flop module. So, that will take care about the remaining modules because we are going to instantiate this. Input clock. This T we are going to derive from the d. So, instantiate 4 instances of T flip flop.

So, the name that we have given for T flip flop is TFF. I will give some another sub name as TFF 0, we have to define this as output, input and clock. So, inputs we can also define this input here t, we can write input [3:0] t. So, t first one is. So, this you have to use the first output, then the input, then the clock.

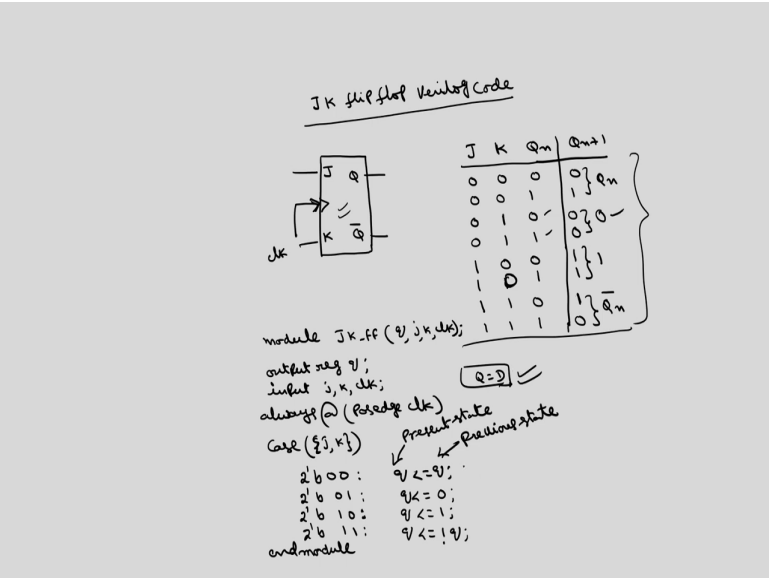
So, for T flip flop 0, output is clock is q 0 input is t 0, clock is external clock. So, for this output is Q 0, input is T 0 and clock is external clock. It has for the next one, input is T 1, output is Q 1, clock is Q 0. T flip flop, TFF 1 output will be q 1, input will be t 1, but in place of clock we have q 0.

Similarly, third flip flop TFF2 output is q 2, input is t 2 the clock for the second flip flop is q 1. T flip flop TFF 3 output is q 3, input is t 3, clock is q 2, end module. This is how you can

construct the higher-level modules from the lower level modules by instantiating the lower level modules. We have considered about D flip flop, from that we have constructed T flip flop, from that we have constructed a ripple carry counter.

You can simply call ripple counter also ripple carry counter also the main program of D flip flop the operation is very simple. So, we can write the code for the D flip flop just simply assigning d is equal to q, but this is not the case of J K flip flop. J K flip flop the output we cannot define like just simply q or q 0 or q 0 bar. So, it has to follow a truth table.

(Refer Slide Time: 21:12)



So, how to write the Verilog code for the J K flip flop? So, we know that J K flip flop if I take only this block because we are not interested in the internals circuit diagram in case of behavioral modelling. This is J, K, Q, Q bar then we have clock. So, what is the truth table of this? J, K, Qn, Qn+1.

If it is 0 0 whatever the previous state will be, the present state when this is 0 0 1 1 only previous state is present state. If J is 0, K is 1 regardless of the previous state output remains in 0, this is independent of this state. Conversely if J is 1, K is 0 regardless of the previous state this remains in 1. For J is 1, K is 1 the present state will complement of the previous state, this is Qn bar.

This is for the master slave J K flip flop you have already derived this stable and we have discussed the internal circuitry of this J K flip flop. So, now, how to write the Verilog code? Module J K flip flop, inputs are J, K, output is q. I will use small letters this is output, inputs are j, k clock output. So, also we have to declared as a register. So, q because this is target output input j, k clock.

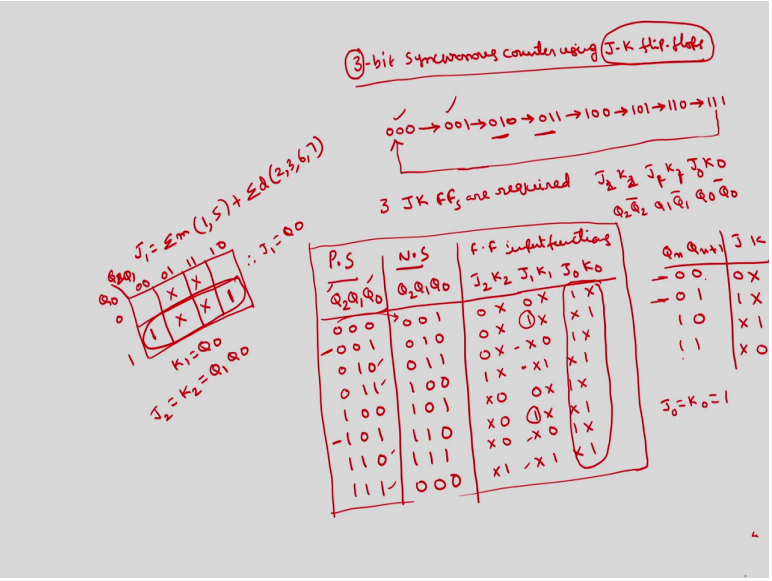
Now, here we cannot write the expression for the output in case of D flip flop like Q is equal to D whereas, in J K flip flop there is no such expression. So, you have to describe this truth table in the Verilog code for that we will use case statement always @ this because I have not given the bubble, you can call as positive edge posedge of the clock case who is going to decide? J , K is you have to put in flower bracket.

Case 2'b0 0, J is 0 K is 0 what will be the output? You have to use the non-blocking statement in case of sequential circuits, q itself. So, this is present state, this is previous state. 2'b 0 1, q is equal to 0 because q 0 here. 2'b 1 0, q is equal to 1, 2'b 1 1 q is equal to q bar, this is complementing operation, end module.

This is about the J K flip flop. So, in case if it is it cannot be expressed as a Boolean expression for the output in terms of inputs, then we have to describe the truth table. Now using this J K flip flop if want to design any circuits you can instantiate these J K flip flops. Using now this J K flip flop Verilog code we can construct a 4-bit or 3-bit synchronous counter.

So, we can instantiate this J K flip flop to design synchronous counter. So, till now I have discussed about the ripple counter. So, ripple counter will act as asynchronous counter. So, which will be having different clocks for the different flip flops. So, in that way the asynchronous counter is slow, the speed of the asynchronous counter is less. To improve the speed, we have to use the synchronous counter where the clock will be common to all the flip flops.

(Refer Slide Time: 27:10)



Now, we will discuss a 3-bit synchronous counter using J K flip flops. So, this design of synchronous counter is not as a straightforward as asynchronous counters, here we have to use excitation table to design the synchronous counter. So, 3-bit synchronous counter as the name implies we have total eight states 0 0 0, 0 0 1, 0 1 0, 0 1 1, 1 0 0, 1 0 1, 1 1 0, 1 1 1, after that it will come to the initial state.

The modulus of these counter is eight, after eight clock pulses we will get the initial state. So, in order to design these, we have to choose the type of flip flop, we can design the synchronous counter using any flip flops here I am using J K flip flops because I have written the code for the J K flip flops so, that I can instantiate this J K flip flop code. So, I am using J K flip flops the number of bits is equal to the number of J K flip flops we require 3 J K flip flops.

You call the names as J 3 K 3, J 2 K 2, J 1 K 1, J 0 K 0 and the corresponding outputs are Q 2, Q 2 bar, Q 1, Q 1 bar Q 0, Q 0 bar. So, in the design procedure is we have to start with the present state, next state, then flip flop input functions. So, the states are Q 2, Q 1, Q 0 what is the next state Q 2, Q 1, Q 0? To get this what are the inputs required? J 2 K 2, J 1 K 1, J0 K 0.

So, this design procedure is exactly similar to the conversion of one flip flop into another flip flop procedure. So, for 0 0 0 present state, after clock signal 0 0 1 is required. If the present

state is 0 0 1, itself the next state is 0 1 0. So, like that for 0 1 0, 0 1 1; 0 1 1, 1 0 0; 1 0 0, 1 0 1; 1 0 1, 1 1 0; 1 1 0, 1 1 1; for 1 1 1, 0 0 0.

Now, how to find out the flip flop input functions? For that we have to use the J K flip flop excitation table which we have already discussed in the earlier lectures. So, Qn, Qn+1, J, K what is excitation table 0 0, 0 x, 0 1, 1 x, 1 0, x 1, 1 1, x 0. So, here Q 2 is changing from 0 to 0. So, for 0 to 0, inputs required are 0 x, Q 1 is also changing from 0 to 0, 0 x, Q 0 is changing from 0 to 1 for 0 to 1 1 x.

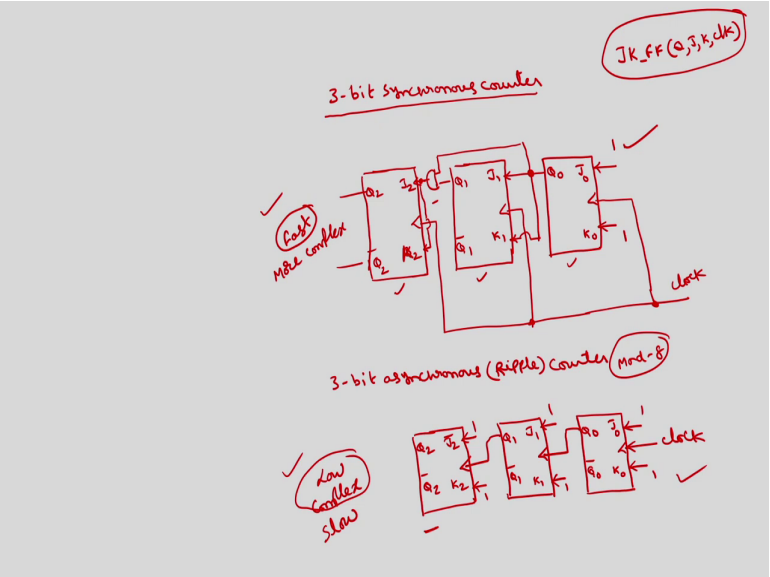
So, like that in the next row, 0 to 0 means 0 x, 0 to 1 means 1 x, 1 to 0 means x 1. So, next row 0 to 0, 0 x, 1 to 1, x 0, 0 to 1, 1 x. So, 0 to 1, 1 x, 1 to 0, 1 to 0. 1 to 0s are x 1, x 1. 1 to 1 is 0 x 0, 0 to 0 is 0 x, 0 to 1 is 1 x. 1 to 1 is x 0, 0 to 1 is 1 x, 1 to 0 is x 1. 1 to 1 is x 0, 1 to 1 is x 0, 0 to 1 is 1 x.

Similarly, last one is all 1 to 0s, 1 to 0 is x 1 x 1 x 1 now what are the Boolean expressions for J 0 K 0 J 1 K 1 J 2 K 2 we can see that in J 0 K 0 columns. So, all the values are either 1s or do not cares. So, if all the values are 1s are do not cares so do not care can also be taken as 1. So, the values of J0 is equal to K 0 is equal to permanently 1.

And if you solve for this J 1 using K map, J 1 is equal to sigma m what are the min terms? So, this is one min term corresponding min term is 1, this is another min term this is 5, sigma d what are the do not cares? This is 2, 3 then 6, 7; 2 3 6 7 this is Q 2, MSB, Q0, LSB.

So, Q 2 Q 1 Q 0, 0 0 0 1 1 1 1 0 0 1. So, min terms are 1 0 0 1 0 0 1 is this min term, 5 is 1 0 1 is this min term, do not cares are 0 1 0 2 0 1 1 3 6 is 1 1 0 1 1 1. So, this is 1, 4 box combination therefore, the value of J 1 is equal to simply Q 0 similarly if you solve for K 1 also we will get Q 0 and if you solve for J 2, we will get J 2 is equal to K 2 is equal to Q 1 Q 0 you can do in a similar manner.

(Refer Slide Time: 34:11)



So, what is the complete design of 3-bit synchronous counter? You have 3 flip flops, this is J 0 K 0 Q 0, J 0 is equal to K 0 is equal to 1 we have just derived this will be having clock. So, one important thing is in case of synchronous counter, the clock is common to all the flip flops, that is why the speed of synchronous counter is more when compared with an asynchronous counter and J 1 K 1 is nothing but Q 0 the same Q 0 is applied here also and J 2 K 2 is Q 1 Q 1 bar is Q 1 Q 0.

So, this Q 1 is this, Q 0 is this you have to connect through the AND gate. This output you have to connect to J 2 as well as K 2, this is the complete circuit diagram of a 3-bit synchronous counter. So, we know that using T flip flop we have derived this 3 bit asynchronous or ripple counter, if I use a J K flip flop only to have the better comparison.

So, T flip flop is nothing but J and K will be connected. So, all inputs will be permanently connected to 1 this is also 1, J 1 K 1 is also 1, J 2 K 2 is also 1. If you take the T, simply T 0 T 1 T 2, 3 inputs are 1s. The only difference is here the external clock is applied, Q 0 will act as a clock for the second flip-flop Q 1 will acts as a clock for the next flip flop.

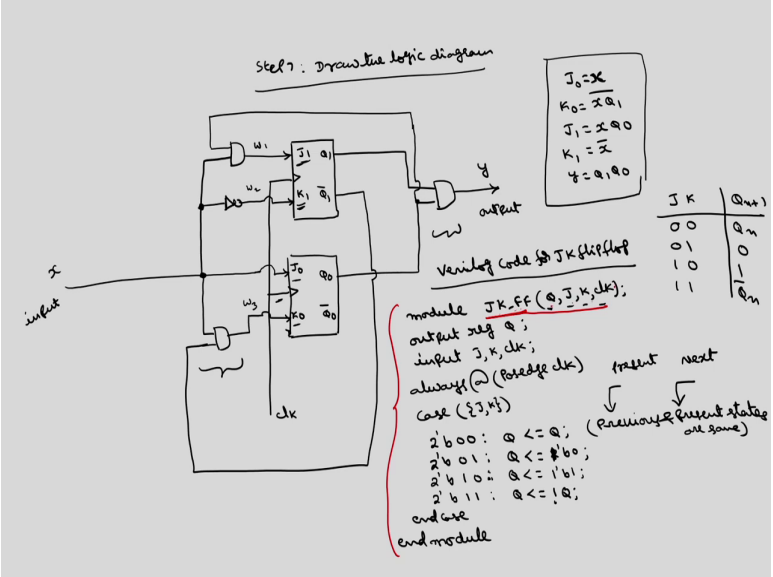
So, this is 3-bit synchronous counter, this is 3-bit asynchronous counter, if we take the relative merits and demerits of this one. So, the complexity of this system is less low complex

because we require only 3 flip flops only no extra additional gate is required here we require additional gate of this.

But this is slow you know this is fast, but more complex then asynchronous. So, in applications where speed is important we will go for the synchronous counters, in applications where complexity is important we will go for asynchronous. This asynchronous counter is mod 8 counter 3 bit or mod 8 if want mod 16 counter you had one more flip flop here, you connect the Q 2 to the clock of that one.

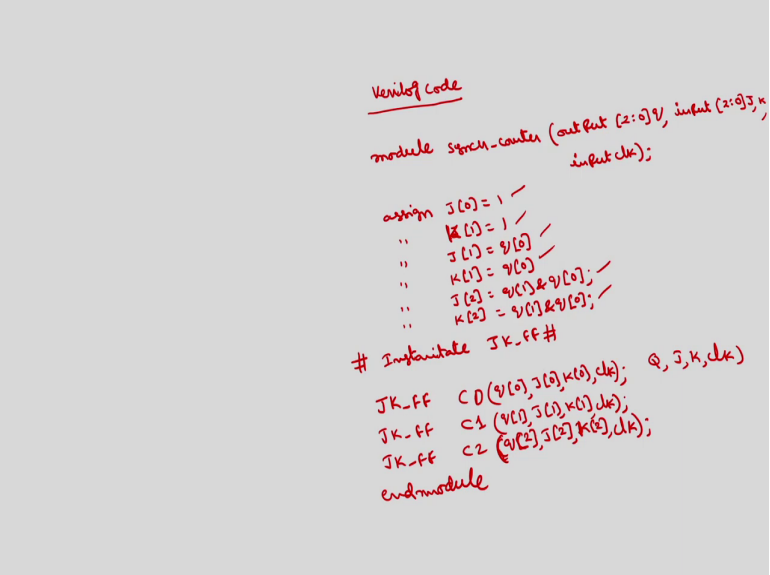
But here we require a design procedure you have to follow the excitation table and then you have to design. So, it's about 3-bit synchronous asynchronous counter now I can write the Verilog code correspond to this 3-bit asynchronous counter by instantiating the code of J K flip flop thrice. So, you have written the J K flip flop with the module you have written as J K flip flop.

(Refer Slide Time: 39:06)



So, J K underscore flip flop, Q K this was the code for the J K flip flop. So, to instantiate this, we have to write J K underscore flip flop with Q, J, K, clock. We saw the name you have used for the J K flip flop. Now, by instantiating this, we have to write the code correspond to 3-bit synchronous counter.

(Refer Slide Time: 39:41)



So, module synchronous counter. So, the outputs will be I will write here itself ,3 outputs [2:0] q, I will call as lowercase letter, input [2:0] j, k; another input is clock. This is initialization. So, J 0 K 0 should be 1. So, I will assign J 0 is equal to 1, K 0 equal to 1, J 1 is you have derived as q 0, K 1 is also q 0, J 2 we have got as q 1 AND with q 0, K 2 also same thing.

Then you have to instantiate. So, J K the name I have given is FF I will give as circuit 1 C 1. So, the format is we have Q, J, K then clock. So, for the first flip flop we can call this J K flip flop, we can give as C 0 if we call as. What is q is q 0 and the inputs are J 0 K 0, but what is J 0, K 0? 1 1 then clock. J K flip flop C 1, for C 1 output is q 1, J 1 K 1, but what is this J 1 K 1? q 0, clock.

The third flip flop C 2, q 2, q 2 is the output, J 2 K 2 are the inputs, but what is J 2 K 2? q 1 q 0, clock. So, see how we can obtain a synchronous counter by instantiating the Verilog code of J K flip flops. So, this is for the counters, counter is actually a simple synchronous circuit.

So, how to design a general synchronous circuit which contains in addition to the inputs and the states, outputs qs, but also it will be having external outputs also. So, that design procedure and Verilog code correspond to that one we will discuss in the next lecture. Thank you.