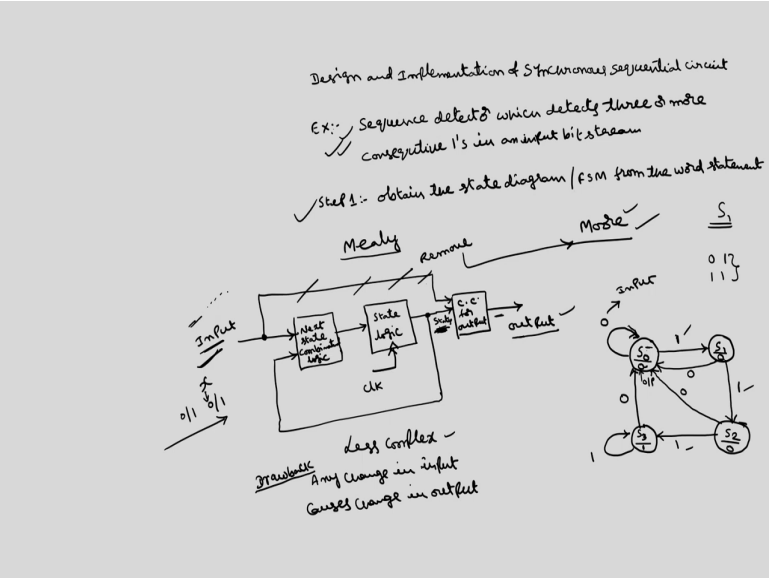
System Design Through VERILOG
Prof. Shaik Rafi Ahmed
Department of Electrical and Electronics Engineering
Indian Institute of Technology, Guwahati

Behavioral modelling of sequential circuits
Lecture - 19
Verilog modelling of sequence detector

In the last lecture, we have discussed about the Verilog code for the D flip flop and then we derive the T flip flop from the D flip flop and then we realized a four bit a synchronous or ripple counter. So, we will take one example of synchronous sequential circuit also, because a synchronous counter we have already discussed in the last lecture and we will use JK flip flop to design this synchronous sequential circuits. And we will see how this JK flip flop can be instantiated to design a synchronous sequential circuit.

(Refer Slide Time: 01:16)



So, I will discuss about design and implementation of synchronous sequential circuit. So, here I will take a synchronous sequential circuit as sequence detector which detects three or more consecutive 1's in an input bit stream, so basically this circuit will have one input where the input bit stream is coming.

So, whenever the circuit detects three or more consecutive 1's, there will be output which will becomes 1. So, how to design a synchronous sequential circuit? So, you might have studied

in your digital logic course here, I will give you briefly the design procedure of synchronous sequential circuits. So, in the design of synchronous sequential circuit the first step is obtain the state diagram. You can also call as FSM; Finite State Machine, from the word statement.

So, you have the word statement is given, you have to obtain the state diagram. So, again in synchronous sequential circuits there are 2 models; 1 is called a Mealy model, another is called Moore model. We have basically, here the state logic. This basically consist of flip flops with some clock. To obtain the next state of this we require some combinational circuit which is called as next state combinational logic.

So, these basically consists of input and also this is function of the previous state. We know that in case of synchronous or any sequential circuit the output is not only function of inputs but also on the previous outputs. This will select the next state. Then the combinational circuit for C, C stands for combinational circuit for output you will get output. So, in case of Mealy model so the output not only a function of the states this will give the states here we have states.

But also, this is function of inputs. This is called Mealy model. Whereas, in case of Moore model, the same thing but this particular line will not be there. If we remove this line will get this Moore model. So, the only difference between the mealy and Moore model is so in case of mealy model output depends upon the states as well as the input whereas, in case of Moore model the output depends only on the states.

So, if you see the relative advantages and disadvantages of this the complexity of the mealy model is normally this is less complex. But, if you take the drawback, if any fluctuation occurs in the input that will change the output also, that is undesirable 1. This is the main drawback of this. Any change in input causes change in output.

That is the reason why in most of the practical applications normally we will use Moore model where the output only function of states is independent of the inputs. So, we will discuss here the state diagram for Moore model.

So, Moore model the output depends only on the states without regard to the inputs. So, the state diagram or FSM for this example which detects three or more consecutive 1's. So, I will

start with a reset state which is called as S 0. How to establish this S 0 state? By using preset and clear signals of the flip flops.

So, the very first bit that we are going to receive from this input, if assume that this input is x, so the first bit can be 0 or 1. Similarly, second bit can be 0 or 1, like that a series of bits are coming. If the first bit received is 0, where does this state will transit where does this state goes and when input is 1, where does this state moves. So, we know that we have to detect three consecutive 1's if a 0 is detected then it will state in S 0 state only.

And the output will be given just below this output will be 0. If the input is 1, it will go to the next state S1, this label represent input whereas, this represent output. So, in any case the output of this S 0 state is 0, because if input is 0 it states here itself, if input is 1 because the first bit, so it as not yet detected three or more consecutive 1's, so the output becomes 0.

So, coming for this S 1, S 1 can receive the second bit as 0 or 1. If second bit is 0, no way to detect the three consecutive 1's, so it will simply go to the S 0 state. And if 1 is received at the second stage it will go to the next stage, because it can detect the three consecutive 1's. Already two 1's has reached. This is 1, but in any case, what is the output 0 only because if this first bit is in at S 1 first bit is 1 at S 1, second bit can be 0 or 1 means, possibility is 0 1 or 1 1.

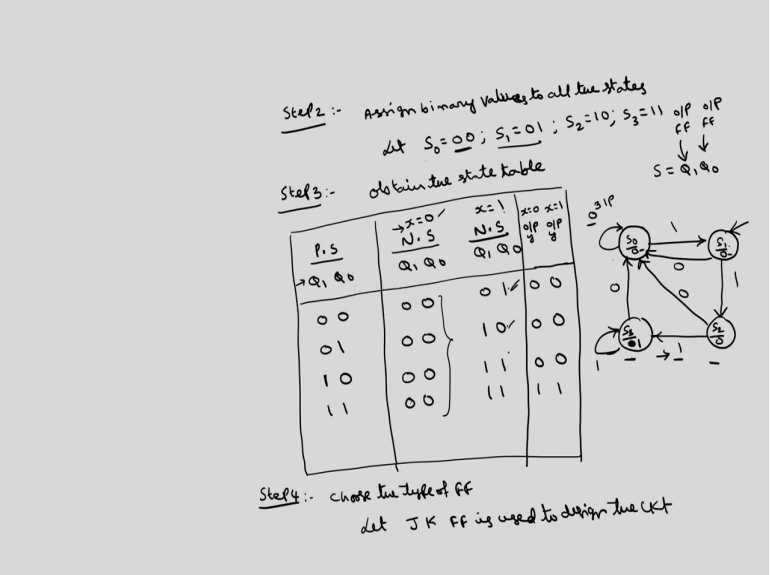
So, in any of the cases it has not detected three consecutive 1's, so the output is 0. Now, coming for this S 2 state, say if the received bit is 0, although two 1's received it simply goes to S 0. If a 1 is received it will go to the S 3 state. But in any case, so this has received only two 1's, so there it can be 0 1 1. So this one will be 0, because it has not yet detected third bit.

So, where the third bit will be detected? Here it is 3. So, already three 1's has reached 1 2 3. So, this will be 1, even if the fourth bit is also 1, it will stay in S 3 only, so the output remains 1. So, if fourth bit is 0, again it will go to the S 0 state. This is the state diagram of Moore model. Here, this output depends upon the state directly, so the state is going to decide the output.

So, in the denominator whatever this value is there that will represent the state. So, if it is Mealy model we require only three states. So, the complexity is less, but the drawback is any

change in the input causes the change in the output. So, you see the first step in the design of any synchronous sequential circuit.

(Refer Slide Time: 12:05)



So, the second step is assigning binary values to all the states. So, here you are assuming, let S 0 is 0 0, S 1 is 0 1, S 2 is 1 0, S 3 is 1 1. Then the next step is obtaining the state table. So, totally four state means we require 2 flip flops. So, we will assume that this each S state is of the form of Q 1 Q 0, because we require for each state we require 1 flip flop. This is output of one flip flop, this is the output of another flip flop.

So, the states I am going to represent as in general instead of writing 0 0 0 1, 1 0 1 1, I will write Q 1 Q 0. So, where Q 1 is MSB, Q 0 is LSB. So, I will call as present state P S stand for present state, Q 1 Q 0, next state not only depends upon the present state, it also depends upon the input also.

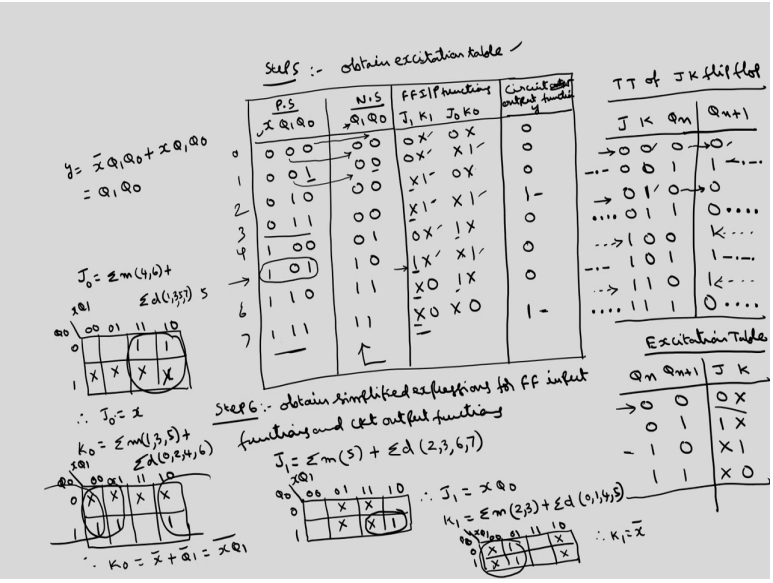
So, next state when x is equal to 0, next state when x is equal to 1. This you can easily obtain from the state diagram. So, I will just redraw here state diagram. The state diagram that we have drawn in the last slide is this one; S 0 / 0, S 1 / 0, S 2 / 0, S 3 / 0. So, 0 it will stay here only, 1 it will go to here, 0 it will go to the S 0, 1 it will come to here, 0 it will go to S 0, 1 it will go to S 3, 0 it will go to S 0, 1 it will stay here itself.

So, how to map this state diagram or FSM onto state table? So, this is your input and we can also write the output. Output, I am assuming the output as y, output for x is equal to 0, x is equal to 1. If present state is 0 0 what is the next state? If x is equal to 0 you see here, next state is 0 0 itself. x is equal to 1 next state is S 1, S 1 is 0 1. So in both the cases outputs are 0 0, here also 0, here also 0.

0 1 means S 1 state, when x is 0 it will go to S 0 means 0 0, when S is equal to 1 it will go to S 2 which is 1 0 in both the cases output is 0 0. 1 0 that is S 2 state, when S is equal to 0 it will go to 0 0 otherwise when input is equal to 1 it will go to 1 1, in both the cases this is 1, this is 0 0. 1 1 when x is 0 it will go to S 0, S is 1 it will remain in the 1 1 only in both the cases 1 1 is 1, because already detected 3.

So, after this stage, 3 detected, so output is 1, even 4 also the output is 1. It is because this is a sequence detector which detects three or more consecutive 1's. Say if we call this output as y, so this is a state table. Now, the fourth step is choosing the type of flip flop. We can design this synchronous sequential circuits using any flip flops. So, I will take a JK flip flop to design the circuit, but you can design using D flip flop also, T also, SR also.

(Refer Slide Time: 18:00)



So, the next step is the important step which is obtaining the excitation table. So, I will take the same state table, I will take this x to this side, so that this becomes simplified. So, I will

take x, Q 1, Q 0, present state, next state because x also I have taken here only one next state will be there, Q 1, Q 0. Same whatever the data previously it is there I will write here. When 0 0 when x is 0 the next state is always 0 you see here.

When x is equal to 0, so all the four states are 0 0 only. When x is equal to 1, the next states are 0 1, 1 0, 1 1. So this is 0 0 so in all the cases the next state is 0 0 whenever x is equal to 0. When x is equal to 1 you can see that this is 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 1.

Now, we have to write the flip flop input functions, because you have chosen JK flip flops, here you have to write the expression for flip flop input functions. Corresponding to Q 1 we have the inputs J 1 K 1, corresponding to Q 0 we have J 0 K 0. Then of course, we have the circuit output function which is y. y also we have already derived this one, y is for x equal to 0, 0 0 0 1 again 0 0 0 1.

This is 0 0 0 1, then 0 0 0 1. Now how to find out these flip flop input functions? For that we require the excitation table of JK flip flop. We know the truth table of JK flip flop is J K, Qn, Qn+1. 0 0 regardless of the whatever the previous state will be the next state 0 1, this is 1. This is simply the previous state and 0 0, 0 1 regardless of the previous state output is 0.

1 0 regardless of the previous state, output is 1. For 1 1 whatever the complement of the previous state will be present state. This is the truth table which we have discussed in the earlier classes. Now what is excitation table of JK flipflop? We have derived the excitation table for D and T flip flop, in a similar way what is excitation table of JK flip flop? So, excitation table represents the inputs required for a given change of state.

So, 0 to 0, we can see here that 0 to 0 is changing here as well as here. So, what are the inputs? This is 0 0 this is 0 1. In any case J is 0 K is 0 over here 1 here. So, it will becomes do not care. This is 0, do not care. Or 0 to 1 also there are two transitions, 0 to 1 is this one, 0 to 1 is this one. So, what are the inputs 1 0 1 1, so 1 is J, K is do not care. Similarly, for 1 0 there are 2 transitions 1 0 is this one.

Another is 1 0 is this one. So, what are the inputs corresponding to this 1 0, this 1 0, this is 0 1 and 1 1 means x 1. Similarly, 1 1 is remaining, 1 1 also this is 1, this is 1, so what are the inputs this is 0 0 1 0, so x 0. This is the excitation table. So, this is required here, because the

next step is obtaining the excitation table. Now, we can see that this Q 1 is changing from 0 to 0.

So, for 0 to 0 what are the inputs required 0 x. So, this is 0 x, Q 0 is also changing from 0 to 0. So, J 0 K 0 is also 0 x. Similarly, here, you forget about the x here. So, 0 to 0 means 0x, 1 to 0 1 to 0 1 to 0 is x 1. So, like that if we fill this one, 1 to 0 is x 1, 0 to 0 is 0 x, 1 to 0 is x 1, 1 to 0 is x 1, 0 to 0 0 x, 0 to 1 is 1 x, 0 to 1 is 1 x, 1 to 0 is x 1, 1 to 1 is x 0, 0 to 1 is 1 x, 1 to 1, 1 to 1 x, 0 x 0. This is the important step obtaining the excitation tables.

Now, the next step is obtain the simplified expressions for flip flop input functions and circuit output functions, that is the expressions for J 1 K 1 J 0 K 0 and y in terms of the states.

Now, you forget about this column. So, you have to express this as a function of the present state and x. So, what is the expression for J 1? J 1 is equal to, so wherever 1 is there, so J 1 is 1 here, this is 1 here, so sigma m. So, what is this 1 corresponding to? So, this 1 is corresponding to this row. So, what is the mean term this is 1 0 1 is 5 this is min term 0 1 2 0 1 2 3 4 5 6 7, these are the min terms.

So, this is 1 is min term correspond to this min term is 5 plus there are some do not cares also. This is do not care which is 6, this is do not care 7, this is do not care 3, this is do not care 2, sigma d do not cares are this corresponding to 2 3 6 7. Say, if we simplify this using K map, we have taken the same order x as MSB.

So, x, Q 1, I will take in the top, you can take in the bottom also I will follow this notation 0 0 0 1, 1 1 1 0, 0 1, 5 is nothing, but 1 0 1 is this. 2 is 0 1 0 this do not care. 3 is 0 1 1 do not care, 6 is 1 1 0 do not care, 7 is 1 1 1 do not care. So, the only possibility is we have only two box combination therefore, expression for J 1 is from here to here x and here Q 0.

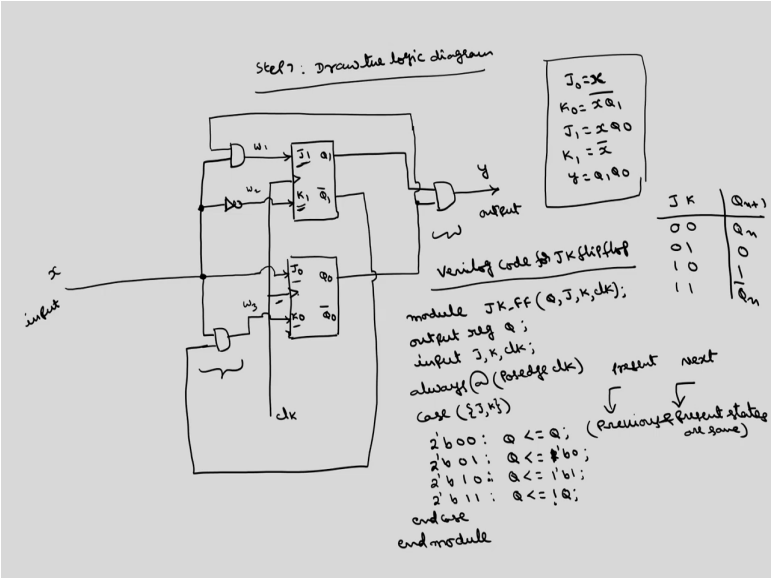
Similarly, you can write down the expressions for J 2 K 2 J 0 K 0. So, what about K 1 sigma m? What are the min terms for K 1? This is 1 mean term 2, 3, do not cares, 0 1 4 5. So, what is the expression? x Q 1 Q 2 Q 0, 0 1 0 0, 0 1 1 1, 1 0, 2 3, is 1 0 0 is this is 1 min term.

3 is 0 1 1, do not cares are 0 1 0 1 4 5 1 0 0. So, K 1 is simply these four-box combinations, this is x bar. Similarly, if you write for J 0 sigma m that is J 0 min term is 4, 6 plus sigma d 1, 3, 5, 7. So, if you simplify by using K map x, Q 1, Q 0, 0 0 0 1, 1 1 1 0 0, 1 4 is this 6 is this.

1 3 5 7, 1 is this, 3 is this, 5 is 1 0 1, sorry this 6 is this. This is do not care, 5 and 7 is this. So, again we have four box combination therefore, J 0 is simply x K 0 sigma m. This is 1 3 5 plus sigma d 0, 2, 4, 6; x, Q 1, Q 0, 0 0, 0 1, 1 1, 1 0, 0 1, 1 3 5.

1 is 0 0 1, 3 is 0 1 1, 5 is 1 0 1, 0 is do not care, 2 means 0 1 0 is do not care, 4 is 1 0 0, 6 is 1 1 0. We have this one four box combination, this is another four-box combination. Therefore, K 0 is equal to, so this four-box combination is x bar plus the other four box combination is Q 1 bar. This is equal to x, Q 1 whole bar. So, you see the sixth step and the last step is drawing the logic diagram.

(Refer Slide Time: 32:22)



So, we have basically two flip flops. This is J 1 flip flop, K 1 flip flop, you have Q 1, Q 1 bar outputs and a clock. This is J 0 flip flop, K 0 flip flop, Q 0, Q 0 bar. This clock will be common in case of synchronous circuits. This can be either positive edge triggered or negative edge triggered.

And we have one input and an output and the expression for the output is y is equal to simply there are two min terms this is correspond to x bar, Q 1, Q 0, I am not using K map because only two mean terms are there plus x Q 1 Q 0 simply this is Q 1 Q 0. So, the output is simply Q 1 Q 0 and say I have told in case of Moore model the output depends only on the states it will not depend upon the inputs.

So, now we have input x. So, what are the expression that you have got? J 0 is x, K 0 is x Q 1 bar. J 1 is x Q0, K 1 is simply x bar and y is Q 1 Q 0. These are the expression that we have obtained in the last step. So, J 0 is simply x. So, there is a x here, which will be directly applied to the J 0. This is x input this is output then K 0 is x Q 1 bar.

x Q 1 bar, x Q 1 is here we are taking this Q 1. x we are taking here this is K 0 and J 1 is x Q 0, x into Q 0. Q0 is here, into x. This x is here and K 1 is x bar. This is x line you have to pass through the inverter this is the complete design of a sequence detector which detects three or more consecutive 1's.

So, here these three gates will form the next state logic and this will form the next output combinational logic. Now, here because we are going to use 2 JK flip flops, so in order to write the Verilog code. I will first write the Verilog code for the JK flip flop then I will instantiate this JK flip flops twice. So, Verilog code for JK flip flop. Module JK_FF for JK flip flop we have the clock input, JK inputs, output is Q, inputs are J, K, clock.

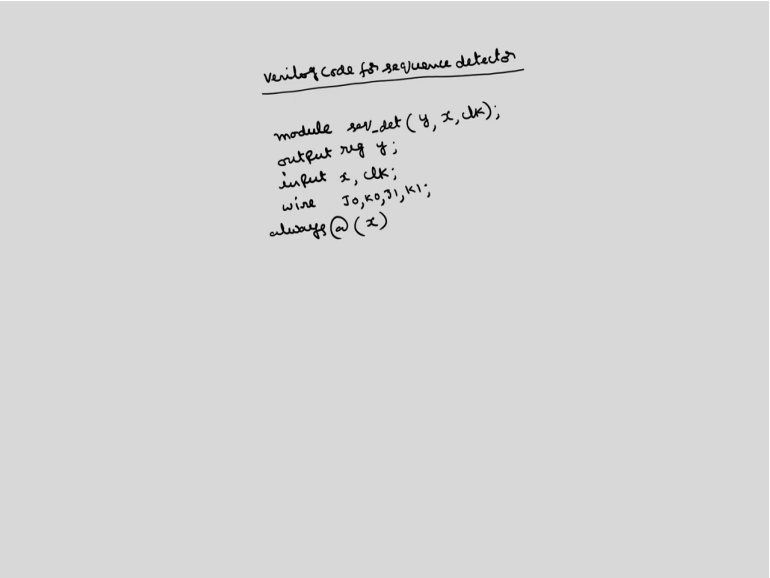
Output we have to define as a register also, Q, input J, K, clock. Always @ , I have not given the bubbles. So, this is positive edge. Posedge clock now unlike this d flip flop where output q=d whereas, here in case of JK flip flop we cannot write the output in terms of JK, output cannot be expressed as a function of JK. So, we know the truth table that we have to implement here.

So, the truth table is J, K, Qn, Qn+1, 0 0 are if I rewrite this simplified truth table this is simply Qn, 0 1 0 1 0 1 1 1, Q n compliment. So, here we have to take a case statement. Case J,K {2'b00} if jk is 0 0 then the output Q will be equal to, we have to use non blocking statement in sequential circuits, simply Q itself; that means, previous and present states are same. Actually, here I will use the two notations one is previous and present state if I call this

itself as a present, this will be next state. You can follow any notation either present next or previous present then 2'b 0 1, Q will be simply 0. I will write in binary form, Q is 1 bit only 0. 2'b 1 0, Q is 1'b 1, 2' b 1 1 Q will be Q compliment.

end case; end module. So, this is the Verilog code for the JK flip flop. Now to write the Verilog code for the sequence detector we can instantiate this JK flip flop code twice.

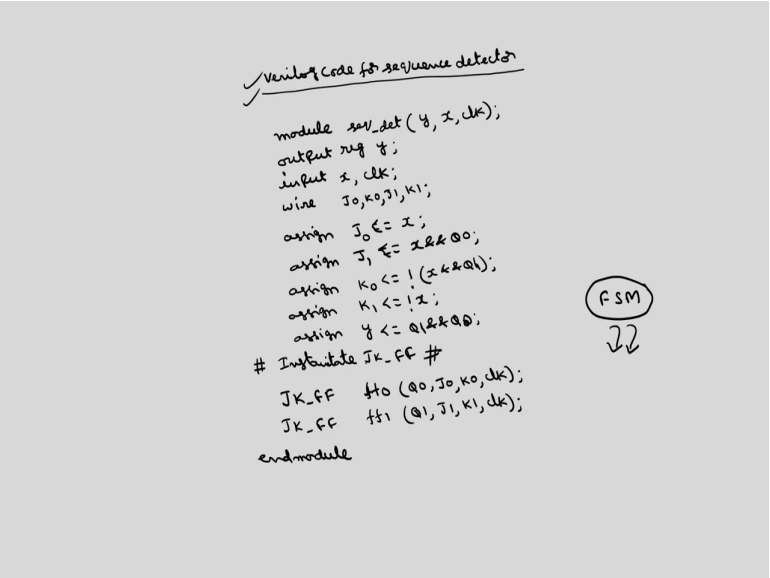
(Refer Slide Time: 40:28)



So, using this relations module sequence_detector. So, for sequence detector overall, we will be having one output y, one input x, another is clock; output you can call as register y, input x , clock.

But we require two wires also, because these are not accessible, four wires in fact. This is J 1 is 1 wire, K 1 is 1 wire, if I call the same the names J 0, K 0. Another you can call as w 1, this is of course x only, this is w 3 or otherwise you can take directly J 1, K 1, J0, K0 itself wire J 0, K 0, J 1, K 1 then always @ (x) because this clock we have embedded in JK flip flop Verilog code or you can directly write the assign here.

(Refer Slide Time: 42:36)



So, next one is assign what is J 0 is equal to x, J 0 is equal to because we will use this as x, assign J 1 is equal to within blocking statement x & Q 0; assign K 0 is equal to x Q 1 or AND gate assign K 1 is equal to x bar. Assign output y, Q 1&Q 0 then we have to instantiate. So, the name that I have given is for JK flip flop, JK_F F, I will call as flip flop 0.

So, the general structure is Q, J, K, clock you have to mention, clock is common Q becomes for 0 flip flop Q 0. J 0 K 0 clock is same for the second JK flip flop if I call this one as FF1 Q 1 is the output, J 1 is the input, K 1 is another input, clock. In synchronous clock is same. This is end module. So, this is how we can write the Verilog code for the sequence detector this is complete design and implementation we have designed first and then we have implemented by using Verilog code.

But this is actually a difficult process for complex systems it becomes very difficult to design using this JK flip flops or any other flip flops following the step by step procedure that I have described. So, there is a possibility to model directly the state diagram or FSM, in most of the complex systems we will draw only the FSM. So, knowing the FSM, how to model using this FSM or state diagrams that we will discuss in the next lecture.

Thank you.