Digital Electronic Circuits
Prof. Goutam Saha
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 47
Moore Model and Mealy Model: Realization of Digital Logic Circuit

Hello everybody. In the last class, we had seen how Moore model and Mealy model are defined and what are the relative advantage, disadvantage. Then, we picked up one specific problem statement. And we try to solve that problem using both Mealy model and Moore model. And in the process, we understood their relative strength and weaknesses.

(Refer Slide Time: 00:42)

CONCEPTS COVERED

Concepts Covered:

❑ State Assignment

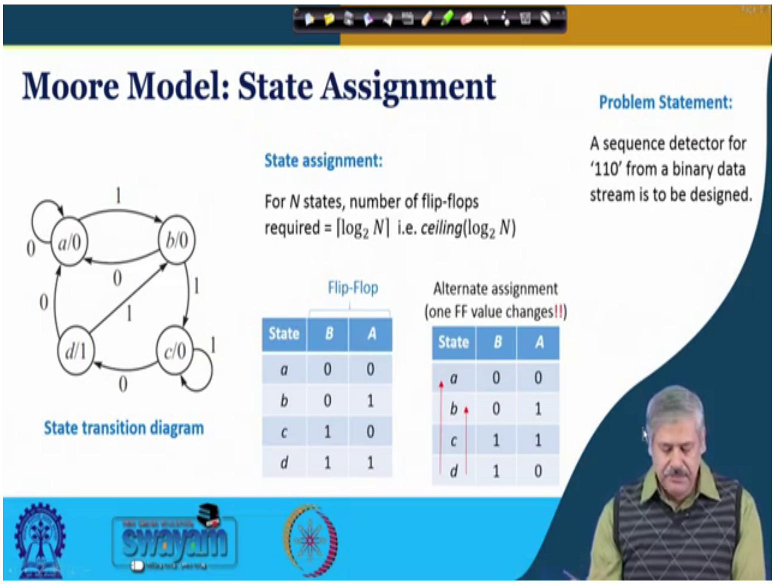
❑ State Table

❑ Design Equations

❑ Circuit Realization & Comparison of Complexity

And in this particular class, so we shall move ahead with that problem statement. And we would like to see how the circuit is finally realized. And we shall look into another part of the comparison that for the same problem statement, the circuit that we get from Moore model and the circuit we get from Mealy model, how do they compare in terms of complexity ok.

(Refer Slide Time: 01:15)



Moore Model: State Assignment

State transition diagram

State assignment:

For N states, number of flip-flops required = $\lceil \log_2 N \rceil$ i.e. *ceiling*($\log_2 N$)

Flip-Flop

State	B	A
a	0	0
b	0	1
c	1	0
d	1	1

Alternate assignment (one FF value changes!!)

State	B	A
a	0	0
b	0	1
c	1	1
d	1	0

Problem Statement:

A sequence detector for '110' from a binary data stream is to be designed.

So, we continue with the problem, you are aware of that problem. The problem was that of detecting a sequence from a binary data stream ok. And this sequence to be detected was 110 ok, this we remember right. And for that the Moore model, we had the circuit, we had the state transition diagram or something like this. And the 4 states were defined in this manner that a, that is initial state, no bit is detected properly, b - 1 bit is detected ok, c was 2 bit detected and properly and d was all 3 bits have been detected properly ok, so that is what we had defined in terms of the states right.

And we had seen that when it is at a, 0 is there, so basically it is remaining at 0. 1 it is going to b. Again at b, when it receives a 1, it goes to c that means to be it is a properly detecting. And then when it is receiving a 0, it will goes to d 3 bits are properly detected, and it is a Moore model. So, output is generated solely from the state value, so at b the output is shown. And the rest of the cases other options other than this different sequence is there in the other path that you see in the state transition diagram ok. So, this we have discussed in detail in the previous class ok, so we can refer to that.

Now, we need to see how the circuit is realized. So, for that we need to do a state assignment, because states we have mention in terms of a, b, c, d right. Now, we have to refer to it in terms of the memory elements, so we shall be using flip-flops, so in terms of the flip-flop values right. So, how many flip-flops should be required, so there are 4 states right.

So, from this formula that ceiling log N to the base 2, where N is a number of states in the state transition diagram. So, by which we can see that 2 flip-flops will be required right. Ceiling means, if it is a fraction, it will be taken to the next higher integer ok. So, if it is 1.2, it will be 2 ok, 2.3 it will be 3, so that is the meaning of this particular operator ok.

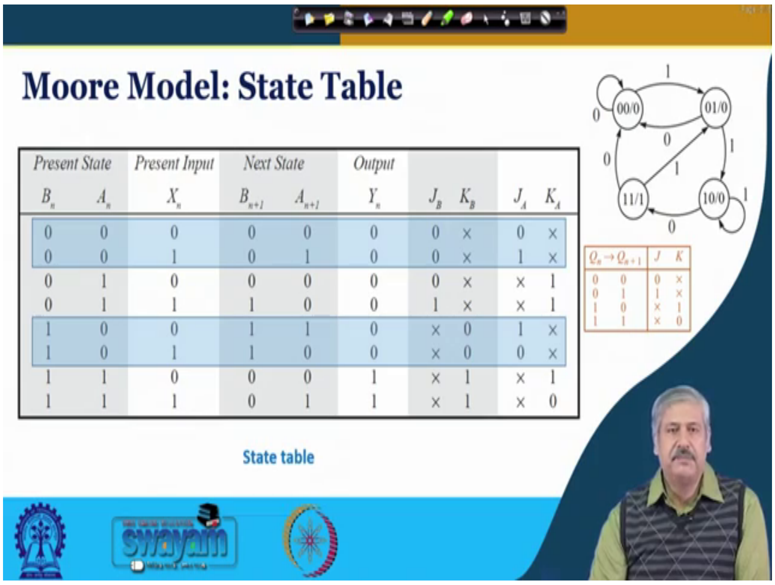
Now, so this 4 states, so a, b, c, d, they need to be associated with 2 flip-flops, 4 different values ok. So, one option is the standard one a, the first one it is coming so 0 0, b is 0 1, c is 1 0, d is 1 1 ok. This is the we can do the assignment and then move ahead with the circuit realizations right, but there could be alternate way of doing things also like and which one will give a simpler circuit less complex circuit.

In the sense that less amount of combinatorial, you know hardware and all, number of flip-flops remains the same ok, so that is something cannot be told in advance that need to be worked out and see, because one more assignment could be that only 1 flip-flop, you know a, b, c, d it is there. So, only one value, 1 flip-flop value changes as we move from a, b, c, d ok, so that could be one thing. So, only 1 flip value changes. So, we can guess that the requirement for changing and the complexity related to this combinatorial circuit, they may be less ok.

So, in that sense we can assign 0 0, then b is 0 1 only one value is changing, c is 1 1 and then d is 1 0 ok. And from d if it is goes to a, it is again is going to 0 0. So, in that sense that could be another way of assigning it, but one need to be careful what I have shown here that you can see from d, it is going to b as well as a. So, in this assignment when it is going from d to a, 1 0 is changing to 0 0, so one value. But, from d when it is going to b, so 1 0 is changing to 0 1. So, two the both the values are changing ok.

So, these are the things because of the interconnections, so that is why I said that we need to you know see it in detail and many different things are there. Otherwise, we can move ahead with any one of them. And realize the circuit which will work ok. So, we consider the first one ok, but you can try as an exercise that how it would look, how the circuit will be you know appear, then with this assignment ok, so that you can take as an exercise.

(Refer Slide Time: 06:34)



Moore Model: State Table

Present State		Present Input	Next State		Output				
B_n	A_n	X_n	B_{n+1}	A_{n+1}	Y_n	J_B	K_B	J_A	K_A
0	0	0	0	0	0	0	×	0	×
0	0	1	0	1	0	0	×	1	×
0	1	0	0	0	0	0	×	×	1
0	1	1	1	0	0	1	×	×	1
1	0	0	1	1	0	×	0	1	×
1	0	1	1	0	0	×	0	0	×
1	1	0	0	0	1	×	1	×	1
1	1	1	0	1	1	×	1	×	0

$Q_n \rightarrow Q_{n+1}$		J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

State table

So, after that what do you do? Once the assignment is done, so this state transition diagram, now we can present in this manner ok, so, 0 0 right, then 0 1, 1 0 and 1 1 ok. And let us consider, we are using JK flip-flop for its realization. So, then we shall invoke, we shall figure out what is the JK flip-flop excitation table from our earlier class, we know that JK flip-flop excitation table is this one.

So, 0 to 0 transition, 0 x 0 cross that means 0, k x can be 0 or 1 any value right, for 0 to 1 1 cross 1 0 cross 1 and 1 1 cross 0 ok. So, this cross here is do not care that means, any of 0 or 1, we will do is it fine, this we all know alright. So, then if we move ahead, so if we try to see it, we go to the state table right; so, this is the present state ok.

And now the input is here, so you have to consider present input as well right. If we have seen that synthesis of counter design problem, there we did not consider the input right. So, it was moving from one state to another with the clock getting triggered ok. And then next state ok and the output; output will be generated solely from this present state values right.

And for this there are 2 flip-flops, so there are corresponding flip-flop inputs. So, these are the different you know columns of the state table is it fine. So, for 0 0, so 0 0, so we had showing it in one block right the current state. So, input can be 0 or input can be 1 is it fine right. And in that case for 0, next state is what 0 0 only from the state transition diagram, you can see 0 0 right ok.

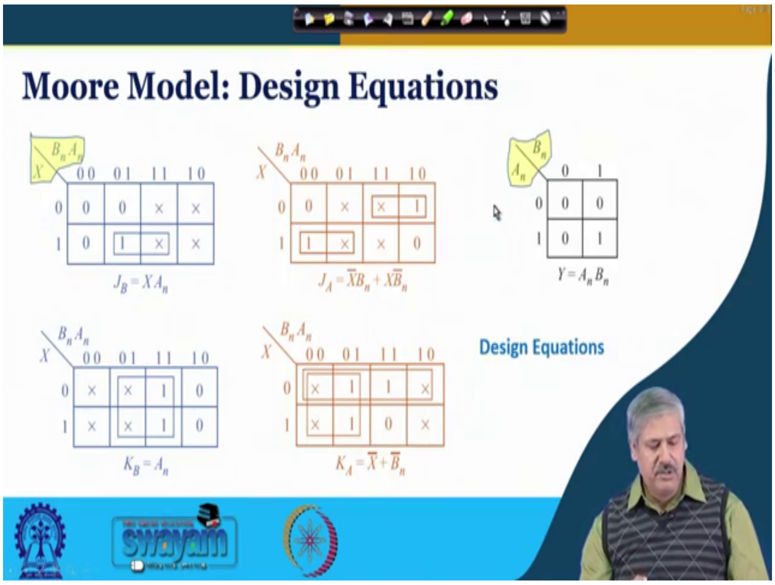
And for 1, it is the next value that is 0 1, we have already done the assignment earlier it was a and b, so now it is 0 0 and 0 1, so this is 0 1 ok. So, if it is 0 1, then there could be two possibilities 0 and 1 for the input 0 1 with 0, it goes to 0 0 with 1, it goes to 1 0 right with 1 0 this one. Again two possibilities with 0 ok, it goes to 1 1 right and with 1, it remains at displays, so, this is at 1 0 fine. And finally, for 0 1 sorry finally for 1 1 right, so we are here for 0, it goes to 0 0 and 1, it goes to 0 1 is it fine.

So, once we are done with this, then the rest is two take help of this excitation table. And for each of this transition B n to B n plus 1 and A n to A n plus 1 right down the corresponding J B K B and J A K A the input that are required at the at respective flip-flops is it ok. So, 0 to 0 0 cross sorry 0 to 0, this is 0 cross, 0 to 0, again 0 cross, 0 to 0 again 0 cross and 0 to 1 0 to 1, we know this is one cross. So, this way you can complete it, is it fine right, so this is that column.

Now, what about the other thing? So, this is your A n to A n plus 1 that we have to consider. So, 0 to 0 0 cross 0 to this is 1, 1 cross we have already seen 0 to 1, then 1 to 0, 1 to 0, this is 1 to 0, this is cross 1 right, this is cross 1 ok, so that is the way it will proceed is it fine right. So, this is the this particular column that also you get. What is left now, these output, how we fill up this particular output.

So, since it is Moore model right, output is always generated from the state, it does not depend on the input ok. So, irrespective of the value of the input. So, basically input is a kind of do not care in this condition. So, you do not bother, it will be totally derived from the state value ok. So, when it is at state 0 0 ok, for either of this options the output is 0, 0 1 0, 1 0, it is 0. And when it is at 1 1 current state, then the current output is 1 1, so that is what you see over here is it fine. So, with this we are able to complete state assignment, then corresponding state transition diagram with assigned state values ok. In terms of flip-flops, then state table using the help of excitation table.

(Refer Slide Time: 12:00)



Moore Model: Design Equations

$X \backslash B_nA_n$	00	01	11	10
0	0	0	×	×
1	0	1	×	×

$J_B = XA_n$

$X \backslash B_nA_n$	00	01	11	10
0	0	×	×	1
1	1	×	×	0

$J_A = \overline{X}B_n + X\overline{B}_n$

$A_n \backslash B_n$	0	1
0	0	0
1	0	1

$Y = A_nB_n$

$X \backslash B_nA_n$	00	01	11	10
0	×	×	1	0
1	×	×	1	0

$K_B = A_n$

$X \backslash B_nA_n$	00	01	11	10
0	×	1	1	×
1	×	1	0	×

$K_A = \overline{X} + \overline{B}_n$

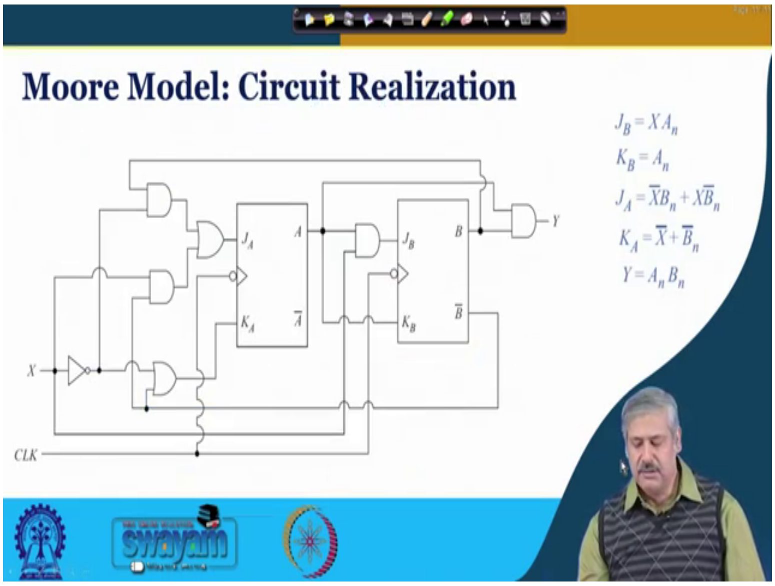
Design Equations

So, next step we will be as we had done in the past is to get the corresponding this simplified relationship for this two flip-flops ok. So, those are the design equations right, so that you can get. So, what we need to do, we need to just map it right. So, 3 variables are there the current state and the current input right. And what will be the current input; current input means, input bit that is coming from here. So, this current input for the flip-flop for which the next state is defined ok.

So, in a particular manner as per as the a state transition diagram is concerned ok. So, in that way we are just mapping it from the state table over here. And once we do that right for J B, once we map it in this manner, if you just for example we take up 1 and rest can be so for J B 0 0 0 1 cross cross cross cross is it ok. So, let us look at it ok, so this is 0 0. So, 0 0 0 1 cross cross cross cross 0 0, then we had gone in this direction for 0 0, this was the 0 1 for x 0 value and it is 1 value, so this is 0 0 0 1 cross cross and this is cross cross right.

So, similarly you will be mapping the other things and then we will we getting the largest group of you know in integer power of two all those things you know the Carno map is simplification. So, by this you get this 4th equations right. And in this case, what you get? In this case, you do not need the consideration for x, because it is solely derived from state value. So, state values are like this A n and B n right. If you put together, when both them are 1, it is 1 so Y is equal to A n B n is it fine. So, with these design questions in hand ok, we move to implement the circuit right.

(Refer Slide Time: 14:02)



Moore Model: Circuit Realization

$J_B = XA_n$

$K_B = A_n$

$J_A = \overline{X}B_n + X\overline{B}_n$

$K_A = \overline{X} + \overline{B}_n$

$Y = A_nB_n$

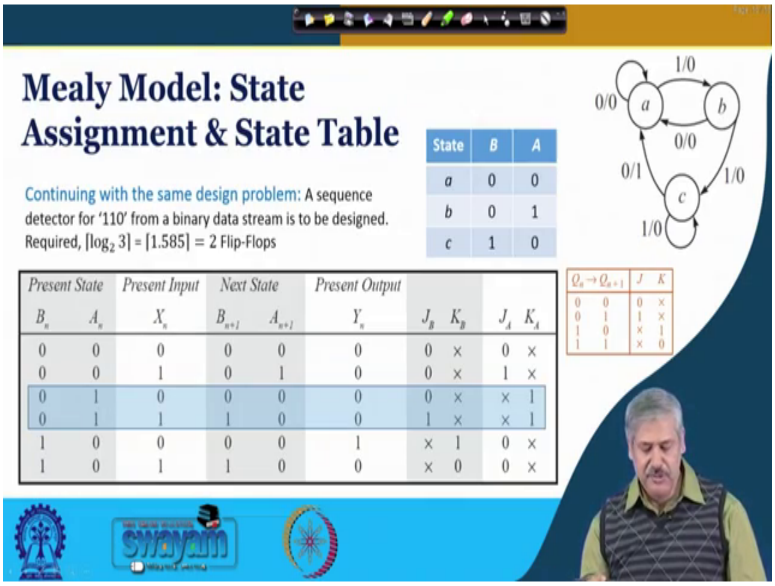
So, these are the equations that has been taken from the previous slide over here right. So, these are the 2 flip-flops J A and K A are the inputs for flip-flop A, J B K B for flip-flop B right. And you can see this is going to X is the clock synchronize synchronous circuit, clock is you know triggering both the flip-flop simultaneously.

So, J B is X A n, so just we verify some of these thing. So, J B is this one right, so X A n. So, this is your A N over here and this is your X. So, you can see that X is coming like this, this is the connection and this is the other one. So, this is your X A n, and K B is A n, so, K B is A n, you can see that K B is A n. J is X bar B n plus X B X B n bar, so, J B sorry that is J A, so that is your J A.

So, you can see that X bar, so this is your X bar right and this is your B n. So, this input this is your X bar X bar and this is your B n. And this input is your X and this input is it is coming from here you can see it is B n, so that is J right. K is X bar plus B n bar ok, so this is the or gate, so X bar is here and B n bar is coming from here. So, you can see the feedback right.

And finally, your Y is equal to A n B n right. So, this is Y, so B n is here and n is here right. So, was the equations are there is any questions are there, even if it looks little bit complex, we are in a position to get the corresponding circuit is it fine right. So, this is Moore model base circuit for that 1 1 0 detection.

(Refer Slide Time: 16:06)



Mealy Model: State Assignment & State Table

Continuing with the same design problem: A sequence detector for '110' from a binary data stream is to be designed. Required, $\lceil \log_2 3 \rceil = \lceil 1.585 \rceil = 2$ Flip-Flops

State	B	A
a	0	0
b	0	1
c	1	0

Present State		Present Input	Next State		Present Output				
B_n	A_n	X_n	B_{n+1}	A_{n+1}	Y_n	J_B	K_B	J_A	K_A
0	0	0	0	0	0	0	×	0	×
0	0	1	0	1	0	0	×	1	×
0	1	0	0	0	0	0	×	×	1
0	1	1	1	0	0	1	×	×	1
1	0	0	0	0	1	×	1	0	×
1	0	1	1	0	0	×	0	0	×

$Q_n \rightarrow Q_{n+1}$		J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

Now, we go to the Mealy model based state transition diagram, and its realization right. Now, one thing that we notice first that number of states are 3 ok. Because, when 2 bits are receipt correctly 1 and 1, then if 0 is receipt ok. In that same clock cycle, we can generate the output ok, so that is the thing right that is the Mealy model, it can give the output based on current state as well as the current input ok, so that we discussed in the previous class, so that is how it looks like this right, so that is why 3 states are sufficient right for this problem.

So, this 3 states, then how many flip-flops would be required of course 2 flip-flops will be required that because of this operator ok, it is 1 flip-flop with 1 flip-flop, you can have 2 states only ok, so, 3 states are there. So, state assignment, so let us do it the this way a is 0 0, 0 1 and 1 0 ok.

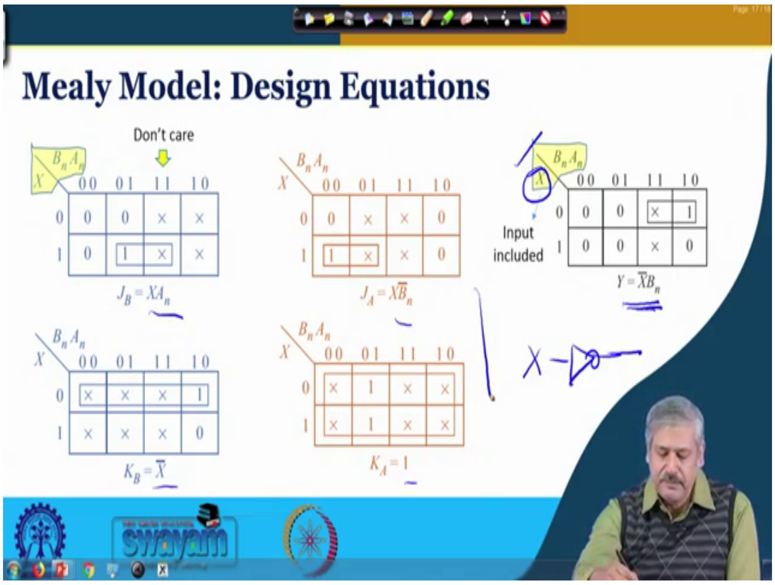
So, again for state 0 0 two possibilities 0 and 1, 0 1 two possibilities 0 and 1 and 1 1 two possibilities 0 0 and 1 right. And accordingly, the state changes are taking place similar to just Moore model right. So, 0 0, the way we have prepare the state table. So, 0 0, it is 0 0, 0 0 it is 0 1 for 1 here. 0 1, 0 1, it is going to for 0 it is going back here, so 0 0, this is going to 1 0. For 1 0, if it is 0 is received ok. So, 1 is the output, you can see the corresponding output is 1 right and next state is 0 0 and 1 0, it is 1 0 ok.

So, present input now the difference that when it is at 1 0, this is c 1 0 right. When it is at 1 0, if it receives 0 ok, it is 1 and if it is receives 1, it is 0 ok. So, under this condition only the input output is 1. So, this is the difference right. And the way we had filled up the JK flip-flop input columns in Moore model, the same thing will do by using the JK flip-flop excitation table, so that has been shown here ok and you can do it yourself also right.

So, what is the other difference that you can see compared to Moore model based state table other than this X is contributing to Y generation. Other than that what you can notice that one pair of column is missing that is 1 1 is not there right, because state B is not available not required ok. The how does it help in circuit realization does it help in any way to reduce the complexity, so let us see. So, this we have noted right. And again we are note down one particular case like say this one J B K B is 0 0 0 1 cross cross ok, so that we had seen before that J B was also cross cross here. So, you will be able to so other cases cross cross cross cross 1 0 ok.

So, let us see one such case, where we had this one K B cross cross cross cross 0 0 1 1 right. So, now we in this particular case came, so we have in this particular case just by comparison. So, this what J A, we are talking about 0 1 cross cross 1 0, then there was cross 1 1 sorry K B cross cross cross cross 0 0 1 1 ok. So, here it is different, because it is going from 0 0 to 1 0 to 0 0 ok, so that is why it is cross 1.

(Refer Slide Time: 20:30)



Mealy Model: Design Equations

Don't care

$X \backslash B_nA_n$	00	01	11	10
0	0	0	x	x
1	0	1	x	x

$J_B = XA_n$

$X \backslash B_nA_n$	00	01	11	10
0	0	x	x	0
1	1	x	x	0

$J_A = X\overline{B}_n$

Input included

$X \backslash B_nA_n$	00	01	11	10
0	0	0	x	1
1	0	0	x	0

$Y = \overline{X}B_n$

$X \backslash B_nA_n$	00	01	11	10
0	x	x	x	1
1	x	x	x	0

$K_B = \overline{X}$

$X \backslash B_nA_n$	00	01	11	10
0	x	1	x	x
1	x	1	x	x

$K_A = 1$

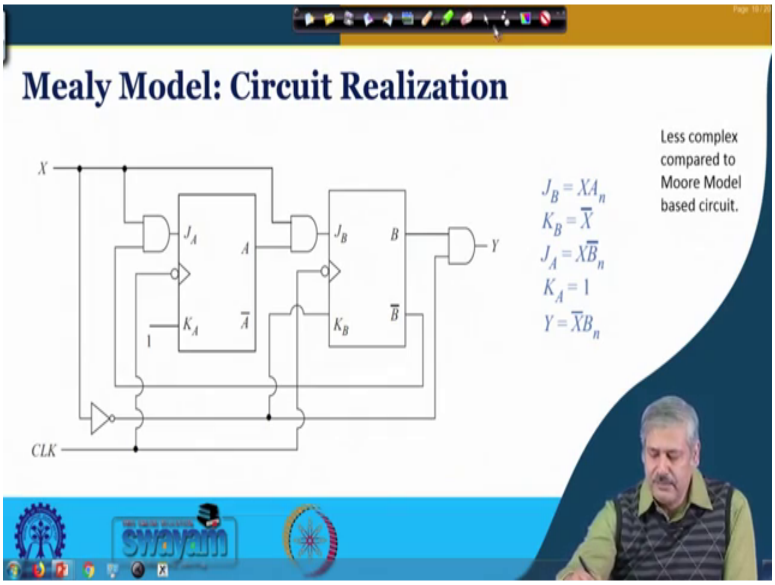
So, what you can see, now for this K B right over here. So, this 1 1, earlier it was 1 1 right, but here you can see that this is cross cross. For K B, it was 0 0, because this state change was different, so that is another aspect of it, but this 1 1 was not cross was 1 1 right. For J J B, it was cross cross because of the way the mixed values were there, but at least what here you can see that 1 1 is not there it is cross cross ok, because 1 1 is not coming this state these not coming into picture right.

So, similarly over here, similarly over here right, so for which you do not have any requirement as was the case in the other Moore model based realization. So, if there was a one here that one need to be covered ok, so since it is not there. So, we do not need to cover it, so that way you can see the final equations that you get are simpler than the one that you had seen in case of Moore model right, so because of which you will be having at least for this part, it let be simpler circuit is not it ok.

But, one thing that we note here that there only 2 such variable were there, now we have got 3 variable coming into place ok. But, even if 3 variables are there, the circuit that you have is the logic is earlier it was A n B n, now it is X bar B n, so that way it is not making too much difference ok. So, and X for other reasons, there is X bar is always getting generated because of the NOT gate ok.

So, even if it is not getting generated, then only one additional NOT gate is required, other than what we had seen before the AND gate required for the Moore model ok, so this is minimal. Even if something because of this additional variable that is there, but in the other cases you can see that there is a relatively you know less complex design equations that we get because of the state the not being present here in the Mealy model ok.

(Refer Slide Time: 22:58)



Mealy Model: Circuit Realization

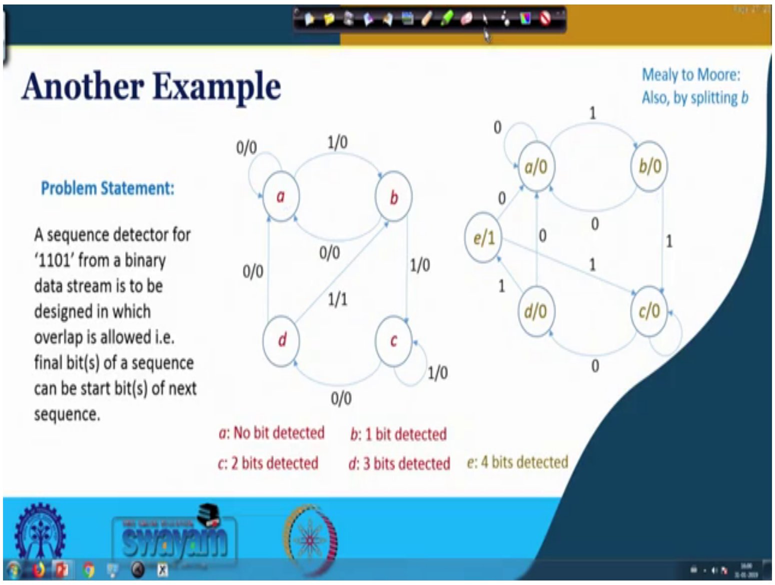
$J_B = XA_n$
$K_B = \overline{X}$
$J_A = X\overline{B}_n$
$K_A = 1$
$Y = \overline{X}B_n$

Less complex compared to Moore Model based circuit.

So, with these equations, now you can see these are the equations right. And the other case if you remember for J A, it was X B n bar plus X bar B n some such thing is not it right. So, you can see that all those things have got you know simplified, I mean more simple ok. So, this is the for the same problem, you have got a Mealy model based circuit like this.

So, usually because of you know less number of states being you know there and possibility of having do not care and all, so you can see. You can when expect that Moore you know complex circuit will be for Moore model and less complex for Mealy model is it fine.

(Refer Slide Time: 23:51)



Another Example

Problem Statement:

A sequence detector for '1101' from a binary data stream is to be designed in which overlap is allowed i.e. final bit(s) of a sequence can be start bit(s) of next sequence.

a: No bit detected b: 1 bit detected

c: 2 bits detected d: 3 bits detected e: 4 bits detected

Mealy to Moore:
Also, by splitting b

So, before we end, we would like to extend this thing on the complexity part, but by being a bringing another example which is in essence extension of the previous example. So, initial have a also a practice of you know developing Mealy model, Moore model, so let us see. So, this is a sequence detector problem on the as we said extinction, but it is now what we want to be detected is 1 1, 0 1 ok. Earlier it was only 1 1 0 so 1 more bit has been added.

So, whenever this 4 bits 1101 is found to be present in the input data stream, we say that the sequence the that sequence we were looking for that has been there ok. And the other thing what has been mentioned here that overlap is allowed ok. So, what does it mean after 1101 ok, if it is non-overlapping that means, you need another 1101 ok.

And if it is overlapping that means, after 1101, again if you get a 101, this one is getting included, so that is the overlap. So, then also you will say, here there will be one output, again here there will be a one output, when you are over here ok, so that means, if it is overlap type, then final bit or bits of a sequence can be start bits of next sequence ok, so that is the thing.

So, let us see if such is the case, how it would work out ok. So, we start with Mealy model ok, the same thing that we the way we have defined. So, a means no bit has been detected properly, b 1 bit, c 2 bits and 3 d 3 bits. Now, we need 3 bits, because 4 bit is to be detected. So, when you are at d that means, 3 bits have been detected properly ok. Then based on the input you take a decision ok, so that is the requirement.

So, at a so initial part will be similar to what we had in the previous case, because it is 110, the first three right. So, a 0 is receipt, it is staying here a 1 is receipt, it is going to b, because it is 1 with properly detected. So, it receives another one, second bit properly detected, it comes to c ok.

But, after 1 if 0 comes, then again you have to restart up ok, so that is why for 0, it is going back to a right. Then at c, if you receive a 0 that means, 110; 110 has been received. So, you come to that is 3 bit correctly received, so you come to d right. And if you receive a 1 here, you stay at c, because after 11, you can if you get a 1, again if you get a 0, then you are doing right that means, you are going right in the you know detecting the sequence. So, you are staying here at c ok.

Now, when you are at d 1 1 0, then if you receive a 0, what happens ok, you have to restart the entire thing is not it, so that means again you need to look for 110 in that incoming bit stream ok, so that is what has been shown here. But, if you are d at d it is 110 and you receive a 1 ok, then it is correctly detected right. So, the output will be 1 that is what has been shown here.

Now, in these output is one, after that where the circuit should go ok. The circuit should go to a state which says that 1 bit is already detected properly, because after that if you get 101, which is then that is you are doing right because of this overlapping, so after the with a 1, it is going to b is it fine. So, this is your Mealy model based circuits, so 4 states are required.

Now, what happens to Moore model based circuit right. So, up to 4 state for a 3 bits detected. Now, you need another state which is 4 bits detected, because the decision is taken from the state value only, unless it is reaches e that means 4 bits properly detected, it cannot generate the output ok.

So, then if you quickly follow, so it is going a, b, c, d, the way we have seen before right, when it is at d ok. If it receives 1, it goes to the 5th state that is e. And if it receives 0, it goes to a because of the same reason the way we have said the you know bit pattern unfolds ok. And when it is at e ok, then it is at e means, it has received 1 0 1 1 ok, this is already received.

Now, when it is at e, it can receive a 0 or 1. If it is receiving 0, then it has to restart that we know. But, if it receives a 1 ok, what does it mean; what does it mean right? That 2 bits are properly detected, please understand this part. One it is at e means, it is already 1 0 1 1 right. Then it is receiving a 1, so that means, 2 bits are property detected. If you receive another 0 1, then the sequence is there ok. So, 2 bits properly detected is your c, so it is going to c is it ok. This part is understood ok, why where it is going to b, it is going to c right.

Because, after d that means, it is c it is it has before that it is received only 0. Here at e means, it is it has received 1. After that it is receiving 1, so two 1s are there ok, so and it is overlapping is it clear right. So, the other thing that we can see that there are 5 states ok and here are four states.

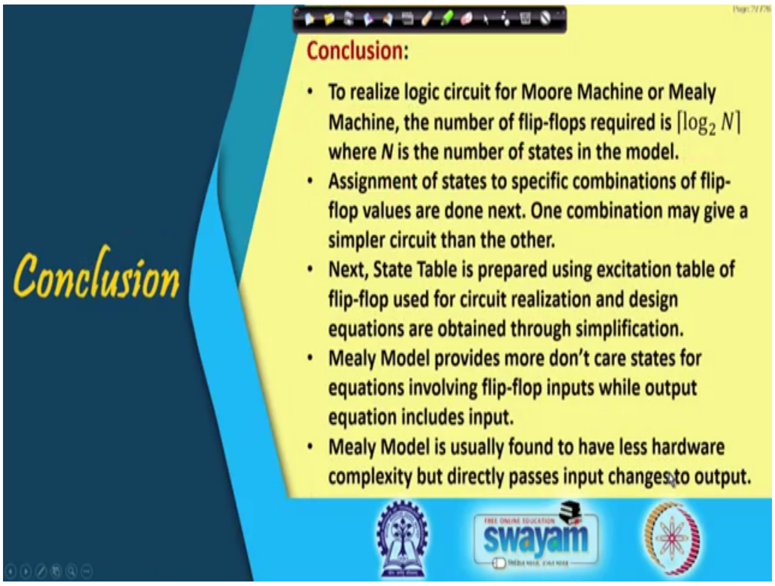
So, number of flip-flops required will be again by that formula, here it will be 2 flip-flops will do, but here 2 flip-flops will not do you will be requiring 3 flip-flops. So, it is not only the combinatorial logic circuit, the there could be a there could be cases, where the number of flip-flops memory elements also need to be more, when you compare you know Mealy model and Moore model ok. So, this usually happens in situations like this, when you are just moving from one particular case like 4 to 5, so at the junctions ok.

And finally, on this particular slide, if you are using those conversion formula conversion rules ok, Mealy model to Moore model and all that we are discussed in the last class, so which one here, I mean need to be split, because it is for other cases it is every other cases fine.

So, you can see that for that finding that candidate, you have to see that a particular state at the input of which there are outputs 0 as well as 1. If it is only 0, there is no issue, you can absorb that 0 or 1 within that particular state right. So, for example said b ok, here these input is output is 0 and these output is 1. So, the b need to be split b need to be split. So, 1 b remains here like this, another b becomes e ok, you can imply that rule. And you can see for yourself as an exercise, the way we are done for one example there in the last class for 3 bit detection. The same thing over here, you can see that if you split, it will be b and e will be coming here alright and let us some the cases are fine ok.

For example, say a this is 0, and this is also 0. So, you can put a 0 inside. So, none of the other cases, there is any such issue only for b, there is the issue. And if you just need to follow the a machine rule and you can do it. And you can come from this side to that side also, I mean Moore model to Mealy model also ok. Whenever you see that from a particular place, you are having a 0's and 1's right.

(Refer Slide Time: 32:54)



Conclusion

Conclusion:

- To realize logic circuit for Moore Machine or Mealy Machine, the number of flip-flops required is $\lceil \log_2 N \rceil$ where N is the number of states in the model.
- Assignment of states to specific combinations of flip-flop values are done next. One combination may give a simpler circuit than the other.
- Next, State Table is prepared using excitation table of flip-flop used for circuit realization and design equations are obtained through simplification.
- Mealy Model provides more don't care states for equations involving flip-flop inputs while output equation includes input.
- Mealy Model is usually found to have less hardware complexity but directly passes input changes to output.

So, to conclude, so we have seen that the to move from the state transition diagram to a particular circuit using Moore model and Moore model and Mealy model and this is a finite state circuit or finite state machine, so often it is called Moore machine and Mealy machine ok. So, number of flip-flops required will be defined by the number of states that is there, so it is log 2 N the ceiling of log 2 N ok.

And assignments of states is something by which one can get to different realization of the circuit ok, I mean if you have two different assignment, two different realization and one may be simpler than the other ok. And from after the assignment, you are going for state table using the excitation table of the flip-flops with which you are trying to realize the circuit. We have seen the example of JK flip-flop, we could have used to SRTD ok, and corresponding excitation table would have be used ok.

Mealy model we have seen because of the possibility having of having mode do not care states the combinatorial circuit the that complexity may be less and also the number of flip-flop required may be less in a in certain cases ok. But, what we have seen that we have noted from the last class also that the input is connected to the output logic circuit the combinatorial logic circuit, so any change in input directly passes to the output ok.

Thank you.