Digital Electronic Circuits

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

Lecture – 49
Circuit Realization from ASM and State Minimization

Hello everybody. In the last class, we looked at ASM chart and given a problem statement, we have seen how we can get a ASM chart for that particular problem statement. And we took up vending machine which has got two inputs as an example, also saw the sequence reactor example right.

(Refer Slide Time: 00:42)

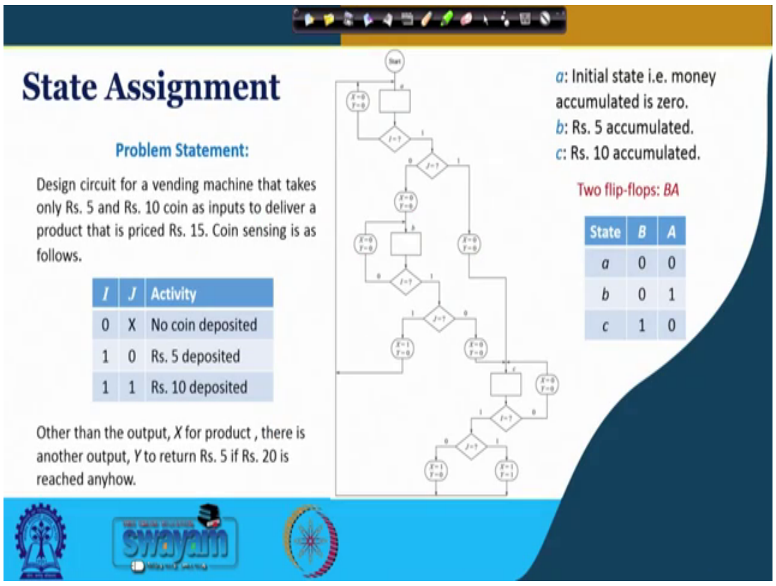
CONCEPTS COVERED

Concepts Covered:

- Circuit Realization from ASM Chart based Design
- Decoder-OR for Combinatorial Circuit
- State Minimization: Row Elimination Method

So, in today's class we shall look at circuit realization using ASM chart and after that we shall look at a decoder-or based combinatorial circuit realization process ok. And finally, we shall move to state minimization and we shall look at wall method called row elimination method ok.

(Refer Slide Time: 01:01)



State Assignment

Problem Statement:

Design circuit for a vending machine that takes only Rs. 5 and Rs. 10 coin as inputs to deliver a product that is priced Rs. 15. Coin sensing is as follows.

I	*J*	Activity
0	X	No coin deposited
1	0	Rs. 5 deposited
1	1	Rs. 10 deposited

Other than the output, *X* for product, there is another output, *Y* to return Rs. 5 if Rs. 20 is reached anyhow.

a: Initial state i.e. money accumulated is zero.
b: Rs. 5 accumulated.
c: Rs. 10 accumulated.

Two flip-flops: *BA*

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

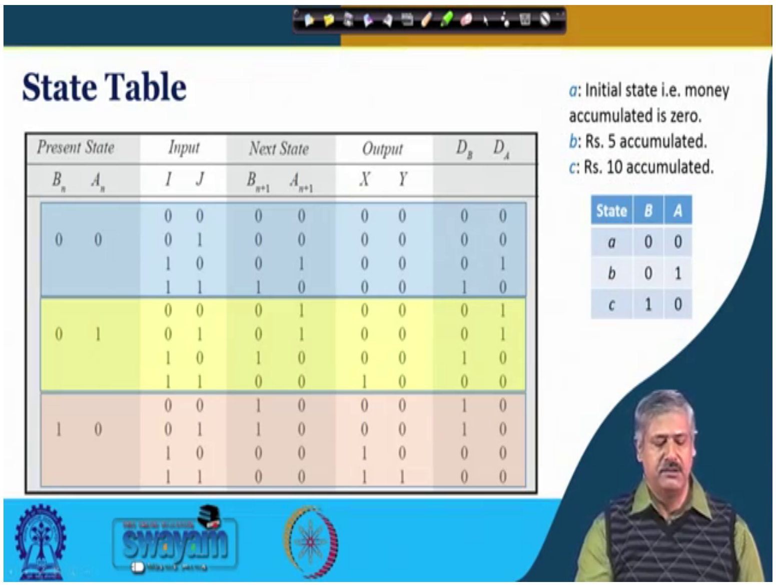
So, for ASM chart based circuit realization, so the problem that we had taken up in the previous class, we continue with that ok. So, that the vending machine problem where, rupees 5 and rupees 10 coin can only be deposited and the product is priced rupees 15. There is a coin sensing mechanism, where if the sensor I is 0; that means, no coin is deposited. Whenever a coin is deposited, it becomes 1 and there is another sensor, if it is 0 when coin is deposited, it is rupees 5 that has been deposited. If it is 1 rupees, 10 has been deposited.

And the output; there were two outputs X and Y. X delivers the product when X is equal to 1 product is delivered, otherwise X remains 0 and Y is equal to 1, rupees 5 is returned because by any chance a it goes to rupees 20 that means, the total money that has been deposited into the vending machine say rupees 10, then rupees 10 or rupees 5, rupees 5, then rupees 10 ok; whatever be the sequence, then a rupees 5 coin is returned to the customer ok. So, this is the problem statement with which we worked and we arrived at this state condition diagram ok. We did part by part analysis every state and this was the state condition diagram ok. So, up to this we had done in the previous class right.

Now, to implement this particular circuit, we need to first do a state assignment ok. So, states were defined as a, b and c; we are already noted. So, we considered that we know 3 states means 2 flip flops will be required. So, 2 flip flops, we name it B and A; capital B and capital A and corresponding states are 0 0 for a; for b, it is 0 1 and for c, it is 1 0.

So, next important task is to move to state table from ASM chart ok. So, what we shall look into in that part ok.

(Refer Slide Time: 03:21)



State Table

Present State		Input		Next State		Output			
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y	D_B	D_A
0	0	0	0	0	0	0	0	0	0
		0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
0	1	0	0	0	1	0	0	0	1
		0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
1	0	0	0	1	0	0	0	1	0
		0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

a: Initial state i.e. money accumulated is zero.
b: Rs. 5 accumulated.
c: Rs. 10 accumulated.

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

So, for every state that has been defined; so Bn An. So, that is when it is 0 0, it is state a ok. So, state a there are 4 possibilities at the input site. I is a 0 0 0 1, 1 0 and 1 1. So, of course, I is equal to 0, then J is 0 1, it is you know does not matter because no coin has been deposited that is the case ok. So, 0 cross that is the way, we have defined it over here ok; over here. Each of this cases what is the next state? Right.

So, for I is equal to 0, the next state is state a only that is 0 0. So, that is what you see over here as 0 0 0 and 0. Is it fine? And if it is 0 0 current state that is state a, I is equal to 1 that means, the coin has been deposited right and J is equal to 0 that means, rupees 5. So, it goes to state b; state b is defined as 0 1 rupees 5 accumulated. So, that is your 0 1 and 0 0 and then, increase the input is 1 1; that means, rupees 10 has been deposited.

So, in the ASM chart, we can see that is going to state c right. So, it is going to 1 0; is it fine? So, from the ASM chart, we can find out for what combination of the input I and J from current state, we go to a specific next state; is it clear right? So, that is the ASM chart to state table might be. Once it is done, rest of the thing is as we had done for sequential logic design circuit ok.

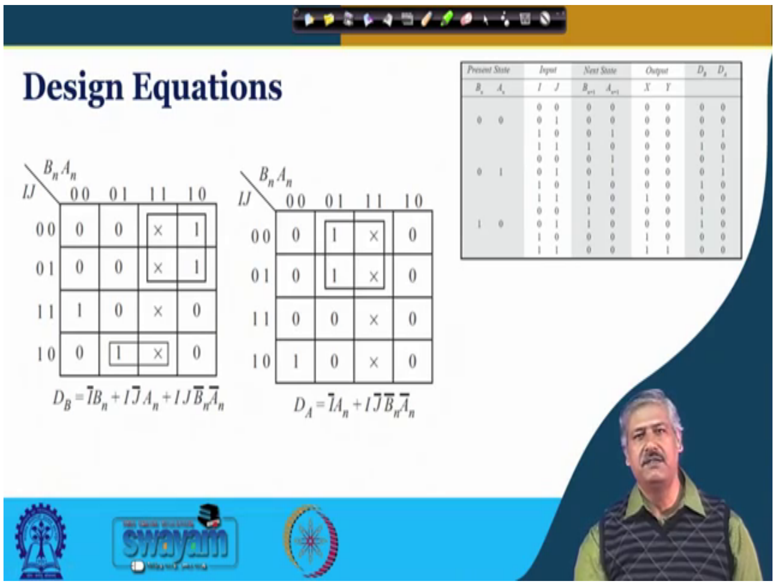
So, next is 0 1; 0 1 is your state b that is the way you have assigned state right. So, these two cases I is equal to 0. So, it will remain at state b on state 0 1 only; state small b only right and when it is 1 0 that means, rupees 5 and has been deposited. Rupees 5 and rupees 5, it will go to rupees 10 state that is c that is 1 0, you can see what here from the ASM chart, we can we can figure it out that path leads to state c right and from 0 1 if it is receiving a 1 1 that means, 5 rupees 5 and rupees 10.

So if we go back, so in the ASM chart, we can see this is state b right. So, this is 1 and this is 1 right. What does it do? It delivers a product; X is equal to 1 and it goes to state a right. So, we can write; we can write when it is at 0 1 and it is at 1 1, it is going to 0 1 sorry 0 0 that is state a is it fine.

Next if it is 1 0 that means, state c right; 0 0 it will remain at state c only; 1 0 it is going to that means, rupees 5 has been you know obtained. So, it is going to state a and delivering only the product. 1 1 that means, rupees 10 and another rupees 10 has been received. It is going to state a; initial state a, you have seen that in the ASM chart and delivering the product and delivering also returning a rupees 5 coin. Is it fine? So, this we can map from ASM chart to state table.

Once that is done rest of the thing is as we have done in the earlier classes for sequential logic design circuit ok. So, mostly we have used JK flip flop. So, for a change let us use D flip-flop and it has certain other significance that we shall see later. So, if you use D flip flop, then the excitation table is very simple right. Whatever is at the input of the flip flop that goes to the output. So, whatever is the next state so that will be the input value of the flip flop; is not it? So, for which whatever is the input values, what you see over here sorry next state values. So, those are the D B values right. Similarly, whatever are the next state values for n plus 1. So, those are the values for D A; is it fine?.

(Refer Slide Time: 08:21)



Design Equations

$$\begin{array}{c|cccc} IJ \backslash B_nA_n & 00 & 01 & 11 & 10 \\ \hline 00 & 0 & 0 & \times & 1 \\ 01 & 0 & 0 & \times & 1 \\ 11 & 1 & 0 & \times & 0 \\ 10 & 0 & 1 & \times & 0 \end{array}$$

$$D_B = \bar{I}B_n + I\bar{J}A_n + IJ\bar{B}_n\bar{A}_n$$

$$\begin{array}{c|cccc} IJ \backslash B_nA_n & 00 & 01 & 11 & 10 \\ \hline 00 & 0 & 1 & \times & 0 \\ 01 & 0 & 1 & \times & 0 \\ 11 & 0 & 0 & \times & 0 \\ 10 & 1 & 0 & \times & 0 \end{array}$$

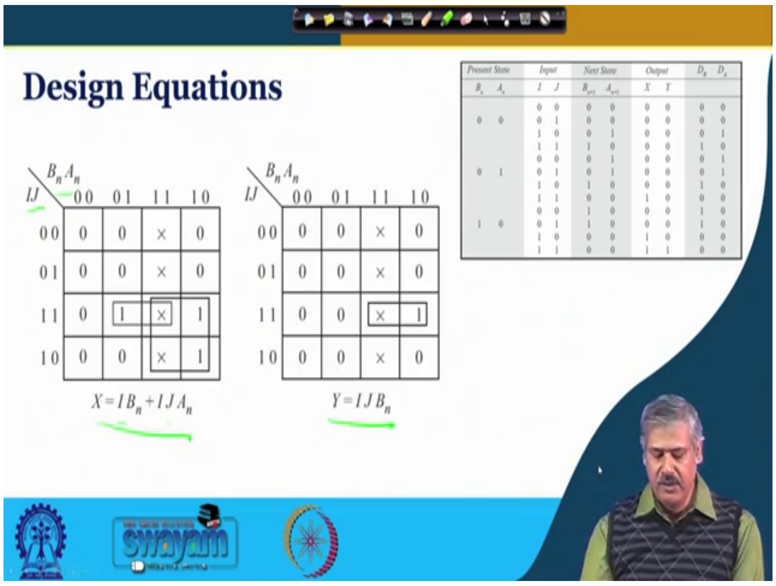
$$D_A = \bar{I}A_n + I\bar{J}\bar{B}_n\bar{A}_n$$

Present State		Input		Next State		Output		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
		0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
		0	0	0	1	0	0	0	1
0	1	0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
		0	0	1	0	0	0	1	0
1	0	0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

So, with that we go to what will be the next step? To get the design equations. So, this input equations right. So, with these inputs these way it has been written 0 0 0 1, 0 0 1 0, 1 1 0 0; this is for D B ok. So, you can see that is the case 0 0 0 1 ok, 0 0 1 0, then 1 1 and 0 0 and then, rest are do not care ok; the rest are do not care ok. Because 1 1 is not there fine. So, the rest are do not care fine. So, then if you form the groups; so, these are the 2 groups you can see and this is another group.

So, these are the 4 variables that are there by which you get this equation. Similarly, for the other case is it ok. So, this is one group for this is for D A, group of 4 members and this is another group which is given by this one ok. I J bar B bar B n bar A n bar, similarly for the rest of the cases this part you know very well; is not it?. So, this is the equation for the a flip flop inputs.

(Refer Slide Time: 09:42)



Design Equations

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

$X = IB_n + IJA_n$

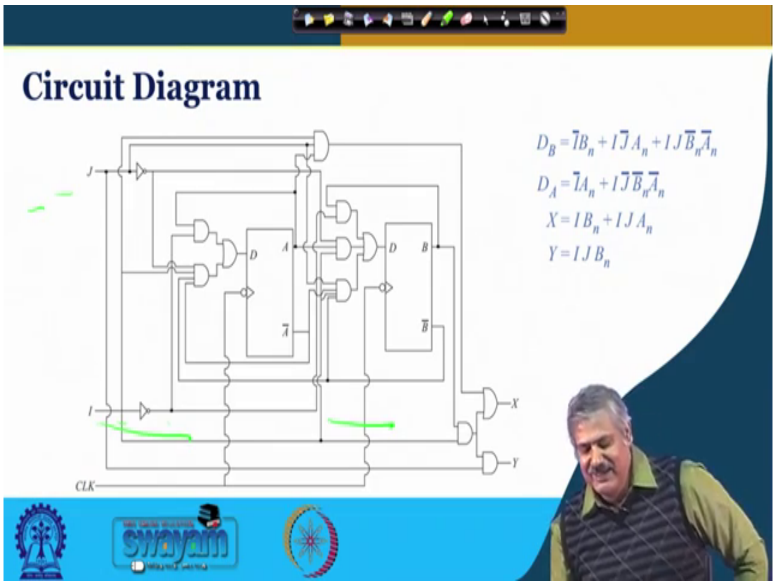
$IJ \backslash B_nA_n$	00	01	11	10
00	0	0	×	0
01	0	0	×	0
11	0	0	×	1
10	0	0	×	0

$Y = IJB_n$

Present State		Input		Next State		Output		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
		0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
		0	0	0	1	0	0	0	1
0	1	0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
		0	0	1	0	0	0	1	0
1	0	0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

And then, corresponding equation for what is that called these outputs right. Since, it is milli model. So, output equation considers the input as well. Is not it? So, that is I and J. Had it been mood model? It would have been solely generate from the flip flop output. So, this I and J would not have been considered right. So, that is why you see that all this input 2 inputs as well as the states are considered over here. So, accordingly this is the equation and you have got another equation for Y over here ok. So, X and Y; so, we are directly mapping it from state table to the design equation following the similar process that you have followed before ok.

(Refer Slide Time: 10:39)



Circuit Diagram

$D_B = \bar{I}B_n + I\bar{J}A_n + IJ\bar{B}_n\bar{A}_n$

$D_A = \bar{I}A_n + I\bar{J}\bar{B}_n\bar{A}_n$

$X = IB_n + IJA_n$

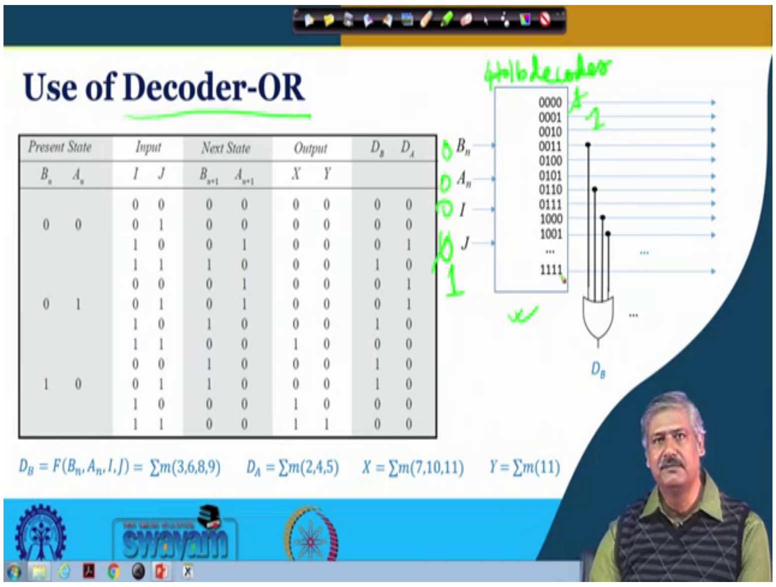
$Y = IJB_n$

So, once we get these equations ok. So, these are the equations that you have obtained. So, next is putting them into the circuit right that we all know how it is to be done. So, if you just take one of them as the example; so, this is your A flip flop right. So, D A; if we look at the D A right. So, this is I and this is I bar right and then, this is your A right. So, this A is coming back over here and this is I bar. So, I A; I bar A n right and this is I J bar B n bar A n bar.

So, this is the four input I will get right which is taking those necessary inputs and then, you have got A OR right. So, this is why which you are getting the D A. Similarly, for the others clock is common right and the output is generated in this manner; is it fine ok. So, you understand how to realize the convert the circuit; just we follow the same steps ok.

Now, to note that compared to JK flip flop, this circuit looks little bit more complicated; is not it because they it provided I mean at least the combinatorial circuit part because it has got less number of do not cares and other things. So, that way the equations that you get, has got more variables, more literals present in the equation ok. So, that is what you see over here ok. Now that gives us you know one opportunity to look at the other options that could be there for realizing the circuit.

(Refer Slide Time: 12:30)



Use of Decoder-OR

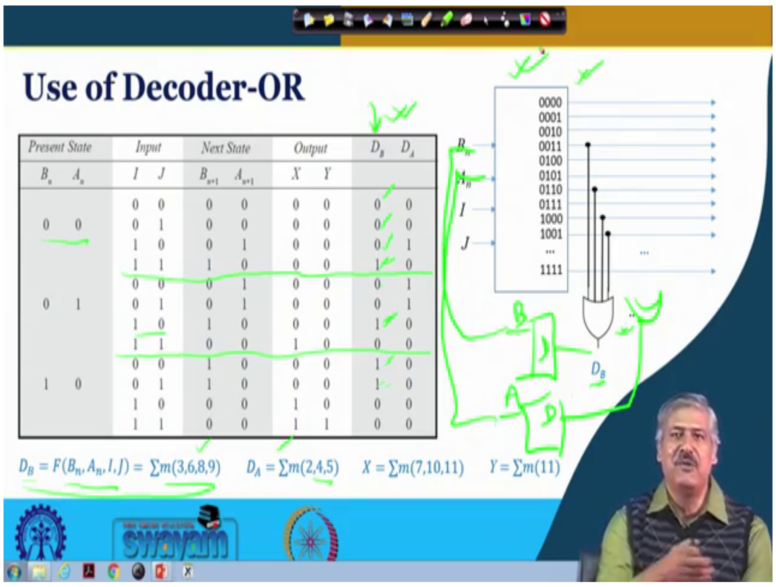
Present State		Input		Next State		Output		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
		0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
		0	0	0	1	0	0	0	1
0	1	0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
		0	0	1	0	0	0	1	0
1	0	0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

$D_B = F(B_n, A_n, I, J) = \sum m(3,6,8,9)$ $D_A = \sum m(2,4,5)$ $X = \sum m(7,10,11)$ $Y = \sum m(11)$

And one such option could be when we are using D flip-flop ok, knowing fully well that the next state is the input to the flip flop right. Then, we can use an arrangement ok. The way we had done if you remember the combinatorial circuit, the Decoder-OR based you know multiple output generation scheme. So, we can employ that for the combinatorial part of the circuit ok. It will not be more simplified, but from hardware point of view it will be easier to get. So, for that we need what? So, this is a 4 to 16 decoder ok. So, this is a 4 to 16 decoder right. So, B n A n I is a these are the 4 inputs and 0 0 0 0 0 1, 0 0 1 0 up to 1 1 1 1, these are the corresponding outputs.

So, depending on the combination present, if 0 0 0 0 is present, so this will be high. All the rest of the things will be 0 ok. If instead 0 0 0 1 is present, this will be high and rest all the outputs will be 0. This is how a decoder works; is it fine? We remember that; so, we use that concept right.

(Refer Slide Time: 14:02)



Use of Decoder-OR

Present State		Input		Next State		Output		D_B	D_A
B_n	A_n	I	J	B_{n+1}	A_{n+1}	X	Y		
		0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
		1	0	0	1	0	0	0	1
		1	1	1	0	0	0	1	0
		0	0	0	1	0	0	0	1
0	1	0	1	0	1	0	0	0	1
		1	0	1	0	0	0	1	0
		1	1	0	0	1	0	0	0
		0	0	1	0	0	0	1	0
1	0	0	1	1	0	0	0	1	0
		1	0	0	0	1	0	0	0
		1	1	0	0	1	1	0	0

$D_B = F(B_n, A_n, I, J) = \sum m(3,6,8,9)$ $D_A = \sum m(2,4,5)$ $X = \sum m(7,10,11)$ $Y = \sum m(11)$

And we can see the state table that we have got ok, we can directly; we can directly convert to a Decoder-OR based combinatorial circuit representation ok. How? So, the D flip-flop that we shall be using 2 D flip-flop that are there. So, this is the corresponding input right and if this is your 0 B n A n I J ok. So, this is your 0 0 0 0 right, for that D B is 0; 0 0 0 1, D B is 0; 0 0 1 0, D B is 0 and 0 0 1 1, D B is 1 ok. That is what you can see. Similarly, for 0 1 0 1 D B is 0; this way you continue when 0 1 1 0 for this particular case this is 1 right.

Similarly, for 1 0 0 0; this is 1, 1 0 0 0 1, this is 1 right. So, what you can see for this particular input combination, if we write D B as a function of B n A n I and J and the 16 mean terms that are getting generated ok. So, these are this is 0 0 1 1; so, 3; then this is 6 right and then, this is 8 and this is 9. So, if you just sum them up right, we will get the value of D B. Is not it?

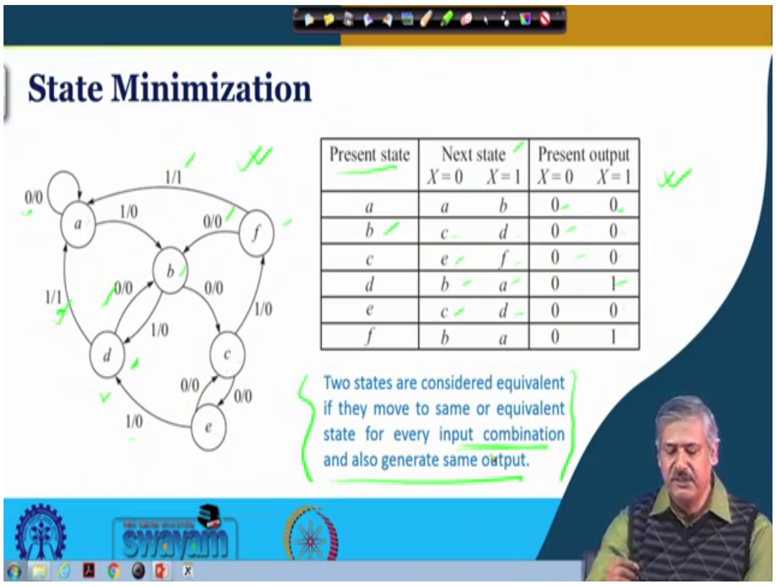
So, this decoder is generating all the mean terms right and we need to a before input or gate over here in the Decoder-OR gate combination. So, this will be the function this you know sigma m 3, 6, 8, 9 which is basically the function the design equation for D B. Similarly, for D A if you just in look at it is 2, 4, 5 ok. So, you can have this another or gate. So, that is why I have given dot dot dot and so and from that, you can get the a corresponding input for D. So, that you fit to the flip flops.

So, you will be feeding that to D flip flop over here. So, similarly over here, you will be feeding to that flip D flip-flop over here. So, this is your D; this is your D; this is your corresponding B and this is your corresponding A and this is B will be fed here and D A will be fed there; is it fine right. So, this is the way you can design the circuit. So, this is the combinatorial part of the circuit.

This the combinatorial part that you had seen which got designed ok. By this design equations and is represented in this manner, this is minimized, this is minimized; but you have to use many gates, but this decoder is comes in a compact you know IC integrate circuit fabricated. So, you can use those output and can make use of multiple or gates. So, similarly X and Y, you can generate from this decoder output also a by appropriate this thing. Is it fine right.

Now, we have seen this Milli model, More model and state transition diagram, ASM chart, realization using standard simplification following some standard simplification or by using this kind of arrangement where decoder can be used to generate the mean terms.

(Refer Slide Time: 18:04)



State Minimization

Present state	Next state $X=0$	Next state $X=1$	Present output $X=0$	Present output $X=1$
a	a	b	0	0
b	c	d	0	0
c	e	f	0	0
d	b	a	0	1
e	c	d	0	0
f	b	a	0	1

Two states are considered equivalent if they move to same or equivalent state for every input combination and also generate same output.

Now, we look at another aspect of this synchronous sequential circuit design which is called state minimization. This is important because often from the problem statement, the a state definitions that we are doing and then arriving at the state transition diagram or the corresponding ASM chart, it could be so that we have actually considered more number of states than necessary. Is not it? If you remember, the way we have defined the states in the sequence detection problem.

So, a knows beta has been detected; properly b 1 beta has been detected and so on and so forth. So, that is from the intuition that is from our understanding of the problem statement, you are d doing it right ok. Similarly, for vending machine problem you are looking at; no money has been deposited, then rupees 5 has been deposited, rupees 10 has been deposited; these are different states ok. So, we made a logical argument of the case and defined these those states right.

So, then will be more complex problems, more difficult to understand right and logically we may unfurl it and get those states; but it is not necessary that those states to begin with or the most minimized one or we as an implementer, somebody gives us a state transition diagram, we have been asked to implement it ok. We can look for whether there is an redundancy and then, if we detail you know can come up with minimized just you know state representation that will help us in reducing cost, reducing time and many other things associated with it ok.

The time complexity of the design and many other aspects that will not be there; is it fine. So, remember the minimized state transition diagram or state machine will be giving you same input output relationship without any you know difference. So, it will not alter the input output characteristics, but only the internal states that are required to generate the circuit, generate the corresponding output in reference to a particular sequence of input that only is getting changed that is only getting reduced ok; is it fine?

So, with this background let us see an example and learn through that particular example how it is done. So, we shall take up one method today and in the next class we will take up two more methods; two more popular methods. So, this is one state transition diagram ok, that has been obtained from some source right and then, we have an job. We have a job to see it whether it can be minimized or not.

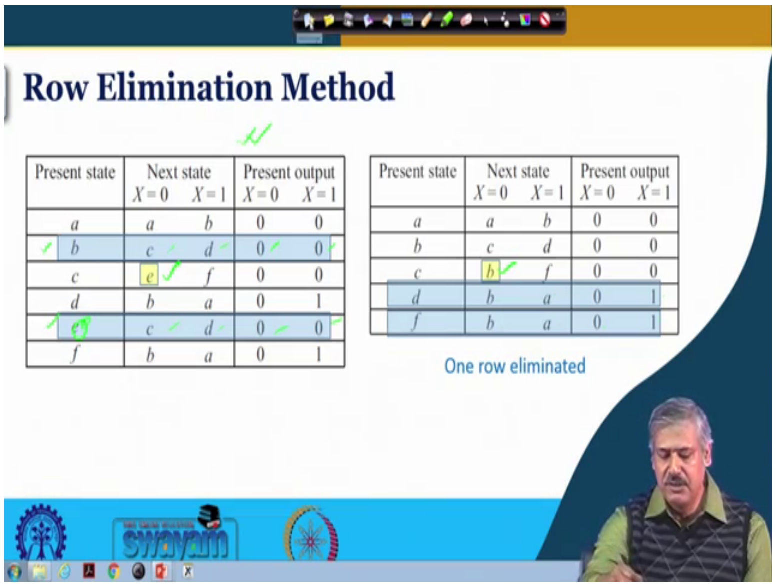
So, we see that a b c d e f 6 states are there and for that the first method that we shall imply we will call is called row elimination method. So, to before that we first understand that two states are considered equivalent; two states are considered equivalent if they move to same or equivalent state for every input combination and also generates same output ok.

So, that is when we shall consider two states are equivalent ok, when they are moving to same state or equivalent state. So, that equivalents we have to see and generate same output ok. So, for that this state transition diagram, we will first move to a corresponding a tabular form ok. So, this is the present state a b c d e f right. So, next state for input on only 1 input is used; this is the Milli model we can see.

So, X is equal to 0 and X is equal to 1 ok. So, the corresponding next state are for a, you can see for X is equal to 0, this is a, for X is equal to 1, this is b ok. What are the corresponding outputs from a, if input is received is 0, output is 0. Input received is 1, output is still 0 ok. So, that is how we get the first row of this table.

Similarly, b we can see; b it is going to c and d, c and d and in each time the output is 0 and 0. For c, it is going to e and f e and f output is 0 0. For d, it is going to b over here and a over there. b over here a over there. Going to b, output is 0 and going to a, output is 1 fine. e it is at e, it is c and d; c and d that is what you can see output is 0 in both the cases. For f, it is going to b and a; b and a right and output is 0 over here and 1 over there; is it ok. So, this is the way to map it in a manner which for which is easier for us to apply the equivalence relationship. Hence, investigate examine redundancy fine right.

(Refer Slide Time: 23:55)



Row Elimination Method

Present state	Next state $X=0$	Next state $X=1$	Present output $X=0$	Present output $X=1$
a	a	b	0	0
b	c	d	0	0
c	e	f	0	0
d	b	a	0	1
e	c	d	0	0
f	b	a	0	1

Present state	Next state $X=0$	Next state $X=1$	Present output $X=0$	Present output $X=1$
a	a	b	0	0
b	c	d	0	0
c	b	f	0	0
d	b	a	0	1
f	b	a	0	1

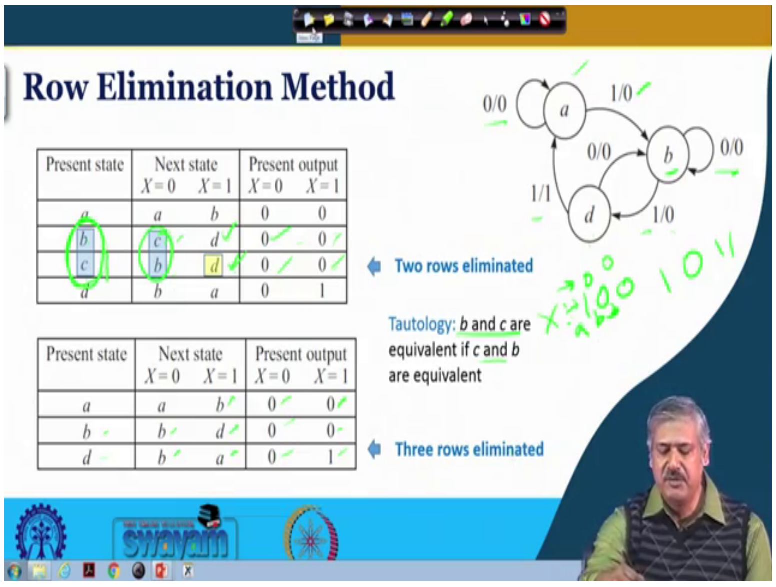
One row eliminated

So, once we have put there and we employ that the particular concept of equivalence, we can see b and c right. So, next state is c and d; next state is c and d output is 0 0 0 0. So, we can say b and e are equivalent; b and are equivalent and we can eliminate one of them. So, that is called one row eliminated. So, the larger you know the one that comes in later in the order. So, e has been removed and b has been retained ok.

And in this table, in this representation wherever a reference of e was there; for example, here since b and e are equivalent, we are replacing it with b; is it fine. So, this is what we have done. One row has been eliminated right and d one that has been eliminated with the equivalent one; reference to that particular state is made with the one that is eliminating, that is b over here.

So, now, we can further investigate and see, that d and f; next state is b and a; b and a and output is 0 and 1, 0 and 1; in each of these cases, it is matching for the corresponding input. So, we can say d and f are equivalent ok. So, we shall return one of them and remove one of them ok. So, f will be removed because it comes later in the order. So, that is what we shall do move to the next slide.

(Refer Slide Time: 25:40)



Row Elimination Method

Present state	Next state $X=0$	Next state $X=1$	Present output $X=0$	Present output $X=1$
a	a	b	0	0
b	c	d	0	0
c	b	d	0	0
d	b	a	0	1

Two rows eliminated

Tautology: b and c are equivalent if c and b are equivalent

Present state	Next state $X=0$	Next state $X=1$	Present output $X=0$	Present output $X=1$
a	a	b	0	0
b	b	d	0	0
d	b	a	0	1

Three rows eliminated

So, that has been done right. So, before that wherever the reference of f was there. So, here was a reference of f. So, f has to be removed with d, so, f has been removed with d. You can see that f has been removed with d ok. After that if you further examine as such there is no such you know two rows where you know exactly some such thing is there. I mean the next state and output are exactly the same, but one thing you can see that b state b it goes to c and d 0 0; c it goes to b and d 0 0. So, this d; this 2 d matches 0 0 matches; 0 0 matches.

So, only where in this place c and b are not matching. So, had they matched? So, if c and b were equivalent, then we can say that this b and c are equivalent, b and c at the inputs I mean the current present state slide present state side. So, next state c and b are equivalent, if present state b and c are equivalent. So, this is from Tautology we can say b and c will be equivalent because the (Refer Time: 27:01) requirement is c and b are equivalent and we are looking for equivalence ok, redundancy and minimizing it to the extent possible.

So, we retain one of them and eliminate the other. So, it becomes 3 rows now; c is removed and wherever reference of c was there, we are replacing that with b right. So, here you get replacing that with b and if we examine we can see that no further elimination is possible. So, now, we have got 3 states a, b and d. a b and d. So, from 6 states we have come down to 3 states by this method and now, if you convert it to the state transition diagram if so required ok. So, you can see that this is a 0 it is remaining at 0 and output is 0. So, this is the one. For X is equal to 1, b and output is 0; it is going to be b output is 0. Then b at 0, it is remaining at 0 ok; output is 0, 1 output is 0 and going to d; 1 output is 0 and going to d.

At d it is for X is equal to 0, it is going to b and output is 0; going to b output is 0 and for 1, it is going to a and output is 1. So, it is going to a and output is 1 right. So, this is the reduced state transition diagram and you can give any inputs sequence random sequence of X say 1 0 0 0 0 1 etcetera etcetera right and you can examine if the constant is a and next state is what will be the next state? It will be the next state will b. And what will be the corresponding output? It will be the 0 right. Then 0, it next state will be b output will be 0.

So, that way you go on doing it, you will find for the previous one and the current one or any you know random sequence that you have taken. So, these output will remain the same; only the next state values will be different ok. Because of the equivalence that are there ok. So, these input output relationship does not alter because of this and of course, you will be realising with less number of flip flops and as I said less complexity, design complexity and many other things how does cost and so, that is the advantage of state elimination. More methods, other methods we shall see in the next class.

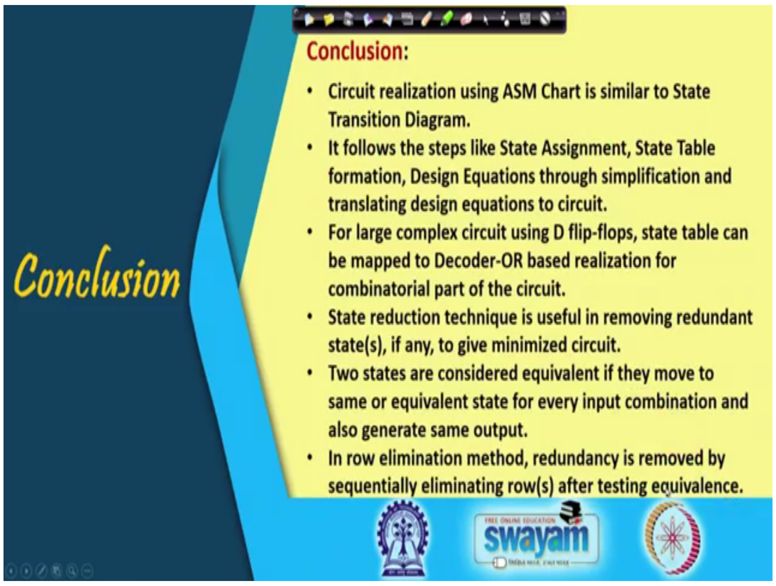
(Refer Slide Time: 29:47)

References

References:

- Donald P. Leach, Albert P. Malvino, and Goutam Saha, Digital Principles & Applications 8e, McGraw Hill

(Refer Slide Time: 29:47)



Conclusion

Conclusion:

- Circuit realization using ASM Chart is similar to State Transition Diagram.
- It follows the steps like State Assignment, State Table formation, Design Equations through simplification and translating design equations to circuit.
- For large complex circuit using D flip-flops, state table can be mapped to Decoder-OR based realization for combinatorial part of the circuit.
- State reduction technique is useful in removing redundant state(s), if any, to give minimized circuit.
- Two states are considered equivalent if they move to same or equivalent state for every input combination and also generate same output.
- In row elimination method, redundancy is removed by sequentially eliminating row(s) after testing equivalence.

So, to conclude circuit realization using ASM chart is similar to state transition diagram method ok. It follows steps like state assignments, state table formation, design equation through simplification method Carno map and finally, translating design equations to circuit. And for large complex circuit, we can have seen that it may make more sense to convert the state table to a Decoder-OR based realization directly from the state table itself and the combinatorial part of the circuit.

And state reduction technique is useful for removing redundant states and two states can be considered equivalent, if they move to same or equivalent state for every input combination and also generate same output. So, both that required and in row elimination method, redundancy is removed sequentially by eliminating rows after testing equivalence ok.

Thank you.