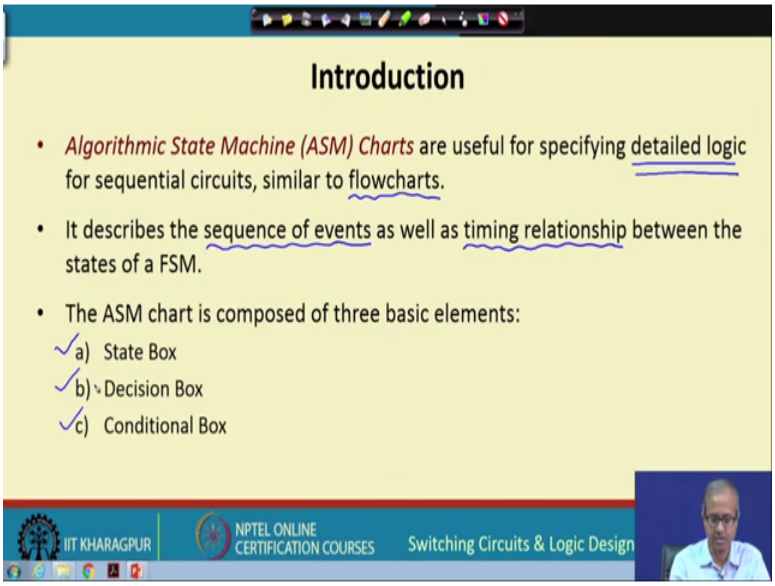
Switching Circuits and Logic Design

Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 54
Algorithmic State Machine (ASM) Chart

In this lecture, we shall be talking about something called algorithmic state machines or ASM charts which you sometimes call, which is a very useful tool for designing sequential circuits, particularly complex circuits. So, the lecture title is algorithmic state machine or ASM chart.

(Refer Slide Time: 00:41)



Introduction

- *Algorithmic State Machine (ASM) Charts* are useful for specifying detailed logic for sequential circuits, similar to flowcharts.
- It describes the sequence of events as well as timing relationship between the states of a FSM.
- The ASM chart is composed of three basic elements:
  - ✓a) State Box
  - ✓b) Decision Box
  - ✓c) Conditional Box

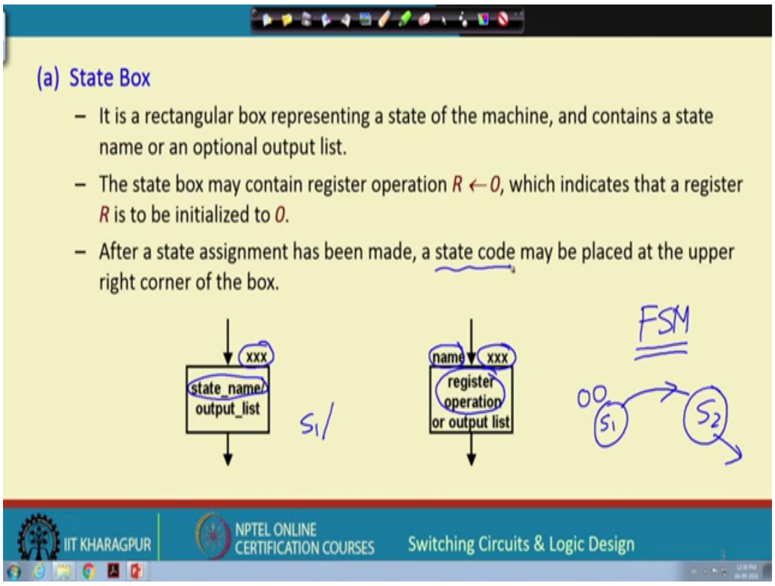
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Let us see what an ASM chart is. But this ASM chart is some kind of a tool you can say, it is a pictorial tool just like a flow chart or a state transition diagram that we use to represent finite state machines something similar, but it captures a lot of information with respect to the FSM we will see. This is quite useful for specifying detailed logic for sequential circuits which I said is somewhat similar to flowcharts.

Now, it specifies two things together. It specifies sequence of events that is reflected in a sequential circuit behavior; also it reflects timing relationship between the various states of the finite state machine. Now, in a finite state machine you recall there are two kinds of pictorial elements, one we represent by circles the states and the transition represented by arrows. Now, along the transition we specify the inputs and the expected outputs.

Now, in an ASM chart there are three kinds of elements which are shown. We shall see this things state box, decision box and conditional box. Let us see the function of these.

(Refer Slide Time: 02:25)



(a) State Box

- It is a rectangular box representing a state of the machine, and contains a state name or an optional output list.
- The state box may contain register operation $R \leftarrow 0$, which indicates that a register R is to be initialized to 0.
- After a state assignment has been made, a state code may be placed at the upper right corner of the box.

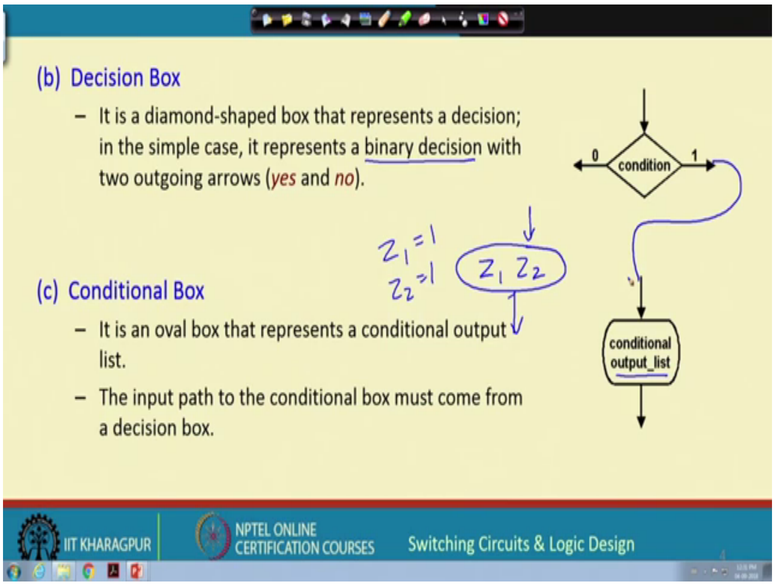
IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Switching Circuits & Logic Design

First the state box. State box is a rectangular box as is shown in this example here which represents the state of an machine. Well, if you think of an FSM a finite state machine where you represent states like this let us say S 1 S 2 and so on and arrows indicating transition from one state to another. So, each of these states is represented in ASM by a state box.

Now, in this state box, you have a state name either you show this state name inside with a slash like you can write S 1 slash. In the output optionally you can mention the output lists or the name you can also show outside the box on the left, there are several conventions, you can use any one of them. And inside the state box you could also contain some register operations to initialize the values of registers like here you can have some register operations optionally. And each of these state boxes can have some state assignment.

Like you see when you specify a finite state machine, you only mention the states S 1, S 2, S 3 and so on, but you think that when you synthesize a machine, when you in the process of implementing it in hardware. You also carry out some state assignment like you may say that this state S 1 will be using this code 0 0. So, in the state box, the state assignment is also specified typically in the upper right corner of the box. This we are calling as something called state code. Well, we will take some example to show how the state boxes are created, but this is how or these are the information it contains. And there can be optional output list also.

(Refer Slide Time: 04:56)



(b) Decision Box

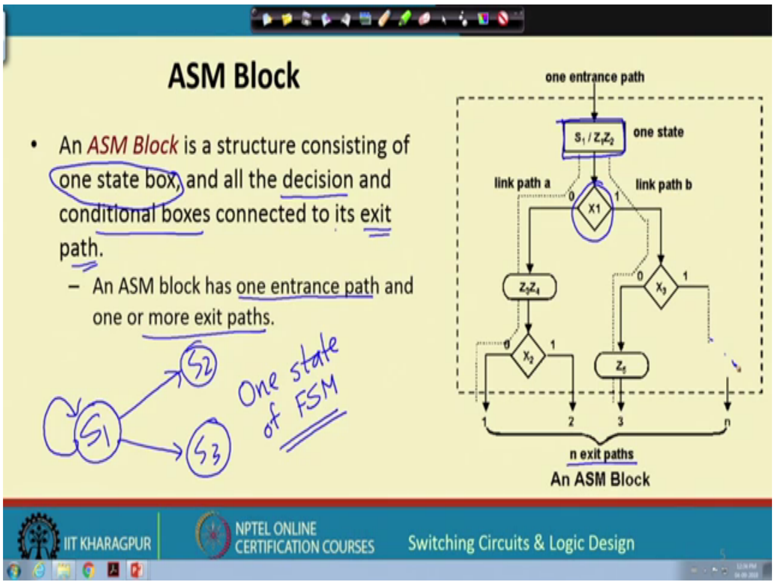
- It is a diamond-shaped box that represents a decision; in the simple case, it represents a binary decision with two outgoing arrows (*yes* and *no*).

(c) Conditional Box

- It is an oval box that represents a conditional output list.
- The input path to the conditional box must come from a decision box.

Then comes the decision box. Just like a conditional box in a flowchart, where if it is true, you go here; if it is false, it comes here. It is a diamond-shaped box, but it can be more general. So, when instead of two outcomes, there can be multiple output nodes depending on how many possible conditions can be there. So, in the simplest case, it came a binary decision as is shown in this diagram yes or no, or 1 or 0 right. And conditional box is a last one; it is an oval shape in box, which contains some output list. Like suppose I create an oval shape in box I mention Z 1, Z 2; it means that these 2 outputs Z 1 and Z 2 must be set to 1. And this input arc for the condition input list must be coming from a decision box maybe from the decision box, it will be coming right, these are the constraints.

(Refer Slide Time: 06:20)



ASM Block

- An *ASM Block* is a structure consisting of one state box, and all the decision and conditional boxes connected to its exit path.
  - An ASM block has one entrance path and one or more exit paths.

One state of FSM

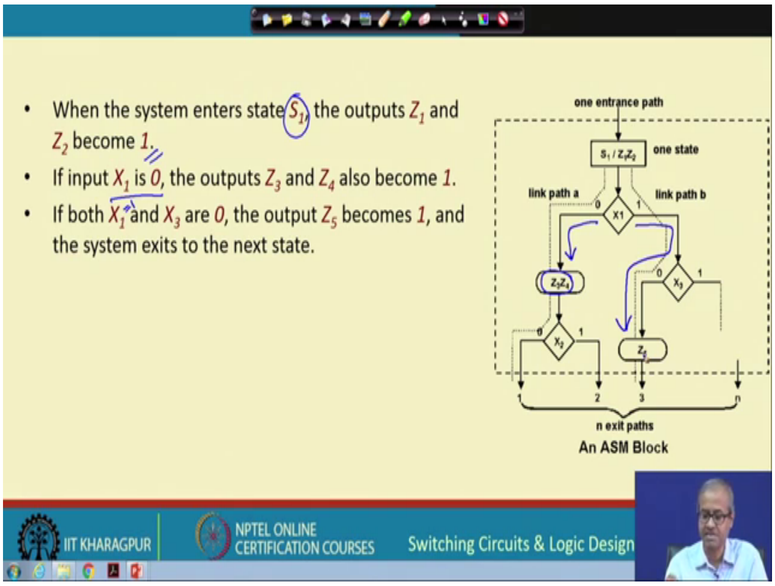
Let us look at a somewhat complete kind of an ASM block that how it looks like this is an ASM block. First thing is that what is an ASM block? ASM block is something that corresponds to one state of the corresponding finite state machine. So, every state of the FSM let us say there was an state S 1, it was going to some state S 2, going to a some state S 3, maybe there was a self loop depending on input combinations. So, every state will be mapped to an ASM block, ASM block is a structure which will be consisting of one state box corresponding to that state.

In this diagram that state box is shown here. So, you see here this state name is given as S 1 with an optional output list Z 1, Z 2; it says that these are the output list of the state box. Now, this you may give, may not give this is optional. Now, inside the state box there can be decision boxes and conditional boxes connected to it is so called exit path. There can be more than one exit path as is shown here; there can be n exit path in general 1, 2, 3 up to n. An ASM block will have exactly one in turn entry path or entrance path, but it can have multiple exit paths.

Now, inside it what kind of things you can have? You can have a decision box like this. Like you check X 1, if X 1 is 0 false you come here; if X 1 is true, you come here which means, if X 1 is false you set Z 3, Z 4 to 1. Then check X 2, if it is 0 you follow this exit path; if it is 1, follow this exit path. Then along this path you can check the condition X

3; if it is 0, you said Z 5 to 1 and exit through 3, if it is 1 you continue with something else. This is how generally an ASM block looks like.

(Refer Slide Time: 09:12)



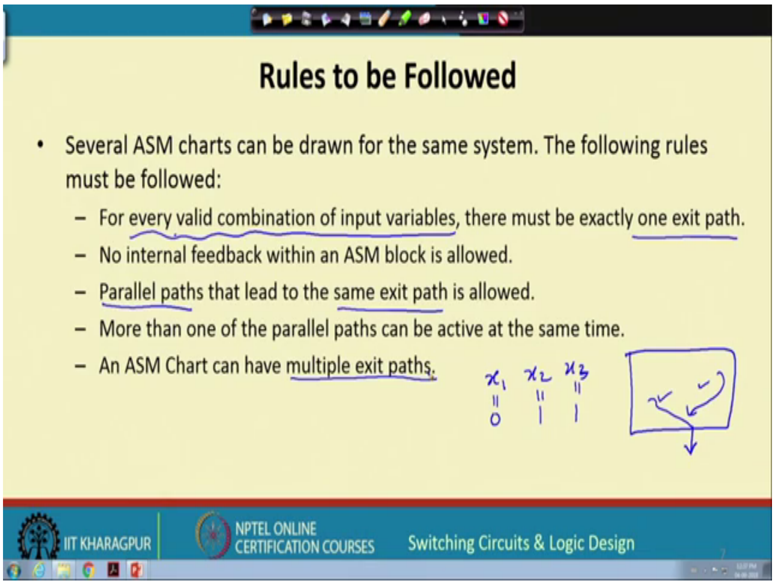
- When the system enters state S_1 the outputs Z_1 and Z_2 become 1.
- If input X_1 is 0, the outputs Z_3 and Z_4 also become 1.
- If both X_1 and X_3 are 0, the output Z_5 becomes 1, and the system exits to the next state.

An ASM Block

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Switching Circuits & Logic Design

So, let us explain this that same example. So, when this system enters this state S 1 from the entrance path, first we have mentioned the two outputs Z 1 and Z 2 here means the output Z 1 and Z 2 are immediately set to 1. Then condition X 1 is checked. If X 1 is 0 which means this path Z 3 and Z 4 there also set to 1; but if X 1 is both X 1 and X 3 are 0 like say both X 1 is 1, this should 1 actually X 1 is 1 and X 3 is 0 you follow this path, then Z 5 becomes 1 and the system exits.

(Refer Slide Time: 10:16)



Rules to be Followed

- Several ASM charts can be drawn for the same system. The following rules must be followed:
  - For every valid combination of input variables, there must be exactly one exit path.
  - No internal feedback within an ASM block is allowed.
  - Parallel paths that lead to the same exit path is allowed.
  - More than one of the parallel paths can be active at the same time.
  - An ASM Chart can have multiple exit paths.

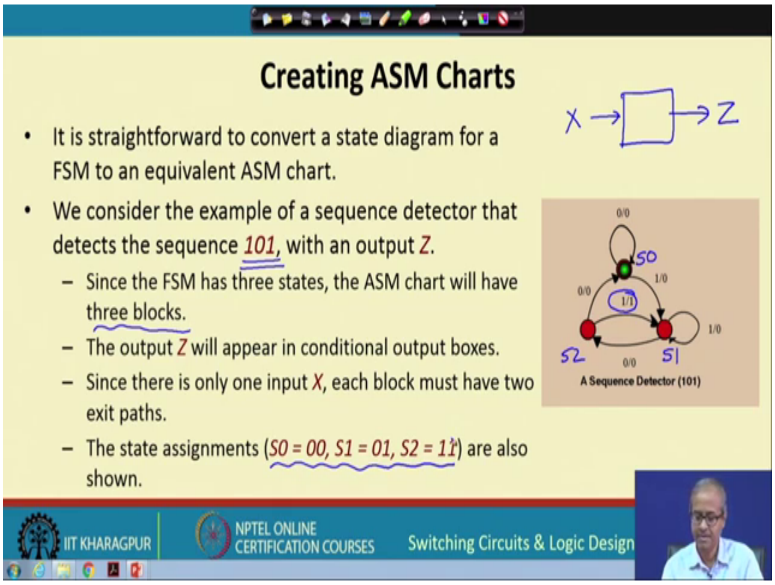
$x_1 = 0$ $x_2 = 1$ $x_3 = 1$

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Now, there are some rules you need to follow when you are creating an ASM chart, you should remember this. For this same system specification, you can have multiple possible ASM charts, you can draw again several different ways, but certain rules must be followed. Like the first one says that for some, for every valid combination of input variables, like let us let us take an example there are three input variables x 1, x 2, x 3, let us say x 1 is 0, x 2 is 1, and x 3 is also 1 0 1 1. For every combination there must be exactly 1 exit path, that means, in that ASM block for this 0 1 1 combination, you must be following the same exit path, there should not be multiple exit paths.

And inside the ASM block, internal feedbacks are not allowed; it should be only from top to bottom kind of transitions. Parallel paths to the same exit paths are allowed, there can be multiple paths going to the same exit path. More than one parallel path can be active at the same time; this can be active, this can also be active such things are allowed. And as I showed in the previous diagram there can be multiple exit paths. So, such rules should be followed when you are drawing or creating an ASM chart.

(Refer Slide Time: 11:58)



Creating ASM Charts

- It is straightforward to convert a state diagram for a FSM to an equivalent ASM chart.
- We consider the example of a sequence detector that detects the sequence *101*, with an output *Z*.
  - Since the FSM has three states, the ASM chart will have three blocks.
  - The output *Z* will appear in conditional output boxes.
  - Since there is only one input *X*, each block must have two exit paths.
  - The state assignments (*S0 = 00, S1 = 01, S2 = 11*) are also shown.

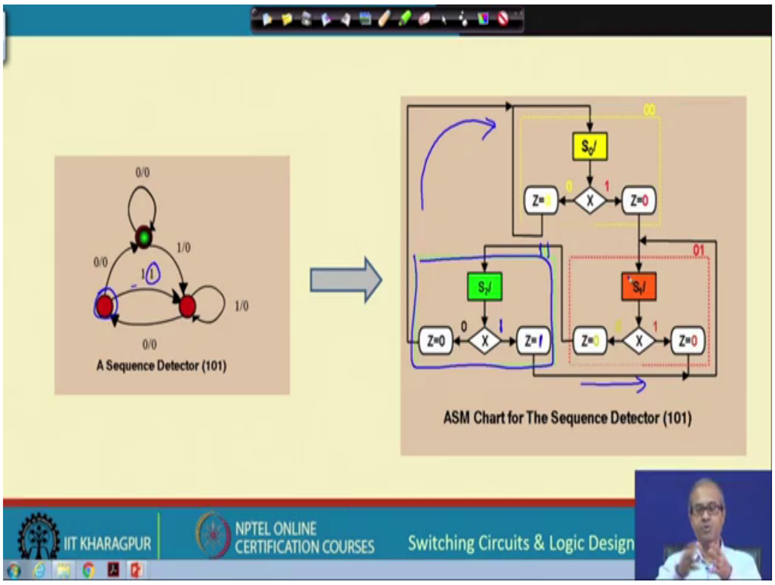
X
Z
0/0
S0
1/0
0/0
1/1
1/0
S2
0/0
S1
A Sequence Detector (101)
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Let us take a small example. So, here we shall show how an FSM can be converted into an equivalent ASM chart. So, we start with a simple FSM description of a sequence detector example, a sequence detector which detects a sequence 1 0 1 in the input sequence. Just look at this state diagram this is the starting state the top one. So, if the input is 1, you move to this state.

So, if the input is again 0, you move to this state. And if you get a third one, you move back to this state with the output 1 that means, you have found out this sequence. Now, this third one means there can be overlapping sequence also like this. The third one means, you have detected a 1 0 1 here, but it can be the start of the next 1 0 1. So, you move back to this state, so there can be another 0 another 1, so this second sequence can be detected again right. So, from here, if it is 0 that means, you have to start from the beginning you go back here right. Now, let us see.

Here once you are trying to detect this sequence 1 0 1, so you have an FSM, where I am assuming there is a single output Z, and a single input X. This X denotes the serial bit stream, and Z is the final output 0 or 1 right. So, the output will be 1 only when 1 0 and 1 is detected right. There are three states in this FSM; this will correspond to three ASM blocks. And let us also make the state assignments. Let us say this state is S 0; this is our S 1; this is our S 2. Let us make this state assignment also 0 0 0 1 and 1 1. Let us say. So, let us show how the ASM chart may be constructed from this FSM.

(Refer Slide Time: 14:39)



A Sequence Detector (101)

ASM Chart for The Sequence Detector (101)

So, here some of these symbols are not visible, I shall show them. Let us again talk about the three state, this is S 0, this is S 1, and this is S 2. So, one by one, let us see. This is the ASM block corresponding to state S 0, this is state S 0. So, the state encoding is also shown here 0 0. So, the output list is default, it just shows the state name S 0. So, you look at this state is the what it does, if the input is 0 it remains in S 0; and if the input is 1, it goes to S 1 in both cases the output is 0.

So, you see if X is 0 left side, you said Z equal to 0, because output is 0. And you go back to this same state, you remain in S 0. But if it is 1, then also Z is 0. And you go to this transition you go to S 1, you move to S 1 right. This is your state S 0, so exactly similar to here. Now, let us look about state S 1, this is your state S 1, your state encoding is shown here 0 1, this state name is mentioned. Now, in this state S 1, if it is 0, it goes to S 2; if it is 1 it remains in S 1 so, let us see. So, if x is 0, then you set output to 0; and you move to state S 2. But if it is 1, then also output is 0; and you remain in the same state remain in state S 1. So, exactly from the FSM you are mapping here.

Now, the last state S 2, this is your S 2 and this is the state encoding 1 1. Now, in S 2 this is your S 2. If it is 0, you go to S 1, you go to S 0, if it is 1, you go to you go to S 1. And here the output is set to 1. Let us see what you have done, if input is 0, you go to Z 0 and you move to here state S 0. But, if the input is 1, you set Z equal to 1 and you move here to state S 1.

So, you see exactly from the FSM, you can have a one-to-one mapping to the corresponding ASM chart. Now, you may ask that why do you need this ASM chart, the FSM was good enough right, but you see this ASM chart is one step forward towards implementing the hardware circuit, implementing the circuit, because you have also done the state assignment. Normally from the ASM chart you can directly design the hardware circuit. So, each state box will correspond to a flip flop and after that some gates you can add you can directly implement. So, it is really easy to map it to hardware that is why sometimes from the FSM, you create the ASM chart from the ASM chart you can create the hardware.

So, in this lecture we have given you a very brief introduction to ASM chart. As it said ASM chart is useful for the complete description of our system, which is more powerful than an FSM, because it can also capture the state encoding. And particularly for more complex system for larger FSMs, this ASM chart is sometimes considered to be an useful tool. So, if you are a system designer, if you are using these tools for designing complex systems, well FSM is of course, one way and you have already seen earlier starting from an FSM, how you can formally synthesize a sequential circuit, but ASM is another route just using ASM also you can generate a hardware. Of course, your hardware may not be that efficient, but for more complex systems it is a fairly simple approach to do that.

So, we have discussed, means almost all the aspect different aspects that you wanted to discuss regarding digital circuit design and synthesis particularly at the gate level and the flip flop level. We talked about synchronous circuits; we also talked about asynchronous circuits. Now, in the next few lectures we shall be discussing on something which is slightly different like once you design the circuits, once you fabricate the circuits in the form of chips, there can be some errors in design, there can be some defects, which can occur during the implementation. So, we need to test our circuits, how to test, what are the different ways of testing these are a few things we shall be discussing during the next set of lectures.

Thank you.