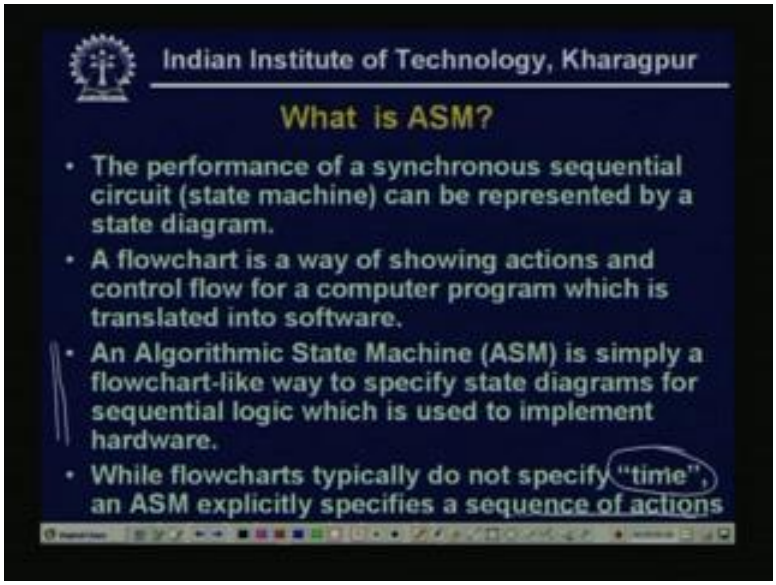
**Digital System Design**
**Prof. D Roychoudhury**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kharagpur**

**Lecture - 32**
**Algorithmic State Machines Chart**

Today, we will read a new type of flowchart to implement the digital hardware called the Algorithmic State Machines Chart or simple ASM Chart.

(Refer Slide Time: 01:02)




So, first we will see what is ASM, what do we mean by algorithmic state machine chart, before that we again we recapitulated the synchronous sequential design. The synchronous circuit, normally it is made up of the combinational circuit with the memory and the memory is a clocked flip flops. Now, the performance of synchronous sequential circuit or simply we can tell the state machine can be represented by a state diagram.

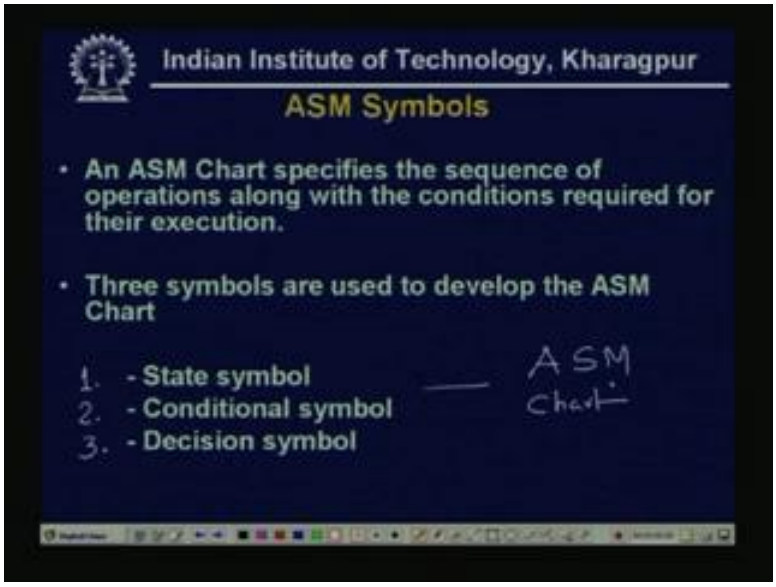
Now, this performance can also be represented by a flowchart, a systematic way, a flowchart is a way of showing actions and control flow for a computer program, which is translated into software. Now, similarly like using the flowchart or what do we mean by software flowchart and the performance of synchronous sequential machine can also be represented by a flowchart and this is called the algorithmic state machine chart.

So, again it is a flowchart we can tell that this is a hardware flow chart, so an algorithmic state machine is simply a flowchart like way to specify state diagrams for sequential

logic, which is used to implement hardware. So, while flow charts, typically do not specify time and ASM explicitly specifies a sequence of actions and their timing relationships, so what we can tell that ASM, this is the basic definition of ASM.

This is in broad sense, this is a hardware flow chart and the difference from our normal software flowchart, that our software flowchart does not specify time, whereas the ASM explicitly specified a sequence of operations. So, this is the difference from a software flowchart or the normal flowchart from the ASM.

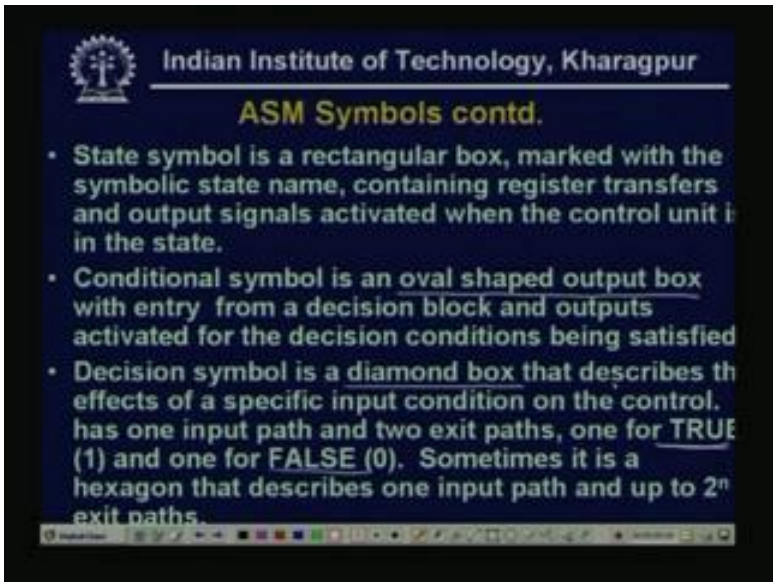
(Refer Slide Time: 03:50)




Now, normally ASM, just like our normal flowchart, ASM have some symbols, so an ASM chart specifies the sequence operations along with the conditions required for their execution. So, here the ASM is used to design a digital system for the problem or the for the whole digital system design, there are some the sequence of operations and this ASM chart specifies this sequence of operations.

Now, normally three symbols are used to develop the ASM chart one is called the state symbol, then the conditional symbol and the decision symbols. So, here only three symbols are used for ASM flowchart or ASM chart.

(Refer Slide Time: 05:09)

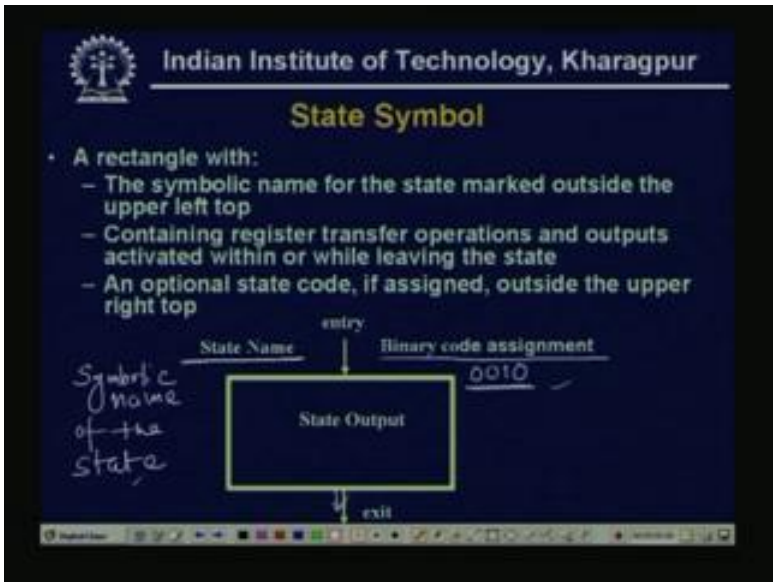



Now, state symbol is a rectangular box marked with the symbolic state name, containing register transfers and output signals activated, when the control unit is in the state. So, this is the rectangular type of box, normally it has some name and some binary coded value is given to the box. Second is the conditional symbol, normally this is a oval shaped output box with entry from a decision block and outputs activated from the decision conditions being satisfied.

So, normally it is a decisional execution or if the condition is satisfied then this decision should be taken, so this conditional symbol is a oval shaped output box. Now, the third the decision symbol is a diamond box, that describes its effects of is specific input condition on the control, that means if this condition is satisfies. Then, this decision is taken means this operation should be executed.

So, it has 1 input path and 2 exit paths 1 for true as it is a decision, so 1 for true and another for false, sometimes it is a hexagon, that describes 1 input path up to 2 to the power n exit paths or more than 1 a bus input and a bus output, we can tell. So, these decision symbols, normally it is a diamond shaped, but sometimes it can be a hexagonal, so these are the three symbols, the state symbols, the conditional symbols and the decision symbols are being used to represent a ASM chart.

(Refer Slide Time: 07:31)

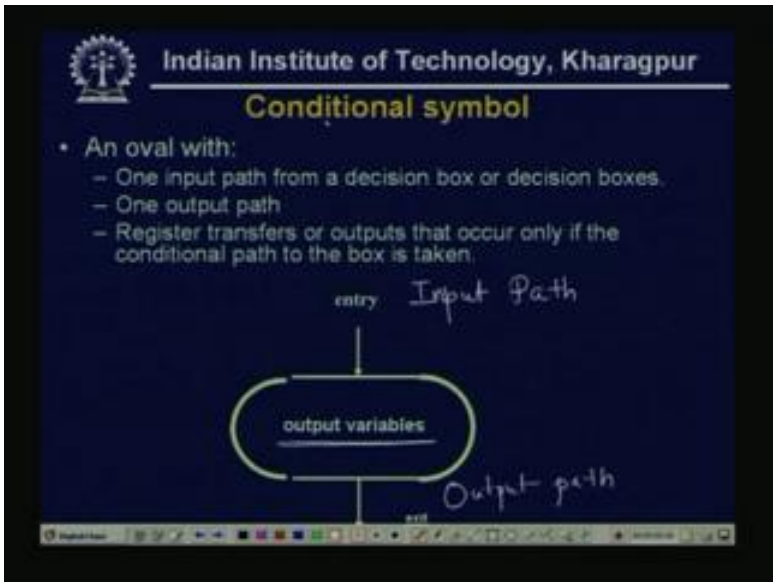



Now, first receive the state symbol, it is a rectangular shaped, now here one state name should be there and the binary code assignment, say if it is a 4 bit say 0 0 1 0, this type of binary code is assigned here. It has one entry and one exit, normally this is the state output, so it is a rectangle with the symbolic name for the state marked, outside the upper left top.

So, statement or we can call the symbolic name of the state, all or sometimes, we can call the logical name, symbolic name of the state. The convention is normally this name is written on the left top. Now, it contains register transfer operations and outputs activated within or while leaving the state and that is the exit point, this is the exit and optional state code, if assigned outside the upper right top.

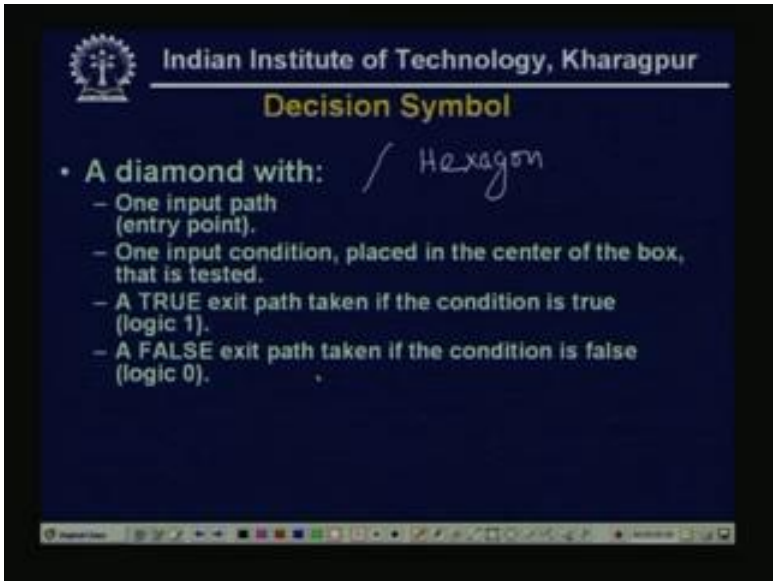
So, this is the binary coded assignment, so this is the logical name and when it is assigned binary, just like our state diagram. So, it is given to the binary coded assignment, so this is our state symbol.

(Refer Slide Time: 09:16)



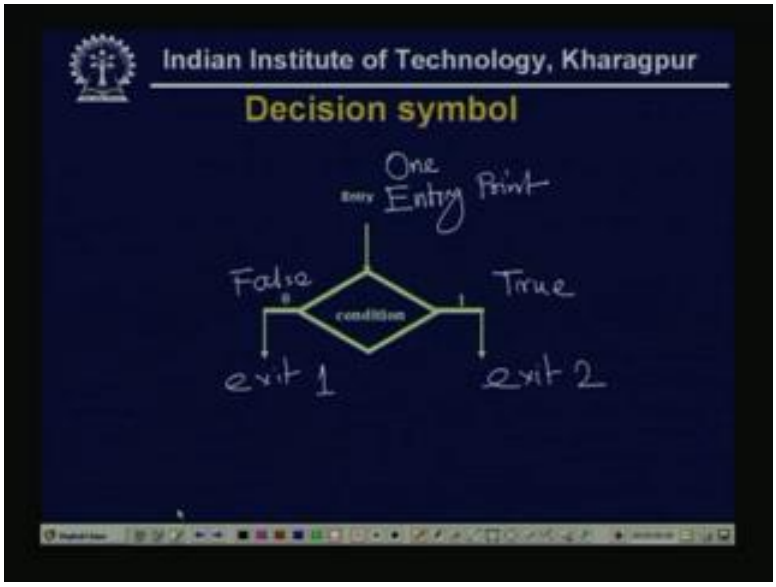

Then the second 1 the conditional symbol, it is a oval shaped box, 1 input path from a decision box or decision boxes, so entry is the input, this is the input path, this is the output path and this is the conditional. So, register transfers or outputs, that occur only if the conditional path to the boxes taken, so these are my initial is the output variables. That means, if it is the condition to be checked, that is the value of the output variable that is there, so this is my conditional symbol.

(Refer Slide Time: 10:20)



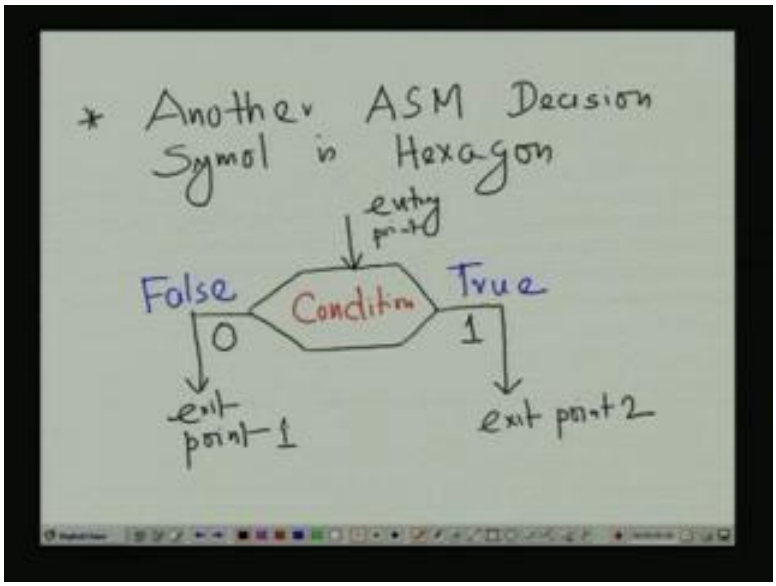
Now, the third is the decision symbol, normally it is a diamond shaped box, sometimes this can be a hexagon, so this is 1 input path, the entry point, 1 input condition placed in the center of the box, that is tested, because it is a decision. A TRUE exit path taken, if the condition is true as it is a decision, so if it is true, the condition is true, then it follows a path, normally call the true path, if the condition fails, then it follows the different path, normally it is called a false path. So, that is the condition is false, a FALSE exit path, so there are one entry point, two exit point.

(Refer Slide Time: 11:17)



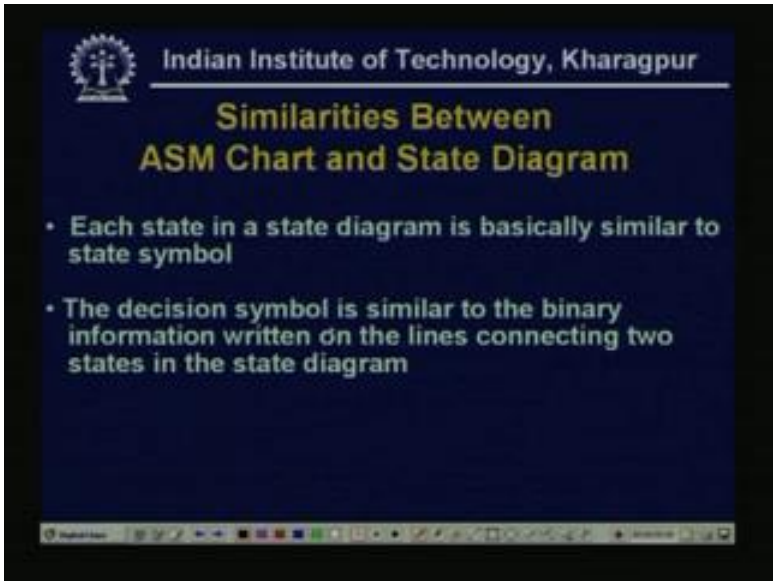

Now, symbol is here this is the one entry point and there are two exit, so if the condition is false, then this is one exit 1, if the condition is true, then it is a exit 2. So, there are two exit point, sometimes another decision symbol is used that is a hexagon.

(Refer Slide Time: 12:09)




Normally, another type of ASM decision symbol is hexagon, normally 1, again, here one entry point and two exit point, this is the condition, if it is false a 0, if it is 1, means true. So, this is the entry point and there are two exit points, so normally these are the three symbols, ASM symbols are used to represent the ASM chart.

(Refer Slide Time: 14:11)

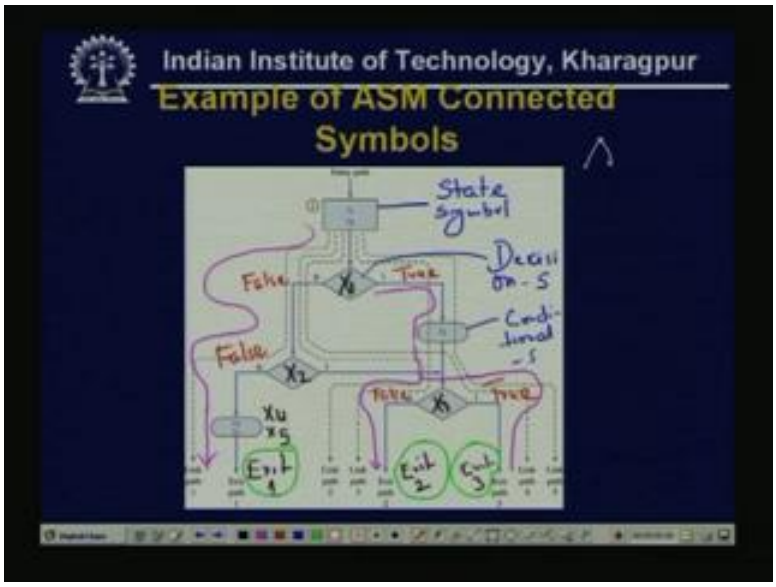



Now, what are the similarities between ASM chart and the state diagram, so each state in a state diagram is basically similar to state symbol, what we have seen the state symbol has also a name, just like our state diagram; a logical name is there. So, state in state

diagram is nothing but the state symbol and the decision symbol is similar to the binary information's, written on the lines connecting to states in the state diagram.

So, decision symbol is the binary information, normally what output it will generate, what will be the value, what value it will transfers. So, that information written on the lines connecting two states in the state diagrams that is nothing but our decision symbols in a ASM chart, so these are the similarities.

(Refer Slide Time: 15:22)




Now, this a simple example, that how the one ASM chart is can be constructed using the three type of symbols, so here we have taken one state symbols, l one entry point is here, so this is one state symbol. Now, this one is a decision symbol, this is a decision symbol, if it is false, if the decision symbols is written as a say X 1 the value. If it is false, then it is the entry point of another decision symbols say called X 2.

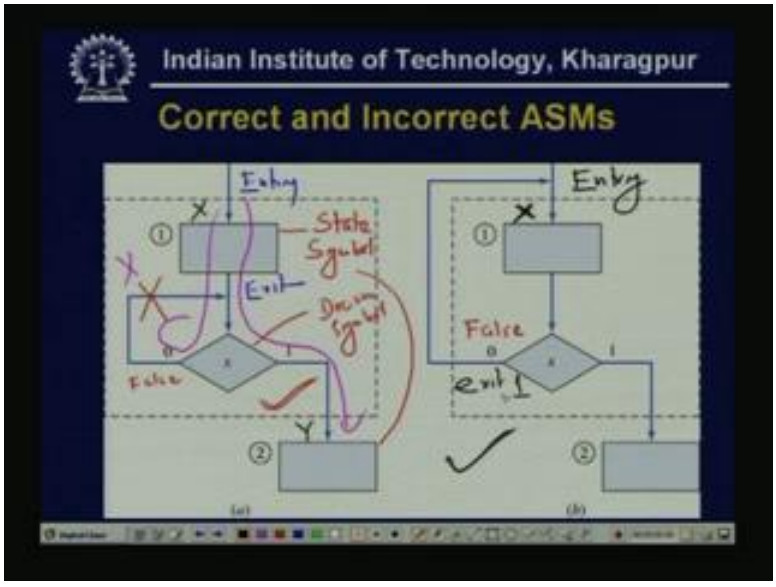
Again, if it is false, then it goes to the conditional output symbol and this is the conditional output symbol, the oval shaped or the output variables are say X 4 and X 5 and this is it has 1 exit path. So, this path we have consider only the left path, means false path, we have we are check the false path, as if this is the false path. Now, again if we consider that the entry path, from the entry path to the first conditional symbol and if it is true, then it will follow the true path or another exit point 2.

It will directly goes to a conditional symbol, so this is one conditional symbol and it has one exit point and it goes to a decision symbol called X 3. Now, again if it is false, then another exit path, it follows if it is true, then another exit path true, so we can tell if we see that how many exit paths are there for this ASM charts. We have seen a false path, so if it is a false false path, then this is only 1 exit path 1, this is a exit 1.

Now, if we consider the true path, then this is first, it is a true path, then this is a conditional symbol, it has only 1 exit path, now it goes to a decisional symbol, so it has 2 exit path, 1 false, so we call this is the exit 2, this is another exit point. Now, if it is a true, it was coming like that, then it is a false path, it follows exit 2 and if it is a true path, then the total true, it will go to a exit path 3, so this is one exit 3. So, there are 3 exit paths of this ASM chart.

So, normally in this way the design is represented by a special type of flowchart consisting of three types of symbols, the state symbols, the decision symbols and the conditional symbols. This is the one example of the ASM connected symbols or the we can tell that, this is the ASM chart, one example of a ASM Chart.

(Refer Slide Time: 21:32)



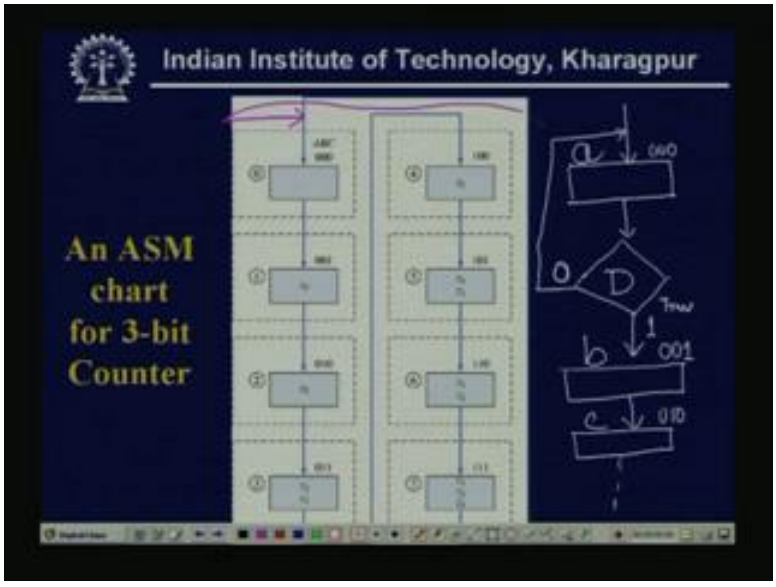

Now, the some correct and incorrect ASM, so here just arbitrary, we have taken some ASM charts, consisting of two type of symbols only, say this is one state symbol, say this is the logical name is X or say the name is the state symbol and it has 1 say x. Now, 1

one entry point and one exit point, it feeds to a decision symbol, it goes to a decision symbol, if it is false, then it cannot feed the same point, this is a incorrect ASMs.

This path is correct, if it is true, the true path, then it is a exit point and it goes to another state symbol, say this is y, so what we can tell, this path is correct, but this is incorrect. So, this is a incorrect symbols are correct, but the connections are incorrect and so the ASM chart is incorrect. So, this exit point cannot be the end same entry point, the problem is here the exit point or entry point are the same.

So, if we remove the mistakes; this is my exit 1 and that means, if this condition is false, then it should go to a state symbol, always it should go to the input of a state symbol, so this is one entry point. So, here the same line cannot be the exit or the entry and exit here, this is the two different thing one entry point and now this exit goes to there and the remaining positions are same, so this is a correct one and this is called the correct ASM chart.

(Refer Slide Time: 25:11)




Now, we take one example of an ASM chart for a 3 bit counter for a 3 bit counter this is a mod 8, so it has 2 to the power three states, eight states, if the state symbols or the logical name is A, B, C, the three things. Then, actually that it will be 0 0 0, 1 0 0 1, 0 1, 0 like that it will goes to 1 1 1, now before that…That means, the input should be like will be first, it will go to a state symbol, say this is A B C and code is 0 0 0.

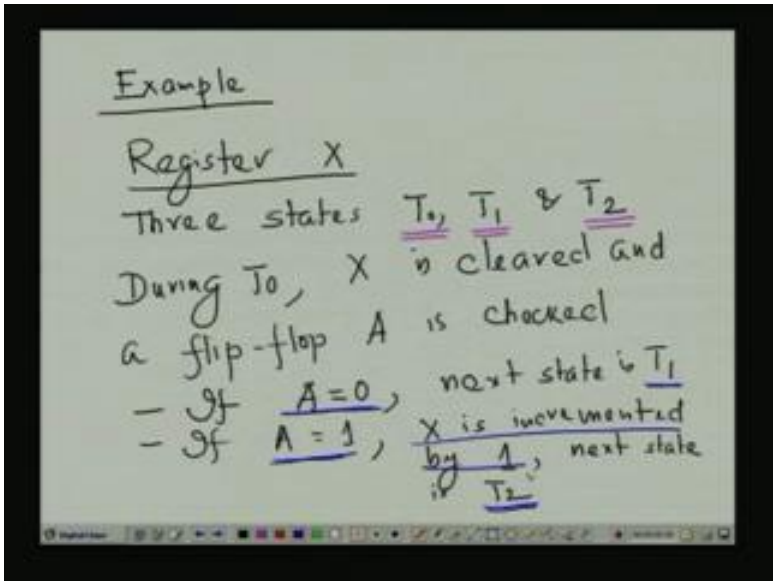
So, it has one entry point, this is my initial thing, now it has one exit point and now I am putting here one decision symbol D, if D is false, means if it is a 0, If it is false, then again it should go, it will start or it will be in the initial state only. If it is 1; that means, this is another exit point, then only it will go to the next state or the other state symbol coded as 0 0 1.

So, if now we think that, we can give a different value different name, say this is a initial state small a, this is small b and then it will, so this is a true path and now as it is a state symbol, it will continue like that, so the remaining portions these are drawn here. So, before actually at the input of this first state symbol, here this it will be added, that means one initial state symbol followed by a decision symbol and if it is 0.

Then, again it will stay in the same state means initial state and when it will proceed or it starts counting, when it is a true. So, actually this is the exit point or if this is true or condition is true or this is the exit 1, then it will it starts counting and in that way we can represent the ASM chart.

Now, before giving the general methodology for the state machine design, we take another arbitrary example, that how actually the timing specification is represented by a ASM chart. Because, that is the basic difference from our normal flowchart or what we call the flowchart to implement the software program.

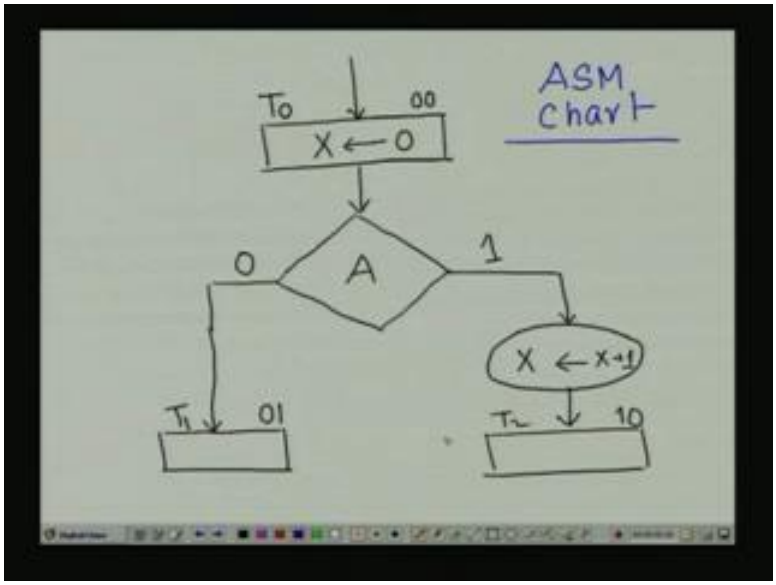
(Refer Slide Time: 29:50)



So, we take one example of an arbitrary ASM chart, say we are taking a symbol register X say we have a register X, there are three states T 0, T 1 and T 2. The machine is such of the synchronous sequential machine that we want to design. During that T 0, X is cleared and means it is reset or cleared or set to 0 and a flip flop A is checked.

Now, if A equal to 0, then the next state is T 1, if A equal to 1, X is incremented by 1. That means, the circuit increments the value of the register X, because X is a register X by 1, so X is incremented by 1 and the next state is T 2. So, this is the simple machine specification. So, we have three states T 0, T 1, T 2 and the conditions are conditions, if A equal to 0, next state is Q 1, if A equal to 1, X is incremented by 1 and next state is T 2.

(Refer Slide Time: 32:58)

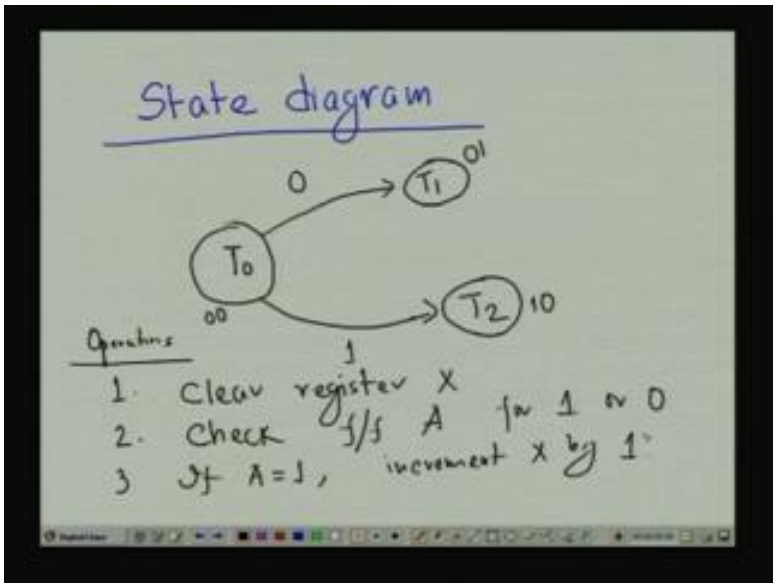



So, if we draw the ASM chart, first one state symbol and the register X is given, this is the states at T 0 as it is a state symbol, it has one entry point and one exit point. Now, it will go to a decision symbol and the decision symbol is A is checked, so the flip flop A is checked, the variable is placed inside the diamond box. Now, we know that diamond box has two exit point then, if A is 0; that means, if this is a false path, it should go to the state symbol T 1.

If it is 1 if A equal to 1, what we have that X is incremented by 1, that means now we need a oval shaped means our conditional symbol, so X is incremented by 1. That means, X is X plus 1 and now it goes to another state symbol called state name is T 2. So, now if

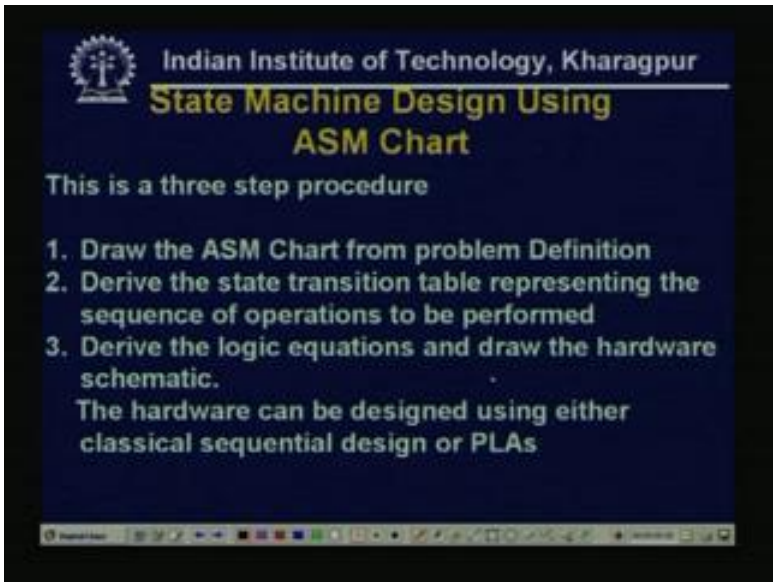
we give a binary code, say we have three states T 0, T 1, T 2, so we need 2 bit to represent the 2 binary code. So, if it is a see 0 0, this is T 0 0 0, then T 1 is 0 1 and T 2 is 1 0, so this is my ASM chart of an arbitrary simple a state machine.

(Refer Slide Time: 35:55)




Now, what will be the state diagram for the this ASM chart, the state diagram will be like the initial state, T 0 coded with 0, then it will go to a 0 1, means T 1, if it is false path, if it is a true path, then it will be a T 2 1, that means this 0 0, 0 1, 1 0. So, this is the state diagram of the ASM and the operations to be performed during the state T 0, that will be the three operations. One will be clear register X, check the flip flop A, for 1 or 0, if 1; that means, if A equal to 1 increment X by 1. So, we have seen that, how the from the machine specifications, how the ASM chart can be drawn and from there the state diagram can also be come out from that ASM chart.

(Refer Slide Time: 38:01)

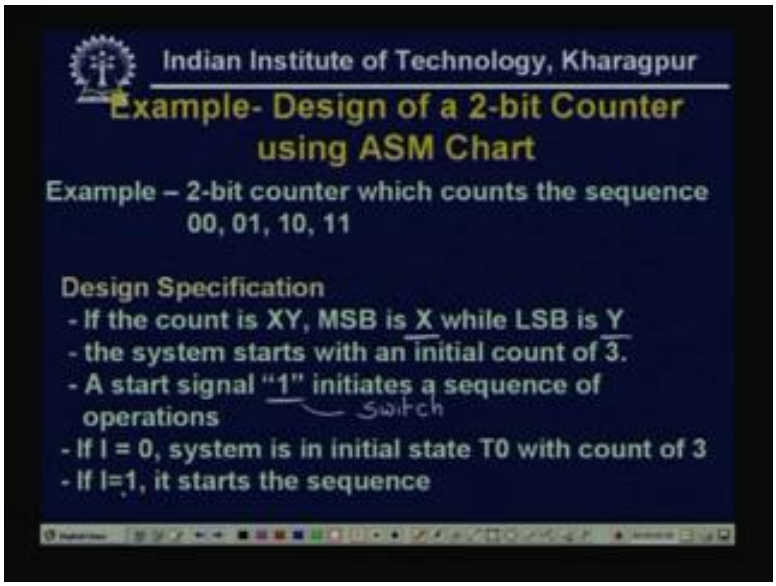



Now, we see that, what are the generic steps for the state machine design using the ASM chart, normally this is a three set procedure, the first step is the draw the ASM chart from problem definition. Problem definition means just now that machine specification I told, that one register X is there a flip flop A is checked, if it is 0, then go to a state T 1, if it is 1. Then, increment X by 1 and next state will be T 2, so this is my problem definition of the problem specification.

So, from there they just now we have seen the how the ASM chart can be drawn, second step is derive the state transitional table representing the sequence of operations to be performed. So, the now the state transition table, representing the sequence of operations to be performed; that means, the list of operations to be performed to execute the machine, we have to derive that thing.

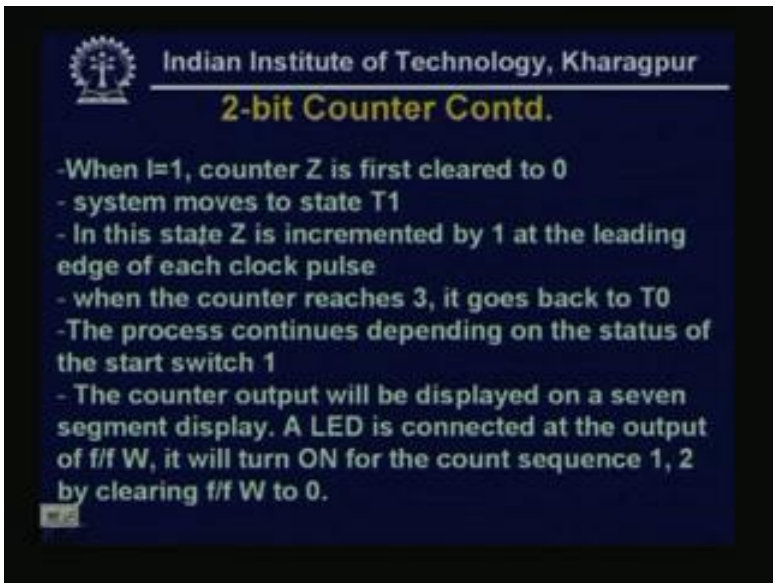
Step 3, derive the logic equations and draw the hardware schematic, so once the logic equations are derived, we know that the hardware schematic can be drawn. Now, the hardware can be designed using either classical sequential design or PLAs, what we have discussed in details, this portion that how the hardware is design or given a set of logic equations, how the hardware can be implemented. So, mainly these are the three basic steps of the design of state machine using ASM chart.

(Refer Slide Time: 40:03)




Now, now we see one example, the design of a 2 bit counter using ASM chart, so to first the problem specification, example is 2 bit counter, which counts a sequence 0 0, 0 1, 1 0 1 1, 2 bit counter. Now, what are the design specs, if the count is X Y, say we represent our convention is MSB is X or MSB is X and LSB is Y, the system starts with an initial count of 3. A start signal one initiates a sequence of operations that means, this is just like a switch, if I equal to 0 system is in initial state T 0 with count of 3, if I equal to 1, it starts the sequence.

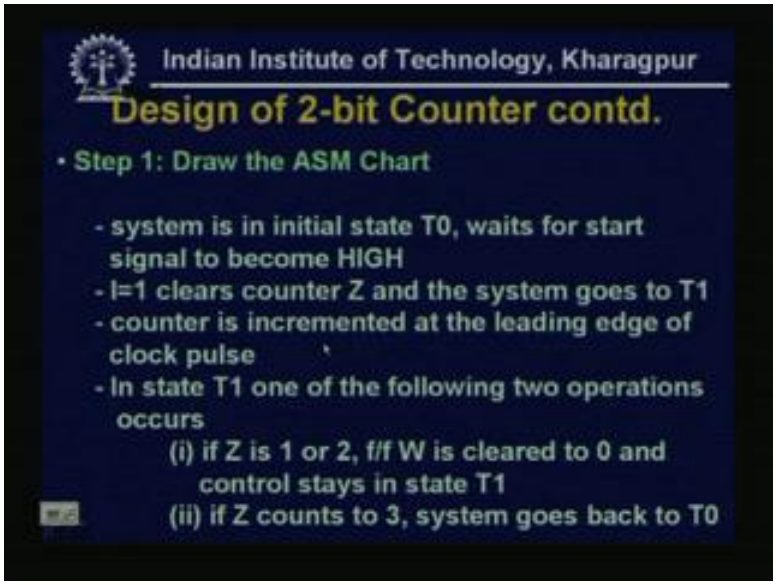
(Refer Slide Time: 41:21)



Now, some more specs that, when I equal to 1 counter Z is first, clear to 0, system moves to state T 1, in this state Z is incremented by one at the leading edge of each clock pulse, when the counter reaches 3, it goes back to T 0. So, it will be all the term, it will be 0 0, if it is only it counts 1 and 2, that will it will change 0 1 1 0, again you when it will be 1 1, it goes back, the process continues depending on the status of the start switch 1.

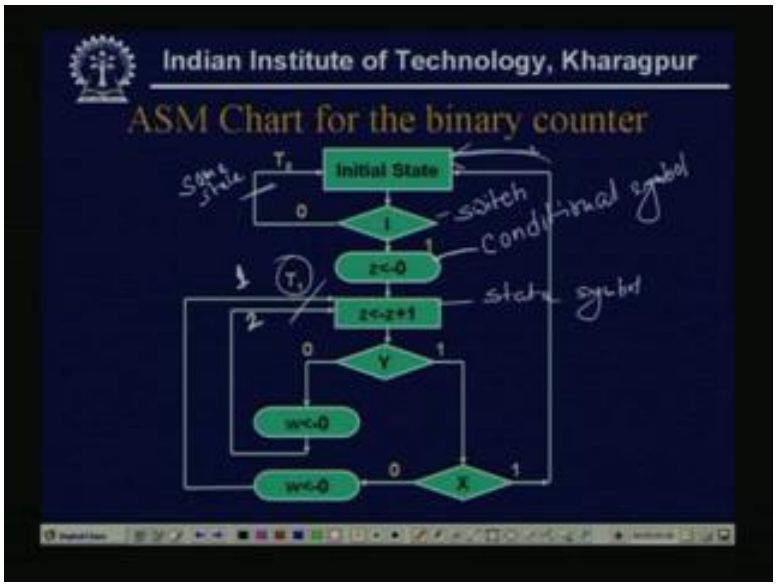
The counter output will be displayed on a 7 segment display and a LED is connected at the output of flip flop W, it will turn on for the count sequence 1 to by clearing the flip flop W to 0. So, these are my this is the design or the problem specs, what I have to design, a 2 bit counter, I want to see the performance of the function of the counter in this way.

(Refer Slide Time: 42:30)




Now, the first step, it has a three step procedure, so first step is draw the ASM chart, the system is in initial state T 0 and waits for start signal to become high, if I equal to 1, it clears; that means, if it is a 1, it clears counter Z and the system goes to T 1. Counter is incremented at the leading edge of clock pulse; instead T 1, 1 of the following two operations occurs. If Z is 1 or 2, flip flop W is clear to 0 and control stays in state T 1, if Z counts to three systems goes back to T 0, because it is a only 2 bit counter.

(Refer Slide Time: 43:29)



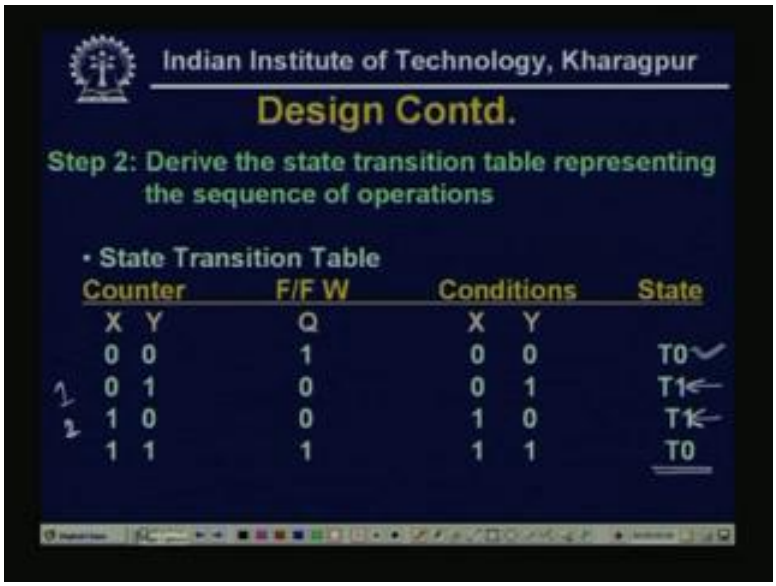

Now, what will be the state diagram for this ASM chart, so the ASM chart just from the problem specification, if we derive the ASM chart or draw the ASM chart, so this is the initial state. Now, this is my switch, we can think that, this is my switch, if it is 0, always it stays in the same state, so this is a state symbol, if it is high; that means, then only it will proceed, proceed means Z is cleared. So, this is my conditional symbol, that Z is cleared to 0, this is the conditional symbol.

Now, it has one entry point, one exit point, now Z at a each leading edge or positive edge of the clock pulse, it will be incremented, so Z becomes Z plus 1, so this is my state symbol. Now, again it goes to a decision symbols, that if Y is 0; that means, if it is false, then the flip flop W is clear to 0 and again goes back to; that means, it continues Z is Z plus 1.

If it is Y is 1, then it comes to another decision symbol and if it is 0, then only W flip flop W is 0 and again it goes to the state T 1, it goes to state T 1, so for this is 1 and 2 count, it is in the state T 1. Now, if it is X is 1; that means, it is a 2 bit counter, so if it counts 3, again it goes back to the initial state, so this is my ASM chart. So, mainly from the problem specification or the problem definition the ASM chart is drawn accordingly.

Only one thing is to be noted that, here a timing specification is always maintained by the ASM chart, which is totally absent in our normal flow chart. There is no time, out specifications, but here always the time out specification is there.

(Refer Slide Time: 47:03)

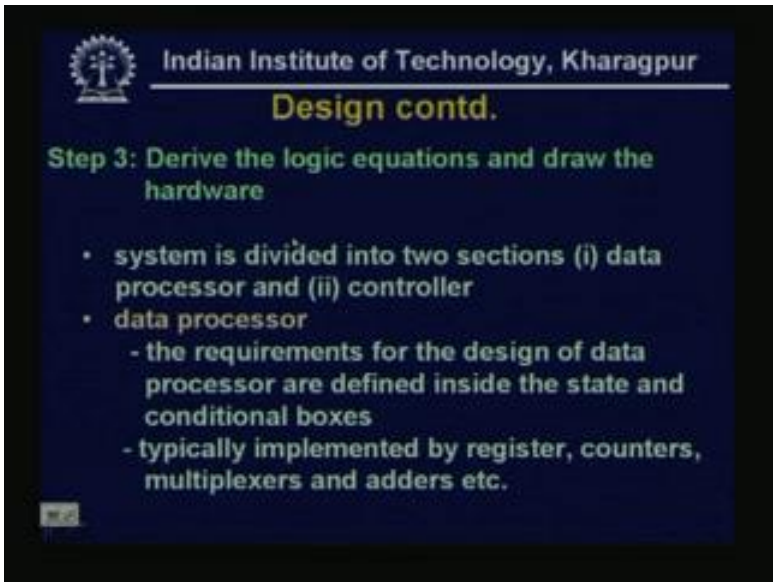



Now, the second step is the derive the state transition table representing the sequence in sequence of operations, so the state transition W, we see that the counter, then it is a W flip flop, the conditions and then, what will be the state. So, if it is 0 0, the flip flop value is 1, conditions to be checked is 0 0 and state is T 0.

If it is 0 1, then; that means, it is 1, then will be in T 1 state, If it is 2, then it will be in T 1 state, just now we have seen, see this is the two things, if it is 1; that means, Y 0 or another is that Y 1, X 0, so this is 1, both the conditions, it will be in state T 1. Now, when it is 1 1 counter value, then the W the flip flop W is 1 and again it goes back to the initial state initial state T 0, so again it will be is T 0.

So, initially at T 0, then it proceeds it is T 1 T 2; that means, for count value 1 2, again when it will be 3, it goes back to the initial state. So these are the four states to be counted.

(Refer Slide Time: 49:22)



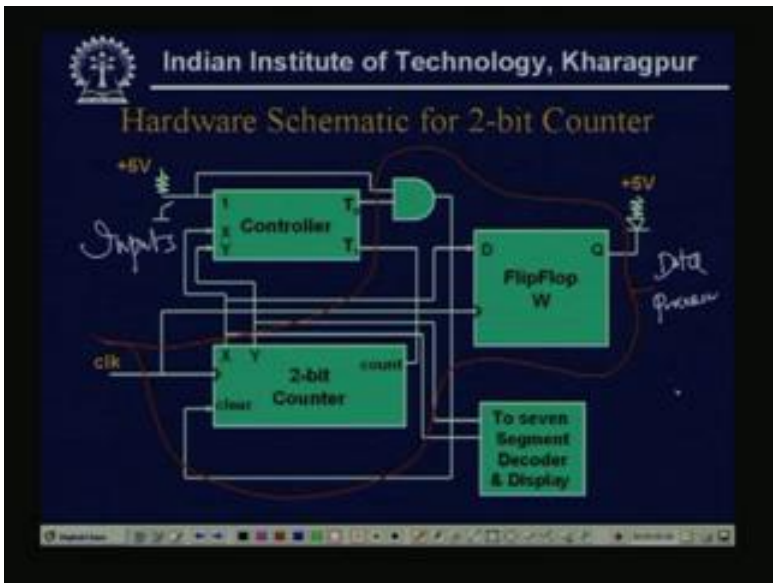

Now, the third step is deriving the logic equations and draw the hardware System is divided into two sections the data processor and controller. So, the data processor the requirements for the design of data processor are defined inside state and conditional boxes and typically these data processors and implemented by the digital components like registers, counters, multiplexers and adders etcetera.

So, here also will see, that the hardware part, that is normally the system is divided into 2 and the data processor will be implemented by this registers, counters, multiplexers, adders etcetera, we will be seeing later.

(Refer Slide Time: 50:26)

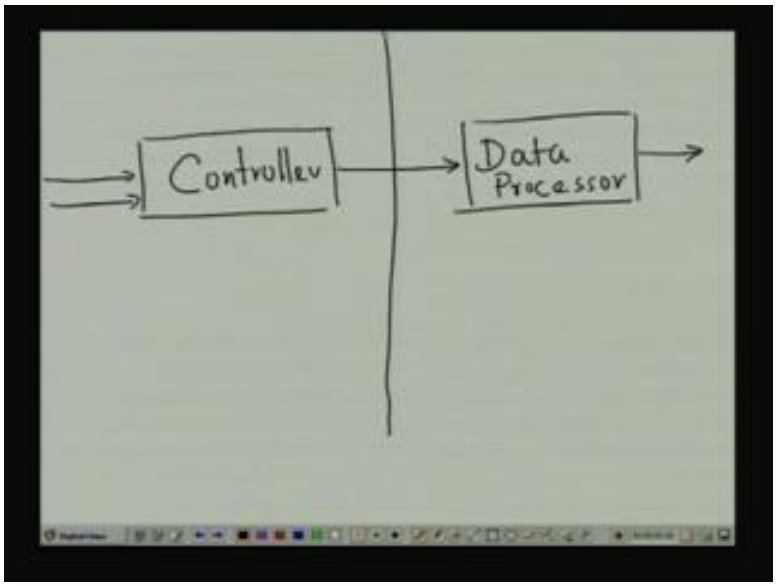

And then, the logic for the controllers is determined from the decision boxes and the necessary state transitions.

(Refer Slide Time: 50:44)




So, this is the hardware schematic for a 2 bit counter, just now what we have seen, see it has two parts, the controller part and the data processing part,

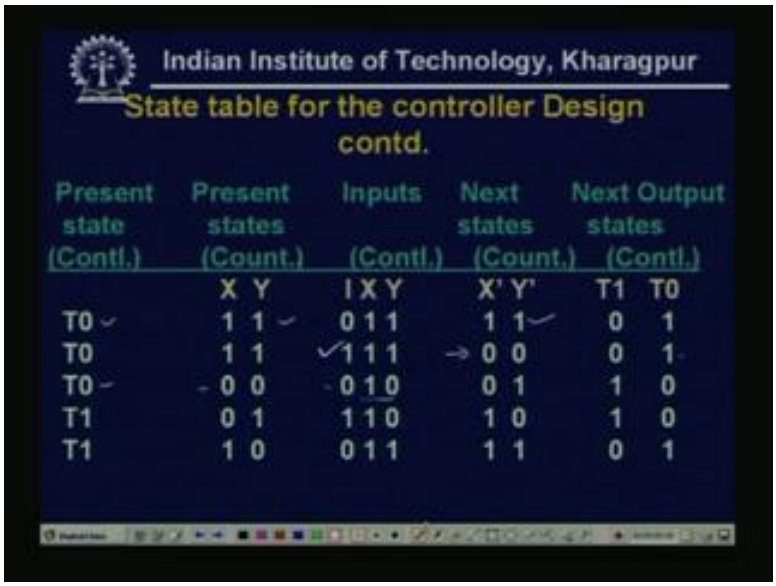
(Refer Slide Time: 51:06)




What, we have see that, if we draw that, it has two part, one is the controller, another is the data processing, the inputs are here and the after processing the output, so actually there will be a partition here. Now, for this particular example of 2 bit counter, that data processor part is the 2 bit counter, Say the 1 2 input AND gate, the flip flop, W, these three constitutes a data processor parts, so these three are the data processor.

And the controller is there,. so the input is coming, these are my inputs X Y to the controller, from here that partition is there this is the data processor and this is for the output. That to a seven segment decoder display, this is only for the counter value, checking the counter value.

(Refer Slide Time: 53:08)

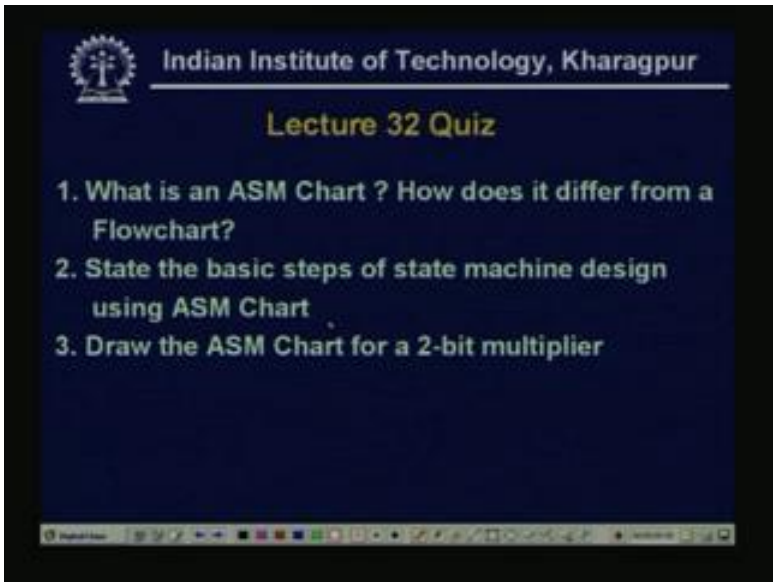



Now, if we see the state table for the controller design, then there will be present state for controller, c o n t means controller and c o u n t means counter. So, present state controller, present state counter, inputs controller, next state counter and next output states. So, present state controller, the initially it will be the T 0 state, the counter value is 1 1 and the inputs coming is 0 1 1, 0 means, it will stay in the same state. So, the next state is again 1 1 And the next output states of the controller is T 1 0, T 0 1.

Now, when only that I equal to 1 it will be changed, so the T 0, X Y is 1 1 1 1 and then only the next state is change to 0 0 and T 1 become 0 T 0 is 1. Now, in this way, similarly the as I describe now, when it is T 0, X Y 0 0, again it is a 0, so it is 1 0, in this way, the state table can be formed.

So, if we see the hardware schematic; that means, that the controller part will we can design and then the data processing part in this way the 2 bit counter with the flip flop that the state tables. We can derive and accordingly the state tables can be implemented by the logic, once we get the tables we can logic equations, we can get from the state tables and then the equations can be implemented by classical techniques or by PLAs. So, in this way, any type of we have seen one small examples of design of a 2 bit counter, that any type of digital systems, that we can design using the ASM chart.

(Refer Slide Time: 56:59)




Now, we see that, today's lecture quiz, the lecture 32 quiz, mainly today we have read the algorithmic state machine chart for designing the digital system. So, what is an ASM chart, how does it differ from a normal flowchart. Here, the flowchart means, this is a normal flowchart or normally the flowchart for the software and state the basic steps of step machine design using the ASM chart.

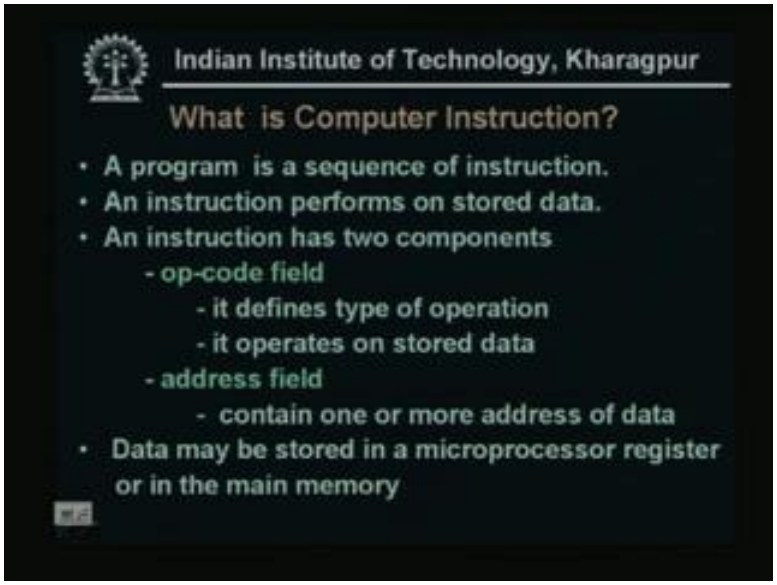
And today, we have discussed the design of a 2 bit counter are similarly we can design any machines a module eight counter or 3 bit counter and which three is draw, the ASM chart for a 2 bit multiplier. So, the three basic steps is in drawing ASM chart, derive the logic equations and from there the hardware implementation, we have to do for a 2 bit multiplier.

**Digital System Design**
**Prof. D Roychoudhury**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kharagpur**

**Lecture - 33**
**Design of Computer Instruction Set and the CPU**

Today, we will start discussion on the design of computer instruction set and the CPU. First we will see, how we can design the computer instructions.

(Refer Slide Time: 57:26)




Now, before stating the design, what do you mean by computer instruction, all of you know that, a software program is consist of a sequence of instructions. Now, an instruction performs on stored data that means a program is nothing but a sequential operation to be performed on the data and an instruction is one single operation on the data.

So, an instruction has two components, normally we call the op-code field and the address field. Now, op-code field defines the type of operation and it operates on the stored data, address field contain one or more addresses of data on which the operation is to be performed and data may be stored in a microprocessor register or in the main memory. So, instruction has two filed and op-code field and the address field, so when we want to design a computer instruction or to represent.