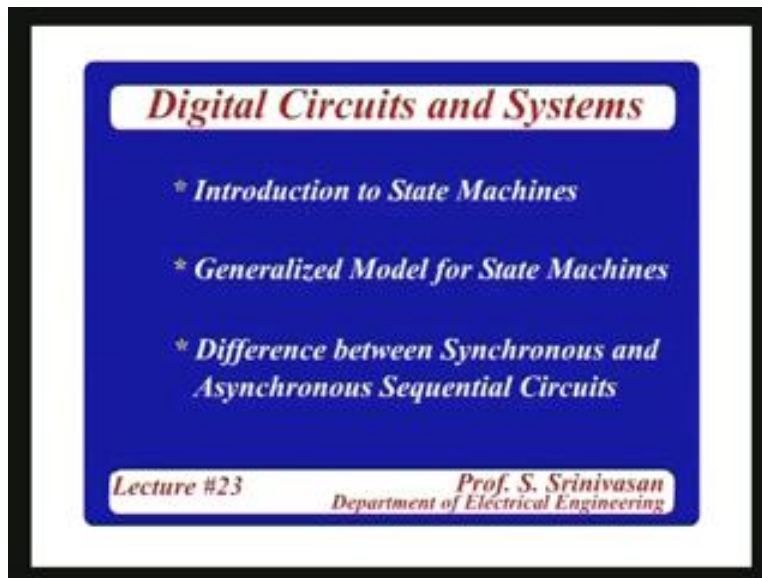


**Digital Circuits and Systems**  
**Prof. S. Srinivasan**  
**Department of Electrical Engineering**  
**Indian Institute of Technology, Madras**  
**Lecture # 23**  
**State Machines**

(Refer Slide Time: 2:13)



Now we have seen building blocks in combinational and sequential logic, some of them are small circuits. For example, an adder can be called a building block if you want to build an arithmetic unit and send them as a circuit then you can think of this as interconnection of gates. So it's not a very rigid definition, the components, building blocks, circuits, subsystems these are all freely used. Most practical systems digital systems are sequential in nature. Of course a sequential circuit will also have a combinational part.

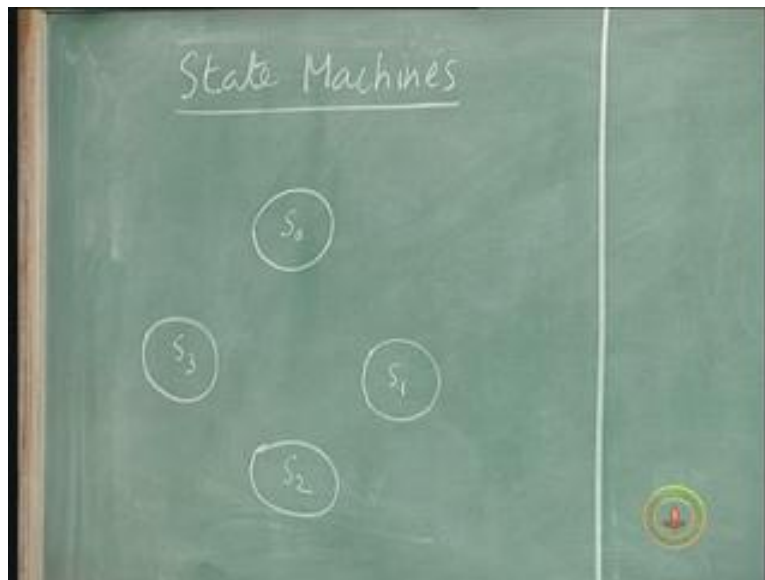
What I am trying to say is it is very rarely we have a circuit which is all combinational occasionally it comes like a code converter, parity checker. But mostly digital systems are combination of both combinational and sequential circuits. The sequential circuits sequential components or building blocks are required to store let us say information. The information need not be always data, information may be a state, a circuit remains in this state has to go to another state based on which state it is in.

For example, if you take a counter, counter is a sequential circuit, it counts from state to state but then the count always has to be retained in order to know what should be the next count. So, a circuit remains in a state, this is called a state, a circuit remains in a state, a particular count is a state and then when the input comes it goes to the next state which is the next count. Take another example like traffic light control traffic and light controller. Circuit has to remain in a particular state let us say green on the main road and

red on the side road, it is an intersection of the main road and the side road, main road as a traffic flow with the green signal and side road traffic is stopped because it's a red light and then it has to go from that to the state wherein we want to go briefly stop the traffic in the main road and let the traffic in the side road pass so the circuit has to know which state it is in, how long it has been in that state before you can make a decision whether it is the right time to change the sequence of lights.

Of course it may not go from red to green, green to red but it may go through amber or yellow as they call it so these are different states. So a circuit or a digital system need not be huge like computers and digital television or flight simulator. Can we have small things that's what I mean, a traffic light controller, elevator control a simple video game, counters these are also systems. So whenever we design a system we need to know what are the various states in which the system is permitted to be in and what sequence of inputs or what inputs should occur for the system to steer from state to state. Generally these are called state machines. State machine is a name given to a digital system. today we will talk about state machines.

(Refer Slide Time: 7:45)



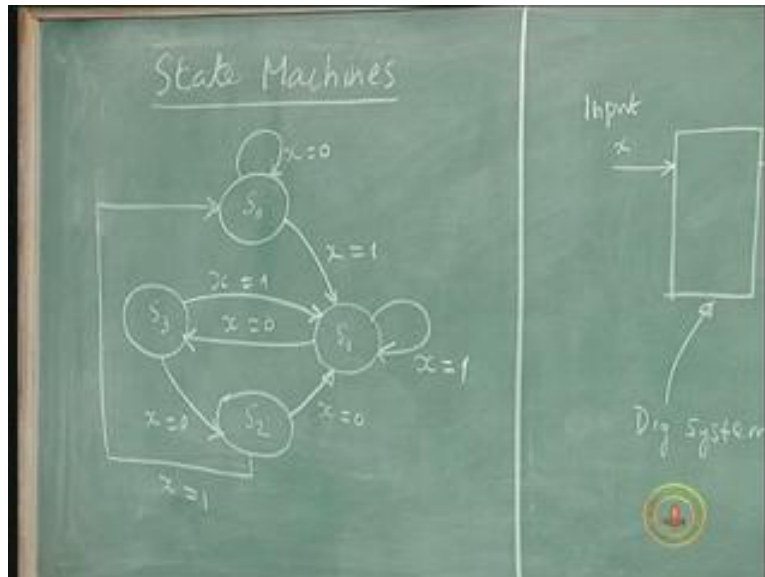
This comes from the good old days of machines, computer was called a computing machine. so everything was a machine. A state machine is a circuit which goes with a system or a circuit whatever as I said again, let us not be too specific about it.

It has several states in which it can exist and it's a clearly defined set of inputs which will steer the circuit or the system from one state to the next state and in each state it can also produce an output. So let us think of this as a simple circuit in which there are four states, I call these states  $S_0$   $S_1$   $S_2$   $S_3$ . So this is the circuit or the system I am talking about. Now let us say there is only a single input called  $x$  which decides what state the circuit goes from a given present state and then it also produces an output  $z$  or  $z$ . So I will call this input variable  $x$  (Refer Slide Time: 8:26) output is  $z$  both of them are binary variables so

it can only take a value 0 or 1 and the output can also take a value of 0 or 1, all variables are binary there is no doubt about it.

What I am trying to say is if circuit is in  $S_0$  it will remain in  $S_0$  and when a particular input occurs it goes from  $S_0$  to  $S_1$  depending on whether the input is 0 or 1. I will just assume that if  $x$  is 0 let us say the circuit does not go from that state  $S_0$  it remains in  $S_0$  on the other hand if  $x$  is 1 it goes to  $S_1$ . This is just an example, it could be the other way or it can go from  $x$  is 0 from  $S_0$  to  $S_3$  and if  $x$  is 1 it can go to  $S_0$  to  $S_1$  or it can remain in state  $S_0$  for  $x$  is equal to 1 and go here for  $x$  is equal to 0 these are all different possibilities based on the system you define. So in each state of the circuit or the system I look at the inputs, what are the inputs possible in that state and what should happen to the circuit when that input occurs or if it does not occur, if input is true what should happen and if input is false what should happen. In  $S_0$  the input is true it goes to  $S_1$  the input is false it remains in  $S_0$ .

(Refer Slide Time: 10:45)



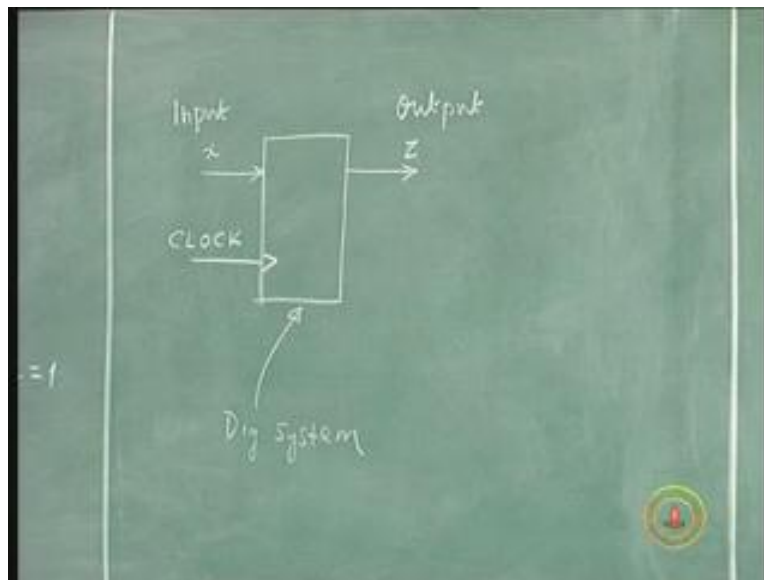
On the other hand just for a change I will say if  $x$  is equal to 1 it remains in  $S_1$  and it goes from  $S_1$  to  $S_3$ . I am writing this in sort of an arbitrary way in order to tell you that these are also possible, don't think that it has to go on a chain or anything like that and from  $S_3$  it may go to  $S_2$  if  $x$  is equal to 0 and from  $S_2$  it may go to  $S_1$ . Let us not even bother about what this system could possibly be, what will be done by this system, what is that the circuit is going to realize. This is a concept I want to bring in a concept called state machine.

A state machine is a machine of course these days you don't call a computer as a machine, in electronics, circuits or system you don't call a machine anymore, to us a machine is a big thing in which something has to move constantly and all that type of a thing high speed and so on. You might call an engine a machine may be or an electrical machine or a huge crane or something you may call it a machine but not a simple IC

which is inside a computer. It is rather difficult to visualize this as a machine but it is a machine in the sense it does certain things. A machine is a mechanized or an automatic device which does things according to certain pattern or certain design. **Let us accept that definition for a machine.** why is it called a state machine? It is not a machine which takes the input and gives the output. It is a machine which can exist in several states and output depends on the input and the state in which it is in. Only if we know the state in which it is in then you can decide on what this input can lead to next.

What is the next state of this machine? You can determine the next state of the machine only if you know the present state of the machine and the input in that state and also not all machines need to have input the external input. The counter can just go from count to count to count just by clock. Because you have a sequential circuit here all sequential circuits have clocks so I will have to also assume there is a clock here. I may not have an external input  $x$ , each clock cycle may go from one state to the next state a counter is one such. supposing I have a 4-bit binary counter going from 0 0 0 0 to 1 1 1 1 and back to 0 0 0 0 I don't need any input except the clock.

(Refer Slide Time: 13:50)



Suppose the circuit is in 0 0 0 0 the counter is in 0 0 0 0 state at the first clock pulse it will go to 0 0 0 1 state, second clock pulse it will go to 0 0 1 0 state, third clock pulse it will go to 0 0 1 1 state and so forth and in the fifteenth clock pulse it will go to 1 1 1 1 and sixteenth clock pulse again it will go to 0 0 0 0. That is also a state machine because it also has states which moves from state to state based on the previous state. What we can be only be sure of is the next state is determined by the previous state. There may or may not be external inputs in addition to the previous state information.

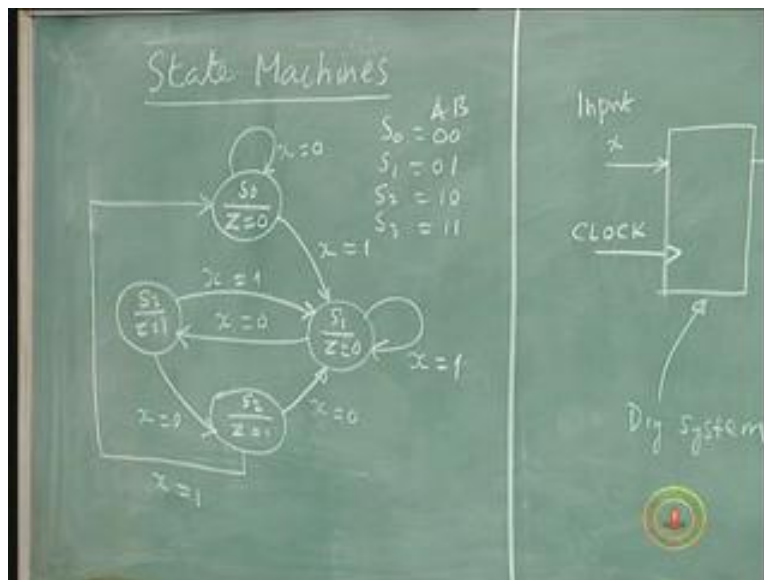
So in order to determine the behavior of this machine or the system or the circuit. I need to know what state the machine is in now, what state the circuit is in now and what input is likely to occur whatever the options of the input are like if the input is 1 what will

happen, if the input is 0 what will happen and there may be more than one input there may be 0 input, there may be 1 input, there may be two input then  $x_1 x_2$  can be two input then  $x_1 x_2$  0 0,  $x_1 x_2$  0 1,  $x_1 x_2$  1 0,  $x_1 x_2$  1 1 there are four different conditions four different states that can take the next value and it can also give you output other than the state the state itself can give an output the counter.

For example, the counter is a simplest thing. A simplest ((15:31)) you can think of is a counter if there is no external inputs no external outputs. If a count is in state 0 that means the count is 0 there are only sixteen states in a 4-bit machine, in a 4-bit binary counter there are sixteen states starting from 0 0 0 0 to 1 1 1 1. First state is 0 0 0 0, second state is 0 0 0 1, with one clock alone without external input it goes from state to state. Is there an output other than the state? No, the state output is same as the output. if the circuit is in state 1 0 0 0 I can say the count is 8.

On the other hand this circuit can have an external output in addition to the state in which it is in. In  $S_0$  z can be 0 as an example. In  $S_1$  z can be 0 again,  $S_2$  z can be 1, in  $S_3$  z can be 1 again. So the basic requirement of a state machine is there should be more than one state in which the machine can exist. There is only one state that is the combinational logic they give the inputs the output is determined. If more than one state is there then it's a state machine. You may have several states and the behavior of the machine can be determined by the path it takes from going from state to state. This behavior can be determined not only by the previous state or the current state in which it is in but also by the external inputs and in each state in addition to the current state information the circuit can also give you the output information which is different from the state information.

(Refer Slide Time: 18:20)



For example, I have to represent these four states so how do I distinguish between state  $S_0$   $S_1$   $S_2$   $S_3$  I need to represent these states as again binary variables so I can for example say  $S_0$  is 0 0,  $S_1$  is 0 1,  $S_2$  is 1 0,  $S_3$  is 1 1 the natural binary sequence. Since I have

already used  $x$  and  $z$  and  $s$  for this let me call this  $A$  and  $B$  as state variables. So there are two variables called  $A$  and  $B$  when  $A$  and  $B$  are both 0 the machine is in state  $S_0$  and at that time  $x$  becomes 0 it continues to remain in 0 0 giving an output of  $z$  is equal to 0.

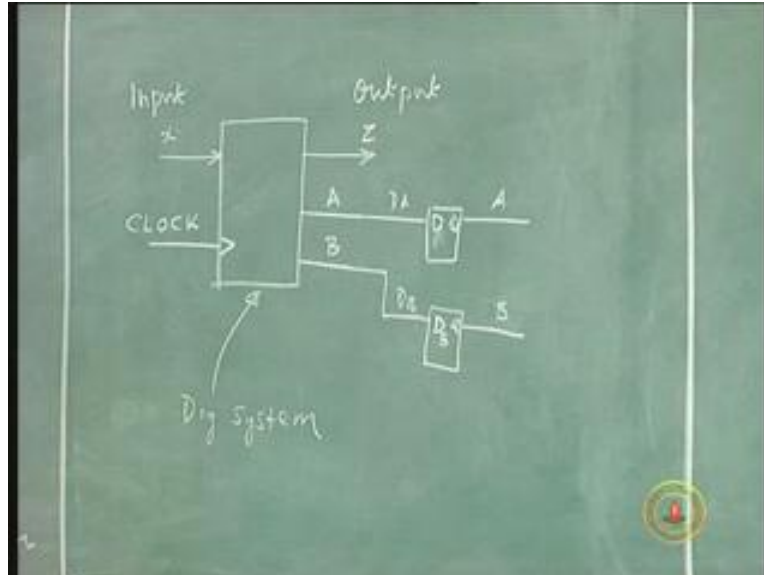
On the other hand when the circuit is in 0 0 and input happens to be 1 it goes to state  $S_1$  which is defined by the this state 0 1 gives an output of  $S$  is equal to 0 again and there again it is tested for  $x$ ,  $x$  is tested,  $x$  is 1 it remains in the same state and when  $x$  is 0 it goes to state  $S_3$  which is called just the value of 1 1 for  $A$  and  $B$  and so forth. So this is the state machine and the representation of a state machine is called a state graph or a state diagram.

Now, coming to this if this state graph of the state diagram represents this system it needs to have four states four and four states are represented by two binary variables because two binary variables can have four combinations 0 0 0 1, 1 0 1 1, these two binary variables are denoted as  $A$  and  $B$  so I would know what state it is in by looking at the value of  $A$  and  $B$ . If  $A$  is equal to 0 and  $B$  is equal to 0 I know the circuit in state  $S_0$  and if  $A$  is equal to 1 and  $B$  is equal to 0 for example I know the circuit is in state  $S_2$  so what are this  $A$  and  $B$ ?  $A$  and  $B$  are the states in which the circuit is and that has to be retained go to the next state, from this state this will go to the next state after one clock period.

Now what triggers this transition, of course input will determine the transition but what triggers the transition is the clock, without clock the circuit does not work. So there is a clock here which triggers the state transition which will be determined by the inputs. The transition will be decided by the input. just because you give the input  $x$  is equal to 0 here it will not go here (Refer Slide time: 21:00). We will give input  $x$  is equal to 1 the circuit is remaining in the state for this during this current clock cycle and you make  $x$  is equal to 1 it will not jump to  $S_1$  but it will go to  $S_1$  only at the next clock cycle. The clock triggers the transitions and the input determines the transitions and the state determines what the output will be there.

That means I need to have these variables  $A$  and  $B$  coming out of this. Since this value of  $A$  and  $B$  are required for me at least for one clock period only then I can determine what state it will be in the next. because let us assume that the circuit has just now reached state  $S_0$  or  $S_1$  let us say, circuit as just now reached  $S_1$  which is 0 1 and input of  $x$  is equal to 0 has occurred, it needs to go to  $S_3$  next. I told you  $S_3$  it can reach only after the next clock pulse or at the next clock pulse edge if you want edge triggering. That means I need to remember the value of this  $S_1$  which is 0 1 for the entire duration of this clock period. Only then at the end of the clock period or at the beginning of the next clock period the transition can happen from  $S_1$  to  $S_3$  because input  $x$  is equal to 0. That means I need to store the value of  $A$  and  $B$  in bits or store them in flip flops.

(Refer Slide Time: 24:00)



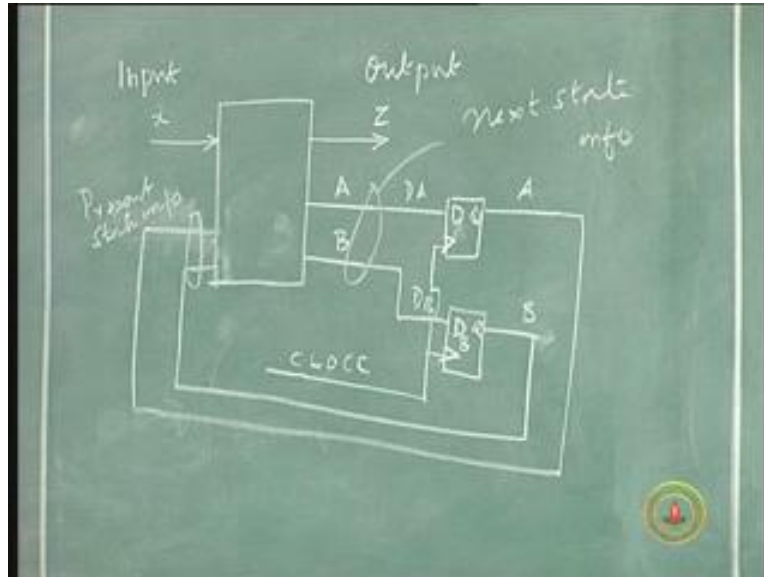
I need two flip flops to store the value of A and B so I have an A flip flop and B flip flop and the value will be given to let us say there are simple D flip flops DA and DB. The circuit is in now state AB which is 0 0 or 0 1 whatever it is and it has to go to the next state, the system should know the circuit is in this state and system should know this is the new input that means I need to give this information back into this state. So I will modify this drawing and not use the clock here.

The A and B values are given back into this (Refer Slide Time: 24:37), this is the present state information which is held by the flip flop along with the input and then it will reach the next state after the clock edge. So I will use this clock here. Therefore basically the whole scheme holds, this is the state machine or digital system or whatever you want to call it. But this consists of two parts very distinctly two parts. This is the sequential part which remembers the present state and makes it available for next transition and this is only a transition circuit which determines what is the present state and what is the input and then decides what should be the next transition so this need not be sequential, this is only a combinational logic. Because I need to know what is the present state and I need to know what is the input so that I can determine what the next state will be, the determined value will be here but it will not yet become the next state, it will become the next state when the next clock pulse arrives. At the next clock transition this information will be moved on to this place that will become the next state information and the next state becomes the present state.

The next state information is fed here (Refer Slide Time: 26:30) or you can say this is the present state and this becomes the next state. So what we will call it, this is the present state information, this is the input information and this is the next state information which is determined by the present state information and the input and it also determines the output of the present state but it becomes the next state by absorbing these values into this flip flop by storing these values into this flip flop and that will happen in every clock.

This becomes the next state information the next state information becomes the present state for the next cycle and then that will be corresponding the input which will go inside the third state and so forth.

(Refer Slide Time: 27:50)

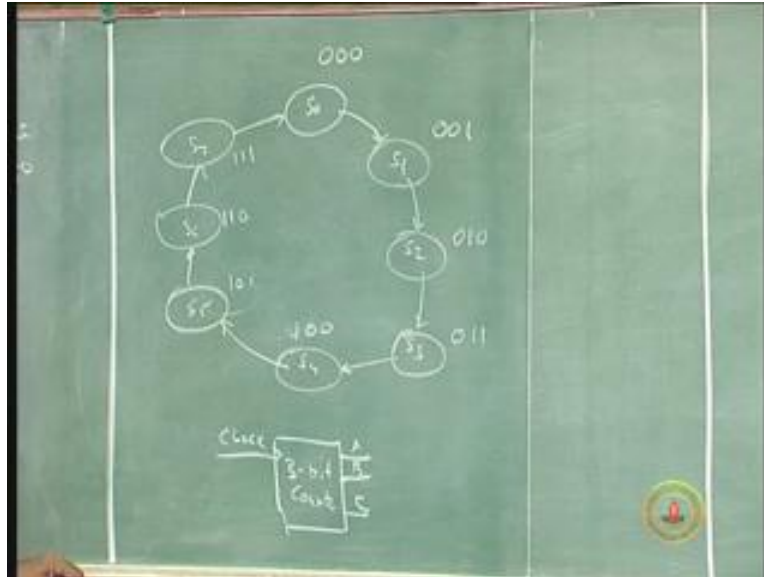


Hence in each state the circuit is in one clock period, for one clock period the circuit is in one state, it is not necessary the circuit has to change state. If  $x$  is 0 for a long time for several clock cycles we have  $x$  is equal to 0 the circuit happens to be in  $S_0$  on  $x$  is 0 it will remain in state 0  $S_0$ . For every clock pulse it will sample the input during every clock transition the input will be looked at, it is called sampling the input, the input will be looked at, the input is 0 so I will remain here. again next clock pulse will say what is the input, it is 0 I will remain here and in next state pulse again I will see the input will be 0 I will remain here, I remain here, here, here until you are going to change your  $x$ , I am not going to move out of this place. And then suddenly you change your input  $x$  is equal to 0 to 1 that is the next clock transition so now you have changed I will also move go to  $S_1$ . So one transition corresponds to one clock period but not necessarily every clock period if there is an external input. If on the other hand there is no external input then every transition will lead to a new clock set that is the counter example I told you.

So if I want to draw a counter state graph a simple counter state graph I will make it a simple three bit counter and I want to go on drawing. So let us say this is  $S_0$  no input, the arrows do not have any condition attached to them  $x$  is equal to 0  $x$  is 1 where there is no external inputs. so this is a counter the only input of the clock the outputs are A B C. Now this is  $S_0 S_1 S_2 S_3 S_4 S_5 S_6 S_7$ . So this is state 000, 001, 010, 011, 100, 101, 110, 111 and back to 000. **Remember that I am writing it outside the circle and not inside the circle.** These are only state values these are not output of course you are treating them as outputs if you want to but it's a state.



(Refer Slide Time: 30:57)



If there is an external input generally the external inputs are written inside the box inside the circle that is a convention. So there are eight states and it keeps going only with the clock there is no external input no external output so in this system I am not defining  $x, z$ . These are three inputs and this A B C will keep going of course this A B C will require to be fed back, this is the counter and since I put a three bit counter inside the flip flops I am not showing those flip flop models because there is no combinational logic here. You know how to build a three bit counter using three flip flops so that is it so there is no question of one more.

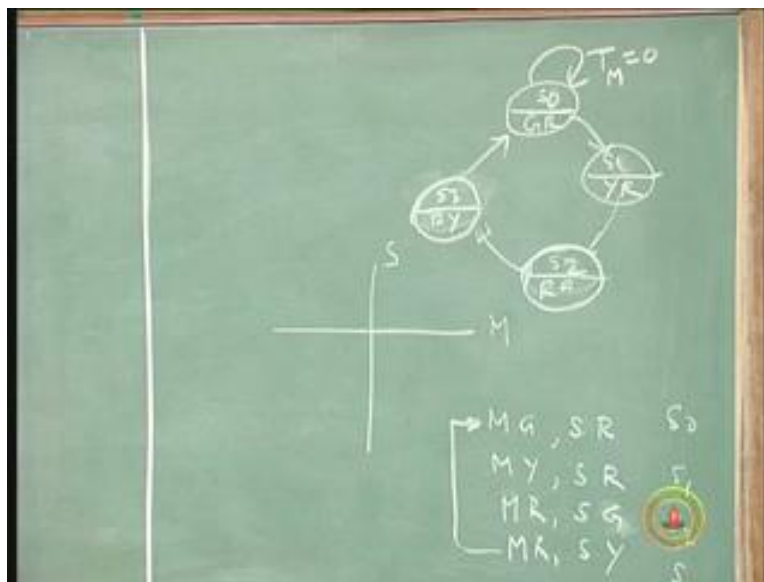
On the other hand here since there is an external input and the current state variables to determine the next state variables and the external outputs we need a combinational logic. The combinational logic determines what should be next state variables what is the value of the next state variables and the output from the present state information and the external inputs. there may be variations of this model,  $x$  may not be there,  $z$  may be there,  $x$  may be there  $z$  may not be there, there may be more than one  $x$ , more than one  $z$  all those variations are possible, more than two state variables, it can be three state variables, four state variables and so on. but this is a generalized model of a state machine.



change. So, in principle, of course you don't do a 30 seconds clock, you will have to get a faster clock and reduce it **I told you how to do that**. so if you have a clock which is changing state only every 30 seconds supposing a 30 cycle period clock for one clock period this will be in this state next clock period automatically this, this so my state machine will be very simple.

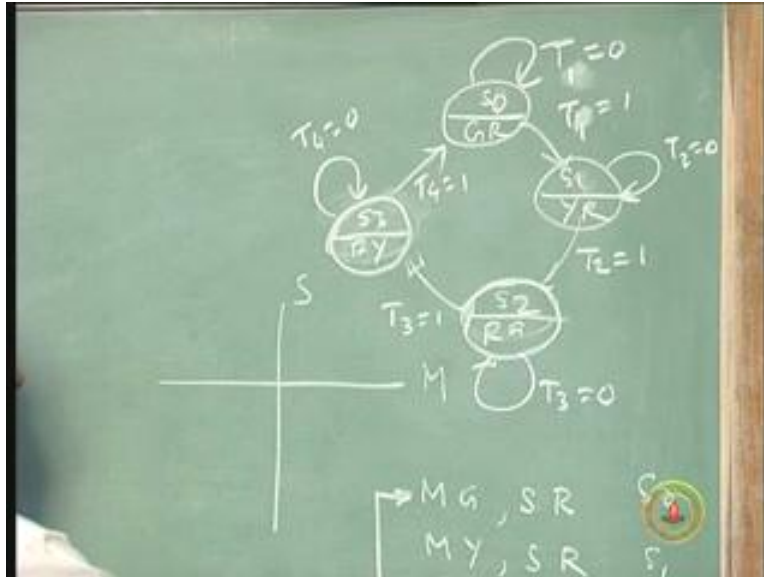
I call this as  $S_0 S_1 S_2 S_3$  and this is MG, SR so I will write GR I will write YR first is the main light second is the side light then RG then RY and I am assuming a clock which is very impractical clock of 30 seconds then there is no need for external input and external intervention and all that.

(Refer Slide Time: 38:40)



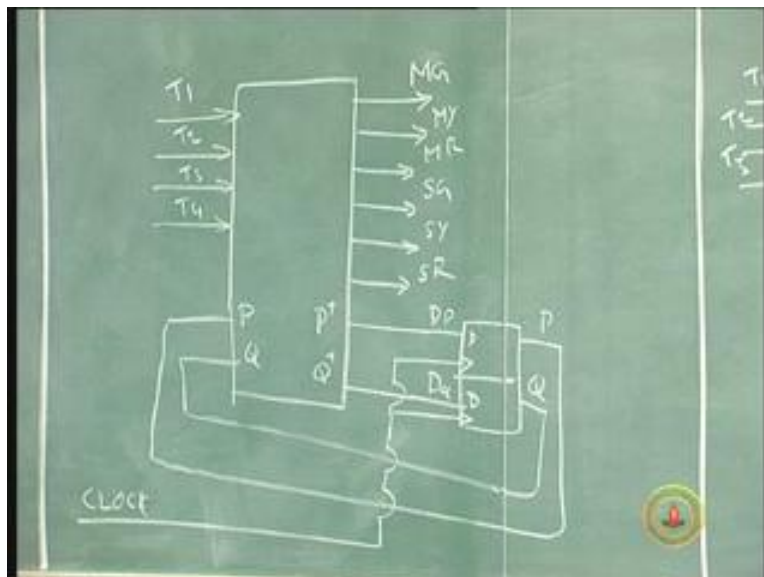
On the other hand I may want the main road signal to be of longer duration for 2 minutes or 1 minute and the side road can be only 20 seconds. In that case I should put a timer and then say if the timer is 0  $T_1$  I will call  $T$  as the timer for the main road, the timer has not reached this value it will remain in that state and if it has reached this value it will go to this state. Here I will have to have a timer again so I will call this  $T_2$  (Refer Slide Time: 39:00) whatever, I am going to call it  $T_1 T_2 T_3$  and the yellow light has to be a fraction of a minute like 10 seconds, 20 seconds then if it has reached that limit you go here then this will be third timer so I can have timer inputs and when that time has elapsed it will go to this and this will remain for a period of  $T_4$ .

(Refer Slide Time: 39:40)



So I am saying this state will be for a period of  $T_1$ , this state will be on for a period of  $T_2$ , this state will be on for a period of  $T_3$ . So I need external inputs which will give the timing events  $T_1$   $T_2$   $T_3$   $T_4$ . When the particular time elapses  $T_1$  will be on, particular time elapses  $T_2$  will be on, another time interval elapses  $T_3$  will be on and so forth. So I can now do the model as  $T_1$   $T_2$   $T_3$   $T_4$ . I need some more space for this I will do it here. And I need outputs these are external outputs called MG, MY, MR, SG, SY, SR these are the six lights I need and how many states the circuit is going to be in, and for four states we need how many state variables because this two state variables can give you four states 0 0 1.

(Refer Slide Time: 42:11)



So I will call these state variables P and Q and call these 0 0 state, 0 1 state, 1 0 state, 1 1 state. That means based on the present state information and the current value of P and Q the next state information will be decided and which lights will be on will be decided. So I will have to get these P and Q outputs into the two D flip flops DP DQ and take it back to P and Q. So I want to call these P to the power + and Q to the power + because the present value of P Q along with the present set of timer information decides which should be the next value of P and next value of Q and what lights should be on and this PQ information will be stored for one clock period and tested every clock period.

Therefore what I am trying to say is whether it's a very simple system or it's a very complex system you can reduce it to a state machine model. The state machine model is a machine in which the circuit present will be represented by several states, the circuit can exist in several states and for each of the states it can give an output need not give an output and each of these states check an input or several inputs so the next state of the circuit will be decided by the present state in which the circuit remains along with the external inputs. The next state information along with the output will be determined by the present state information and the present state inputs.

In trivial cases there are no external inputs but the state machine is a counter. Again a trivial case example would be that there is no external outputs other than the state variable output which is the counter. The counter doesn't have external outputs but only state variable as outputs, no external inputs except the clock.

A combinational logic is a trivial machine with one state no flip flop. a single D flip flop is a one state machine two states on and off or a traffic light controller or this model (Refer Slide Time: 44:10) or anything. Now all we have to do is think of various things you want to design; various systems and subsystems and circuits you want to design and try to model the behavior of that into a state graph.

So a word description of the problem is very clearly given to you as a set of specification. The set of specifications has to be modeled as a state graph and once you have a state graph I can reduce it to this form of and how to implement this we have to see of course, how to make a state graph, how do you convert these into required inputs, what inputs are required and all that but the most important part is this state graph drawing. Once you have given the specifications of a circuit or a subsystem or a system if you are able to reduce it to a state graph rather not reduce it represent it as a state graph then the rest is easy.

**So in the next few classes I will teach you how to implement a given state graph.** Later on we will see how to formulate state graphs based on specifications. Given a state graph how do you translate into this? How do you translate this into this, how do you translate this into this (Refer Slide Time: 45:44) that you will have to first see then you have a tool a procedure, after that all you need to do is to formulate the state graph properly for a given problem. You have to understand the problem specifications clearly and form a state graph and if you know how to implement a state graph as a hardware then you know how to build it, how to design it and how to build it and test it. So that is in crux with the

digital system design theory and these are all synchronous circuit as I said clock dependant. The clock triggers the transition of the state from one state to the next so this is called a synchronous sequential circuit. In asynchronous sequential circuit there will be no clock to steer some events will drive the next state. Suppose I am in a present state you can define a particular event to happen and that event will take you to the next state and you define another event to happen that will take you to the next state and so forth. that is a little more difficult to analyze as well as design **we will not do it now.**

So first we will start with synchronous sequential circuits wherein there will be a clock and always you can rely on the clock to work the clock precision as I said the clock work precision and at the clock transition the system will transit system will change its state. So all you have to do is to make sure that the new inputs new values are put here before that, before next clock pulse arrives you take all the time in the world to determine the combination logic delay because I am not going to do anything until the clock arrives and then only it can go to the next state. **So we will see many examples in the coming classes.**