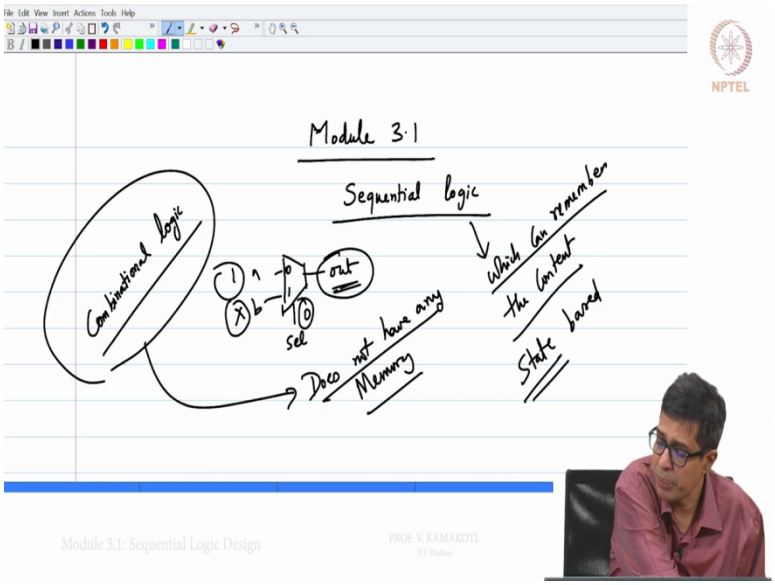
**Foundations to Computer Systems Design.**
**Professor V. Kamakoti.**
**Department of Computer Science and Engineering.**
**Indian Institute of Technology, Madras.**
**Module 3.1.**
**Sequential Logic Design.**

(Refer Slide Time: 0:21)




So, welcome to models depend one and in this module we will be talking about sequential logic. So what have we been talking so far? We have been talking so far about something called combinational logic. So, what is combinational logic? The output of the circuit depends upon the current combination of the inputs. So we have seen adder, we have seen multiplexer, let us say a multiplexer. So there is a select input, 0, 1, a, b. So the current value of the output depends upon the combination of the inputs.

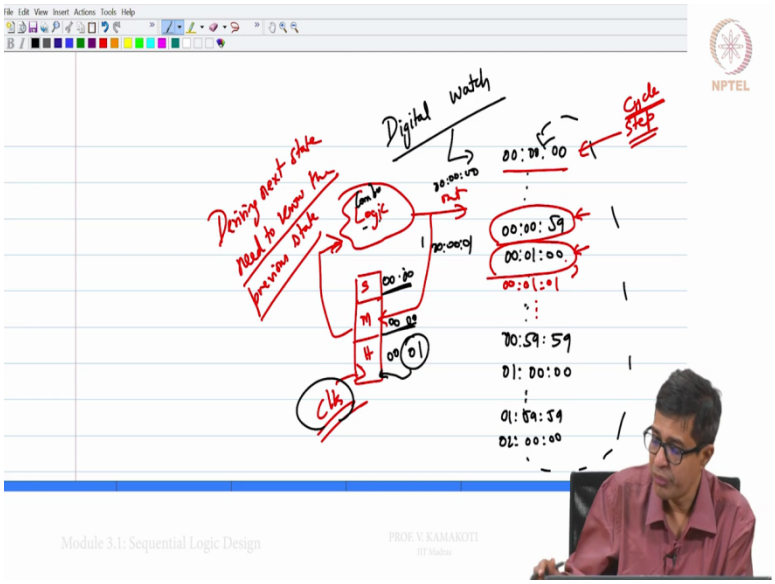
Like for example select is 0 and a is 1output is 1, whatever b be b can be anything, since selective 0, okay… So how is out of determined, I buy the current combination of the values of the inputs, that is why we call it as a combinational circuit. Now the point with the combinational circuit is that it cannot remember anything of what happened before. Currently what is the input, it will just, that combination of inputs it will take can give you an output, it does not have the capacity to remember something that happened say before this step or some time earlier in this. So that is why it is called the combinational logic.

So, essentially a combinational logic does not have any memory. But the system that we are trying to build needs memory, that is we need to remember what happened just, at least just

before what the previous step. Let us take a would one example of why this memory is needed. Right, so before we go into that, sequential logic is a set of circuits which can remember data, remember the content or whatever you call. I can remember, whatever was that in the previous cycle or previous instance, I can basically remember it in this instance.

So this is basically what we call as the sequential logic. Now sequential logic is needed for building any system because system is actually, the system, the computer system that we are seeing are basically state-based. State is nothing but the values of the different elements in the, just before, right, at any point of time, the value of the elements of the particular circuit for the state of the circuit. Now, I need to remember the previous state for me to calculate the next state.

(Refer Slide Time: 3:22)




So we will take some examples in this direction, where state becomes, a very simple example where state becomes important. Let us see, let us take the Digital watch that you have. So what are the digital watch do? It basically starts with 00:00:00 hours, then it goes on till 00:00:59 seconds, then it becomes 00:01:00 and so on. So, this will go on till 00:59:59, then 01:00:00, this will go on till 01:59:59, then it will become 02:00:00, like this it will go back to this, right.

So this is an example of a digital watch. Actually to make a working digital watch, what I need to do? For deriving next state, deriving state, I need to know the previous state. I need to know the previous state and that is very very important. If I do not know the previous state, then I cannot basically do this digital watch. So, like let us take this example, 00:01:00

comes, because in the previous state I had 00:00:59. 00:01:00 will not follow anything other than 0:00:59. So I need to remember this state to derive the state, I need to remember this state to derive what we call as 00:01:01 and so on so forth. Okay.

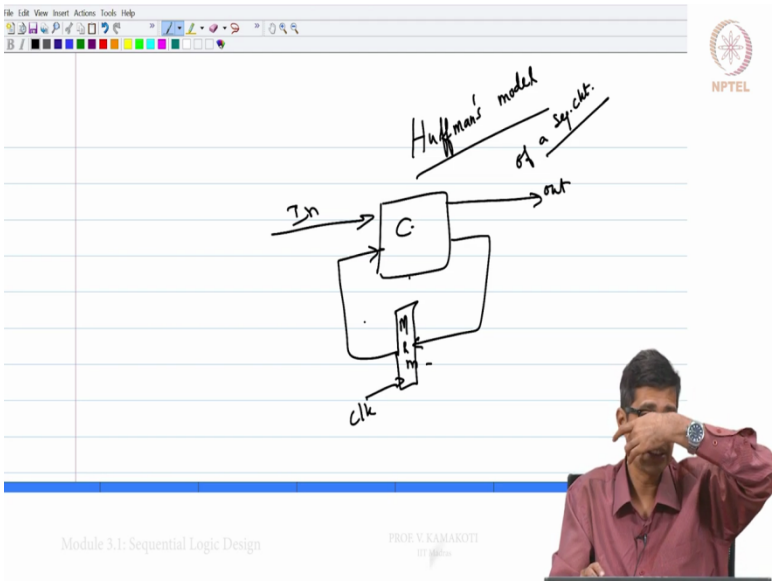
So I need to know the previous, so this is a very simple example where I need to know the previous state for me to progress to the next state. So that means the previous state has to be stored in some value and that will be used. So let me say that each one is a is a step, sometimes you can call it as a cycle, each one is a cycle. So, in the 1st cycle I will have 00:00:00, in the next cycle I will have 00:00:01, like that it will become 00:00:60, sorry, 00:00:59, then it will become 00:01:00 and so on so forth. Okay, right.

So I need to have, I need to remember 3 things, the seconds, the minutes, and the hours and this goes through a logic through which it is fed an output goes here and the output is displayed. So, I store the seconds, minutes and hours and that is given as an input to the logic, right and that gives an output and it gives me, gives the next state here, so I will have a clock. This is called a clock. So what will the clock do?

Every time it will advance, so initially, the values will be 00:00:00, this will be input to this logic, this logic will generate 00:00:00 and it will generate 00:00:01 on this line. So, this will feed this logic and the logic will feed the external world, while it will also try to give, come and try to set this to the next stage. So the 00:00:01 will be here and when will this 0, so these 2 are same, previous state and current state. When will 01 enter here, when the clock is given next. So every clock pulse, you say, clock is, there is some external clock, every clock pulse, this value will keep changing.

So it will change from 00:00:00 to 00:00:59, then after that it has to become 00:01:00. So there is some logic which will take the previous tenant compute what should be the next state. This is basically a combinational logic or combo logic. So this is how it sequential machine is from, so a general form of a sequential machine is called Huffman's model of sequential machine.

(Refer Slide Time: 8:32)




In the Huffman's model of sequential machine what you have? You basically have a combinational circuit, then you have the flip-flops or whatever memory, we will call it memory, here is the output, this is input. This is a generic Huffman's model of a sequential circuit. So I have a combinational block, I have some memory here, so and then I have a clock here. So, we start with 00:00:00, that is fed here, there is no external input here as far this circuit is concerned for the clock, there is no external input, so it immediately came back as 00:00:01 and it was waiting here.

So with the next clock pulse it became 00:00:01 and that is fed here, it comes as 00:00:02, 2-3, 3-4, 4-5, etc, it just keeps rotating here. So this is how the clock basically, the wall clock that we saw or the digital clock that we saw, this is how we work. Necessarily, when I went to make a digital system, I need to have a combinational block and a small memory or scratchpad, which will be storing the previous state of the circuit.

So, any circuit which has these memory components or memory cells inside, those circuits are called sequential circuits. Right. So we will see more on the sequential circuits in the next module. Thank you.