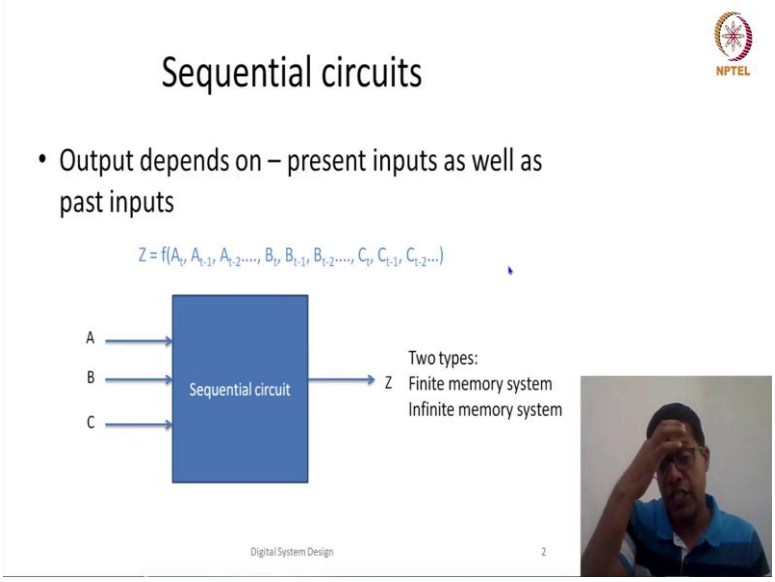
Digital System Design
Professor Neeraj Goel
Department of Computer Science Engineering
Indian Institute of Technology Ropar
Lecture 40
Introduction to Sequential Circuits

Hello everyone, today we are starting this new module sequence circuits and in this particular module we will design building blocks of these sequential circuits.

(Refer Slide Time: 0:31)



Now, let us first define what is sequential circuit whatever we have studied so far whatever type of circuits digital circuits we have studied there, the output depends on the input. So, as soon as the input would change there would be after a certain delay maybe, but after a certain delay output will also change, but output does not depend on any of the previous inputs, output depends only on the present input. So, those kinds of circuits are called combinational circuits.

Today, we are going to discuss another set of circuits which are sequential circuits where output does not depend on only on the present input, but also on the past inputs. So, to understand this, let us understand let us say that we have a system which has A B and C as three inputs and it has Z output.

So, if it would have been a combinational circuit, then Z would be a function of A B and C and A B and C all three would be the present inputs. But in case of sequential circuit, I would define Z as function of present value of A past value of A and all the past values of A similarly, present value of B and all the past values of B. Present value of C and all the past

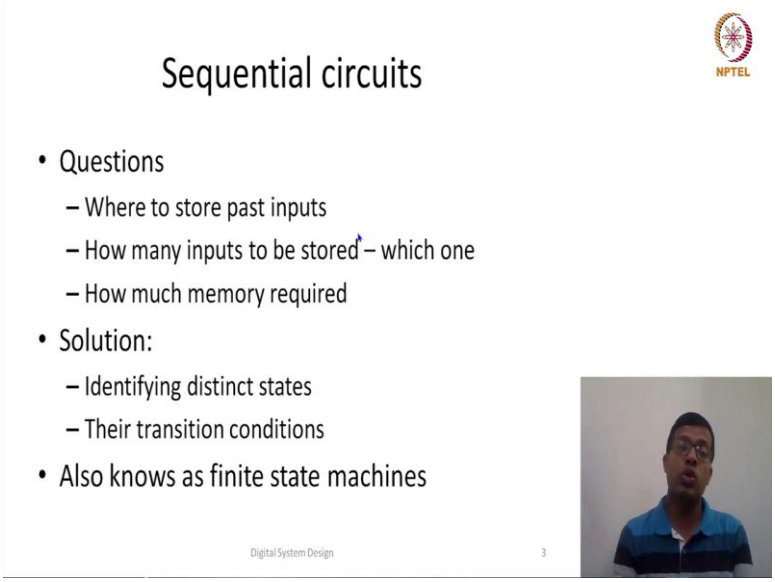
value of C. So, when we say t these are all recorded discrete time. So, at t equal to 0 that means current time the value of A would be AT and at one unit before the present time that means T minus 1 the value of A would be A T minus 1. So, for so and so forth for all other values B and C. So, essentially my Z depends on the current input AT, BT and CT but also depends on their previous changes.

Now, based on that you can assume, you can see that the system the sequential system has to be of two different types finite memory and infinite memory. Finite means how many inputs we have to store or do we need to store infinite inputs or in other words, does it depend on certain number of previous inputs or does it depend on all the previous inputs?

So, if it depends on all the previous input then I can call it infinite memory system. Infinite memory system does not mean that system, this sequential circuit will have infinite amount of memory but infinite memory system means that it is able to memorize infinite series of time series data or that particular input.

So, since starting let us say A input whatever was the first input from first input to the current input it depends on all of them then it is infinite memory system, but finite memory system depends on only on the couple of previous but finite number of previous inputs. So, if it is so, then we have again certain number of questions which arises in our mind.

(Refer Slide Time: 4:11)



That means, a sequential system is able to if it responds to the previous inputs also it depends on the current input also, the question that will come in our mind is where do we store all these past inputs and how many past inputs should I store? Which one should I store? That

will make the designers life miserable because, let us say a function, if it depends on infinite number of inputs.

So, the digital system designing itself will become complicated. We have already seen as we go from number of variables from 3 to 4, complexity of K map increased from 4 to 5 complexity of K map further increased, but from 5 to 6 will become unbearable. So, as the number of parameters of a function will grow, the complexity to optimize that function will grow along with that designing them will also be a very costly affair.

So, but here we are talking about a system which would be able to store infinite input, if it is infinite it is able to represent infinite inputs, then the complexity is going to be huge. And how we will be able to design system what will be the utility of such systems? These are all the questions which are interesting questions, we need to at least think about think upon these questions.

So, one of the solution to store all these infinite input. So, of course, this is another challenge that if we would like to store so many of the previous inputs, the amount of memory which would be required in each other circuit will also be huge, how much should we keep, how much area or how much cells we should keep for that memory.

So, the solution is essentially to find out that if we have infinite amount of previous inputs, then maybe we can categorize those previous input into certain categories, these certain categories how do we identify those categories, that would be the topic of this particular module going forward. So, these categories are called states that let us say whatever could be the number of infinite number of inputs, but we can possibly categorize all those input combinations into certain number of definite states, distinct as well as definite.

So, if we can represent all the infinite inputs, all the previous infinite inputs for a one particular signal, we can if we can categorize all of them into certain distinct categories, certain distinct states, then, it the solution is easy, the solution is easy then in that case, what we can do is we can possibly store only those states.

And we will also try to remember that when we will go from one state to another state or basically in other words, what input combination will come so that I will go from one, system will go from one state to another state. So, if the number of states number of such categorization is finite, then we can design such a system easily.

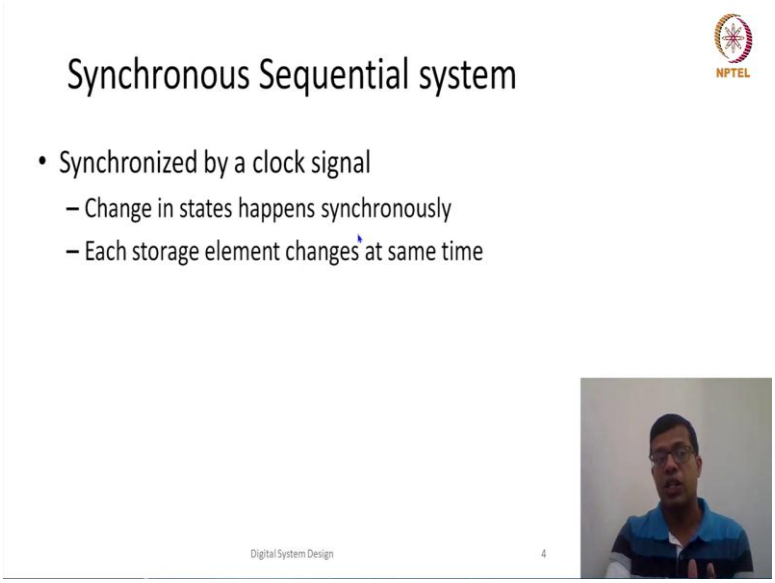
And because the number of states are finite, and we are calling them states, so, these designs the sequential machines, sequential design sequential circuits are also called finite state machines. So, when you will read automata theory at some point of time in your bachelor's degree then also you will come across these terms, where these finite states are there and there is a transition from one state to another state.

So, overall we could understand that if I want to design a system, if I want to design a sequential system, which depends on not only the present input, but also the past inputs, then I will summarize all the past inputs for a particular signal into some finite number of states, if I am able to define that into finite number of states, I will call such as designing a system as finite state machine.

So, these states would require some memory element, so that I can store the state. I need not state store all the infinite inputs the answer of this particular question, where do we store our past inputs, we have to store them in some sort of states and those states would require some storing elements.

So, those elements we will try to design in this module and how many inputs to be stored it depends on the categorization. If categorization says that these are the finite states, those we will store and memory also depends on how many states are there how many bits, how many bytes would be required to store those states would be the memory required.

(Refer Slide Time: 9:39)



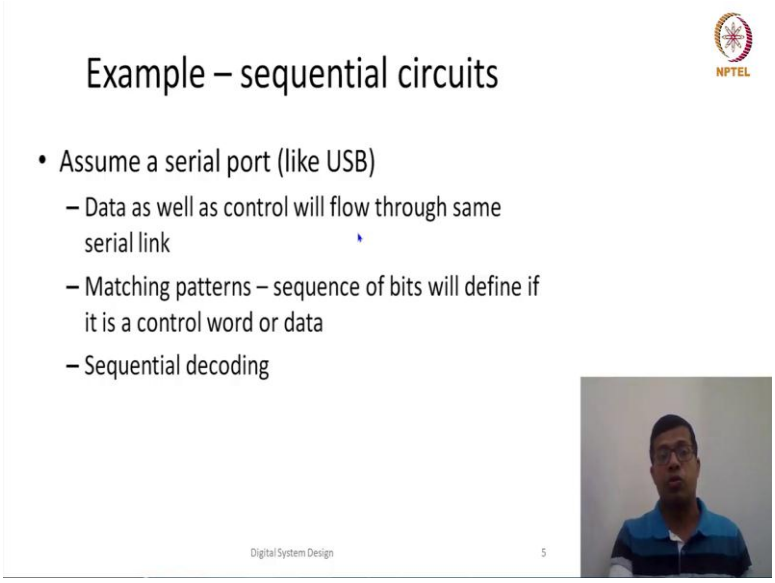
Now, further it has been observed that when the states will change from one to another, if all these states are changing at one common control signal called clock signal, then it is called

synchronous sequential system. So, in synchronous every signal is synchronized with the clock signal.

So, whenever clock will change, then all the state changes would happen and our input will also change usually during the same time. So, output will also be synchronized with respect to clocks, if everything is synchronized with respect to one control signal or a clock that particular control signal is called clock signal, then the system, the sequential system is called synchronous sequential system.

Most of the sequential systems we will find they are they are synchronous in nature mostly because they are easy to design they are easy to understand less number of bug possibility, issues or bugs or faults, they are less probable if the system is a synchronous sequential system. So, to again understand such kind of a synchronous sequential system or in general sequential system let us let us have one particular example. So, that we can visualize at least what we are going to get in this particular module.

(Refer Slide Time: 11:11)



So, all of us have used USB drives or USB sticks, USB in been various form. USB keyboard, USB mouse, the USB is a very popular protocol. And you have also seen how the port looks like so, in general a USB port has 4 pins 1, 2 is the pair of like VDD or power and ground and 2 signals 2 wires to transmit input or output. So, basically 2 pair of wires 1 pair of wire for power and ground, another pair of ground for transmitting data, one pair of signal can transmit only one bit at a time.

So, that is why these kinds of protocols these kinds of ports are called serial port, serial because whatever number of inputs are there, let us say you want to transmit 1 kilobyte from your USB stick to your computer, this 1 kilobyte will go bit by bit. Now, as you will see, as soon as you insert your USB drive into your computer, your computer will detect that it is a phone it is a smartphone, sometimes it will detect that it is a keyboard or a mouse or it will say it is a camera or it is a USB stick, it is a storage device.

So, how does it come to know all of those things because this USB port is not only used to transmit the data, which is stored in USB drive, but also it will tell the system about some other information for example, what is it type of device, how much data to be transferred. So, these are called control information. So, all of this information has to be transmitted and one by one, 1 bit at a time.

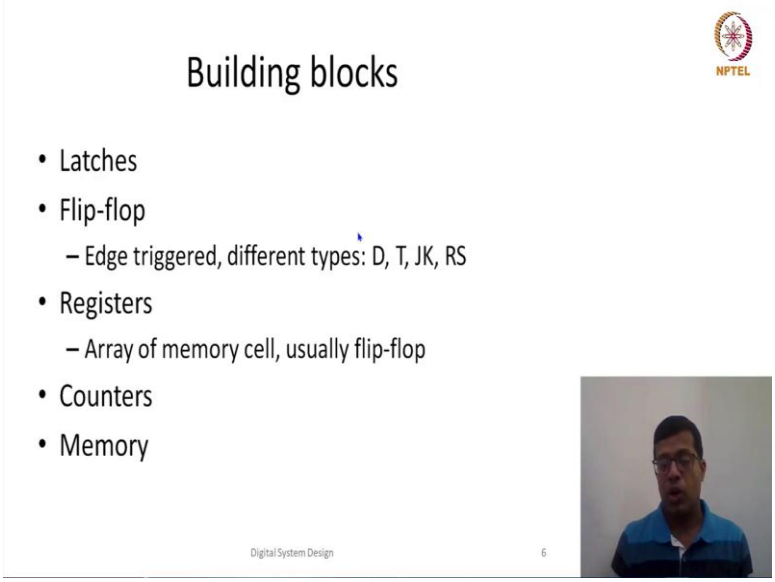
Now, if so much of information, different type of information need to be transmitted, so I would require some sort of a sequential circuit, which will be able to remember that which bits have been transferred already, what type of bits were they? Whether that was a control signal, whether that was a data or whether that control means that this this particular data this this particular signal is for querying the device, what type of device it is, to querying whether the data is correct or not.

So, all of these control questions will also be transmitted through USB and we need to do all of this pattern matching, fixing and basically determining that what should do next, what should your laptop should do next. So, we would require a lot of memory it depends on which port what was the previous bit what was the previous-to-previous bit, so signal is same wire is same, but it has to have some finite memory. So, that it can remember that what was the previous input based on that it can decide what would be done next.

So, in this way you are you are making decision at every point of time you are making decision based on whatever was a past input on the same port. So, this is kind of a one motivating example to understand to use or to study sequential circuits. Here we are not talking whether this particular sequential circuit is a synchronous sequential circuit or non-synchronous or asynchronous sequential circuit.

But we at least understand that some memories required, some state information is required, some categorization is required that what should we do if the previous pattern is this. So, in this whole module by the end of this, so we will classify or like sequential circuits we will study in two different modules.

(Refer Slide Time: 15:40)



In this first module, we would be talking about the building blocks of a sequential circuits. So, as you see that the major building block of a sequential circuit is the memory element where we are going to store all the state information or the information regarding previous inputs, these memory elements form the basic building block of sequential circuits, so this would be our first focus point in this lecture.

Now, these memory elements would be designed either we using latches or flip flops and there are various categories of latches and flip flops. So, latches could be RS or reset set latches, it could be D latches, similarly flip flop could be H triggered and essentially it could be D, it could be T flip flop, JK flip flop, RS flip flop and once we understand the basic elements memory limits like latches and flip flops, then we will also use registers, so basically array of memory cells or array of flip flops to create more elements, so that we can use it you know sequential circuits.

And other related circuits is counters and memory can be designed not only using latches, but other technologies also. So, we have to understand if we had to design some larger memory element or larger memory structure, how do we go and design so, this this particular module will talk about these basic building blocks. Once we will know these basic building blocks then in next module, we will talk about how synchronous sequential designs can be used that those techniques can be used to design any even real-life applications or different kinds of applications.