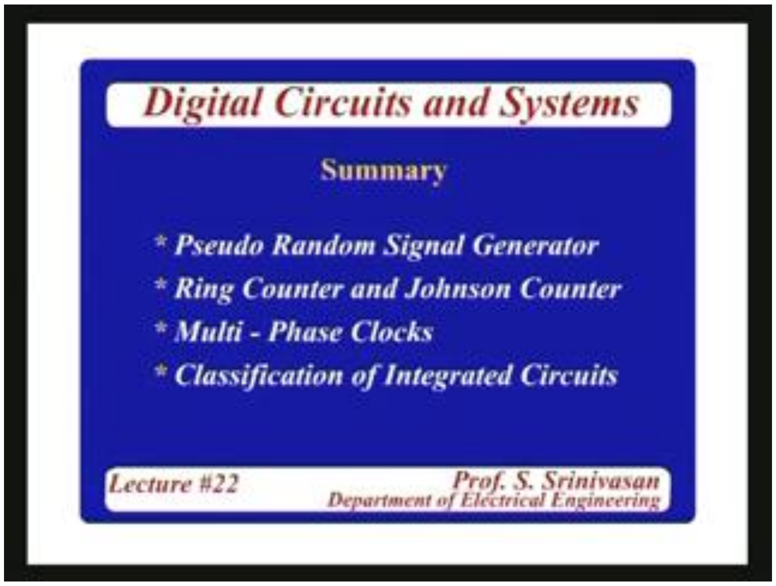
Digital Circuits and Systems

Prof. S. Srinivasan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Lecture # 22
Application of Shift Registers

(Refer Slide Time: 2:03)



Digital Circuits and Systems

Summary

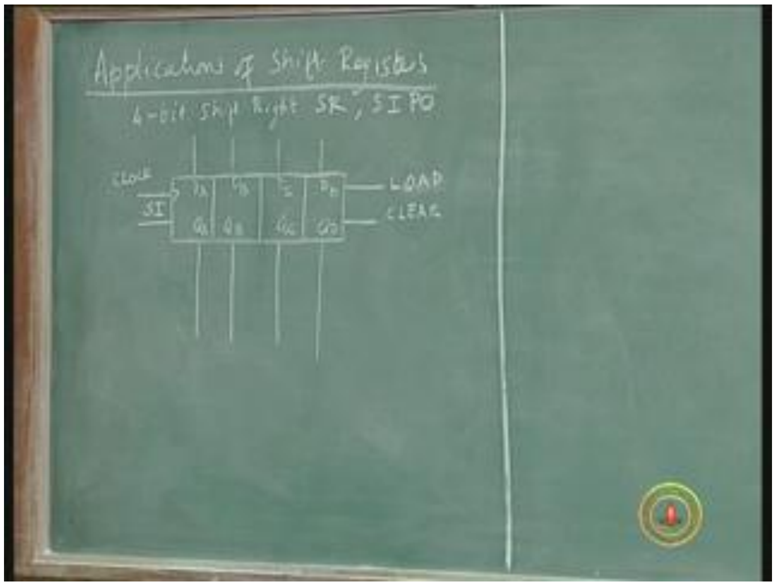
* *Pseudo Random Signal Generator*
* *Ring Counter and Johnson Counter*
* *Multi - Phase Clocks*
* *Classification of Integrated Circuits*

Lecture #22 — Prof. S. Srinivasan, Department of Electrical Engineering

Today we will see some of the applications of Shift Registers. We talked about flip-flops, counters and registers. Registers are group of flip-flops in which you can store data. Then we said the registers can have a shifting feature so that once you put a data you can shift that data either to the left or the right and these are called Shift Registers. There are some other applications, of course we saw applications of flip-flops in counters, likewise an application for registers the most prominent applications of registers in the memory storage, to use them as storage elements. Shift Registers are sometimes used for other applications. I thought we will see some of those applications today, the Shift Register applications.

Let us for example take a 4-bit Shift Register with a shift right feature, so it's a 4-bit shift right Shift Register. Let us assume this is serial in parallel out that means you can put the data serial in parallel out does not mean you cannot put parallel data, parallel input also will be there with a load in feature otherwise normally data is put through serial. So this will have the data in the clock of course clock either positive edge or negative edge doesn't matter, the input is called serial I SI serial input and the clock will shift the bits. We should also have load and clear features because we want to start with a pattern that also you can do by putting the data one at a time and shifting it that is one way of loading and the other way of loading is to give parallel inputs and then give all the bits at the same time. So these are parallel output these are Q all these are A B C D so it is QA QB QC QD and D so DA DB DC DD.

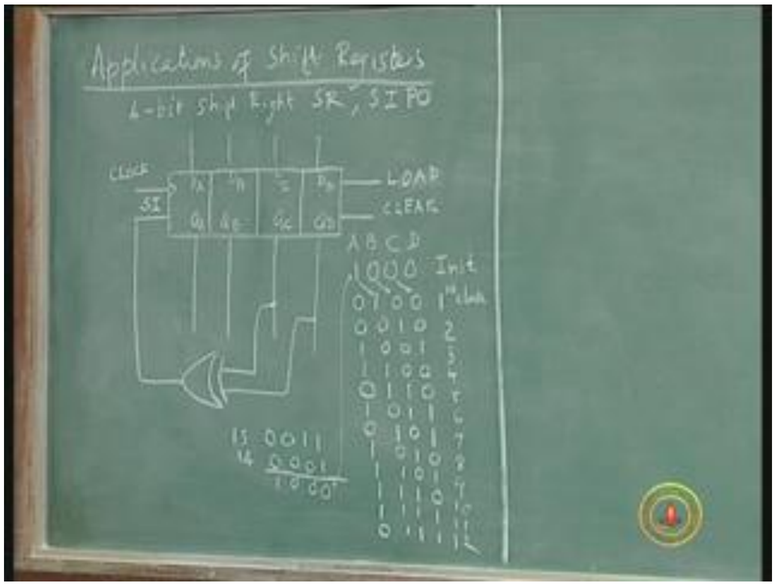
(Refer Slide Time: 5:53)



First you put a pattern in this load it of course you want to clear it later on you can do clearing, then you clock it and as you clock it the bits get shifted as one position to the right that's why it's called Shift Register to the right. So what I am going to do is to take the output of this and this into an Exclusive OR gate and feed it as the input. So every time I clock it this bit goes here, this bit goes here, this bit goes here, and the Exclusive OR of these two gates is loaded as a fresh input. So, as you clock it this will go on. So let us assume that initially I am able to load a pattern of 1 0 0 0 initial pattern but you can assume anything, I am just assuming this.

So to start with I load 1 0 0 0 then at the next clock pulse at first clock pulse what will happen is this 1 gets here, this 0 gets here, this 0 gets here (Refer Slide time: 7:40) and a new bit will be added to the input which will go into QA and since this is the Exclusive OR of C and D what will be the Exclusive OR of C and D? It is 0 so that 0 will get into this position. Therefore again I shift it after the second clock pulse and this will become 0 1 0, this also 0, and after third clock pulse one fourth clock pulse, fifth clock pulse and what will be the new bit, it is 0 again, 1 Exclusive OR of these two so all you have to do is to shift these three bits here and get the Exclusive OR of this here (Refer Slide Time: 8:43). Now after this this will be 1 0 1 0 0 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 then it will be 1 1 1 0 I will write here after this it will be 0 1 1 1 0 then 0 0 1 0 1 0 0 0 which is this.

(Refer Slide Time: 10:05)



So by just putting one initial pattern, this is only an arbitrary pattern it doesn't have to be 1 0 0 0, you can put any other pattern. And again shifting every time the uplift clock pulse the bits get shifted one bit position to the right and a new bit is added to the left then again is shifted and so forth. So what happens is how many different patterns are here? This is the starting pattern so 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 patterns again it repeats. So, after the first clock second clock third clock fourth fifth sixth seventh eighth ninth tenth eleven twelve thirteenth clock fifteenth clock and this is 0 so fifteen clock pulses so the frequency of repetition is fifteen clock pulses or fifteen clock durations. After fifteenth clock pulse the original pattern is restored.

Now there are only sixteen possibilities with four bits so I get all the four except one, I don't get 0 0 0 0 if I get 0 0 0 0 what will happen is it will keep repeating so I should not put a 0 0 0 0 to start with because when you put 0 0 0 0 it gets shifted to 0 0 0 0 again it gets shifted to 0 0 0 0. So, if you leave 0 0 0 0 so you load anything other than 0 0 0 0 I get all the other fifteen patterns and then it gets repeated. Such a circuit is called a pseudo random noise generator. It need not be necessarily noise but it can be pattern you can call it pseudo random generator.

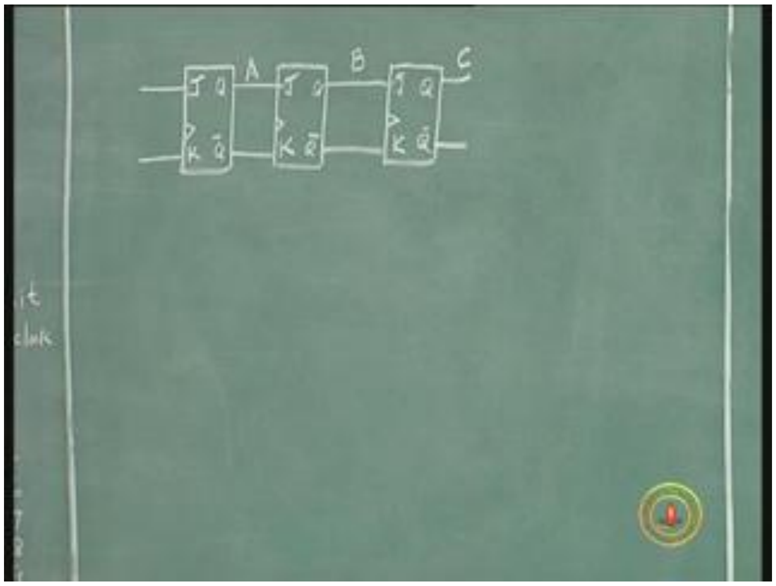
What I mean is suppose you want to have a verification pattern you put a bit pattern which is accepted and for that bit pattern you know the sequence that has to come so you can know whether there is any error in the transmission. If you agree on the initial pattern both at the sending and receiving end for this given pattern you know the sequence of these patterns I think you call it pseudo random sequence generator which can also be used for generation of noise so if you want put it this way. So the advantage of this is I can check the transmission over a period of time so I agree on the initial pattern both at the receiver's and sender's side and then start sending data and as I receive data I check with this and if there is a change that means there is something happening to the transmission.

Why it is called random is because the pattern can be also kept secret in the sense even though all the fifteen patterns will come all the fifteen words all the fifteen different patterns will arise the sequence in which they will come depends on the initial pattern, if the initial pattern is different then you will have a different sequence so again it is fifteen. So it is not completely random and after fifteen it will be repeated so that way it is pseudo random.

Pseudo random means semi-random it's not completely random. You know that fifteen patterns are going to come and if you know the initial pattern you can guess the other patterns so that's why it's called pseudo random. But if you have this pattern that is given as a test pattern I can use it for testing. I can give the initial pattern and send it and test whether I am receiving the other things in the same order. It is one of the important applications of a Shift Register, pseudo random sequence generator.

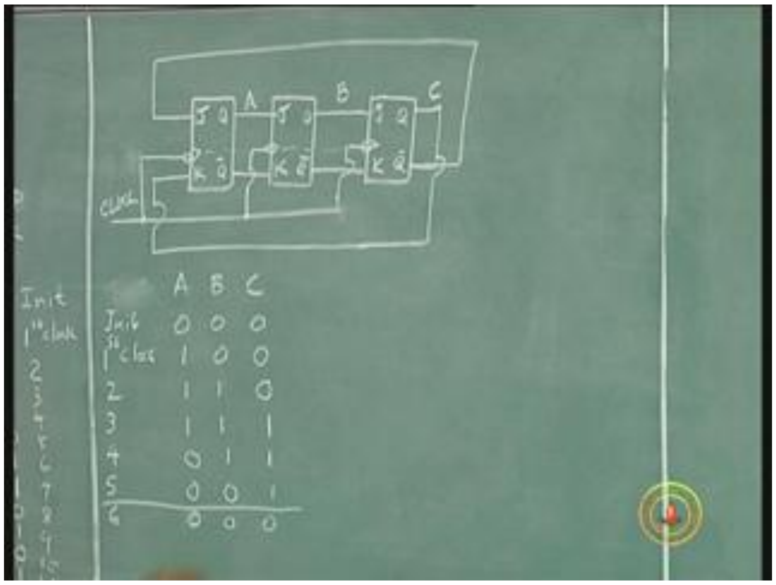
And I can now have different values as I said the starting value so that the patterns will be repeated but not in the same sequence or I can tap instead of here I can do here I can do here (Refer Slide Time: 14:56) I can take this, this and this and Exclusive OR of three inputs I can do, I can do all sorts of variations I can get different sequences and different patterns so I can have a simple test design one is for testing the system and the other is for establishing the reliability of the communication between two points. If the initial test gets through the other side properly then you know that the channel is well established and you can start sending the transmission. This is one of the important applications of Shift Register.

(Refer Slide Time: 16:30)



We will talk about another application of Shift Register and just for the purpose of explanation I am going to put them as three different flip-flops. Let us have JK flip-flops. So I will call these flip-flops A B C and these are clocked together, I will not show the clock separately because they are the same thing, I am just showing this drawing or I can show it separately also doesn't matter. What I am going to do is to connect this j to this Q bar and K so you have three JK flip-flops output of first JK goes to second JK, output of second JK goes to third JK and output of third JK goes to input of first JK with a twist Q and Q bar. This is a Shift Register in the sense that the output of this gets shifted into this, for every clock pulse this output goes into this, this output goes into this, that's why it's called a Shift Register.

(Refer Slide Time: 20:20)

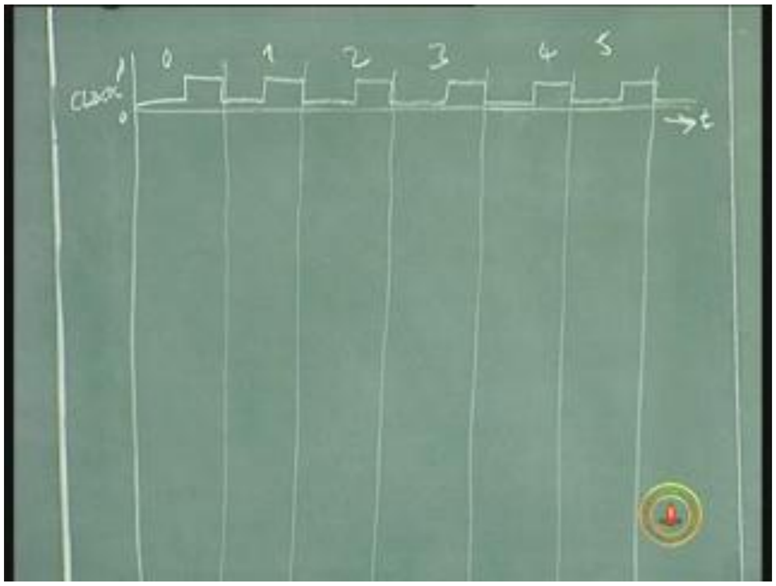


	A	B	C
Init	0	0	0
1st clock	1	0	0
2	1	1	0
3	1	1	1
4	0	1	1
5	0	0	1
6	0	0	0

Now, after the first pulse because the C which is 0s C bar which is 1 is connected to J where J becomes 1 and K becomes 0 after the first clock pulse. I put 0 0 0 I shift it, one clock pulse shifts this 0 into this, this 0 into this, this 0 not back into this but this 0 is connected here in this one. So after the thing the next time the pattern will be 1 0 0. After second clock pulse it will be 1 1 0, third clock pulse it will be 1 1 1 and fourth clock pulse it will be 0 1 1, fifth clock pulse it will be 0 0 1 and for sixth clock pulse it is 0 0 0 which is the same as the original. So, in six cycles of clock I get the pattern 0 0 0, 1 0 0, 1 1 0, 1 1 1, 0 1 1, 0 0 1. Now if you draw the waveforms for this how is it going to look like?

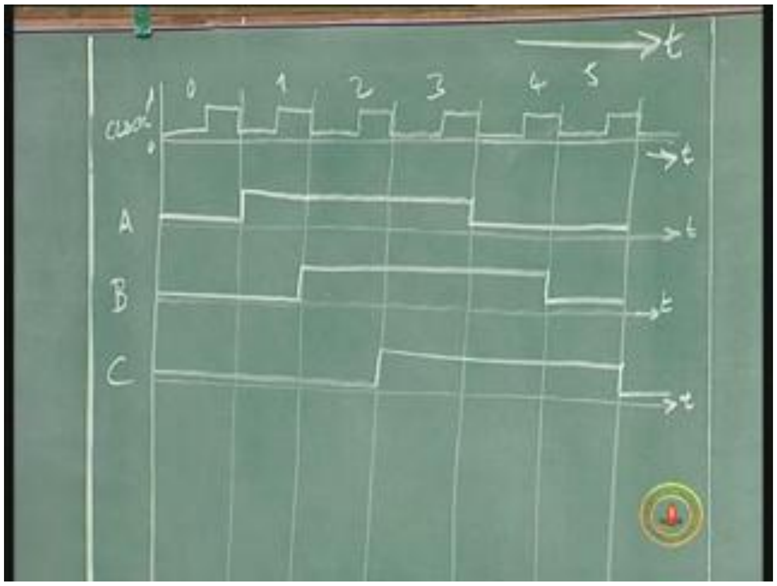
I will draw the waveforms here. Here it is clock clock clock clock, this is 0 of the first clock pulse, (Refer Slide Time: 21:10) second clock pulse, third clock pulse, 0 and this is clock time axis.

(Refer Slide Time: 21:30)



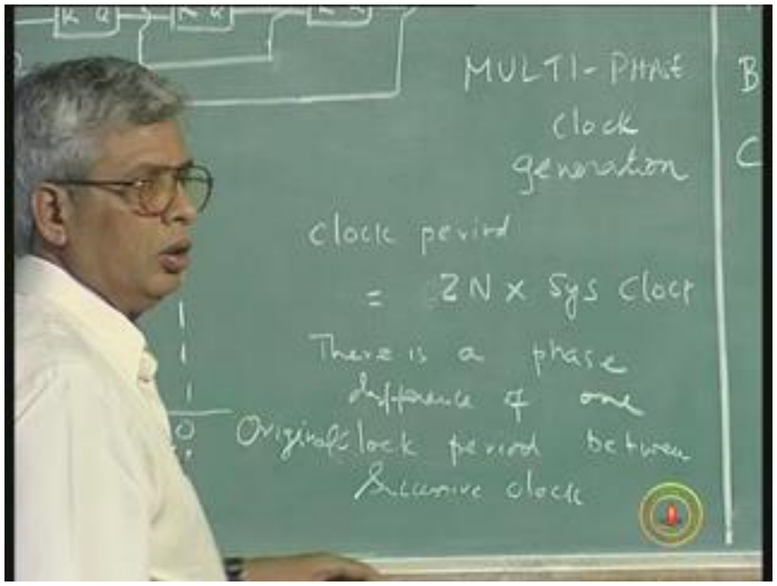
Now if we look at this we can see that initially it is 0 0 0 then A becomes high for three clock pulses and low for three clock pulses and high for three clock pulses and low for three clock pulses it repeats. Similarly for B it remains 0 for first two clock pulses then three and then next three clock pulses 0, next three clock pulses 1 and so forth it repeats. C becomes 0 to start with and for first three clock pulses it remains 0, second three clock pulses 1 then again 0 then again 1. So when you draw this for A B C pattern A would be 0 then becomes 1 remains 1 for three clock pulses and then remains 0 for the rest of the period. this is the output A I will not show the timing separately in the time axis B would be first two clock pulses will remain 0 then next three clock pulses will be 1, C would be 0 for first three clock pulses and 1 for the next three clock pulses.

(Refer Slide Time: 23:20)



Therefore I am dividing this clock by, what is the frequency of this compared to this? Time period of this is six times less, for example (Refer Slide Time: 23:50) this period is equal to six periods of this clock, the period of this clock is one sixth or I can put eight four flip-flops will become eight so the period of this clock is n times the period of the original clock where n is twice the number of flip-flops so I should say 2n. The period of this clock is 2n times the period of the original clock where n is the number of flip-flops that you use for this. That is one thing which you can always get by a division. But more importantly if you want to use this as a separate clock of different phases there is always a phase difference of this much, one third of a clock period, one third of a clock period between this and this, this and this and between this and this (Refer Slide Time: 24:50). I can get a clock period it's called multi-phase clock or multiple phase.

(Refer Slide Time: 26:10)

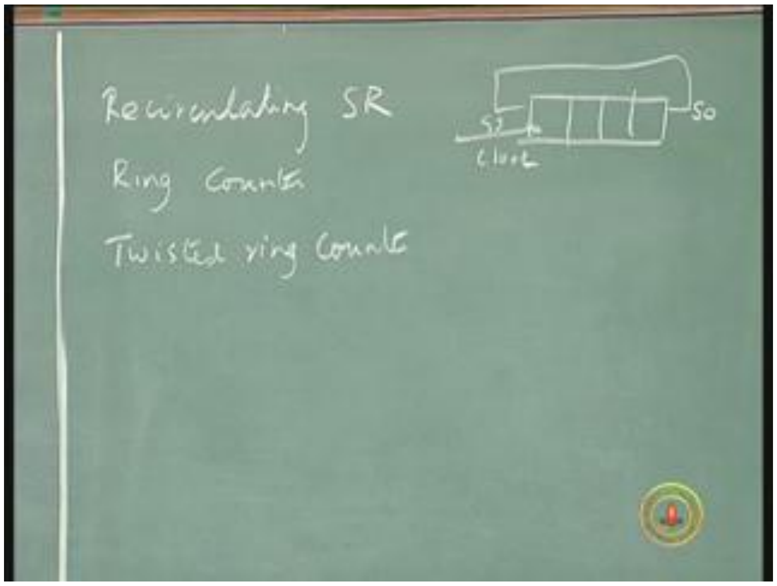


So I have the clock period which is 2n times the system clock period we will call this system clock and there is a phase difference between two successive waveforms of one original clock period between successive clocks. Sometimes it is useful so I want to have some lights going in a sequence. Suppose I turn on lights and turn off lights using this, this light will be on at this point and it will stay on for three clock periods and then it will become off, this light will be turned on (Refer Slide Time: 26: 25) after a delay from this light again it will stay so before that it will go, so first this will light up then this will light up, then this will light up and this will go and this will go and this will go.

Therefore it is some sort of an application like that where you need multiple phases, it's called multi phase clock, you can use it for some system. This is also called a counter. Basically what you do in a counter? Counter will give you a clock pulse divide the clock by a factor of 2 by every flip-flop frequency is divided by 2 in every flip-flop stage so the same similar thing here. This is also a counter but the only thing is the output of this counter we twist and then connect it. In a counter the output of 1 goes to the next, output of second goes to the third, output of third goes to fourth and output of fourth goes back to 1. And without twisting it if I do you can take it as an assignment exercise you connect Q back to J and Q bar J back to K.

Of course you put a 0 0 0 nothing will happen and anything other than 0 0 0 you put and see how it is, it's a Shift Register basically that's all, 1 0 0, 0 1 0 then 0 0 1, 1 0 0 so it will only repeat it is a Shift Register. That's called ring counter a counter wherein we use the output of the last flip-flop back to the input of the first flip-flop is called a ring counter or recirculating Shift Register they call it, you circulate again and again in the same pattern. But this counter is slightly different from the ring counter it's called twisted pair ring counter.

(Refer Slide Time: 29:40)

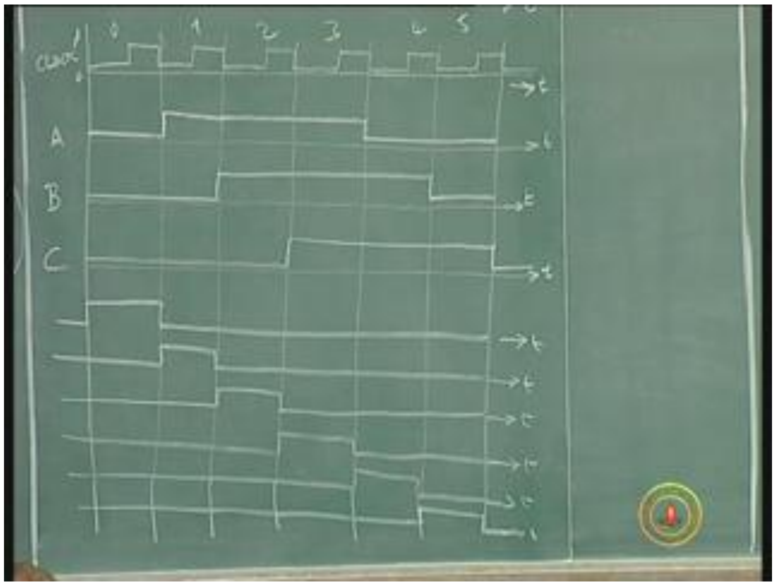


So we have recirculating Shift Registers, ring counter, recirculating Shift Register is same as the ring counter in the sense I have this flip-flops (Refer Slide Time: 29:12) serial output is fed as serial in, put a pattern and keep circulating it that's what will happen in a ring counter. Put a pattern and connect it back even if it is a D flip-flop for that matter, you don't need to use JK. If you want you can use JK too. But this Shift Register which uses twisting of the output back into the input is called twisted ring counter.

What we have seen is twisted ring counter used for multiple phase clock generation and the period of the clock will be one sixth of the period of the original clock with a phase difference of one clock period between two phases. This counter is also called a Johnson counter. Again it's used commonly for multiple phase generation. If you want to have lights turning on and turning off in a particular sequence you can do this, this is one example.

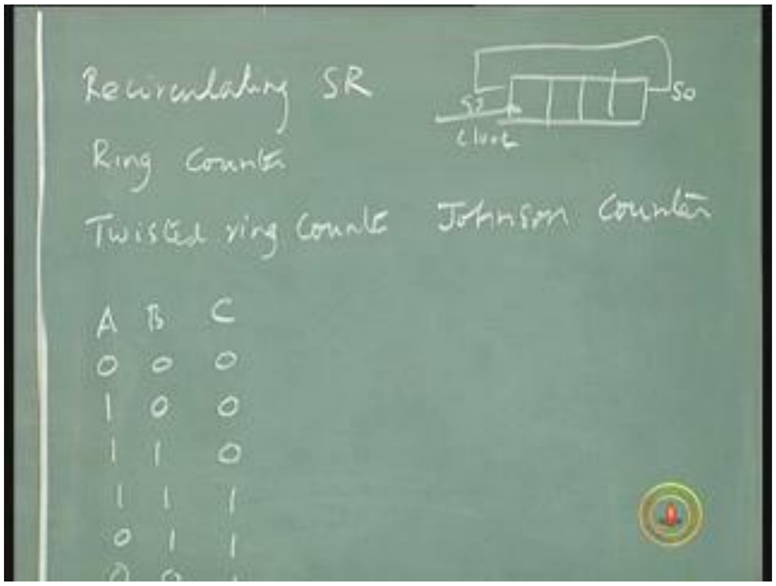
Now, if I don't want this phase to be of this duration one period of clock but I want only one clock period with different phases, that is I want a clock to be on, I want one pulse to be on for this period, one period another clock to be on for this period as we have six periods and suppose I want pulses like this (Refer Slide Time: 31:19), what I am saying is take the same light example, this light is on at this point in time and remains on for three clock periods and then switches off, this light is on after the next clock period but remains on for three more clock periods and so forth. I have three lights turned on, this turns on, this turns on and I can have them on for half the clock period and for other half clock period it starts disappearing and then becomes like this.

(Refer Slide Time: 32:00)



Instead since I want six phases I should have six LEDs or six lights which will be turned on each only for one original clock period. That means I want something to be on for this period alone and off for the rest of the period. It can be a motor, it can be a light, it can be something which you want to just activate for this clock period, another only for this clock period, another only for this clock period, this, this, and this so from these waveforms I should be able to generate these waveforms easily because if you look at this A B C we can also write as we started with 0 0 0 then we made it 1 0 0, 1 1 0, 1 1 1, 0 1 1, 0 0 1 and back to 0.

(Refer Slide Time: 33:35)



Recirculating SR

Ring Counter

Twisted ring Counter Johnson Counter

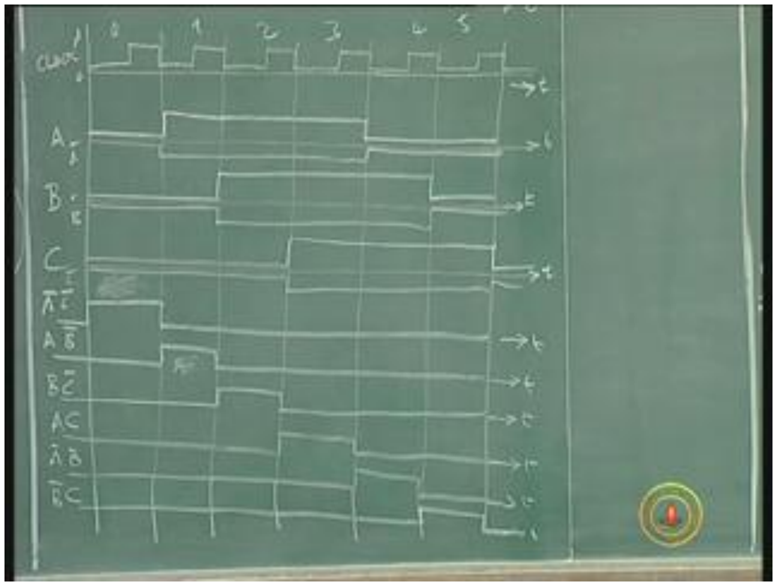
A	B	C
0	0	0
1	0	0
1	1	0
1	1	1
0	1	1

Suppose I take the complements also at the same time so each of these flip-flops have Q and Q bars so I can call this A B C this as A bar B bar C bar so similar to A B C I can draw waveforms for A bar B bar C bar which will again have the same on period for three clock pulses, off period for three clock pulses but that will be complemented to the corresponding clock, A bar will be complemented to A so inverse of A, B bar will be inverse of B and C bar will be inverse of C so again on for three clock period off for three clock period total clock period being six times the original clock period. Now I can see the pattern such that I want something to be on only for one clock period, now you can look at this for example, if you take B bar and C bar or 1 only for the first clock pulse, let us call this the first clock period second third four five six (Refer Slide Time: 35:08).

Suppose I say A bar or B bar C bar, B bar C bar is 1 only for this clock period, there is a change here, you can have A bar B bar AB bar, the first clock period will be on, A bar C bar and not B so A bar C bar is on only for these two clock periods, for this clock period A and A C bar only for this clock period only for this clock period is on high, another change here A B bar, what is third one? B C bar and fourth one will be A C will be there A C will be on only for this, this one would be A bar B and A bar B here I think it is the complement of this, this will be B bar C most probably B bar C complement operations.

Hence, now all I have to do is to get the complement of these waveforms A B C A bar B bar C bar and then use AND gates so in that case the first clock period is on A bar C bar of course I am not showing C bar, I am not showing A bar B bar C bar you can imagine that, it will be the opposite of this (Refer Slide Time: 38:32) and this will be A bar, this will be B bar, this will be C bar. So these waveforms can be clubbed to form A bar C bar here A B bar, B C bar, A C, A bar B, B bar C so I have now variety of choices. I can either have three phase clock with the same period of six times the original clock period so I can have a light on overlapping, three lights on, first light will be on, second light will be on after sometime, third light will be on after sometime and then the first light will be off, second light will be off, third light will be off so you can have this kind of a display.

(Refer Slide Time: 40:00)



If I want to have only a light on for this period first light and then it will go on and second light will be on and it should go and the third light should be on so there are six lights it should be on for one clock period each of the original clock then again it will repeat. Thus all these applications are possible by Johnson's counter or twisted ring counter which is a basic modification of the Shift Register as I said because Shift Register basically is the output of one stage fetched into the second stage and shifted to the left or the right depending on the sequence.

The other two applications we started the other day was the parallel to serial converter. Suppose the serial data is coming I am giving examples of data coming through telephone lines in bit form and I want to assemble it as words serial to parallel conversion or you have the computer generated data which is available as parallel data bytes or words and you want transmit it through a telephone line you have to shift it serially and send it. So other applications are then parallel to serial converter and serial to parallel converter.

So, Shift Register is a very very important building block of a sequential system or any digital system. these are all building blocks of digital circuits, first we saw combinational building blocks, gates, adders and all that, we saw sequential building blocks flip-flops and using flip-flops we saw how to build counters and how to build registers, how to build Shift Registers. Shift Register is a very frequently used component in many digital systems.

I have given you some applications today like the pseudo random sequence generator, the Johnson's counter and then the parallel to serial and serial to parallel and things like that as we talked about the other day. So with this we are more or less completing the discussion of the basic building blocks. We have to talk of some other building blocks in combinational domain, multiplexers and all that. What I have to do now is to talk of systems.

We will use multiplexers, decoders and other combinational building blocks which are not basic building blocks like gates. Gates are the basic building blocks in combinational, flip-flops are the basic building blocks of sequential but I took a flip-flops application and developed counters and registers and Shift Registers like we did in the case of gates we took the basic building blocks of gates and developed adders, subtracters and all that. So now a whole lots of things can be added, these are basically built using the basic building blocks, combinational basic building blocks are gates and sequential building blocks or basic building blocks are flip-flops so now we can use all these things to build subsystems or second order type of complexity and then using them you have to build systems. So we have to get into the act of how to define a system to synthesize it, how to build it. Before that we will just deviate for a minute and stay here now.

Why did I now start from gates and finished gates and then instead of doing the other things like multiplexers and decoders and other programmable arrays which are all combination of building blocks switched to sequential in order to give you all the basic building blocks in both combinational and sequential basic tiles like a LEGO block I told you basic tiles are there it's up to you to play with them and then build castles. Now, why did I do that is because technologically these are all of the same complexity.

Generally we define systems as small scale ICs, all these are Integrated Circuits that is made of silicon but then built as one single circuit, a gate, even gate is an Integrated Circuit in the sense gate is not a component; the basic component of a gate is a transistor may be. a field of a transistor MOSFET as they call it Met……. Semiconductor Filed Effect Transistor or BJT Bipolar Junction Transistor, sometimes you may use extra components like resistors occasionally capacitors. So, using all these things you build a basic building block which is an Integrated Circuit because all these things are integrated, the circuit is integrated in a small single piece of silicon crystal in which you etch this by lots of means.

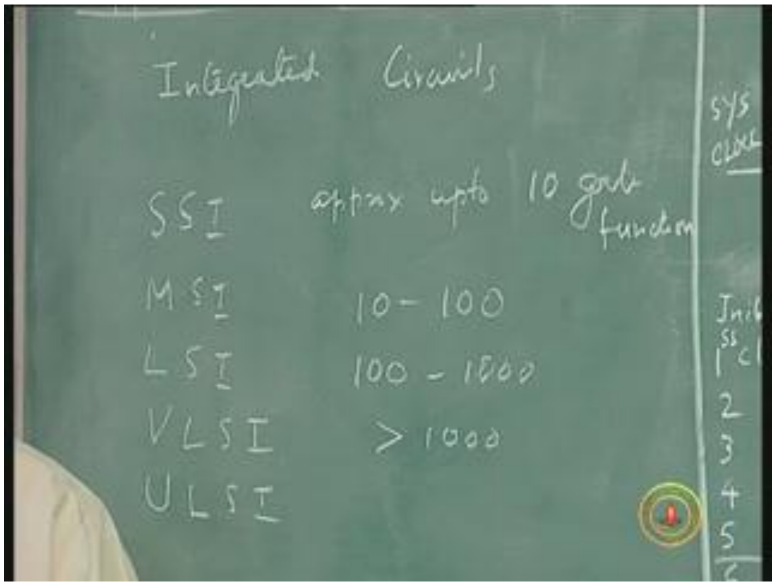
Etching is a very crude term today, earlier it was done by chemical process but today we have so much precision required in all these things, we have lithography using lasers, using electron beam, molecular beam and all that, lithography, so that you can precisely cut. Even in surgery earlier people used to cut open and then do the surgery and then stitch back but today everything is done by precision beams, laser beams and so forth. Likewise in a building block also we have transistors and gates and things like that because the technology has grown very very precisely they are put together in different places and formed in a single circuit these are called Integrated Circuits.

Now, general Integrated Circuits are any circuit in which several components are both active and passive. Active components are transistors, diodes, FETs etc and passive components are resistors, capacitors, and inductors.

Of course active (( )) are easy to make but passive (( )) are little more difficult, resistance is ok, capacitors are little more difficult both in terms of the size and the precision values and inductors are even more difficult. But we will see all those things later on as how they build all these things. But these integrated components Integrated Circuits are classified into small scale Integrated Circuits, medium scale Integrated Circuits, large scale Integrated Circuits and very large scale Integrated Circuits.

A small scale Integrated Circuit is a circuit in which you have about ten gates in one circuit. A gate is an equivalent function and don't think a gate has to necessarily be a gate in the sense, I told you, a gate will have a few transistors and a couple of resistors. So, if you think of that as a basic building block like five transistors or two transistors or something in that order as one unit of gate you call it a gate function. It is actually not a gate as per say but it's a gate function and up to ten gate functions in a single circuit is called Integrated Circuit, these are all approximately.

(Refer Slide Time: 47:45)



Integrated Circuits

SSI approx upto 10 gate function
MSI 10 - 100
LSI 100 - 1000
VLSI > 1000
ULSI

sys clock
Init
1
2
3
4
5

Any circuit which can do an equivalent function of ten gates or up to ten gates like the AND gates, OR gates, suppose you draw a simple combinational logic circuit system few gates here and there equivalent of that ten gates so up to that we call them as small scale integration and then from 10 to 100 approximately these are approximately thumb rules nobody defines this, these are not numbers like Planck's constant or something or anything but it is just to give you the idea of complexity. Somebody gives a circuit you say how many gates are there in that circuit, you give me a circuit and ask me how many gates I say thousand gates but it doesn't mean thousand gates you put, it means that this circuit will do the things which can be done by about thousand gates that's what we mean. From 100 to 1000 it's called LSI and beyond 1000 it is called VLSI but of course today people use ULSI but I don't believe in all that Ultra Large Scale Integrated Circuit but then anything big falls between VLSI and ULSI and all that. Somebody says beyond 100,000 it is ULSI and all that it's all individual's opinion.

So this is basically a structure and why I finished all these basic building blocks and combinational or sequential is because whatever you have done up to now can be generally classified as small scale circuits. When you put more of this gates the functional complexity increases and they become medium scale Integrated Circuits and then you put more and more of them then it becomes large scale Integrated Circuits. So we have now understood the basic building blocks bricks and we will use those bricks to build the subsystems which are medium scale like multiplexers and other things and in the case of 4-bit full adder which may be qualified to be a MSI, similarly a 4-bit Shift Register or 8-bit Shift Register may qualify to become a MSI. then we will do MSI functions and then how to use them directly instead of gate every time because if you are giving a very very large circuit like a microprocessor you can design you cannot go to the gate level and design because a microprocessor may have 300,000 gates or 100,000 gates like that so I can't give you a circuit diagram with 100,000 gates connected to them, it is mind boggling so you have to think of higher levels of abstractions it's called level of abstraction, higher level of abstraction. I will talk about MSI being directly used, LSI being directly used to get VLSI and all that.

So starting from next lecture using these building blocks of both combinational and sequential we will see how to build systems which are meaningful in sense of system design. We have basically smaller systems to start with and later on probably we will do one or two bigger systems. Just to get you some ideas we will start with small circuits.