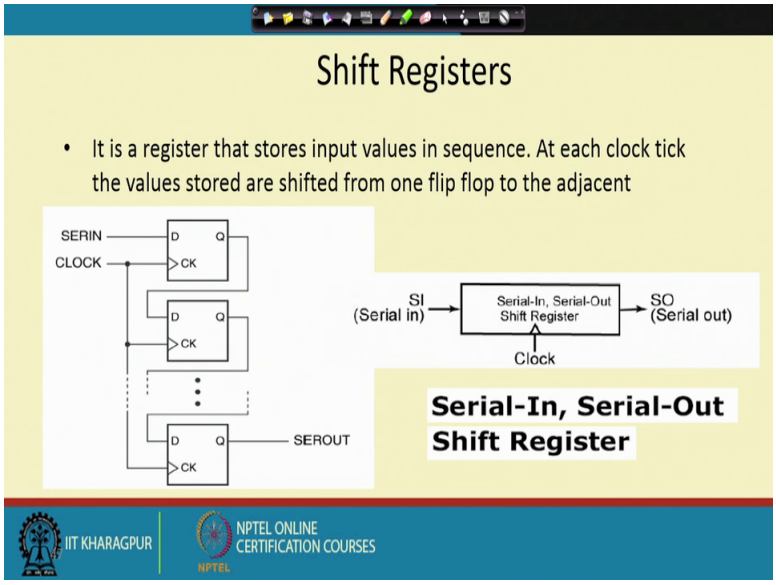
Digital Circuits
Prof. Santanu Chattopadhyay
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture – 32
Sequential Circuits
(Contd.)

So shift registers: so these are special type registers, where instead of loading the value parallel into the register we will load it in a serial fashion.

(Refer Slide Time: 00:26)



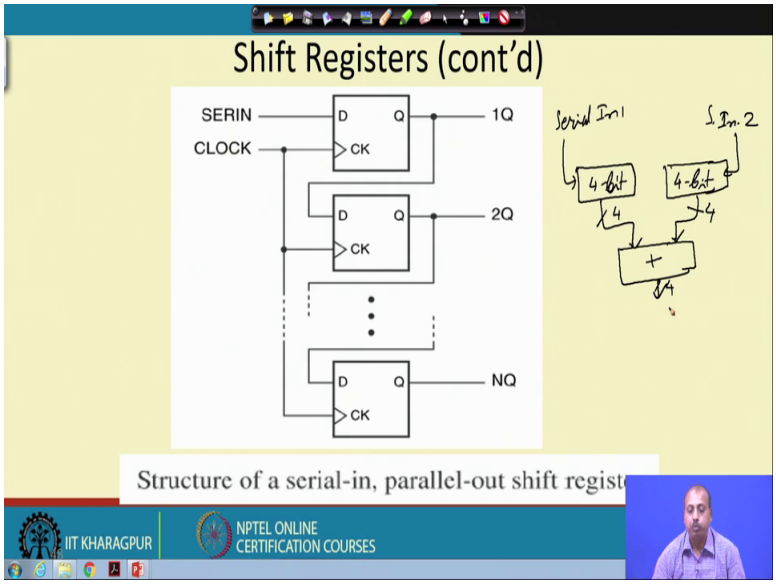
So, here a number of flip flops have been connected in series as you can see here. So, these Q output of this first flip flop is fed to the D input of the next flip flop, similarly Q output of the next the second flip flop goes to the Q input of the third flip flop. And that way we can connect n such flip flops.

So, in a parallel load register this n values the values into these n register n flip flops will be loaded parallelly, so you need n parallel pins to be there in the chip by which you can load feed that n bit pattern. But in many cases what happens is having so many number of pins is difficult in a chip. So, in those situations this serial register they come into use like we have got a single input pin which is serial input. So, here whatever bit we give in the first clock pulse, so that bit get us last on to the first flip flop and whatever was there in the first flip flop gets transferred to the second flip flop in the chain and so on

So, as a result if there are n such flip flops connected in a chain like this, then to load the a particular bit pattern, so we have to shift the patterns n times. The bits are to be fed serially n times an every clock 1 bit will be put into the register. So, the bit for the last flip flop should be loaded first, so that in a in the total shifting process that bit finally reaches the final flip flop.

So, this way we have got serial IN serial OUT type of shift register as it is shown here, so we have got a serial OUT pin. So, output of the last flip flop is coming out as the serial output from the register.

(Refer Slide Time: 02:06)

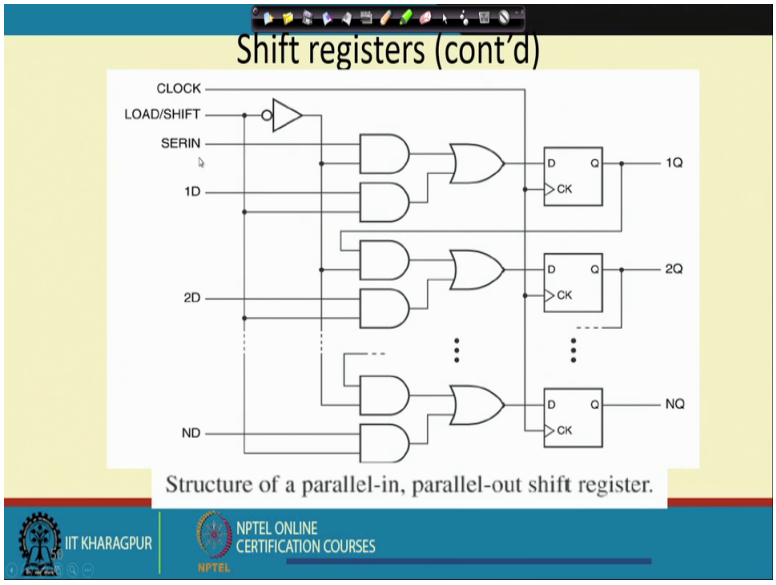


So, we also can have serial input parallel output type of shift register, where the input is coming serially but output is available in a parallel fashion. So, this may be useful in some cases where we need the parallel output; for example, if we have loaded 2 bit patterns into 2 shift registers and just want to sum them up.

So, we have got 2 shift registers like this, so this is 1 shift register this is another shift register and then the individually there may be 4 bit 4 bit. So, they are shift registers so they have got one serial input line. So, this is serial input 1 and this is serial input 2; serial input 2. So, they are coming so after the 4 bits have been loaded, so they may be given to an adder 4 bit adder and that at that time this 4 bits are feed parallelly.

So, these are 4 bit lines coming from the 2 flip flop 2 registers and the result is also 4 bit. So, this type of cases so we need serial input parallel output type of configuration so that is also used in some situation.

(Refer Slide Time: 03:21)



Then we have got a parallel input serial output type of configuration. So, here the values are loaded parallelly, so these load control when this load is equal to 1 then this load signal comes here as a result this AND gate, at the output here also you will get the value whatever value is given at pin 1D so that comes here. And since this is coming through this load shift and inverted, so load shift line equal to 1, so this will get a 0 here. As a result this or gate outputs finally this 1 D value and that gets loaded into the D flip flop.

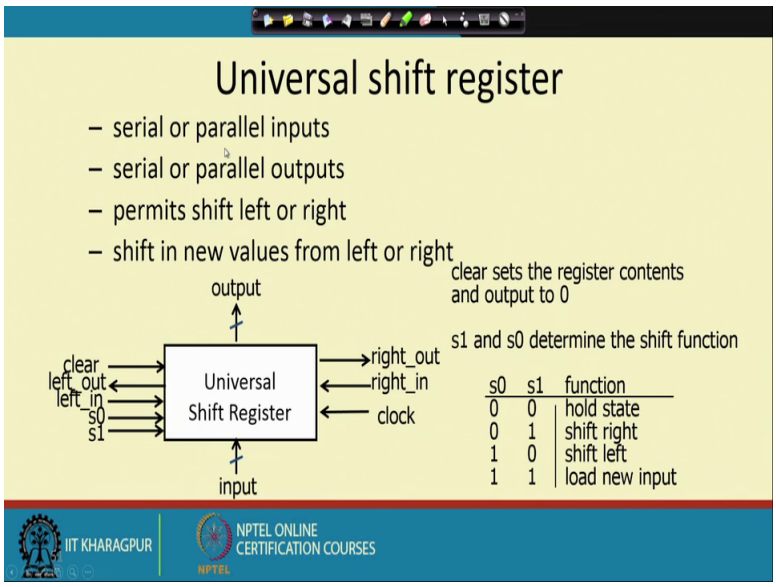
On the other hand the second bit pattern that you have loaded parallel given it parallelly that also gets loaded into the second flip flop, so that way we have got the parallel in type of configuration. Now it is a serial OUT like, if you want to shift out the value, so you have to give the shift you have to make the load shift line equal to 0, as a result this output of this inverter will be equal to 1. So, whatever is coming as serial input so that will be coming to the first D flip flop, so this serial input bit comes to the first D flip flop, output of the first D flip flop is coming as the input of the second D flip flop. So, now it behaves as a serial shift register serial input serial output shift register.

So, we have got this parallel input serial output type of shift register configuration. So, in this you can have different types of shift register configuration, we can also have parallel

in parallel out type of shift register. So, here you can have this parallel input like 1 D 2 D up to ND. So, n possible data bits that you want to store into the register and then we have got this load shift lines. So, if load shift line is equal to 1 then the value from this 1 D 2 D to ND, so that will get loaded into this flip flop.

On the other hand if this load shift line is equal to 0, then whatever is coming in the serial input so they will be loaded into the D flip flop. Also I have got this parallel output available so this for getting the parallel output I do not need to do anything because, I just have to have this extra n number of pins in the chip where this Q outputs are available on those pins, so you have got this those n bits. So, this is basically a parallel input it also has got a serial input and then it is a parallel output shift register. So, it is a parallel input stroke serial input and parallel output shift register. So, this way we can think about different types of shift register configuration and they are used in different applications.

(Refer Slide Time: 06:08)



So, the most generic concern structure of this shift register we can think about is the universal shift register. In universal shift register so it should be able to serial or I should be able to choose between serial or parallel mode of inputs serial or parallel mode of outputs, I should permit shift left or shift right. So, in this diagram so if I say that this flip flops are drawn in a horizontal fashion, so this is the left most flip flop this is the right most flip flop.

So, the shifting that I am doing here is from left to right, but it may be the other way also we can connect these Q output to the previous D flip flops input, similarly finally so this Q output to the to this D input and this serial input line can be connected to the last D flip flop.

So, as a result I can get a right shift type of left shift type of configuration, so that actually gives us the nothing like we can permit shift left or right and we can also shift in new values from left or right, though that is you can you should be able to put a new value into the register either from the left side or from the right side. So, an universal shift register should have so many controls.

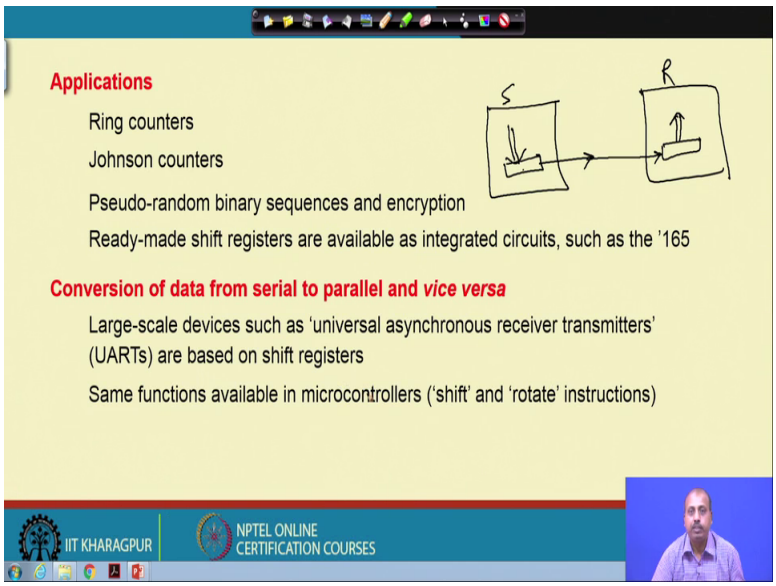
So, that why the parallel input output facility, so the this these are these are the parallel lines parallel input and this is the parallel output I should be able to have this left serial IN and right serial IN then this the left serial OUT here left out and right serial OUT right out and the clock signal is definitely needed there is a clear signal. So, that if it is given so it will clear the up all the flip flops, so it will be register content will become 0 and I need to select between these 4 modes ok. So, there here you see that the register can operate in any of these 4 modes. So, I need to select between those 4 modes and for that in need 2 mode bits S 0 and S1.

So, it may be like this that if S 0 S 1 both are 0 0 nothing happens the flip flop the flip flop contents remain unchanged, so this is the whole state for the register. If it is 0 1 then we are doing a shift right. If it is 1 0 then that control may translate into the shift light shift left command for the shift register and 1 1 it may be for loading new input. So, this way we can think about different types of control different modes of operation for this shift register and we have to set this S 0 S 1 pins accordingly to get the job done.

(Refer Slide Time: 08:49)

So, you can extend the designs that you were doing here ok. So, that it can take care of this is S 0 S 1 controls you can introduce into this design and get the whole thing done. So, I am not showing that design explicitly because the design will become a bit complex, but this is nothing very difficult, so you just have to draw the corresponding combinational logic that should feed the individual flip flops and that will be giving us the full functionality. Rather we will be looking to the applications of this shift register like in which situation we can use the shift registers.

(Refer Slide Time: 09:28)



So, there are several uses some of the sometime we can use it for counter design, so this ring counters and Johnson counters so will see them later. There are some application in pseudo random binary sequences and encryption. So, pseudo random binary sequence basically generate some random number in many applications so we need random number generator.

So, when you are writing a program you in software, so you know that there are normally the operating system provides you with the routines like random number generator, but how those random number generators are working. So, it may be that you have got some random number generation algorithm that runs, but while you are doing in digital circuits how can you have a random sequence.

So, this shift registers they can be used for generating some random sequence and then once you have got this random sequence generators, so they can be used for data encryption type of application. So, there are also readymade shift register that are available in integrated circuit such as the 165. So, they are shift registers are available as chips so you can use it directly.

Another application is conversion of data from serial to parallel and vice versa. So, large scale devices such as universal asynchronous receiver transmitter UART, so they are based on shift register. So, this universal asynchronous receiver transmitter or UART, so they come as a separate chip or they come as part of some processors also; like what happens is that if you want to transmit some data over from one system to another system. So, if you are using a parallel type of transfer then there are large number of lines which has to run parallelly over may be a long distance. But instead of that if we have got less space and we can use a single line because serial if I transmit the data serially then the only one line is needed for doing that and data bit transfer.

So, that many times we need this serial transmission and that is asynchronous in nature because, the 2 system at the 2 ends that we have of that transmitting line so they are operating at their own clocks, so naturally that is an asynchronous system. So, this type of model there are this type of systems we need this module which is known as UART ok, so that essentially use as a this shift register.

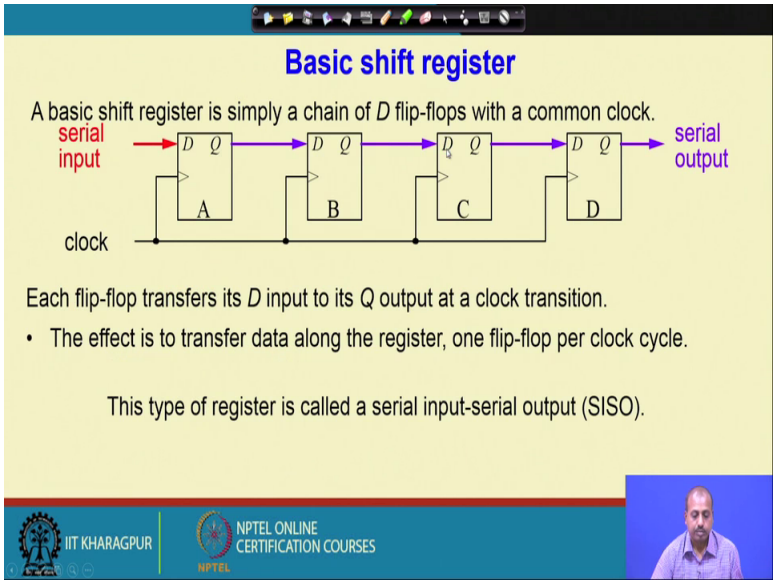
So, from the source if this is the processor from where I am this is the source processor from where I am sending it, so there is a special register here ok. So, the data the

processor loads this register parallelly the value is loaded here, but from here the data is transmitted serially.

So, you need a parallel input serial output type of register, similarly at the receiving end so you have got a similar type of register. So, where this serial line is connected and it outputs the parallel line. So, this is a serial input parallel input serial output type of com combination that we need here and at the receiving end we need a serial input parallel output type of combination, so both the cases so you need some shift register for doing the thing. Some functions that available in microcontrollers or microprocessors where the shift on rotate type of instructions.

So, there also you need to shift the bits of a register by number of places and there also we will see that we will we will have to reduce this type of shift registers. So, this will be more clear when you go to our discussion on microprocessors particularly 8085 microprocessor. So, then it will be clear further.

(Refer Slide Time: 13:18)

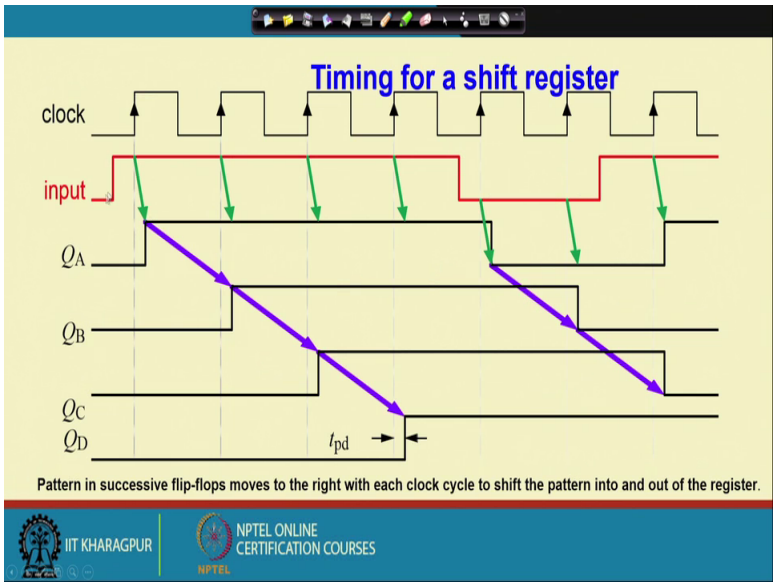


However, so basic shift register from our understanding you see that it is nothing but a chain of D flip flop with a common clock. So, clock is going everywhere and this serial input line is connected to the D, Q output of the first flip flop is connected to the D input of the next flip flop and it goes like this. So, each flip flop it transfers it is D input to it is Q output and at a clock transition; when clock there is a when the when the clock signal

makes a transition. So, this from serial input it is transferred here similarly from this Q it transferred to the next D, so that way it goes.

The affect is to transfer data along the register 1 flip flop per clock cycle. So, it is going 1 flip flop per clock and this type of register is known as serial input serial output register or SISO this is the one terminology which is quite often used, which is SISO register serial input serial output register.

(Refer Slide Time: 14:13)



So, timing behavior is like this so if this is the clock signal that we have got and suppose we have got an input pattern like this ok. So, it was initially low then it is high for some time then it is again becomes low for some time then again high for some time.

Then this Q A that is the first flip flop, so it is fed directly from the serial input, so at the clock transition so it looks into the value like here there is an edge, so this value is sampled and the Q A makes the transition accordingly. So, there is a delay associated with the flip flop so this is the delay that we have. So, after that Q A becomes high then at the further at further clock edge further rising clock edge Q this input is still 1. So, it remains high and it goes on like that so it goes on like that.

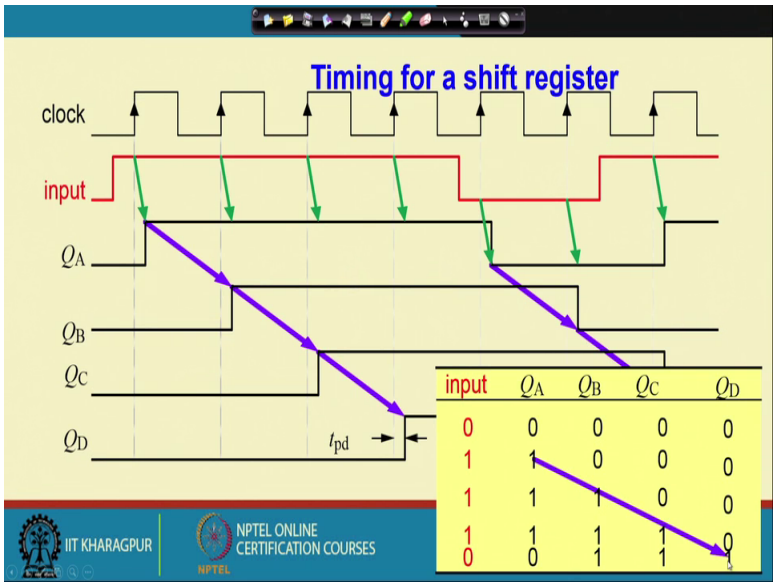
So, similarly at the third clock so it goes high and it when the third clock step it goes high. So, it is also having this input as high, so this value is sampled here so this way it goes on and finally, when it at this particular clock edge it finds that the input is low. So,

the Q A output becomes low and it now it becomes low and continues with the low value till the next clock edge where it is again low, so this is sampled at that time.

If you look into the second flip flop the B flip flop then at the first clock edge it finds that Q A is 0. So, it remains 0 at the second clock edge it finds that Q A is 1, so it becomes high and it goes on like this. Similarly Q C it at the previous clock edge is so it finds at Q B is low. So, till this clock that is no change but at this clock find the Q B is high. So, it will be making a transition like this and it will continue like that.

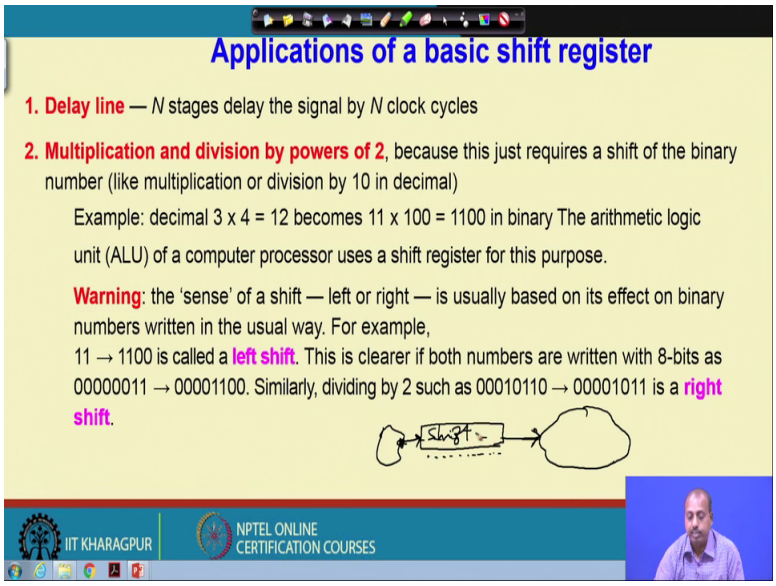
Now, this way so we have got this propagation delay TPD, so this is from the clock edge after how much time the transition will occurs. So, that is the TPD or the transition time ok. So, pattern in successive flip flop moves to the right with the with each clock cycle to shift the pattern in to and out of the register, so that happens in this shift register.

(Refer Slide Time: 16:37)



So, you see how this 1 is getting transmitted so it is shown here. So, this 1 is actually transmitted to this Q D after 4 clock cycles and that has to be taken into consideration that after 4 clock cycle the value will come there.

(Refer Slide Time: 16:56)

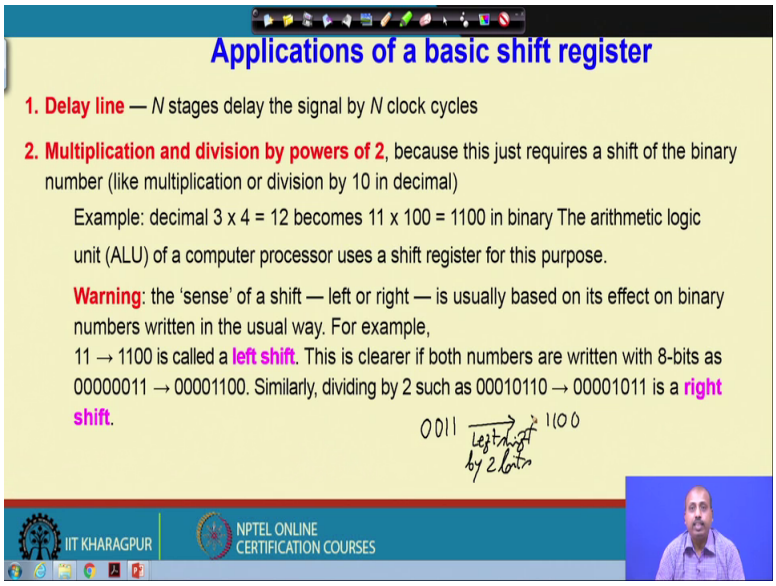


So, another some applications of this basic shift register are like this delay line, like if you want to put an N stage delay that is when a particular clock edge comes. So, you want that the circuit or the actual circuit should get that signal after some time after some N clock pulse this ok. So, you put an N state delay and then the signal will be coming here. For example what I mean is suppose this is my actual system and the signal comes from some other systems as the signal comes here, but what you need is that when this signal is generated after some delay the signal should be available here.

So, what you do in between you put some shift register of sufficient number of stages ok. So, you put a shift register of sufficient number of stages, so that this signal gets transmitted through this shift register, and then finally, it comes to this register. So, we can have this in this way this shift register behaves as a delay line in our in this particular case.

Now, another application is multiplication and division by powers of 2. So, you see suppose we are doing a multiplication by 3 by 4 that is 12. So, it becomes 1 3 is 1 1 and 4 is 1 0 0 so it is 1 1 0 0 in the binary arithmetic. Now you see this when we are multiplying 3 by 4 so it is nothing but we are doing a shifting of this 1 1 towards the left by 2 bits, so this is the left shift operation so what I mean is if I if I take if 4 bit number and 4 bit number representation then this 3 is represented as 0 0 1 1.

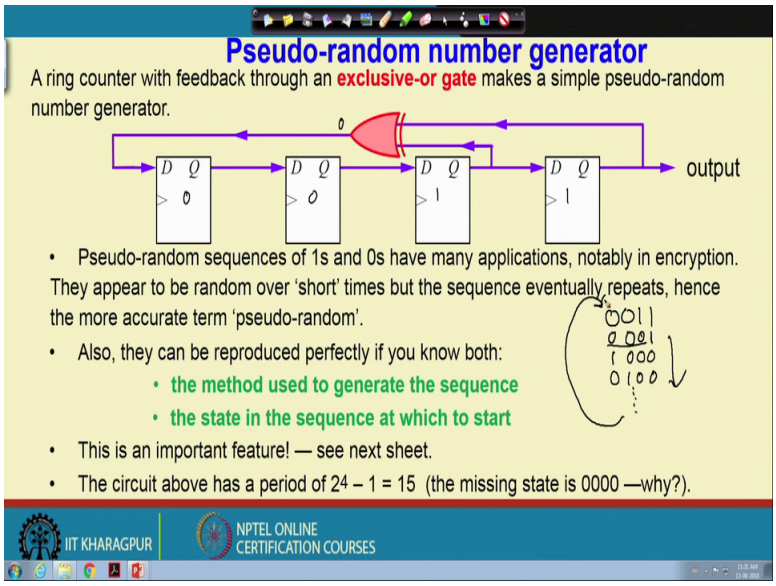
(Refer Slide Time: 18:40)



Now, when I am multiplying by 4 what I am doing is that I am shifting left by 2 bit position. So, left shift left shift by 2 bits. So, if we do that so it will become 1 1 0 0 so that is essentially multiplication by 2. So, this multiplication by 2 for powers of 2 is a very common operation in computer application. So, there we can do that by shifting left by (Refer Time: 19:15) by N bits on the other hand if you are doing a division. So, division by powers of 2 then it is nothing but right shifting the content by appropriate number of bits.

So, this sense of shift left or right so it is based on that it is effect on binary numbers written in the use value; for example, this 1 one 2 1 1 0 0 so this is called a left shift and so this is numbers written in 8 bit format is like this and then you transmit to a shifting towards left. So, this so 2 ones get left shifted by 2 position so it becomes 0 0 0 0 0 1 1 0 0 and similarly when you are dividing by 2, so this is this factor becomes like this so this is a right shift operation. So, this is a left shift and right shift operation that we have.

(Refer Slide Time: 20:07)



Now another application that we have of this shift register is known as pseudo random number generator. So, this is the shift register but only thing is that so these Q output of the last flip flop, so it is connected back to the first flip flop via some XOR gate and this XOR gate it has got some of the bits tapped from this shift register stages and connected to it. For example, in this case the last 2 flip flop output, so they have being XOR and then fed to the first flip flop.

So, this if you now if you start this D flip flop, if you initialize this D flip flops with any value other than all 0, then what will happen so it will be it will be based on the these values in this these 2 flip flops, these shift value will be determined accordingly it will it will generate a sequence. For example, we can if we say that we started with the pattern say 0 0 1 1 that is these 2 flip flops are 0 and these 2 flip flops are 1, then what will happen so these 2 ones, so they will be XOR here so you will get a 0 at this point. So, at the next clock so this is a shift register otherwise. So, by the shift register operation, so these 3 bits will be 0 0 1 and this bit will become 0.

Similarly, in the next clock so this 1 1 will be going out. So, you will be getting 0 0 0 so this 3 zeros will be right shifted by 1 position and this 0 and 1 they will be XOR and coming as input to the first flip flop, so it will become 1 0 0 0 0. So, at the next clock so you will get this 1 0 0 0 here and then this 0 0 will be XOR and you will get a 0 at the

first flip flop. So, this way it will go on generating a random sequence first it is 3 then 1 then 8 then 4 after generating a random sequence.

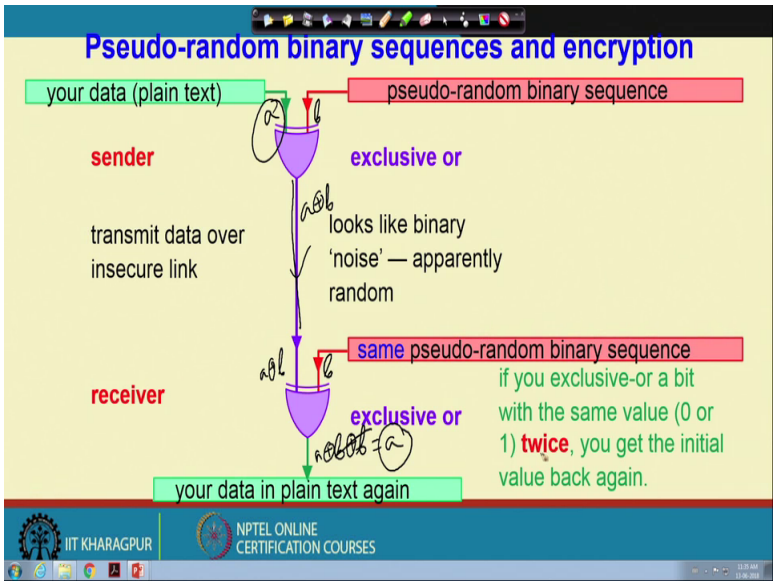
So, this pseudo random sequences of ones and zeros have many applications particularly in the field of encryption will see, they appear to be random over short time period and because the sequence will eventually repeat. What will happen is that in this way if you to if you trace out this sequence then after generating all the 15 nonzero pattern, so it will again come back to this points. So, it will start generating from the beginning.

So, that pseudo random that is why it is called a pseudo random sequence not a truly random sequence because after sometime the sequence repeats itself, so it is a pseudo random sequence and they are they can be produced reproduce perfectly if you know the following thing the method used for generating the sequence, that is what is the structure of this what is the structure of this shift register. That is, but in particular what is the where from which pints from which flip flops you have taken inputs to be XOR to be fed back to the first flip flop

So, if you know that information and you know what is the initial pattern that you have fed; so given these 2 information you can reproduce this sequence at any point of time ok. So, that is why we must the method used to generate the sequence which is basically this XOR tapping points and the state in the sequence at which it is start. So, that is the first value of the of the shift register and this in this particular case after fifteen cycles it will be return it will be repeating and the as I said that the state 0 will not come because, if you load 0 then it will always remain in the 0 value.

So, it will not be having any other pattern here, so it will always be at 0. So, it cannot generate the sequence 0 apart from tat it will generate all other patterns.

(Refer Slide Time: 24:08)



So, what do you do so this pseudo random sequence generate they have got application in encryption. So, in encryption what happens is that you have got some plain text which is combined with some key value and accordingly it will generate some encrypted result ok. Like in this particular case suppose this is your plain text is coming, I am considering the plain text also as a bit sequence although it is coming here and you have got pseudo random binary sequence, so that is generating the pseudo random pattern.

So, the successive bits that are generating so you are you do XOR of them and then you see; what is the output since it is XOR. So, naturally you the nature of this plain text will be lost and you will see something like binary noise ok, so which you which will appear to be apparently normal. So, at the sender end we do it like this we XOR this plain text with this pseudo random binary sequence generated by the shift register and then we just after doing the XOR we start transmitting over this channel.

So, at the receiving end what you do you use the same pseudo random binary sequence. So, you are using shift register where this feedback points are same to the first XOR gate. So this, the same structure is used here and also you start at the same initial pattern. So, once you start at the same initial pattern then the same sequence of random numbers will be generated here and that you can XOR with your plain text. So, there that so since this if it is your plain text is say A and this if your plain text is A and this is B. So, at this point I am getting A XOR B.

Now, what is happening is that here I am using the same pseudo random sequence generator starting with the same initial pattern. So, this is basically B so at this point so this is A XOR B and at this point I get A XOR D XOR B. So, these B XOR D is equal to 0 so I am getting back A.

So, whatever A you are transmitting you are getting here, but in between in this while it is getting transmitted through this channel. So, this is encrypted as A XOR B so this is the property that if you exclusive or bit with the same value 0 or 1 twice you will get the initial value back, so that is the property of XOR. So, that is used for this encryption algorithm. So, this pseudo random sequence generator it has got a application in this encryption algorithms.