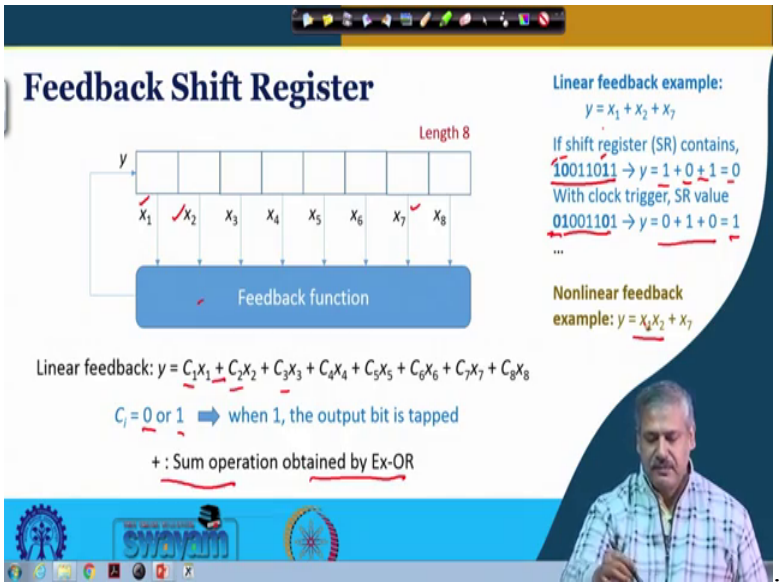
Digital Electronic Circuits
Prof. Goutam Saha
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 39
Linear Feedback Shift Register

Hello everybody. In the last class we saw certain application of shift register and there we saw that a feedback of shift the serial out as serial in can give rise to Sequence generation Ring counter, Johnson counter. We shall look at what is called Linear Feedback Shift Register in this particular class and its different uses.

(Refer Slide Time: 00:37)

(Refer Slide Time: 00:45)



So, by linear feedback shift register what we actually mean that the shift register outputs say it is say length 8, you will consider x1 x2 x3 like up to say x8, ok.

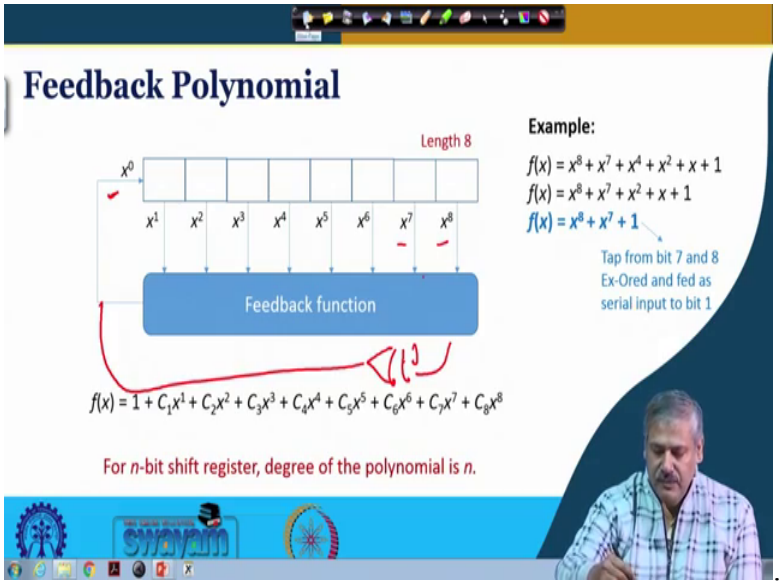
So, it is going to a block where there is a feedback function and that feedback function is generating an output, and that output is getting feed as serial in and when we say linear, so this general case linear means this function has got a linear relationship, the equation that you are having is a linear relationship, ok. So, in this case we write it in this manner C1 x1 C2 plus C2 x2 to C8 x8 and these coefficients are either 0 or 1, is a binary world. So, it is 0 or 1. So, 0 means there is this value is not used, ok. So, in analysis there is no tap from this shift register output to the this feed feedback function generation logic, ok.

So, that is not there and if there is 1 present that means, that particular tap is present. So, that is how this CIS are defined either 0 or 1 tap is present or tap not present and this plus is the sum operation which is obtained through Er-OR. So, this is the way the circuit is developed and if we look at an example, so y is x 1 plus 6 2 plus say x7, right. What does it mean? So, x1 x2 and x7.

So, these three taps are used, these three taps are used and they are these three inputs are summed up here. These three outputs of this shift register is summed up here and fed as input to the input y as serial in. For example, if the this tree study is initially loaded with this value 1 0 0 1 1 0 1 1 ok, then x1 x2 and x7 are these value the one in the bold, right.

Then y will be 1 Ex-ORed with 0 Ex-ORed with 1. So, the value is 0. So, 0 will fed back. So, with clock trigger what will happen. So, this 0 will come here. So, everything else will get shifted 1 0 0 1 1 0 1 will come over here. So, at that time your x1 x2 x7 are the one in the bold. Again here 0 1 0, so 0 1 0; so, now the value is 1. So, next value one will be getting it, and rest of the values will be shifted, and when we say non-linear that means other than sum operations. So, this is x1 and x2, then sum x7. So, these are the things that comes under non-linear feedback which we not we are not discussing here; our topic is Linear Feedback Shift Register.

(Refer Slide Time: 04:11)



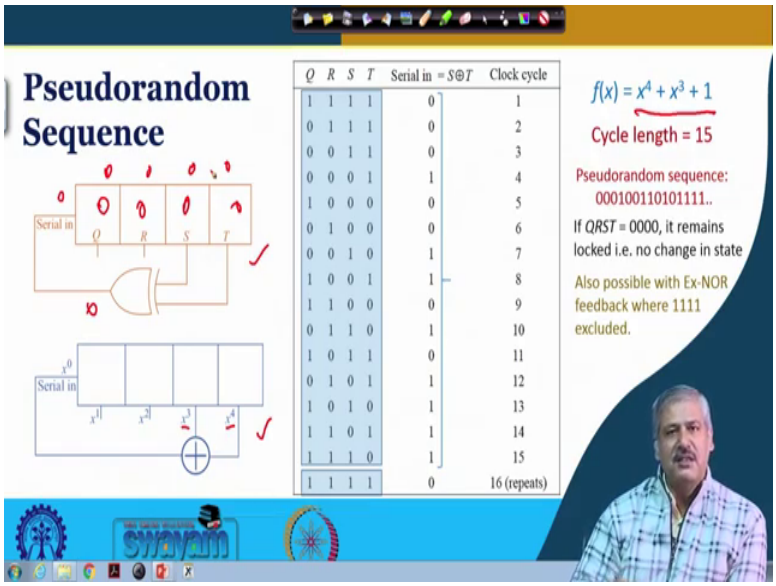
So, what we have? We have seen that is redefined in a manner by which we introduce the term feedback polynomial. So, in these instead of x1 x2 up to x8, we represent the output of this shift register of length 8 as x to the power 1, x to the power 2, x to the power 3, x to the power 4 etcetera up to x to the power 8 as if you know each one of them is delayed by 1 8. So, delayed by 1 8 the delayed by 2 units d into d square. So, d cube d 4. So, this is this is the idea, ok. So, in this sense you have got x to the power 8 up to here, right and the input to it, it is not delayed. It is the out it will come to the output after one clock trigger. Isn't it? So, this input is represented as x to the power 0 or 1 x to the power 0 or 1.

So, this one you will see in the polynomial that the function that is you know getting represented and usually in many cases since it is length 8 requirement is of that of length

8. So, x a x to the power 8 is also would be visible if it is of in. So, if it is of length n, so x to the power n is also expected to be there, but there could be another way the polynomial can be generated without that and in between wherever the connection is there; that means, this C i values are 1. This type is there otherwise it is 0, right. So, for n bit shift register with the degree we say will be of n 8 bit, here the degree is 8 and these are some examples and if it is x8 x7 and x1, ok. So, x8 x7 and x1, so x 8 x 7 and 1 is this 1. So, what does it mean actually?

So, these two are getting Ex-Ored and fed as input. So, this is the meaning of this polynomial. Is it ? From bit 7 and 8 are Ex-Ored and fed as serial input to bit 1.

(Refer Slide Time: 06:47)



Now, such an arrangement can give rise to can generate pseudo random sequence, sequence generator we have seen. So, where we had seen a particular length if it is 8 bit, it is know it bit long it was right here we see what happens when we connect in a particular manner; so here that is 4 bit shift register. So, this 4 bit shift register, this is s and t is observed it is fed as serial in. So, in terms of polynomial feedback polynomial just now we have defined this is x to the power 3 and this is x to the power 4, and this and this input is 1. So, we will be representing this as x to the power 4 plus x to the power 3 plus 1, clear.

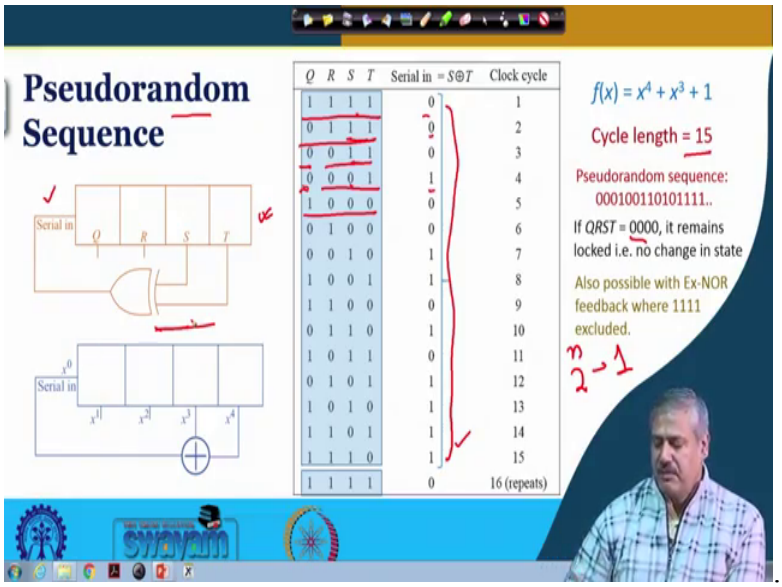
This arrangement can be re-written as this the meaning is this end, ok. So, if this thing happens, then let us see how the with clock these states will evolve and what will be the

serial in each case. So, it is if it is initialised with say 0 0 0 0 what happen if it is all 0 0 0 0. So, 0 0 Ex-Ored output is 0, ok. So, this 0 will be fed in. So, again next value will be 0 0 0 0; so it will remain locked. So, it is not to be initialised with 0 0 0 0. So, let us consider instead of all getting cleared, all was pre-set ok. All the flip flops where pre-set and it was initialised with 1 1 1 1. If there was no pre set option, then it was really or parallelly loaded, but somehow we have initialized the shift register with a value 1 1 1 1 ok, then what will happen this S and T are Ex-Ored 1 and 1 is Ex-Ored. So, serial in will be 0; so, next value will be 0 1 1 1.

After that this 1 and 1 again this 1 and 1 Ex-Ored, it is 0. So, again 0 comes and the rest of the things gets shifted; so it is 0 1 1. So, then this 1 1 again 0; so this is 0 0 1. Now 0 and 1, this is 1; so this is 1 0 0. So, this way you will continue you can work it out and you can see you can see unlike this sequence generator which was of you know you know length 4, in this case this output which is the you know you can take as final output of it continuous up to a length which is 15.

After that again it starts repeating. After that it again it starts repeating and the numbers you can see you here these 0s and 1s, it is what is known as pseudo random, ok. It is not exactly random in the sense because it will repeat after some time and if you have number of these n bit shift register, the value of n is large then of course you can understand that the length will be very large.

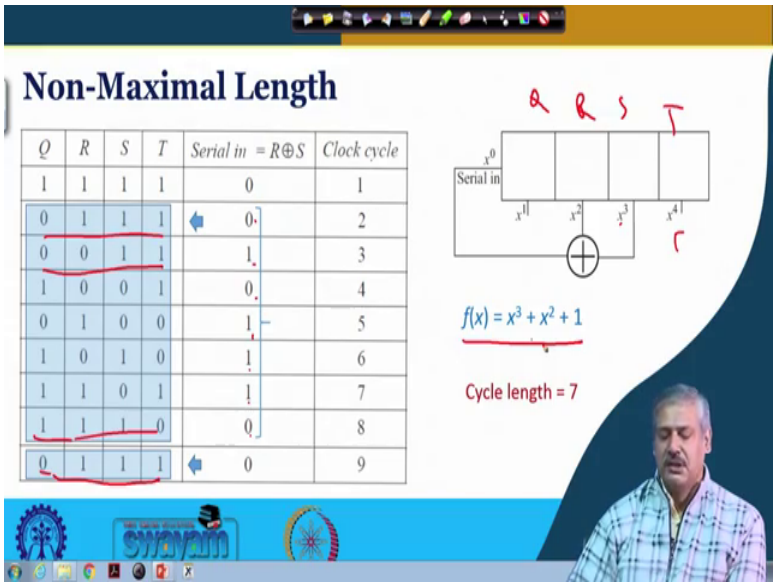
(Refer Slide Time: 10:07)



And the formula that is used for such case is the maximum length that is possible is 2 to the power n minus 1 2 to the power n minus 1. All these 0 0 0 0 that that particular state is not acceptable, ok; rest of the things are getting. So, this rest of the things are getting circulated, ok. So, with a higher value of n you can understand that these will get represented after a long value after this sequence will be very long, and almost equal probability of 0s and 1s that is present in this particular sequence, clear.

So, this is a one particular way you can generate pseudo random sequence using shift register, and you require minimum you know hardware you know additional hardware and you can move very fast, ok; these things can be done in very fast manner.

(Refer Slide Time: 11:11)



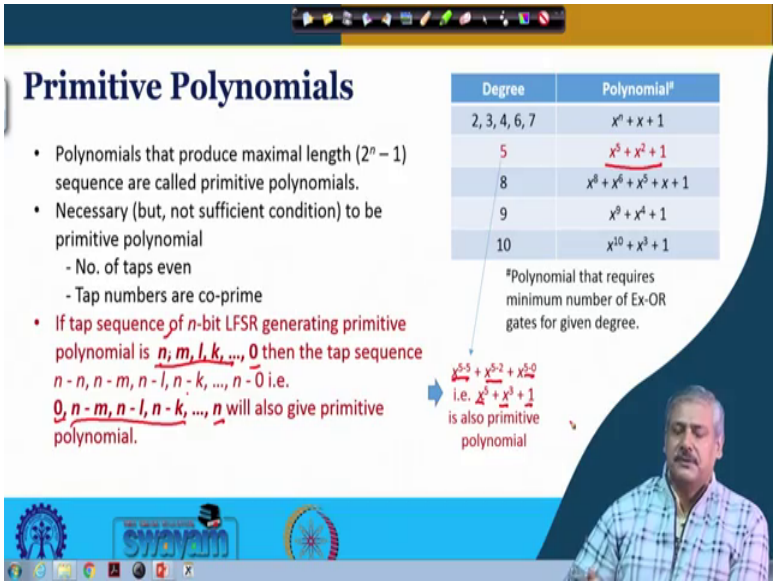
Q	R	S	T	Serial in = R⊕S	Clock cycle
1	1	1	1	0	1
0	1	1	1	0	2
0	0	1	1	1	3
1	0	0	1	0	4
0	1	0	0	1	5
1	0	1	0	1	6
1	1	0	1	1	7
1	1	1	0	0	8
0	1	1	1	0	9

Now, it is to be noted that the feedback that you take any kind of feedback we will not give maximal length, ok. So, for example, again for a the 3rd 4 bit shift register, we are talking about instead of x 4 and x 3 if it is x 3 and x 2, there Ex-Ored and fed as serial in. So, the corresponding polynomial is x3 x to the power 3 plus x to the power 2 plus 1 and we initialize it with say 1 1 1 1, then this serial in is 0. So, 0 1 1 1 and that way you continue.

We shall see that over here it becomes 1 1 1 0, ok. After that again this 0 comes over here, this is T ok, this is Q R S T, this T comes here. So, it becomes 0 1 1 1. So, this 0 1 1 1 again repeats here. Of course, after that it will repetition will start. So, 1 2 3 4 5 6 7. So, this cycling becomes 7. Earlier when it was taken from x 3 and x 4 x to the power 3 and

x to the power 4 and it was 1, it was maximal length that was 15 possible 0 0 0 0 0 was ruled out, ok. So, it will remain long. So, this is to be noted.

(Refer Slide Time: 12:53)



And that gives us brings to the discussion of discussion on primitive polynomials. So, not all polynomials will give us maximal length. So, d 1 that gives us the maximal length are called is called primitive polynomial, ok.

So, the example that you can see; so these are the polynomials that requires minimal number of minimal number of Ex OT gates for a given degree; so, provided a table. So, for degree 2 3 4 5 6 7 ok, so x to the power n plus x plus 1 that will give a primitive polynomial; this is not the only one. I shall tell more about it shortly. So,, but this will give you a primitive polynomial, right. So, for example if it is a 7, so x to the power 7 plus x plus 1, 1 is the input that is coming here. So, 7th flip flop and the 1st flip flop output right Ex-Ored together and fed as input, it will give a maximal length and maximum maximal length will be 2 to the power 7 minus 1.

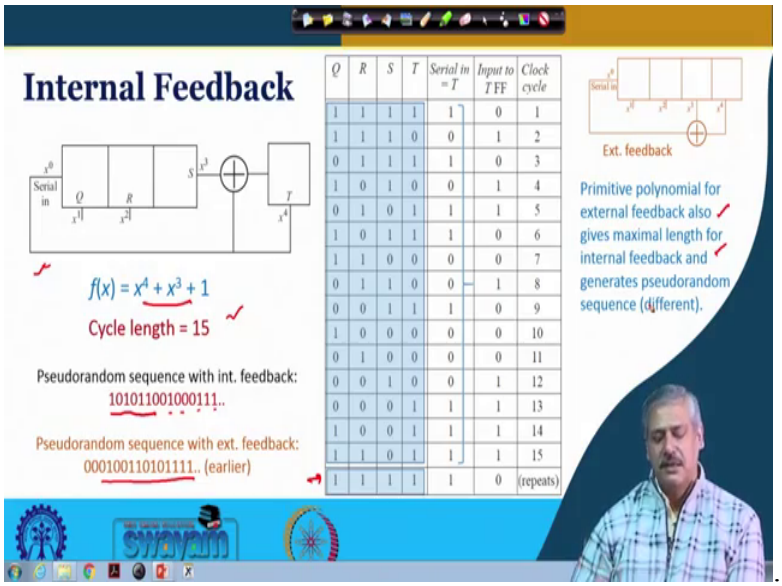
So, that is how it is there. So, for 5 x to the power 5 plus x to the power 2 plus 1; so, 8 9 10, so this is what has been arrived at after checking with different combinations. Now, in each of these cases when you look at it right and what you can find that the necessary condition is not sufficient; when you look at it, right and wat you can find that the necessary condition it is not sufficient, but it is required that number of taps are even ok, number of taps x to the power n and x. So, x that is x to the power 7 and the 1st flip flop

and 7th flip flop 5th and 2nd 8th 6th 5th and 1st ok. So, 9th and 4th 10th and 3rd ok. So, we have shown it up to say 10 degree, but extend and you see that always the number of taps are even.

And the other important thing is the tap numbers are co-prime like 7 and 1, 5 and 2, 8 6 5 1 9 4 10 3 relativity between them, ok. There is no common division other than 1 ok. So, that is a necessary condition, but it is not a sufficient condition. The other thing important here is if the tap sequence of n bit linear feedback shift register generating primitive polynomial is in n m l k and 0. N means that is if it is a degree 8, the last flip flop, so that is n and 0 is the input that is coming here to x to the power 0. So, n and 0 are there in between. These are the different taps, then if this that gives a primitive polynomial, then n minus n, n minus m, n minus l, n minus k, n minus 0 right which the first one will give you 0 and the last one will give you n. So, basically they do remain, ok.

And other value is just n minus the other one, the particular tap number ok. That will also give a primitive polynomial that will also a maximal length sequence and pseudo random number getting generated of the particular length. For example, in this case if x to the power 5 plus x square plus 1 is a primitive polynomial, then n minus n is x to the power 5 minus 5 minus 2 and 5 minus 0, right. So, x to the power 5 plus x to the power 3 plus 1 is also a primitive polynomial. So, if you take it from 5 and 3 x or it and feed it as input, that will also generate maximal length. In this case 2 to the power 5 minus 1 log sequence which will repeat pseudo random sequence.

(Refer Slide Time: 17:29)



So, what we had discussed so far constitutes external feedback, ok. So, there can be linear, there can be internal feedback also if provision is there by which these polynomials can be generated and implemented, ok. So, internal using internal feedback if we are implementing the polynomial x to the power 4, x to the power 3 plus 1 which was maximal length in case of the external feedback which you have already investigated, so then it would look like. So, x to the power 4 and x to the power 3, there Ex-Ored and fed as input to the flip flop over here and of course, x to the power 4, this is coming over here as 1, x to the power 0.

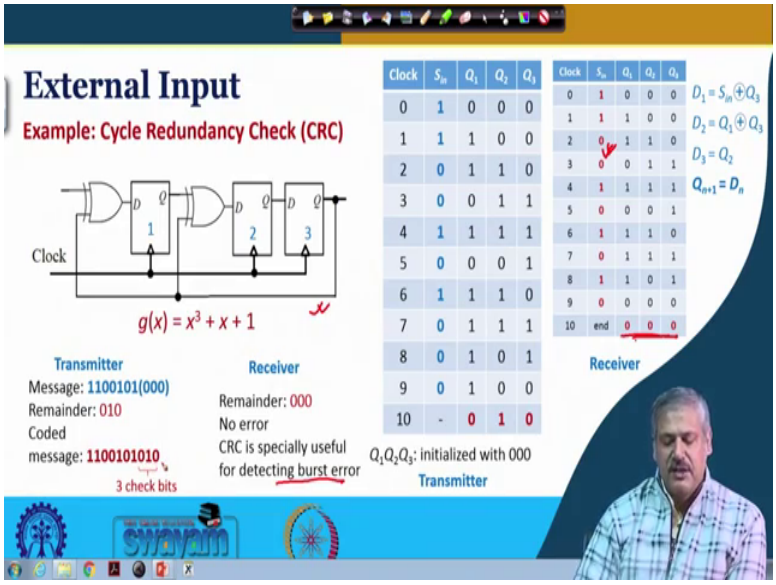
So, this is how it would look like with internal feedback and with this internal feedback again if you look at an initialisation which say 1 1 1 1 right and then, you can see for first few cases you can examine. So, here is S and T are Ex-Ored. S and T are Ex-Ored 1 and 1.

So, the output is 0; so S and T, so this output is 0. So, 0 will be fed back here. So, next clock right this is becoming 0 and was 1. One is feedback here 1, 2. These other two 1s are getting pushed, ok. This 2 1 is getting pushed here, this one is coming over here and 1 1 Ex-Ored is 0. So, that is getting 0. Is it ok? So, next this 0 will come over here like this, right. This 2 1 pushed over here and now 1 and 0 Ex-Ored is 1, ok. So, this one, this one Ex-Ored 0 is coming over here. So, this is the way it will evolve and if you rub this you can see that it is coming 2 4. Once again after 15 clock trigger after 15 clock pulses,

ok. So, this also x to the power 4 plus 6 to the power 3 plus 1 gave maximum length for external feedback. For internal feedback also that we see that it is giving maximal length so, but in this case this pseudo random sequence that is getting generated is 1 0 1 0 1. So, whatever you see over here and for the other case we saw something else, ok.

So, that number getting generated and fed as you know serially could be different, but maximum length is there from both the cases, ok. So, this is one important thing. We take note of that. Primitive polynomial for external feedback also works as maximal primitive polynomial for internal feedback and generates pseudo random sequence, clear.

(Refer Slide Time: 20:51)



Now, we look at again, this particular arrangement configuration. Earlier it was actually involving only from internal values and the feedback that is getting generated.

Now, the feedback is there in addition we are putting an external input, and would like to see if it is of any use. Yes it is of important use in control as error control code generation, and this is used in what is called a Cyclic Redundancy Check code, CRC code ok. So, I shall take you through an example and see how it is useful. So, the polynomial here used x to the power 3 plus x plus 1, right. So, that means this is 1 2 3. So, this one this one, first one it is fed back, this x 3 x 1. So, this two internal feedback this is fed here and this one that is coming over here that is getting Ex-Ored with the incoming data stream which we call as a message which you want to code. Is it clear?

So, x 3 x plus x Ex-Ored here going here and 1 is coming over here which is now getting Ex-Ored with the incoming data stream, right. Now the message that we are having let us consider it is a 7 bit message. These flip flops are initially all cleared, right in this register shift register, and to this 7 bit long message we append three 0s, we append three 0s signifying the values that are present here, signifying the initial values of this shift register, ok. Now, this is the input data stream 7 bit and 3 appended with three 0s, ok. So, now how this particular circuit evolves with time with clock pulses?

So, this is the clock pulse 0, this is serially so initial value is 1. So, this is the one that you see in here. So, these are all three 0s ok, then what happens this 1 and 0 right. This is 0. So, D1 is serial in and Ex-Ored with Q3 serial in Ex-Ored with Q3, D2 is D2 is this one, Ok. Q1 and Q3 getting Ex-Ored and D3 is just whatever is Q3 output that is going as Q3 Q2.

And always we know for D flipflop whatever is the D and that is becoming Q n plus 1. Next clock value 1 2 hello, ok. So, we can examine in it in this manner. So, the first case serial in is 0 serial in is 1 and Q 3 is 0. So, the value is 1, right. So, what will go to Q1 in the next clock is 1 ok, then Q3 and Q1 Ex-Ored is 0, Q3 and Q1 Ex-Ored is 0. So, next clock it becomes 0 and we Q3 will get, Q3 whatever this is there in D3, D3 gets Q2. So, this is the one that will be there. So, the next value will be 1 0 0. After that 1 comes as the serial input. So, 1 and 1 sorry 1 and serial in and Q3, so 1 and 0; so 1 comes here, then in Q1 and Q3, right. So, 1 and 0, so this is 1.

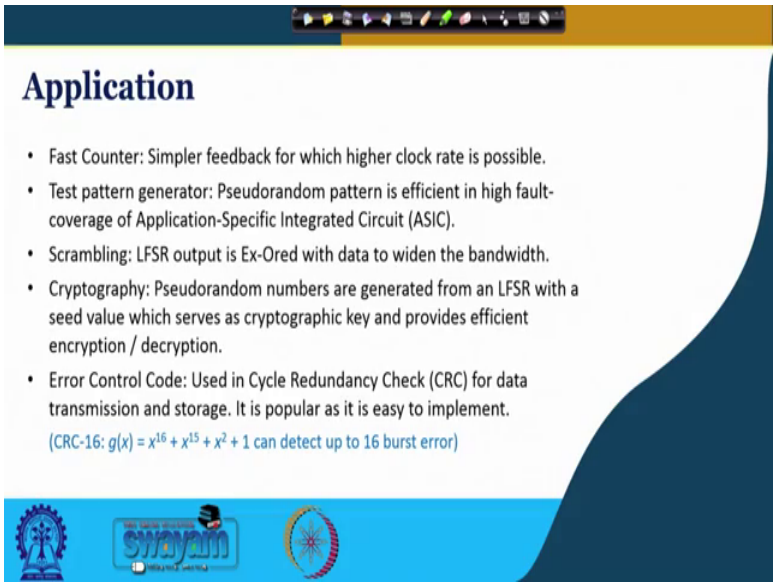
So, 1 comes here in the next clock and Q2 comes to Q3, in the next clock. So, that becomes 1 1 0, then 0 comes here. So, that way it continuous and you include these 0s also, ok. So, 0 to 9. So, 10 clock pulses are there. Once we complete this 10 clock pulses right, the value over here what we see is 0 1 0 that is the reminder, ok. So, that remainder now attach as check bits to this 7 message bits, ok.

If the message bit was less than that, then or more than that will continue accordingly, but always will be in our scheme of things; these three 0s, these three 0s will be there right. So, now the this is what is getting transmitted, right. This message bits are getting transmitted at the receiver end. What we will what will you do? So, receive end now these bits whatever is there will be given as input. The circuit will be the same and it will be initialized with 0 0 0, and we will publish in the same manner, exactly the same

manner the way it has published at the transmitter end and at the receiver. When if every bit is appropriately received, no error it is there, then the remainder will become 0 0 0, ok. If the reminder is anything other than 0 0 0 ok, then that means there is error here, there is an error here in the incoming stream, ok.

And this CRC code is especially useful for detecting bust error. Bust error means because the noise and all more than one bits consecutive bits have has got have become wrong ok, so it has gotten two consecutive bits. So, this is the CRC code generation and then, it is at this receiver side its detection and this is where the check bits are coming and there this shift register is useful.

(Refer Slide Time: 27:43)

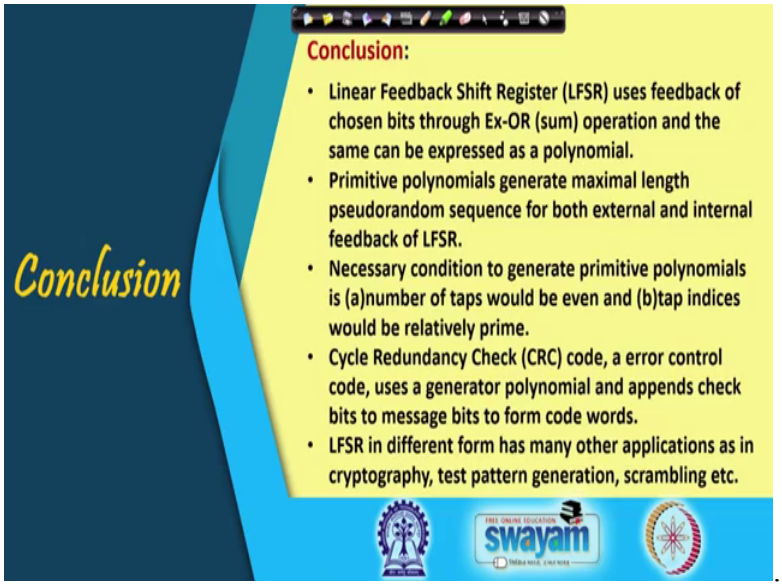


There are many applications of LFSR. So, here I have listed a few and majority of them from from its ability to generate pseudo random sequence if the counter does not require states to be like 0 0 0 1 0 0 1 0 that kind of you know these sequential numbers you know increment is not required. Any kind of you know states can be there and then, a first counter can be made of sufficiently long length.

So, that is it in with n bit we can go up to 2 to the power n minus 1 and the other use of it is in test pattern generation of which can be used for testing the fault in application specific integrated circuits. Scrambling is another use of LFSR output. The pseudo random sequences can Ex-Ored with the message bit and the resultant signal pattern would look more noise like, and the reverse can be done at the receiver end to get back

the original message and it is used in cryptography also where the initial seed from which the pseudo random sequence is generated can serve as a cryptographic tree and can be used for encryption first and at the receiver side for description and you are already seen its use in error control code, the cycle redundancy check, the check bit why it has been placed and CRC 16 is one popular such thing, CRC 32 is also there. For example, x to the power 16 plus x to the power 15 plus x square plus 1, it could be one such polynomial and it can detect up to 16 burst error and many other types of errors.

(Refer Slide Time: 29:43)



So, with this we conclude. We have seen that LFSR can be used and to generate a through primitive polynomials pseudo random sequences of sufficiently long length, and primitive poly polynomials D2 have even number of pairs and tap in. This is need to be relatively prime. This is a necessary condition, not sufficient and CRC code is useful for your control as error control code and LFSR can be used in cryptography and scrambling test pattern generation and many other areas, ok.

Thank you.