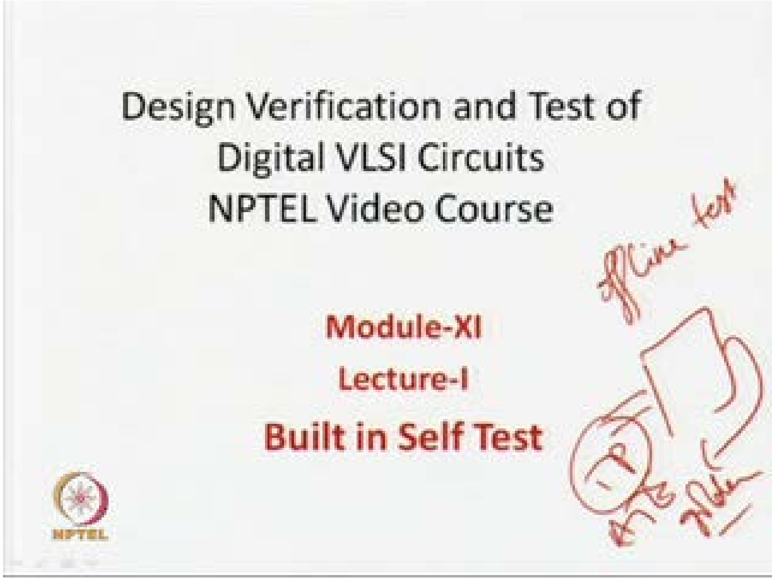
Design Verification and Test of Digital VLSI Designs
Dr. Santosh Biswas
Dr. Jatindra Kumar Deka
Indian Institute of Technology, Guwahati

Lecture - 1
Built in Self Test

So, welcome to the next lecture on the next module on the velsiting part of the NPTEL course. So, within till now, we were discussing with talking mainly about different type of apply to sequel emphasizing come on equation side we test pattern and all those things. So, if you actually want to come all this things we can quite mantel that is we call actually offline testing.

(Refer Slide Time: 00:42)



Design Verification and Test of
Digital VLSI Circuits
NPTEL Video Course

Module-XI
Lecture-I
Built in Self Test

So, we can call it is an off line testing. So, in this what happened we have certain parameter we apply the some testing through an at and this golden response spantual compared. So, that is in like all the over test we have the as you want to apply that is to be determine for we found algorithms like the algorithms than we have found a sequence are give to you what do you call this time expansion method.

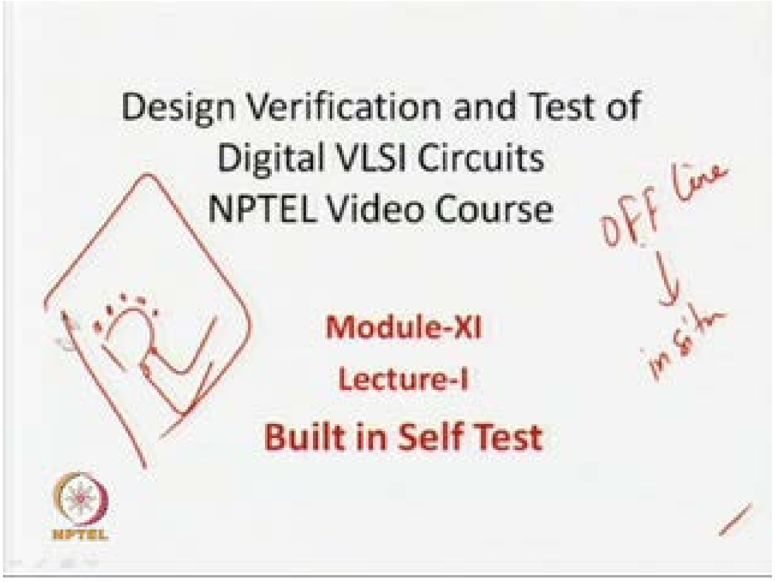
Sometime we use the algorithms along with design testable scheme call for the scan chain and also we see the random pattern. But, as a hold what was an idea which idea if have fabricated if we put all this some items and as this golden response and your job is done and you can sell up in the market.

But, this assumption you can take and if I have what will going to say these all pre 90 kinds of a thing but, as over take as over integration mechanism because of this more certificated we are going to sub micro designs. So, may as the problem arise that is once you manufacture chip that is to be market you cannot guarantee tangible for the right time on the expected life time, that means what some of the forms even may be can be in circuit operation and it means develop say after to the 1 year or some 6 months of operation.

Now, where ever your circuit is expressed the complex board now something design toward in a deep part which will circuit on the on working in complex system like a mother board or you can say that complex mother board what you call the lop top on mother board of your mobile. So, one of the cheap is not working now it is rarely purpose on engineering to talking with external points and working finding out exactly the cheap is not workings.

Because such type of a failures are very common with on micro designs and individually they in the any parts of their fabricated than we sold in the market but, after operating from season say 6 month that is within the lifetime very we will in the life time that is very problems any of the . So, takes lot of time because due to find out over which chip is wrong.

(Refer Slide Time: 02:50)



Design Verification and Test of
Digital VLSI Circuits
NPTEL Video Course

Module-XI
Lecture-I
Built in Self Test

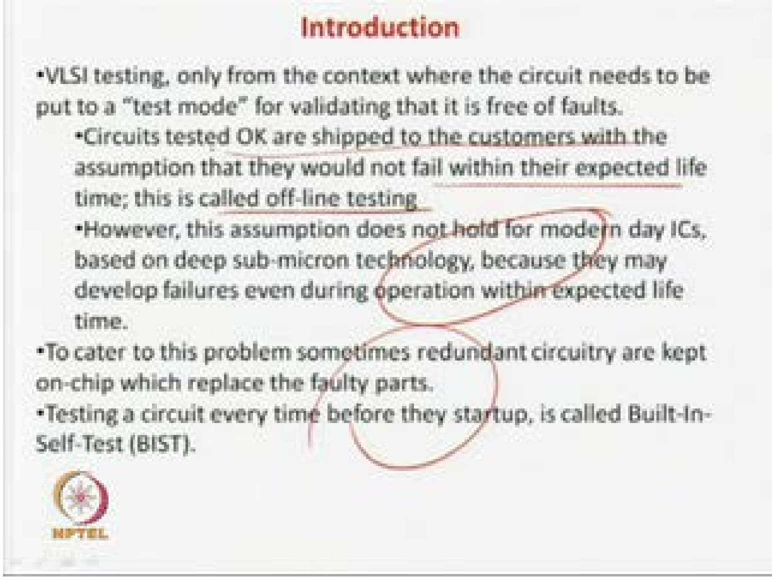
OFF line
↓
in situ

Then, you have to replace the chip so, than may be the now we cheap is already fabricated in the board see cannot apply this pattern in all those things this is lot of problems include the testing. So, the next level of the testing offline actually the offline testing discuss all listed in off line now there is something called in c 2 in c 2 that means already call chip least in the board and now what have still it is operating 5.

Because, that is already something problem in there system. So, what will know because already placed like in this is your mother board and chip is already fabricated and lubricated and solden over here so, where this is operating here this fabricated and over here because it operates individual chip use be can in part of that some actually make in life so difficult.

So, now what is the solution what are you can do. So, what are the requirement inclusion test is important the important arise because of the micro this falls which can occur we will of a the fabrication and total being test it will free they can occur when the system is operation for we replied it. So, now on chip that is the inside the test could be requirement for all circuit nowadays.

(Refer Slide Time: 04:06)



Introduction

- VLSI testing, only from the context where the circuit needs to be put to a "test mode" for validating that it is free of faults.
  - Circuits tested OK are shipped to the customers with the assumption that they would not fail within their expected life time; this is called off-line testing
  - However, this assumption does not hold for modern day ICs, based on deep sub-micron technology, because they may develop failures even during operation within expected life time.
- To cater to this problem sometimes redundant circuitry are kept on-chip which replace the faulty parts.
- Testing a circuit every time before they startup, is called Built-In-Self-Test (BIST).

So, that means what is actually call built in self test that is we have going to see in today lecture. So, what it says so that means what the idea will has to some that arrangement in the both that can apply some status and that can analyzed response and can tell that this circuit is fine and that circuit is fine and ore in the in other words once we have circuit to be the offline test to be fine using an a eighty that is to be market.

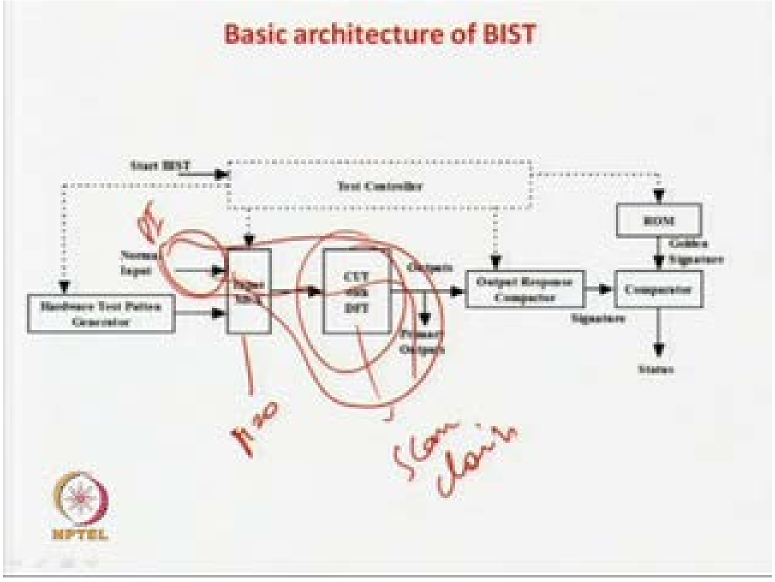
Now, I put it on mother line board putting on a machine this step now every day we reaming chip started as operation that be before boot to computer before you start advance your mobile phone every chip will test . So, obviously that the testing part what you call so that testing cannot be as exhausted as you are doing on a because is sophisticated never you think huge memory the suffocated machine is a huge memory where you can apply ur patterns where you can also the comparative response where you can come fast. But, of and put it on chip so, what we have saying that what we start operating chip you can test a partially for you can say that I can applies the very important test that I can get estimate with the debt or that circulate is working fine or not.

So, that the given problem become varies in this case. Like for example, that excision chip can board your circulate is not working fine than what can you do the building surface that is actually in two testing that is you can say the part of the test buttons are already you have used for off line testing it can now be applied from average circuit use on chip that is the separate circuit chip which you all used for testing the circuit it test for application responsible for that analysis. So, obviously sub part of the certain part will be applied in all the chips before for the test starts the operation and we aim any one of the chip not working fine.

So, very it is the pin point this is chip for the these are the chips where there is the problems so that can be rebirth and we can pin point with because do not have all the chip, already put in the board, so you cannot apply any external elements external testing but, they having the it is not required but, because your test application circuit and test analysis circuit already in the chip.

Now, you can chip on the chip testers which is actually called built self test this is called builty which is already built on why this chip is not working and find we can easily pin point that the chip is not working fine.

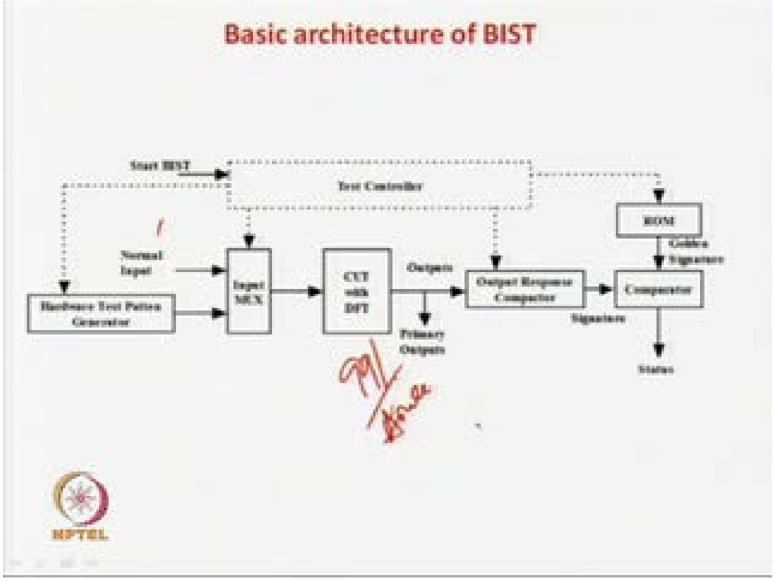
(Refer Slide Time: 06:15)



Basic architecture of BIST

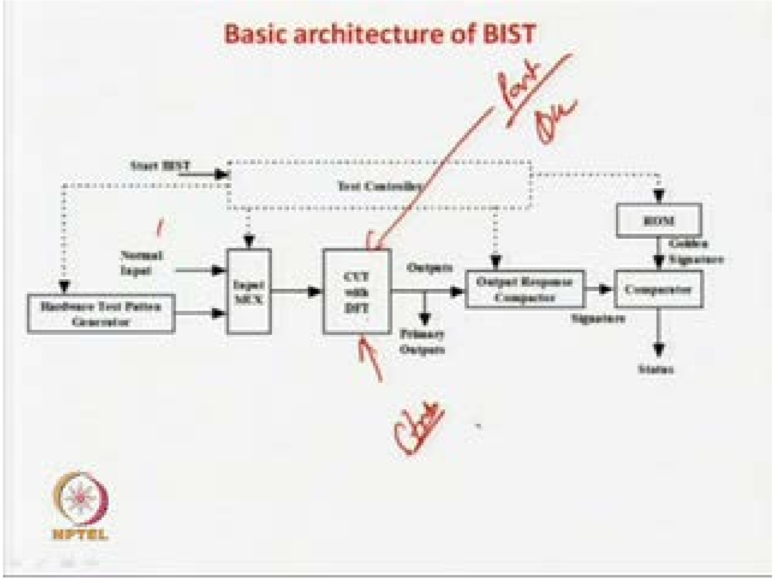
So, you can animate the chip and could have achieve. So, your test time these are very simple and easy for testing in an any testing are a board level or by the system level. So, that is what to be introduction this tested are ship to the costumer with the assumption with a extracted life time these are offline test these are already discussed but, in modernization these may not hold because during of operation these are may be same to gather to this problem we test before circuit the time bottom and putting some circuit all chip that is actually called built in self test. So, now will come to basic architecture of this how it works you see. So, this is actually your circuit with some you can this scan chain.

(Refer Slide Time: 06:59)



Basic architecture of BIST

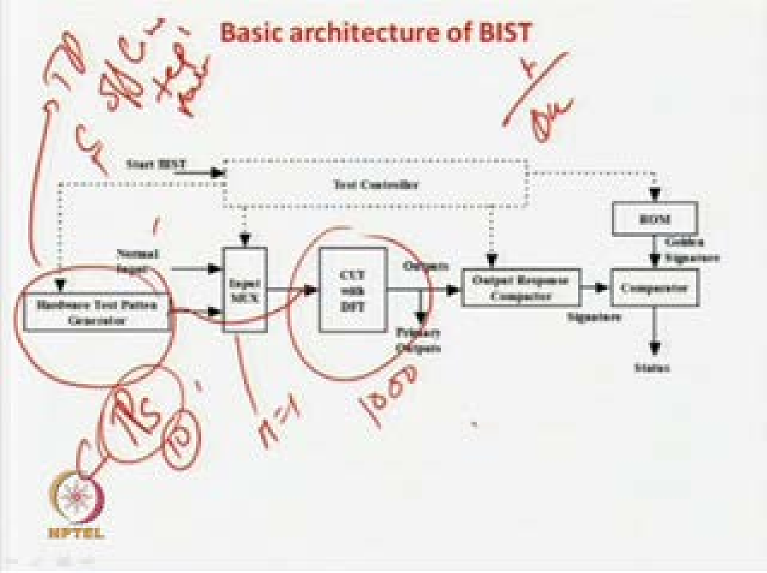
(Refer Slide Time: 07:00)



Basic architecture of BIST

So, this input will go by the mark so in test mode again this mode equals to 0. Now this operation these are primary output. So, this is what you can thing that this is on this small part of the circuit which is actually normal circuit. Now, what happen so that actually we can say in other hand and other blocks to that many other blocks but, that area is not very high. So, in we discuss all that why the area is not high so this actually consume 99 percent of the area main circuit and all the other parts are very small in size whereas, so this part of 99 percent so this area let on later you can the chip this circuit part of the chip. So, this part of circuit of gain an verified fantastic fine in tested by in sold in the market up to off line testing. Now, it can happened that every time an before you start operation that every time you start and so falls or it may have some falls develop will circuit operated.

(Refer Slide Time: 08:03)

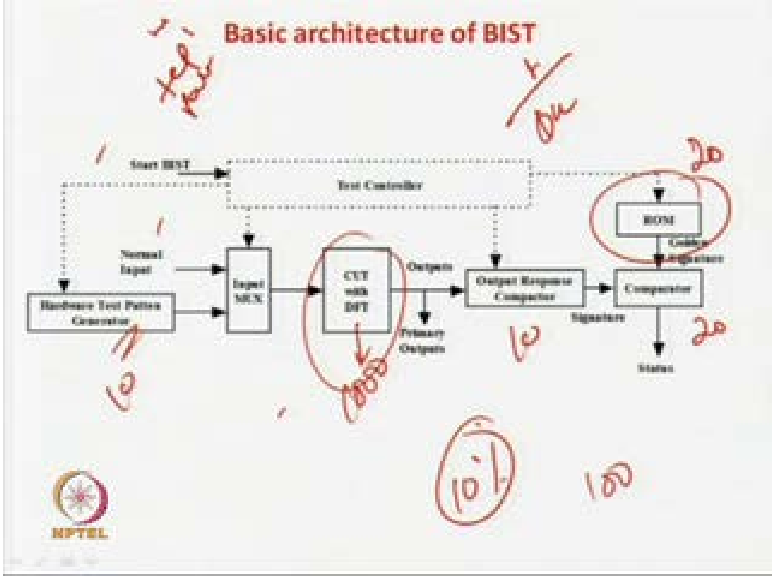


Basic architecture of BIST

Now, as all this circuit fabricated in the chips in the both are where there is a problem. So, we have due to see to testing reveal where we have the chips as the circuiting the both you want to test them. So, so some extra circuit will be required and to do that. So, why is actually become hard ware testing strategy are need to this is actually continue so the test pattern they are got him the already scanchered what is the generation that have what have to be case these are the test patterns which have already generated.

Now, you have to the observe that the stage will the generally applied in a 88 is the huge expensive machine big machine is a lot of memory and it is very powerful that means this now this is not main circuit and obviously say 100, 1000 unit you cannot have a 10000 unit circuit the test it is the infusible and it will not appreciate that. So, what we have to do that have already test pattern of the area may be certain and 100 marks it will 1000 it can be area founded so what is act meters do will actually some have a random is nature obviously if rate will generate all the test pattern for generate because the memory card and others see you can generate important this pattern otherwise, you can say that we are going to test this circuit for the where few to the most important patterns on the test patterns generated by the all got him on other words you on the term but, detects a this term in terms this parent detects 5 and so 1 you use all those things that we detects more number of because you are limited by in a by the an hard ware requirement of the test pattern generated by. So you will say some percentage test for the pattern which are using in the off line testing.

(Refer Slide Time: 09:46)

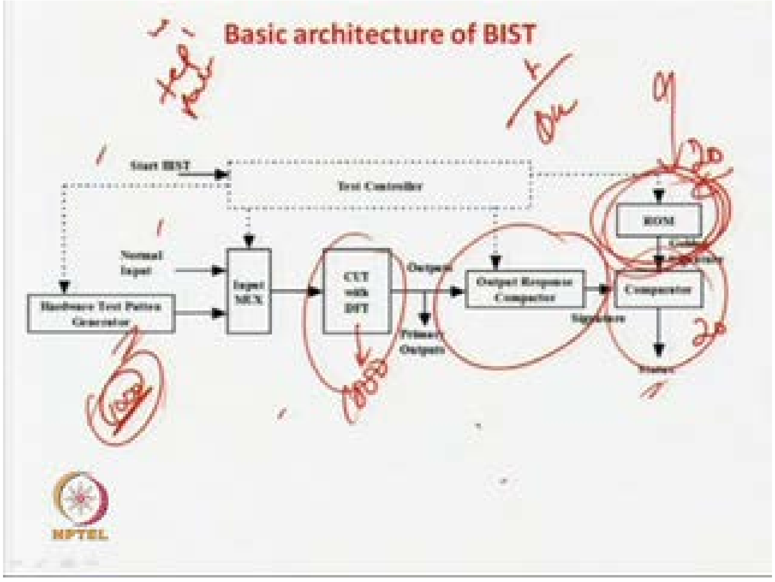


Basic architecture of BIST

So, now when you test mode is 1 than this pattern will off line this circuit these are actually off set of off line test pattern. So, this pattern actually off set of off line in off line this test patterns . Now, what happens so off set they will be applied now there is I will tell you these are in the raw modules so, what do you store actually golden opportunity response for the expected response from this for this . Now, we can see again this chip circuit on the area is 1000 unit.

So percentage of the general circuit. So, including all the block this is say 1000 area over it some units than all of the test controller the is that for including apply say only 100 units of area. So, we can think that fly use what you can say some of this happen 100 test there may be 10000 pattern will be often you say that very good pattern for efficient test more number of falls and you generate and if you want to store golden responds to corresponding of all one is this pattern so they are size to be very very high so that you cannot do.

(Refer Slide Time: 10:22)

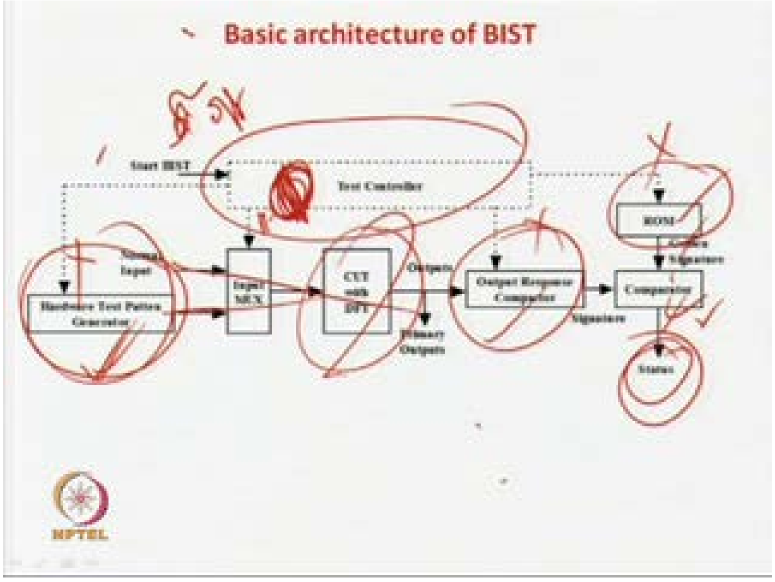


Basic architecture of BIST

So, exact response for this 1000 test pattern are whatever cannot be store so for what they want do you we are something call a response formula pattern that is whatever, response using the compotator comprise them by compressing them one something call aliasing which will see in this lecture, so aliasing means what like compression will losing compression for some time you may compression lousing as efficiency if may got compression could not they could they call because, we call compression in this architecture that is we called losing that is air presenting in the they have compression because that have it means and high not is possible because if you have compression test all the that is the generator you call store the answer what you can call the golden pass in the rom.

The ROM size is very very high and you cannot do that. So, you have to compact it and very compact report they can be pass the compact test pattern and output compact but, patterns you can have the response and you can compact of the our output than the compact if this access we have all these examples and along with all these there is something call a test something called a something called a test controller.

(Refer Slide Time: 12:20)

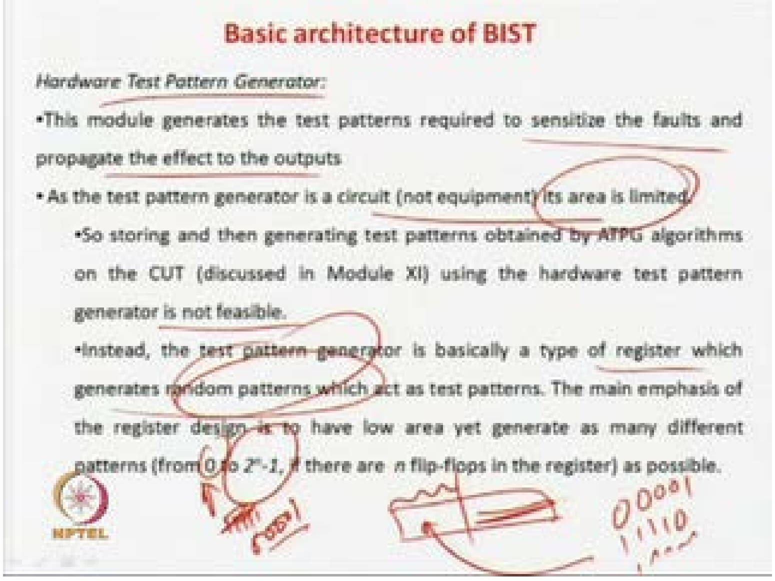

Basic architecture of BIST

So, now what is the test controller whenever this circuit start will be the operation if you say what this circuit have a signal can what will do it will it aim equal to 1 because select this pattern it will ask it to generate the random patterns it will ask into compact responses as well as along with compression start. And whenever you say that why this is done you can state the another fly is 5 you will go for the normal operation some circuit than you can say that disties are when control are making aim equal to 0 it will stop its operation it will activated you will act this is also activate the rom.

Because, one more thing we to understanding that whenever circuit is doing a operation so only this part of the circuit will be radding unnecessarily this 1 this 1 so, do not separation will be an execution there will be lot of part circumstance. So, of all these so whenever you want to this up grids you can make equal to 1 this is can operating staging all these start you get results status was this is done when you control equal to 1 and than.

So m equal to 0 it will may normal input will go through and all other part circuit will be this part may be loss that is aiding this one why this basically the architecture. So, what here 1.0 you observe than some how you make this long and small size and also the be small size those if you compare to at the hardware also the ROM actually analyzed these response but, it is very big machine lot of a memory power so we cannot may not able delete the number of test pattern all this pattern remedial by the test time.

(Refer Slide Time: 13:21)



Basic architecture of BIST

Hardware Test Pattern Generator:

- This module generates the test patterns required to sensitize the faults and propagate the effect to the outputs
- As the test pattern generator is a circuit (not equipment) its area is limited.
  - So storing and then generating test patterns obtained by ATPG algorithms on the CUT (discussed in Module XI) using the hardware test pattern generator is not feasible.
  - Instead, the test pattern generator is basically a type of register which generates random patterns which act as test patterns. The main emphasis of the register design is to have low area yet generate as many different patterns (from 0 to 2^n-1, if there are *n* flip-flops in the register) as possible.

Because if you are using too many test pattern than the test time will be higher and you are so eighty cost will increase but, requirement size and all this is not a much of concern is a very much concern call the circuit it will put on if the mains are giving some area restriction some 1000 unit of their nobody alive use a built within we have a one thousand area it will area of the normal circuit. So, this area should very small some 8 percent and 10 percent 5 percent small are the better area over compact this circuit so that is why very critical to design which circuit have to applied are you compact them, so that is why very important so that is will see so as your concerned lecture.

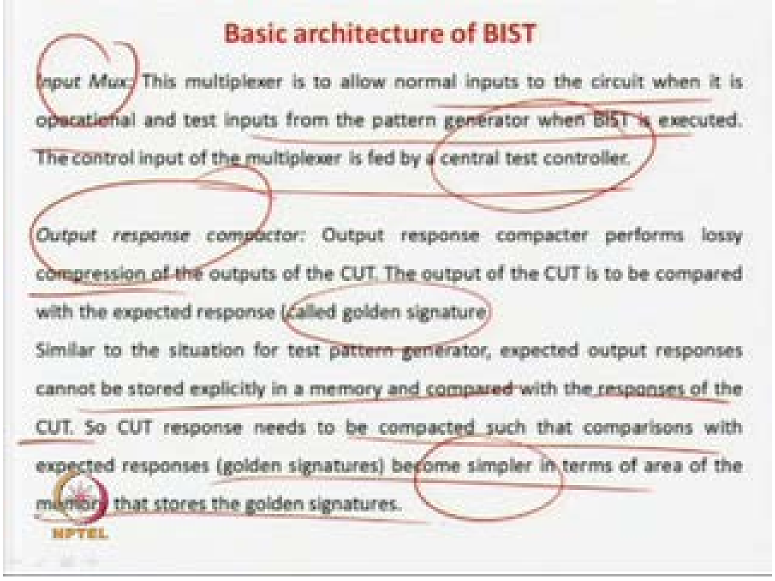
So, now to let us some see what we have discuss so these are actually hardware test pattern so what happens so this strategy we need the patterns going to sensidial and property with the output and this same test pattern which would have develop the and this time expansion method or whatever you can call the repent individual but, in this case this may not the equipment this circuit so the area is limited. So, it will generate all random pattern say from0 to 2 n minus one generally 0 is not generated we will see the case. So like say 1 1 1 1 1 2 to the bar n minus 1 all patterns it will generate kind of the thing 0 0 0 1 to the but, whatever you require.

So, you can make the register of this random pattern in such a way so that you are mean whatever I mean it start generating there is something called a seed as we see which will determine the sequence of pattern of random pattern it can generate all pattern in between all 0, 0, 0, 1 to 2 the bar n minus 1 that is 1 to the bar n minus 1 so and you see pattern 0, 0, 0, 1 then 1, 1, 1. 1, 0 then 1, 1, 1 these are the 4pattern you want see you can arranging such a way.

So, that in initial few runs initial you get all three require 3 patterns and then the other random patterns would be generated it not required for you so after that these 3 or four steps when you get all the results all the patterns you can reset and again you can restart it if anybody require so but, we will see that the the random pattern is not generated in specific three or four patterns you require is generate all random patterns from 1 to the bar n minus 1.

Now, we have our duty is that you can generate the random pattern whatever patterns you require occurs initially before the undecided or do not required for the random pattern. So once you all the require random patterns are generate you can reset the random pattern generate stop this kind of a thing. That is the basic philosophy.

(Refer Slide Time: 17:27)



Basic architecture of BIST

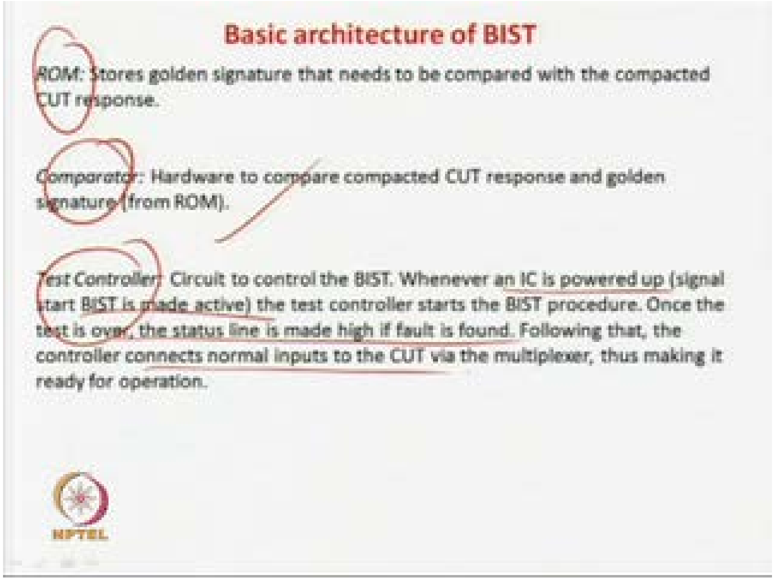
Input Mux: This multiplexer is to allow normal inputs to the circuit when it is operational and test inputs from the pattern generator when BIST is executed. The control input of the multiplexer is fed by a central test controller.

Output response compactor: Output response compacter performs lossy compression of the outputs of the CUT. The output of the CUT is to be compared with the expected response (called golden signature)
Similar to the situation for test pattern generator, expected output responses cannot be stored explicitly in a memory and compared with the responses of the CUT. So CUT response needs to be compacted such that comparisons with expected responses (golden signatures) become simpler in terms of area of the memory that stores the golden signatures.

We will see with the examples. We have also seen an input multiplexor, so input multiplex or is nothing but, it actually normal input to the circuit when circuit is operational and test input pattern generator when bits is executed the control of the multiplexer is a what you call central test controller. Central test controller we already seen so so output response compacted so the output response compact or into the lossy compression as I already told you why do you call lossy compression because if you using non compress structure when for the all the random patterns which was using as test pattern the output response will have store to the rom.

So, the Output response mean for all will be very every high the large size of the we are very much restrict what can you call the golden signature very large in that case this is not compacting it but, similar to the situation for test pattern generator the Output responses cannot be explicitly stored in a memory and responses cannot be compare because it will make the size very very high. So, what we have to do so we have compact it, so that golden signature signature become simple in terms of area of the memory that should be golden signature. So that is you compress so that whatever we doing compare will be smaller in area size.

(Refer Slide Time: 18:28)



Basic architecture of BIST

ROM: Stores golden signature that needs to be compared with the compacted CUT response.

Comparator: Hardware to compare compacted CUT response and golden signature (from ROM).

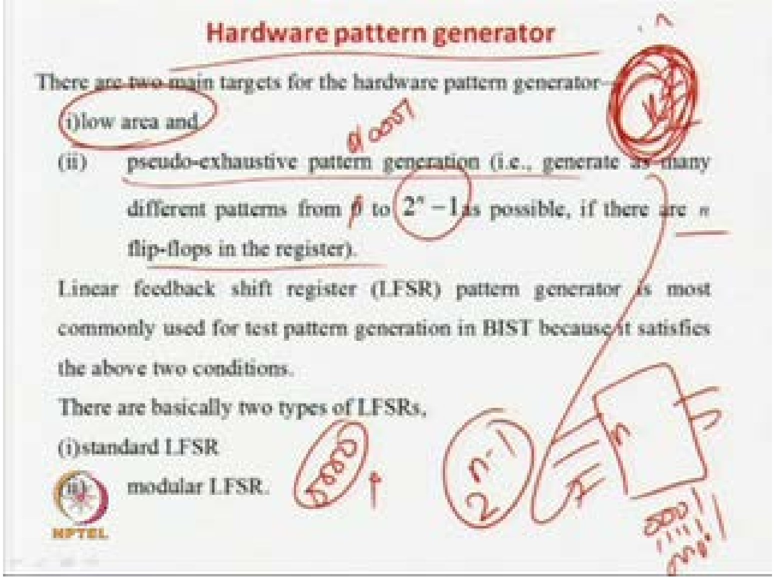
Test Controller: Circuit to control the BIST. Whenever an IC is powered up (signal start BIST is made active) the test controller starts the BIST procedure. Once the test is over, the status line is made high if fault is found. Following that, the controller connects normal inputs to the CUT via the multiplexer, thus making it ready for operation.

So, in size raw material will be less but, it is a lossy compression some time we find that we are actually loosing some of the test coverage because of this comp ratio but, go about the because you cannot have a legacy we cannot have a extency of using a huge area in this case. The next element is ROM which stores the golden response to the compact again the golden signature herein this case is the compacted signature.

So, the comparator comparator is a digital compact the output response of this compressed output response of the circuit will be golden signature in the ROM, so it is equal it is a yes, if they is a failure it will say that it is the fault. And the test controller is the controller of the circuit so whenever the IC is powered on if the this signal is made high then this procedure is started was the Bist status is high or low the where ever you know that there is no fault in the circuit then you can connect the circuit in the normal input to the marks and its operation did you find that is there in problem in the circuit in the bits circuit is not allowed in they our circuit to operate it will ask it will given a message for this part of the circuit is not working in the mother board this chip is not working so you replace the chip and then all your otherwise circuit chip cannot operate your otherwise this system.

So, that is the another advantage of this with one or chip circuit you system is not set operating that in the middle male function you may loosey our so it will not allow the circuit or your system to boot up say these is the problem here you will better recover it and replace the chip and then start the operation.

(Refer Slide Time: 20:00)



Hardware pattern generator

There are two main targets for the hardware pattern generator—

(i)low area and

(ii) pseudo-exhaustive pattern generation (i.e., generate as many different patterns from 0 to 2^n-1 as possible, if there are n flip-flops in the register).

Linear feedback shift register (LFSR) pattern generator is most commonly used for test pattern generation in BIST because it satisfies the above two conditions.

There are basically two types of LFSRs,

(i)standard LFSR

(ii) modular LFSR.

So, that was about the basic this architecture that and about the requirement of this. So, now what we are going to see there are two major parts of your circuit here 1 is your Bist test for pattern generator and another part is 1 another part is about here is response is a digital course this controller is nothing but, the controller circuit it depends upon the for something it can be designed using digital circuit fundamentals and optimization computer is digital circuit.

So, only two important parts which you do not know we have learn into testing module is the and the response compactor and the aliases. So, these are the only 2 modules which other can be recollected or you can go back and look for your diesel circuit fundamentals and you can do that.

So, what is a random test pattern generates. So, what do you see you say if you have a circuit like this say say it has any inputs and some output and we know that the expected 2 and minus 1 kind of test patterns are possible here say that 0 is not possible when cy 0 is not possible to be a flat so, let us assume that there are n test pack are input size and the hardware size minus one patterns is possible so along they say you want these test patterns to be applied.

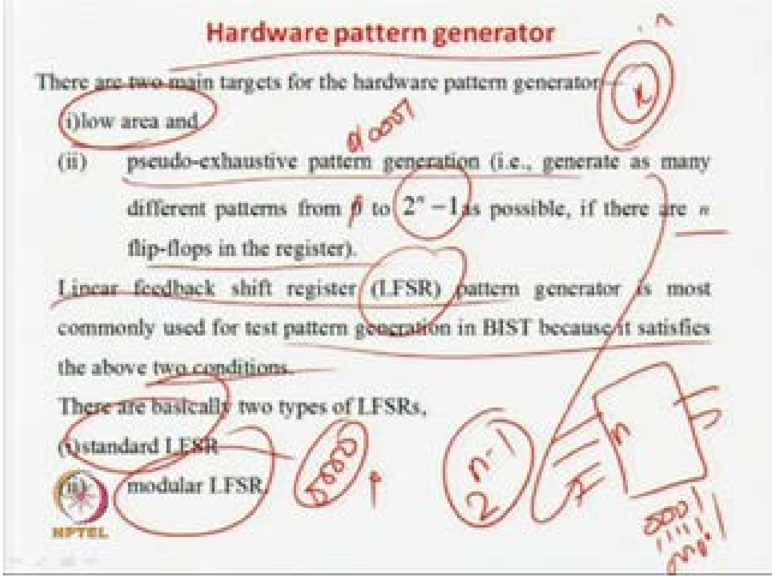
So, this k test pattern which you want to apply which you whatever is to be applied to analyze the circuit for generation in Bist. Now, what you have to do if I generate a circuit we generate all these key patterns one by one will be very very rash as we see in the example but, now what we can do the other way is that there is something very much important is it has a low area over wide. So, some circuit will take very very less area over it which actually cause random exotic pattern generator which generates at most almost pattern all for actually 0 is 1 that means 0 will be generate so 1, 2 to the part 1 in there are n flip-flops in the register flip-flops register is there.

So, we can start from 0, 0, 0, 0, 1 and all patterns will generate among them and they will be in random fashion there is actually exotic is also called random pattern because these patterns are generated random with depending on initially seat initial seat means you have flat a register with some values of all are 0.

We are referring to start generating the patterns all are random it will called the sugar random because it will depend on the nothing all pure random because it depends upon the seat and connections. So, it has sugar random patterns and among this sugar random pattern some of these sugar random patterns are 1, 2, 3, 4 something exist will generate and among them the k patterns will obliviously be there which will there which will be applied to this 1 and if you want to apply this 1 specific order like k 1, k 2, k 3, k 4 which are among this k it is applied which require very sophisticated circuit like initial pattern you say 0, 0, 0, 1 and 1, 1, 1, 1, 1 and 0, 0, 0, 1, 0 one something like this.

So, if you want to generate this the 3 patterns which are the patterns, so you that you can implement you can apply using 1, 2, 3, 4 for be registered and some combination is implement we have to this is a fast output this is a next output this is a next output so, you easily go for diesel circuit design if the state may so these are the outputs you can do that you know that you have to go for optimization of the gears and all so there will be some gears will be one in area very high favor that can do with any which 100 or 200 you know.

(Refer Slide Time: 23:27)



Hardware pattern generator

There are two main targets for the hardware pattern generator

(i)low area and

(ii) pseudo-exhaustive pattern generation (i.e., generate as many different patterns from 0 to $2^n - 1$ as possible, if there are n flip-flops in the register).

Linear feedback shift register (LFSR) pattern generator is most commonly used for test pattern generation in BIST because it satisfies the above two conditions.

There are basically two types of LFSRs,

(i)standard LFSR

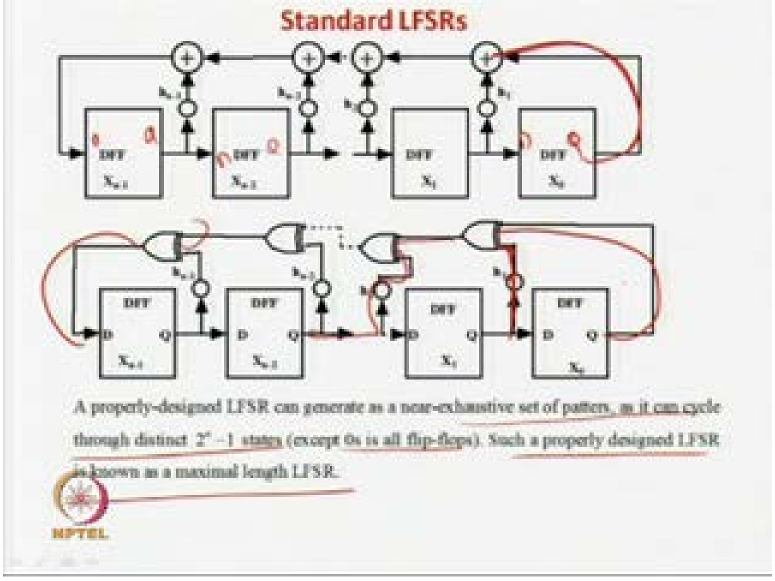
(ii) modular LFSR

It is such a high number of Bits, so your area of one of this counter what have you call test patterns it will be very very high. So, what will do is that use the pseudo-exhaustive pattern random this pattern of random generate the, so it will generate all the random pattern with this but, whatever you do want to select this case k important for once but, now one important thing here is this somehow we will see intelligently you can use starting one so that your k input which here is meaningful term as fast as possible.

So, that basically such type of register are called Linear feedback shift register is almost commonly for all these because these satisfied two conditions. So, LFSRs should know the term is called Linear feedback shift register is a feedback over here so they are used in this case because this generate possible in all random patterns all are possible between minus 1 is random and the depends on the seats so we are controlling the seat you can make the resistance in such way that patterns are important first than the patterns are .

And there are 2 types of LFSRs that study 1 by 1 one is the standard LFSRs and is a modular LFSRs. So, we will see that but, basic idea is that they are very small in area and secondly what and secondly what is the alignment and 1 page and the other advantage here is that it is a the area over it will be very small one thing it will generate all possible pattern fashion into the random pattern. So, you can take whatever is required and that can be changed by using the seat we will see 1 by 1.

(Refer Slide Time: 24:55)



Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patters, as it can cycle through distinct 2^n-1 states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

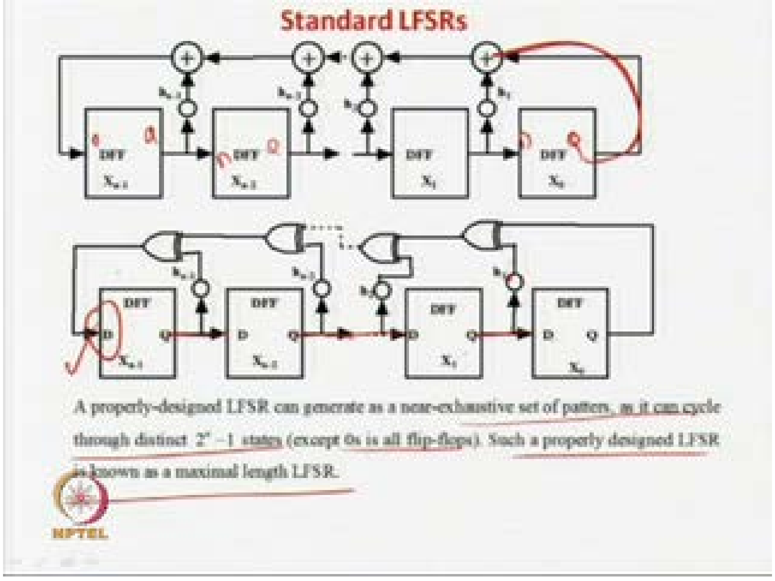
So, there is 2 2 types the standard LFSR and modular LFSR in a feedback shift register. So, this is a standard LFSR, so why we call this linear feedback register . So, now we will study the in up the detail of that we can see what will happen what is basically happening.

So, you see this is a register so there is being a so already deep this is Dn Q this is again D n Q . Now, what you can see that why you call the linear feedback flip register and again d n q so the output of the last you can call 0 whatever x 0 this is connected to this 1 so, all this places are nothing but properly designed LFSR is nearly exhaust the say the pattern it can cycle through the on stage except is all 0s and the maximal length LFSR we will come to that.

Now, see what we are doing so what is the idea here and so the output of the connected to this one of the feedback and you can and this h one which actually we will see the connection for that the idea is that this Q is a 2,1 input of the (()) and the last flip of the

the other input is . Then the output of gets will be the here the output is the 2nd last flip-flops so this one will be connected to this 1 and so fore and the first flip-flops will have this one. So, first flip-flop will complete our output the input will be having a feedback from itself and as well as therefore, for this 1.

(Refer Slide Time: 26:32)

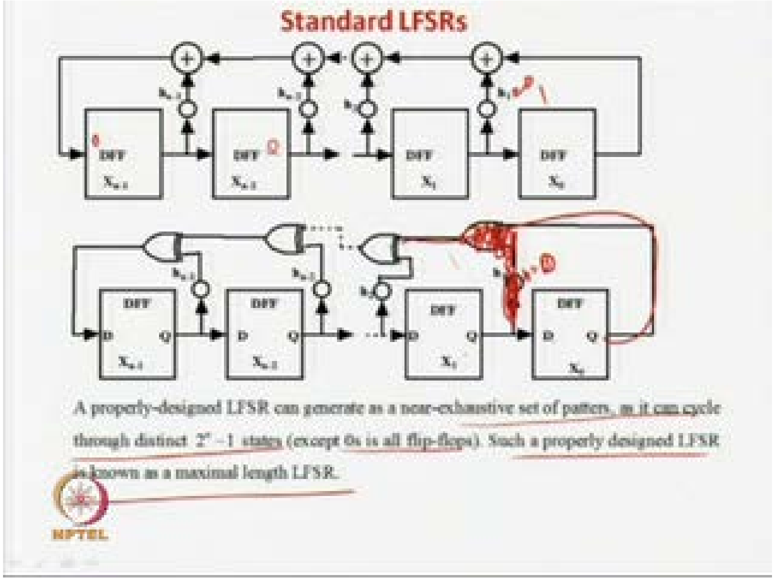


Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of pattern, as it can cycle through distinct $2^n - 1$ states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

Now, it is a important thing over here so these are generally architecture of a Linear feedback because this output is depending on this feedback this for this case the input of the first flip flop in that explorer is 1 minus 1 depend on the feedback. But, for the other cases but, for the other cases these are directly connected this output will be here this output will be here this output will come to the input of the next 1.

(Refer Slide Time: 26:57)



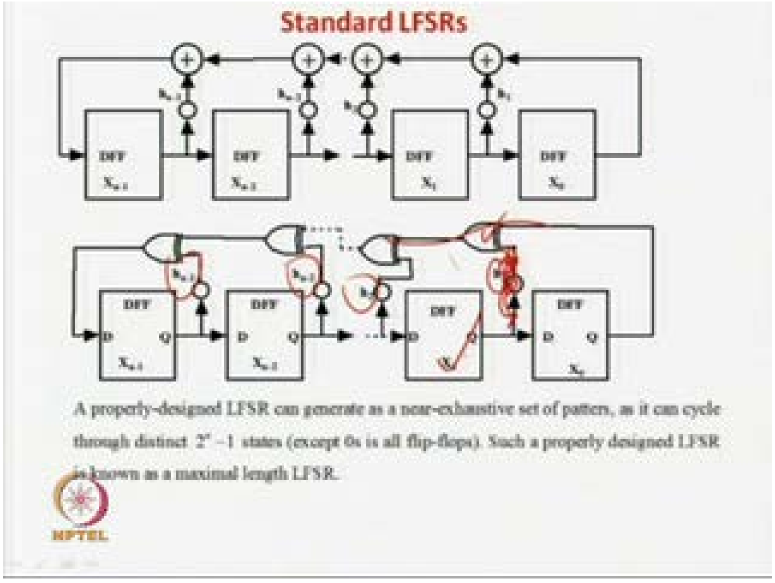
Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patters, as it can cycle through distinct 2^n-1 states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

So, this output will control the but, the input of the first will determined the feedback and you can usually the feedback . Now, it is not mandatory that you can see, something inches are there h minus 1 minus 2 dot dot h 1. So, they can be either 0 or 1 so, what do you say that if I say that h is equal to 1 that means what this this feedback is flip-flop this multiple will be this will be there and this thing will be there.

But, if it can also be the case that they do not require the feedback from this register of the register this place, so if we say that we do not require a feedback from this kh minus what do you call this h minus 1 flip-flop we can do is that this things will this will be not there h 1 will be. So this is not there this feedback directly connected to this 1 you see the X 1 so your X 1 is equal to 0 so this is not there at all this feedback is directly connected to this 1 you see that h 2 is equal to 1. So, your X 1, X 2, X 3, X 4 etc can be 0s and 1 so, if it is 1 than this feedback will be the if it is 0 feedback will not be use other it will be a bypass of that 1.

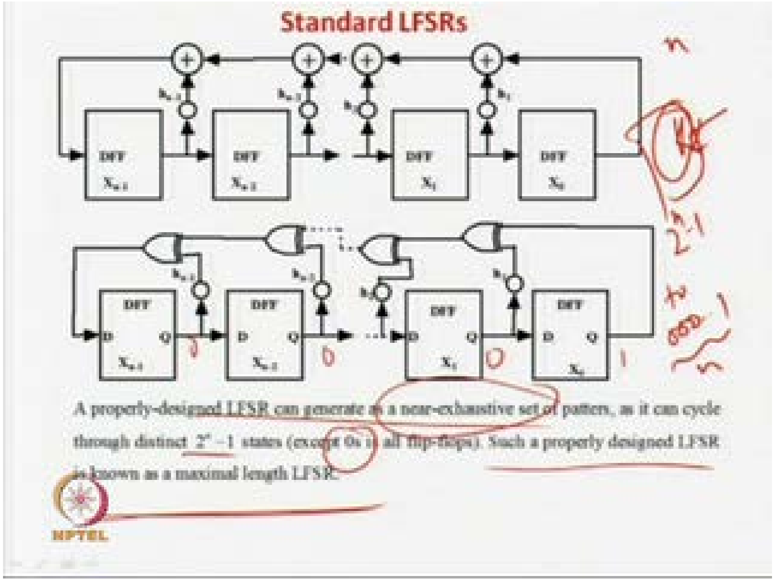
(Refer Slide Time: 28:03)


Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patterns, as it can cycle through distinct $2^n - 1$ states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

So, by changing this X 1, X 0, X 1, X 2, X n minus 1 so making them 0 and 1 you will get different type of LFSR. So, by on this will have some polynomial as we see something characteristics polynomial for this 1 so, whatever form whatever other way by changing this X 1, X 2, X n this one having different values of 0 and verse feedback is inward.

(Refer Slide Time: 28:30)


Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patterns, as it can cycle through distinct $2^n - 1$ states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

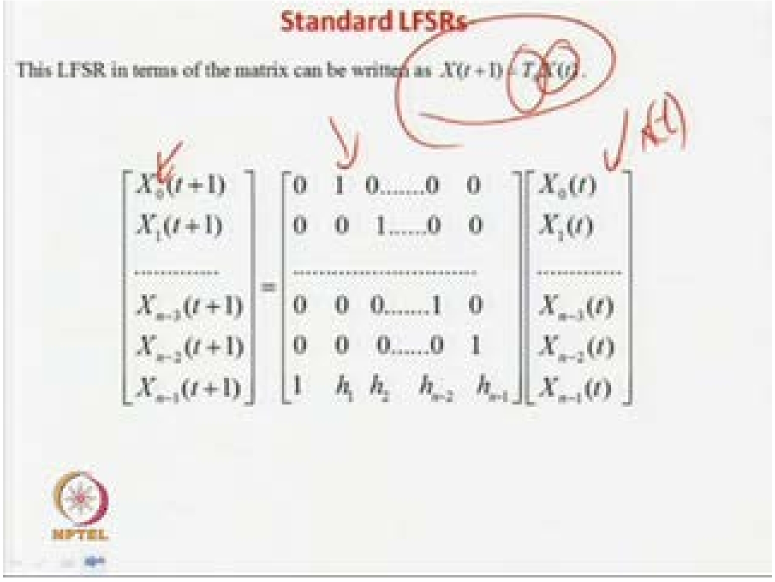
If X 1 is this in the feedback if it is 0 than it is not use in the feedback. So it will be directly connection over here and this thing is not there. So, by using this X 1 used to 0 and 1 you can get the different configuration of this LFSRs. So, that is what saying properly designed LFSR properly designed means using of proper values of X 0s and X values 0. You can generate a LFSR which will generate a patterns as it can cycle from minus from 1 states 0 is on there.

Such properties are called that is what we call . So, what we require we require some say k test patterns to a generated out goes we will see how will generate. So, now we even choose properly the values so in inputs it will choose what is the value of this 1 than you can generate 2 to the power n minus 1 to than would the minus lets exhausted manner near exhausted because 0 is the are included and it would be the random pattern.

And depending on the first seat like initial like registers in some direct vaules if you some error different you can say that this is 0, 0, 0, 1. So, this can 1 starting point so, this called the seat using different seats you can get the difference and patterns change or the cycle through minus 1 to all 0s and so forth what you can say that from the seat and you can circle the circle in would be the n minus 1 including 0s now this is called the maximal length LFSR because it is a full cycle.

But, sometimes we can do not make a full cycle start because you might k is important to generate. So, using different configuration of this X1, X 0 you can also go for call maximal length. So this cycle will be less than the minus 1 will not generate in all patterns in between 1 to X minus 1 but, generally what we do is we use a maximal length of generate because do not know the . So, what we do is equal we generally use a decimal length of LFSR and then we use a seat such way the important patterns is equal they offer first than these are unnecessary test for the after those in the cycle so what is important is general and what do you call this linear feedback register. So, we will see by this an example the things will be more clear and sometimes we want to you can say this we can said this a X 1, X 0, X 1, 2, 3, 4 we can say very carefully and you can have a non exhaustive and non maximization you can design such a way only in cause you require on generated by . So, that is why its you lot of the flexibility and the nothing but, only some exhausting. So, an all 1 X 1 which will not be one way some of them will be 1and some of them will be 1 0 now the area over a and only some of the exhausting nothing more than that why nothing more there there is require because even only if you want a feedback from the particular area otherwise is not the...

(Refer Slide Time: 31:40)

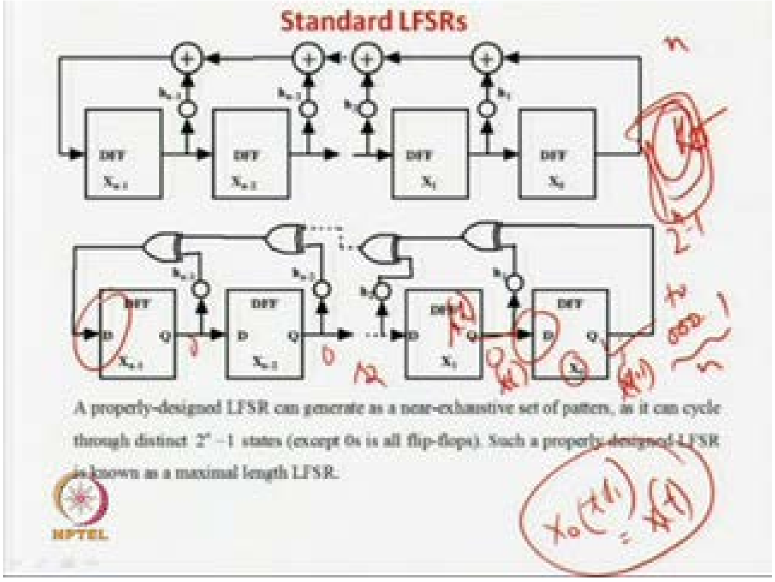


Standard LFSRs

This LFSR in terms of the matrix can be written as $X(t+1)=TX(t)$.

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ \cdots\cdots \\ X_{n-3}(t+1) \\ X_{n-2}(t+1) \\ X_{n-1}(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0\ldots\ldots0 & 0 \\ 0 & 0 & 1\ldots\ldots0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0\ldots\ldots1 & 0 \\ 0 & 0 & 0\ldots\ldots0 & 1 \\ 1 & h_1 \; h_2 & h_{n-2} & h_{n-1} \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ \cdots\cdots \\ X_{n-3}(t) \\ X_{n-2}(t) \\ X_{n-1}(t) \end{bmatrix}$$

(Refer Slide Time: 32:20)



Standard LFSRs

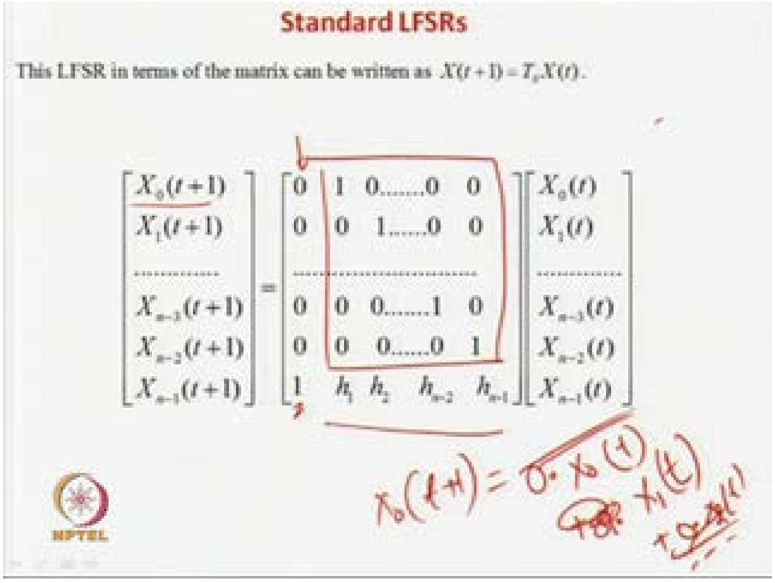
A properly-designed LFSR can generate as a near-exhaustive set of patters, as it can cycle through distinct 2^n-1 states (except 0s is all flip-flops). Such a properly-designed LFSR is known as a maximal length LFSR.

Now, this whole thing can be represented by a matrix. So, always a so what is there so generate in a matrix because Xt plus 1 is depending on the present value of the matrix and the present input value. So, Xt plus 1 is equal to t X into X (t). So, these are in to X of that is the present value of and these is your max value for all these (()) and this is the characteristics of the is equal to a tX.

So, will see how it can be generated, so you see so what is the idea so idea here is that so idea here is that if you look it very observe if you observe it very carefully the so what is the idea the input of X naught is the output of X 1. Similarly, the output of X 2 you can say is controlling the input of X 1 and so fore.

So, we see what I represent say that what I X 0 plus 1 T plus 1 what is the X0 t plus 1 will be actually equal to what is the output of X 0 t plus 1 will be the the input of that is input d so the output of Xt plus 1 will be equal to equal to actually Xt this 1 so, whatever time period plus 1 will be the what is if be dial you of d at time t what is the value of d X it is the output of X 1 a time. So, what you can say that the X 0 of the time t plus 1 is nothing but, equal to X 1 value of time t. Because, the output of X 1 is equal to of X 0 and at t plus a output will come here. So this is the similarly, it can interpreter for all other accepts the the flex of depends of lot of feedbacks.

(Refer Slide Time: 33:29)



Standard LFSRs

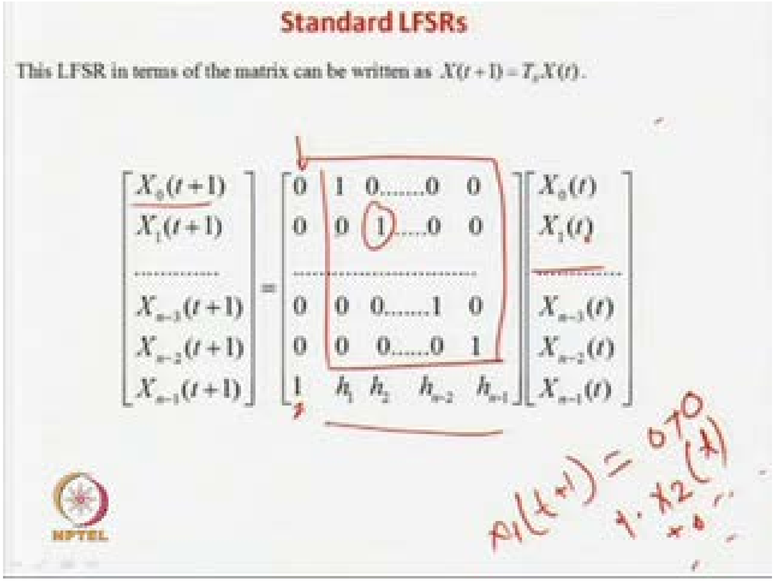
This LFSR in terms of the matrix can be written as $X(t+1)=T_s X(t)$.

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ \cdots\cdots \\ X_{n-3}(t+1) \\ X_{n-2}(t+1) \\ X_{n-1}(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0\ldots\ldots 0 & 0 \\ 0 & 0 & 1\ldots\ldots 0 & 0 \\ \cdots & \cdots & \cdots\cdots & \cdots \\ 0 & 0 & 0\ldots\ldots 1 & 0 \\ 0 & 0 & 0\ldots\ldots 0 & 1 \\ 1 & h_1 & h_2 \quad h_{n-2} & h_{n-1} \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ \cdots\cdots \\ X_{n-3}(t) \\ X_{n-2}(t) \\ X_{n-1}(t) \end{bmatrix}$$

$x_0(t+1) = \cancel{0\cdot x_0(t)}$ $1\cdot x_1(t)$ $+0\cdot x_2(t)$

So, that is why by using the matrix, so what will find out so we have to observe that this is nothing but, a diagonal matrix is there. So, the first column that the only 1 it is 1 this is diagonal is only 1 and here is actually one that h 1 h 2 dot dot that the h in minus 1 this is whether you want to have feedback or you do not want to have feedback. So, now what is the value of X naught t plus 1 we, can easily see that X is equal to the 0, 0 into 0 this is equal to 0 into X 0 T last this 1 at this1 at this 1 into X 1 of T plus of sum 0 into X 2 of n dot dot dot already 0 always things will be the eliminated and we will have only X naught t plus 1 is equal to 1 is that is . So, similarly you can find out

(Refer Slide Time: 34:29)



Standard LFSRs

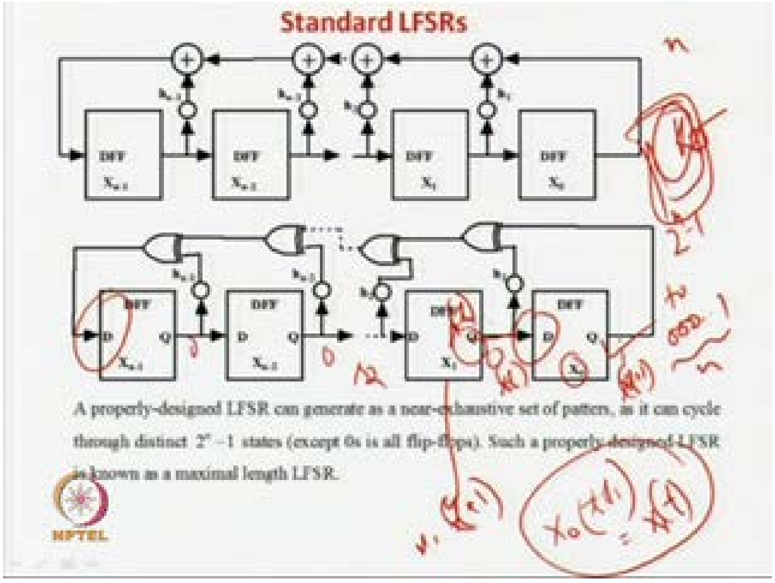
This LFSR in terms of the matrix can be written as $X(t+1)=T_s X(t)$.

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ \cdots\cdots \\ X_{n-3}(t+1) \\ X_{n-2}(t+1) \\ X_{n-1}(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \ldots\ldots 0 & 0 \\ 0 & 0 & 1 \ldots\ldots 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 \ldots\ldots 1 & 0 \\ 0 & 0 & 0 \ldots\ldots 0 & 1 \\ 1 & h_1 & h_2 \quad h_{n-2} & h_{n-1} \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ \cdots\cdots \\ X_{n-3}(t) \\ X_{n-2}(t) \\ X_{n-1}(t) \end{bmatrix}$$

$X_1(t+1) = 0 \cdot X_0 \ldots 1 \cdot X_2(t) \ldots + 0$

what is the value of this X 1, X 1 of t X 1 that is this one it is actually 1, so this is equal to this will be the 0 0 factor this one will be multiplied by this one will be the 0 factor so this 1 will be multiplied by this 1, so it will be 1 into X 2 of T of plus all 0 will be there so X naught t plus 1 that will be Xt 2.

(Refer Slide Time: 34:52)

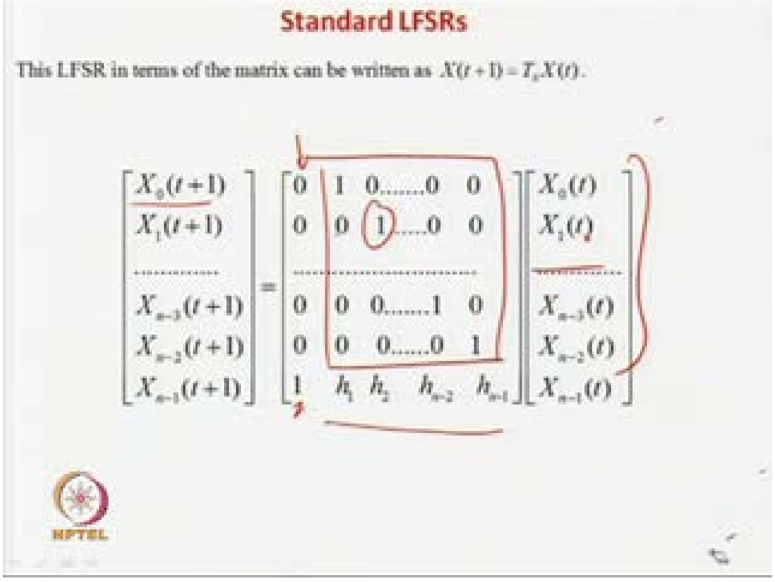


Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patters, as it can cycle through distinct 2^n-1 states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

$X_0(t+1) = X(t)$

(Refer Slide Time: 35:03)



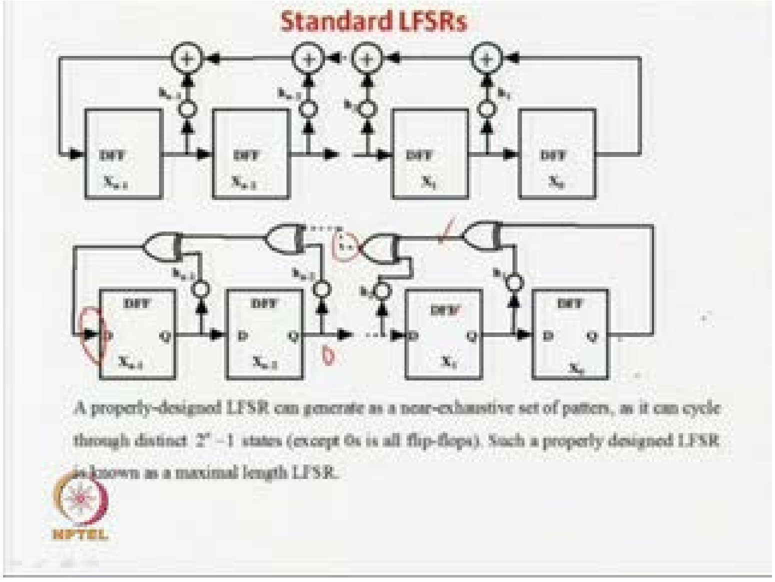
Standard LFSRs

This LFSR in terms of the matrix can be written as $X(t+1)=T_s X(t)$.

$$\begin{bmatrix} X_0(t+1)\\ X_1(t+1)\\ \cdots\cdots\\ X_{n-3}(t+1)\\ X_{n-2}(t+1)\\ X_{n-1}(t+1)\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0\ldots\ldots0 & & 0\\ 0 & 0 & 1\ldots\ldots0 & & 0\\ \cdots & \cdots & \cdots & & \cdots\\ 0 & 0 & 0\ldots\ldots1 & & 0\\ 0 & 0 & 0\ldots\ldots0 & & 1\\ 1 & h_1 & h_2 & h_{n-2} & h_{n-1}\end{bmatrix} \begin{bmatrix} X_0(t)\\ X_1(t)\\ \cdots\cdots\\ X_{n-3}(t)\\ X_{n-2}(t)\\ X_{n-1}(t)\end{bmatrix}$$

That is while so what is the output of this value of t plus 1 so X 1 T plus 1 it is equal to 1 will be X 2 these value is coming from a . So, similarly, we can find out that for all these the equation holds now, only you have to worry about the first flip-flop because its comprises the feedback so that is it we have to see 1.

(Refer Slide Time: 35:12)

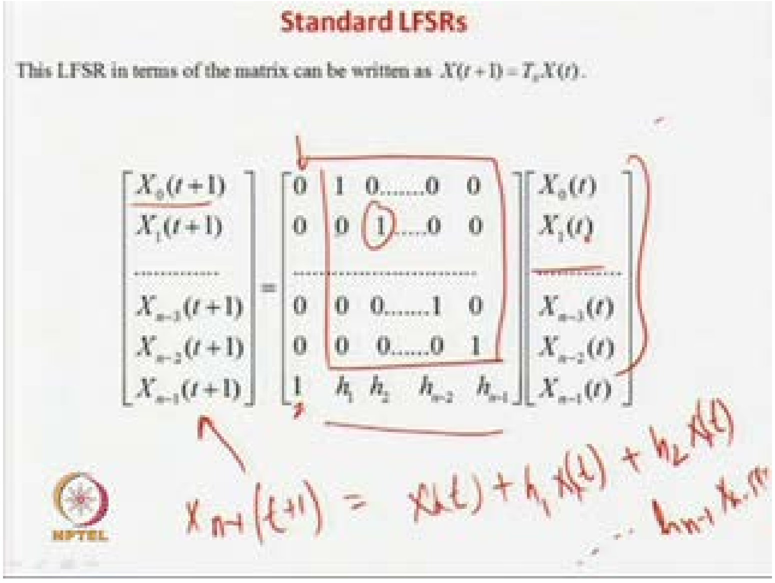


Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patters, as it can cycle through distinct 2^n-1 states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

So, this is the last flip flop we can see over here that we can see in the last fee for that. So what is the dependent on it is dependent on X naught and then last it h 1 in their then only it will be the each 1 it will be the so whatever will be the care right, so that the X naught will be there this 1 X are with the then X 1 of T and it will be if the equation is something like this input is dependent on this output exhorts with this one is h 1 of 1 similarly, the output exhorts with this output it will be the 0 similarly, this one with exhorts this 1 this is so output if something is to not will come there will not be considered let us see and again will come back.

(Refer Slide Time: 36:07)



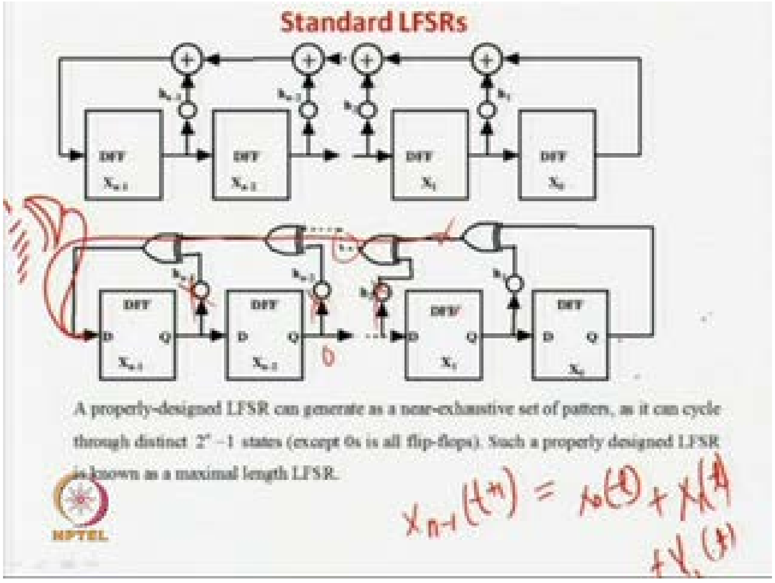
Standard LFSRs

This LFSR in terms of the matrix can be written as $X(t+1) = T_s X(t)$.

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ \cdots\cdots \\ X_{n-3}(t+1) \\ X_{n-2}(t+1) \\ X_{n-1}(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \ldots\ldots 0 & 0 \\ 0 & 0 & 1 \ldots\ldots 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 \ldots\ldots 1 & 0 \\ 0 & 0 & 0 \ldots\ldots 0 & 1 \\ 1 & h_1 & h_2 \quad h_{n-2} & h_{n-1} \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ \cdots\cdots \\ X_{n-3}(t) \\ X_{n-2}(t) \\ X_{n-1}(t) \end{bmatrix}$$

$$X_{n-1}(t+1) = X_0(t) + h_1 X_1(t) + h_2 X_2(t) \\ \cdots h_{n-1} X_{n-1}(t)$$

(Refer Slide Time: 36:08)



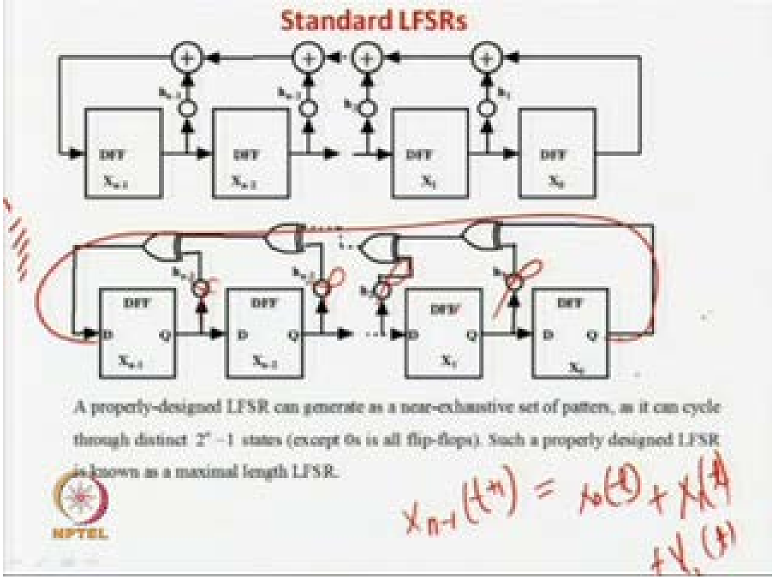
Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patters, as it can cycle through distinct $2^n - 1$ states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

$$x_{n-1}(t+1) = x_0(t) + x_1(t) \\ + x_2(t)$$

So, X n minus 1 so there is last flip-flop X n minus 1 is T plus 1 this 1 is equal to but, this X naught t this 1 is the correct this one is multiplied by correct plus h into a task is nothing but, . So, this 1 is h 1, X 1 e plus h 2 into X 1 t dot dot dot h n minus 1 then your X n minus 1 T. That means, what you see if somewhere X is 0 this X is 0 this will not be used in feedback where this part will be gone it will be till early other way . So, if even if say we considered all these things are 0 for this 1 is 0 this 1 is 0 this 1 is 0 we assume that. So, all these things there so what will be xn minus 1 xn minus 1 t, 1t plus 1 it will be nothing but, X 0 of T this will have to consider and it will only this 1. So it will be plus this is x on each 1 that is actually 1.

(Refer Slide Time: 37:18)



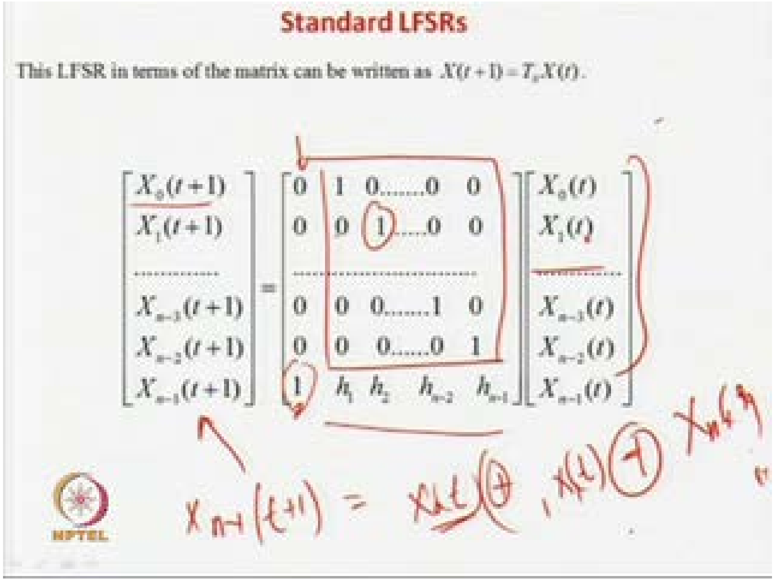
Standard LFSRs

A properly-designed LFSR can generate as a near-exhaustive set of patters, as it can cycle through distinct $2^n - 1$ states (except 0s is all flip-flops). Such a properly designed LFSR is known as a maximal length LFSR.

$x_{n-1}(t^{+}) = x_0(t) + x_1(t) + x_1(t)$

Now, so we can eliminate this 1 it is now 1 so it is 1 dot x 1 of t. This is the case all other 0 so, this is directly . But, if you want to use another over here, so you have to again put another extra here so it will be again x 2 . So, wherever it is age value is equal to 1, so you are using 1, 1, 1, 1 one more excess solvence. So, actually x will very big the x r over connected over here and there all the inputs coming to this one. So, which ever want to give feedback to the first flip-flop that age has to be made a want and it will be included in the equation. But, if you see is nobody wants to see to the feedback then what happens then on the only this fast this feedback will already be there all other swift not be there. So, the output of the first flip flop will be actually X naught. So, on this it will remain along with all the sequel that is what is reflected by this line.

(Refer Slide Time: 38:15)



Standard LFSRs

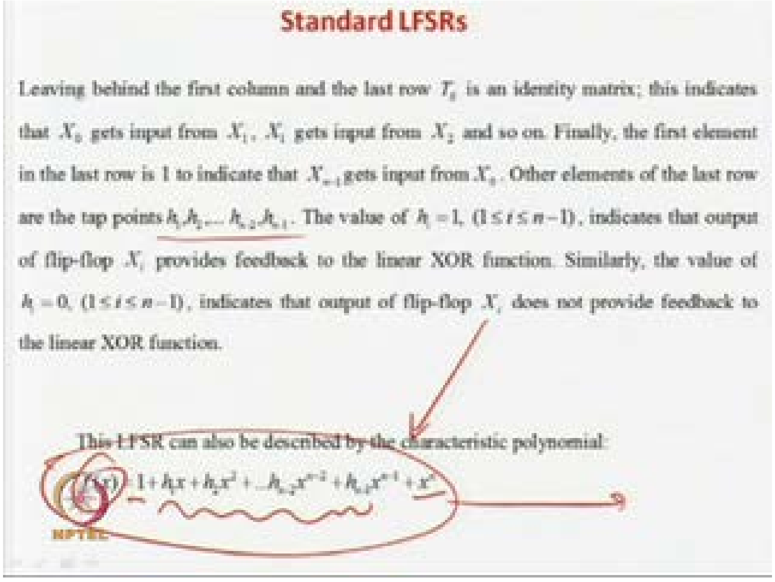
This LFSR in terms of the matrix can be written as $X(t+1) = T_s X(t)$.

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ \cdots\cdots \\ X_{n-3}(t+1) \\ X_{n-2}(t+1) \\ X_{n-1}(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \ldots\ldots 0 & 0 \\ 0 & 0 & 1 \ldots\ldots 0 & 0 \\ \cdots & \cdots & \cdots\cdots & \cdots \\ 0 & 0 & 0 \ldots\ldots 1 & 0 \\ 0 & 0 & 0 \ldots\ldots 0 & 1 \\ 1 & h_1 & h_2 \quad h_{n-2} & h_{n-1} \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ \cdots\cdots \\ X_{n-3}(t) \\ X_{n-2}(t) \\ X_{n-1}(t) \end{bmatrix}$$

$X_{n-1}(t+1) = X_0(t) \oplus , X_1(t) \oplus X_{n-1}(t)$

So, it is x naught Xn minus 1 t plus 1 is equal to fast that is why this is always kept as 1. So, whatever may be that is it nobody has give feedback to the first flip-flop actually last 1 will be the X naught will have to give it so, that will so X naught will be will always be there and this part with depend on whether h 1 h 2. So, if first 1 wants to give the feedback it will be 1 so, it will h plus nothing but, x 1, h 1 is 1 it will be once it will be there, so this is your feedback so, X naught t plus X 1, X 1 t p. Then, all others may not be liking to give best could be all 0 and may be this h 1 minus 1 minus 1 wants to give feedback. So it will be other X all the h 1 n minus 1 h 1 so you will X n minus 1 so this is also a given. So, this is how these matrix for also generate the and also represents this circuits.

(Refer Slide Time: 39:03)



Standard LFSRs

Leaving behind the first column and the last row T_s is an identity matrix; this indicates that X_0 gets input from X_1, X_1 gets input from X_2 and so on. Finally, the first element in the last row is 1 to indicate that X_{n-1} gets input from X_0. Other elements of the last row are the tap points $h_1, h_2, ... h_{n-2}, h_{n-1}$. The value of $h_i = 1$, $(1 \le i \le n-1)$, indicates that output of flip-flop X_i provides feedback to the linear XOR function. Similarly, the value of $h_i = 0$, $(1 \le i \le n-1)$, indicates that output of flip-flop X_i does not provide feedback to the linear XOR function.

This LFSR can also be described by the characteristic polynomial:

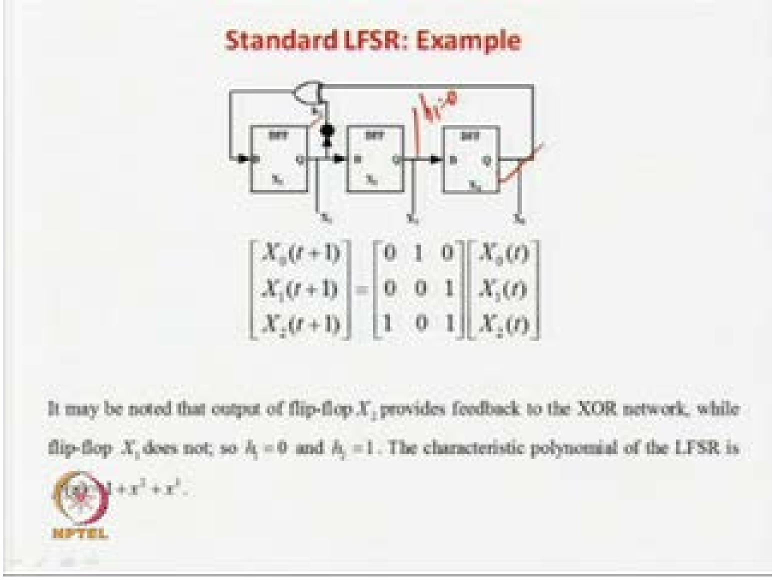
$f(x) = 1 + h_1x + h_2x^2 + ... h_{n-2}x^{n-2} + h_{n-1}x^{n-1} + x^n$

Whether, there are two two representation of your what you can call this the standard LFSR. So, now so we are not going to this very depth very well not going to that it is the very in depth here which is available so we sales that then we can say that so one more thing will say that they will also we can that can be represented by a function. Now, if you have this can have function representation of this one so, what do you how do you right the so we say that fX is equal to 1 is always there an excel is always there.

So, if you have this are so this is the fix will see how it we can say key this how it happens. So, these that may the characteristic polynomial we say that it is 1 and X n minus 1 can be last it is except it is the link of the n so this another factor involved in excel and (()). So, wherever you put have h is equal to 1 so, this X factor will be there. So X square will be there and wherever it is not there it is 0 that factor will be not there.

So, this same can also represented by a what you can call can characteristic for the . Then, they are lot of theory who will give the reference so we can see the they some reference to why now it will be what is the characteristic of polynomial which can this is the characteristic this also defined in which can be derived from this matrix. So, if there they they lot of theory its we can say that this characteristic polynomial cause of something trying of the some and all these are there we are not going to there can be theory because it will take you for very debt analysis of with polynomial for function all those things or characteristic function and all so, I mean so these kind terms are there an so many other examples are exhaustics theory which are not going in to but, given deistic by characteristics polynomial these theory will tell you whether this free flower this it is vary say whether it is going to exhaustics say of that is test patterns where it will be said become through 1 the n minus 1. So, if you choose the values of of the then you can find out the characteristic for no me the such as it will generate all possible patterns how to the n minus 1, 2, 1 an so for that they are sum for to choose the eight too the different way.

(Refer Slide Time: 41:13)



Standard LFSR: Example

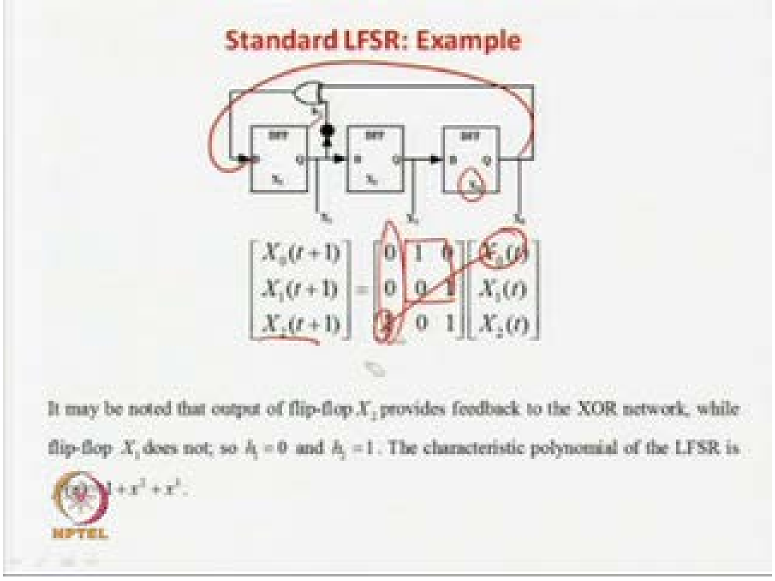
$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

It may be noted that output of flip-flop X_2 provides feedback to the XOR network, while flip-flop X_1 does not; so $h_1 = 0$ and $h_2 = 1$. The characteristic polynomial of the LFSR is $1 + x^2 + x^3$.

NPTEL

So, the characteristics for the polynomial in change and then you can determine that it is not going to generate the exhaustive set of patterns will not cycle through exhaustive test patterns. So, although those queries are there which can be easily determined by to get the characteristics for polynomial. So, depending depending on your equal way say that in accordingly and using the theory is an easily check whether it will rotate to this what you can call this exhaustive set of test patterns or not. So, this is generally whatever I discussed about the matrix that will the depending on the h 1 and. So 4-th this flip-flop may provide a feedback or may not provide a feedback so, that is the what will by the by explained about, the matrix about, this will go through this is the of the characteristics polynomial. So, mean not going to the theory but, by using the theory but, the on the characteristic polynomial we can easily find out the whether your exhaustive cause of the it will not go for with a can exhaustive generation and so 4-th. So, will start with an example. So, let us see the this is the LSFR right, so this is the x 2 x 1 and x 0 so this is there and you can see that these guy is not giving us feedback only this guy is an h 1 in this case is do anything so this is the feedback their so only h 2 is 1 this guy 0 h 1 . But x 0 always give the best feedback we cannot is an so no feedback is there. So, some LFSR in this case..

(Refer Slide Time: 42:38)



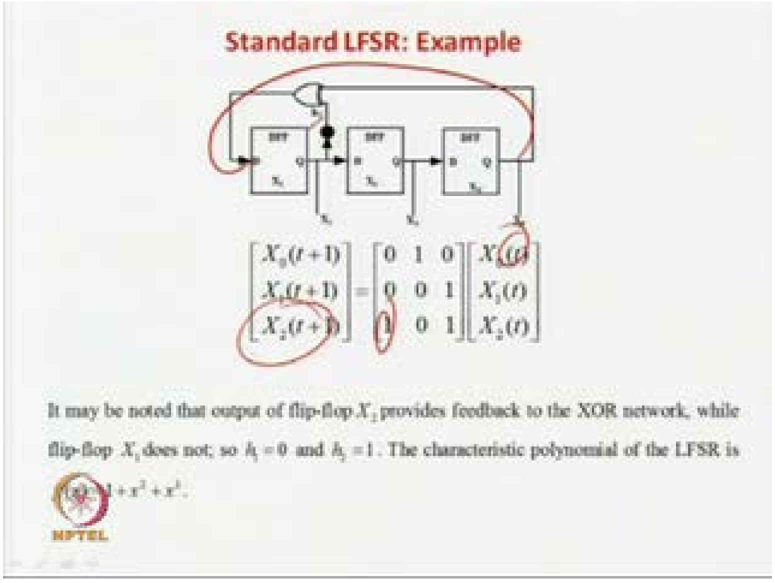
Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

It may be noted that output of flip-flop X_2 provides feedback to the XOR network, while flip-flop X_1 does not; so $h_1 = 0$ and $h_2 = 1$. The characteristic polynomial of the LFSR is $\phi(x) = 1 + x^2 + x^3$.

Now, what is this the matrix so you know that this is the case this is the magonal matrix. So,this is the case we for this call the column the last call the will be 1 by this last beet is the 1 because in the always no that exids that it is the must give up a back to the so these are to be care as the as the 1,2 because X 2 this multiplied by this act so this factor by multiuplaying the this actor so always we are going to get.

(Refer Slide Time: 43:03)

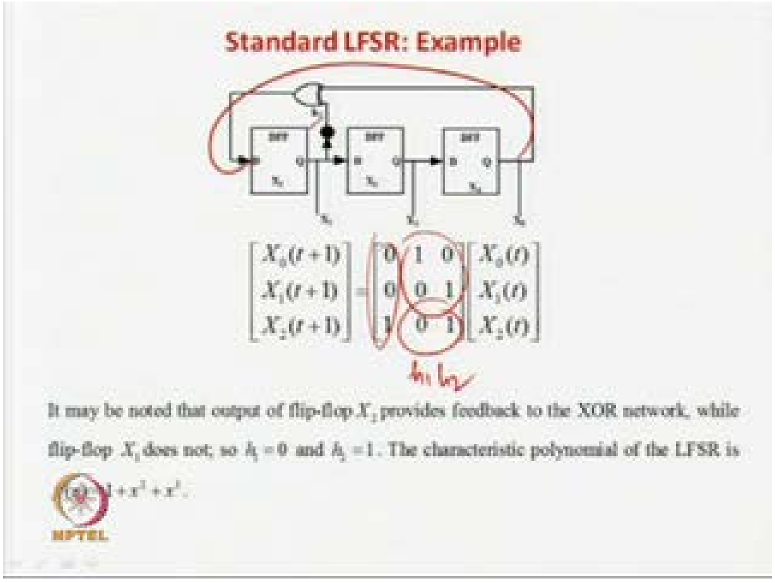


Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

It may be noted that output of flip-flop X_2 provides feedback to the XOR network, while flip-flop X_1 does not; so $h_1 = 0$ and $h_2 = 1$. The characteristic polynomial of the LFSR is $1 + x^2 + x^3$.

(Refer Slide Time: 43:07)



Standard LFSR: Example

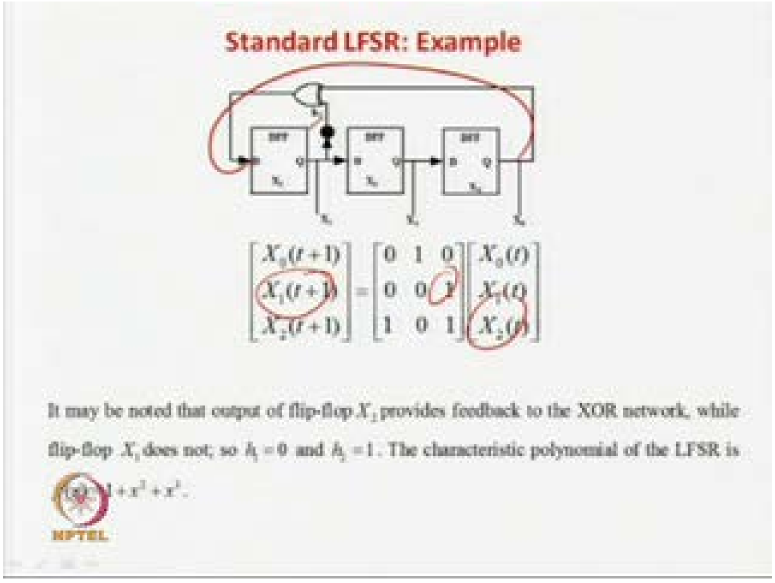
$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

h_1 h_2

It may be noted that output of flip-flop X_2 provides feedback to the XOR network, while flip-flop X_1 does not; so $h_1 = 0$ and $h_2 = 1$. The characteristic polynomial of the LFSR is $1 + x^2 + x^3$.

So, this will be this one always keep the this is the 1, so the implies the X 2 t plus 1 will be determined by N naught this is the minimum key that is required. So, now let us see he represents he can while diagonal matrix this is the 1 and this is 1 and h 1 is used put us h 2 is the (())

(Refer Slide Time: 43:29)

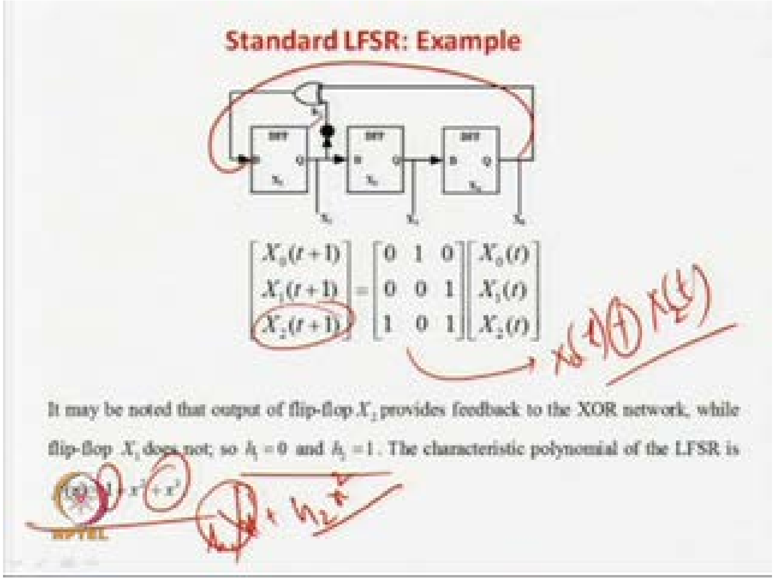


Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

It may be noted that output of flip-flop X_2 provides feedback to the XOR network, while flip-flop X_1 does not; so $h_1 = 0$ and $h_2 = 1$. The characteristic polynomial of the LFSR is $f(x) = 1 + x^2 + x^3$.

(Refer Slide Time: 43:30)



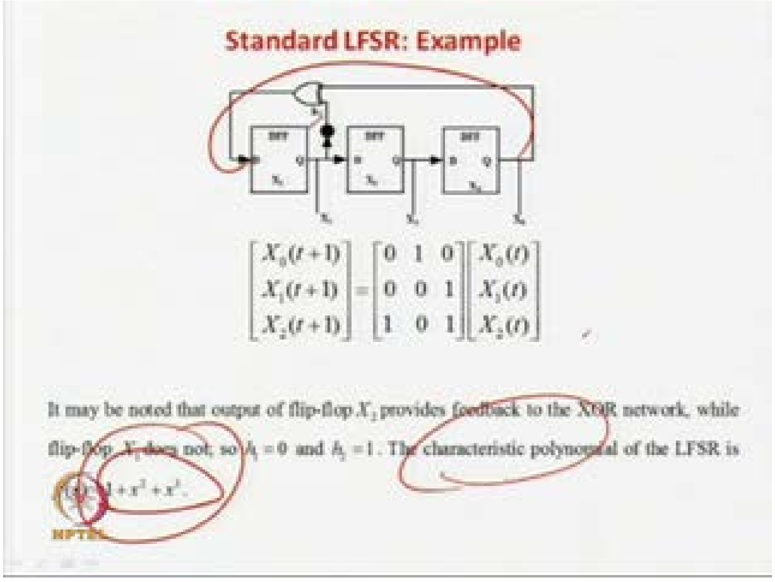
Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

It may be noted that output of flip-flop X_2 provides feedback to the XOR network, while flip-flop X_1 does not; so $h_1 = 0$ and $h_2 = 1$. The characteristic polynomial of the LFSR is $f(x) = 1 + x^2 + x^3$.

So, you can easily see that what is the case so X naught the p plus p plus an is nothing but, X 1 product of this 1 X naught p plus 1 is nothing but, X 1 than the we can say that X 1 t is nothing but, X 1 muliple that x 1 by nothing by X 2 so this is nothing but, the X 2 and in the final case that is this last 1 this 1 is nothing but, is equal to the X naught t plus 0 will not be there o this one and this 1 . Now, this is the matrix with the also we can go for the what you can call this 1 this characteristics what will say.

(Refer Slide Time: 44:15)



Standard LFSR: Example

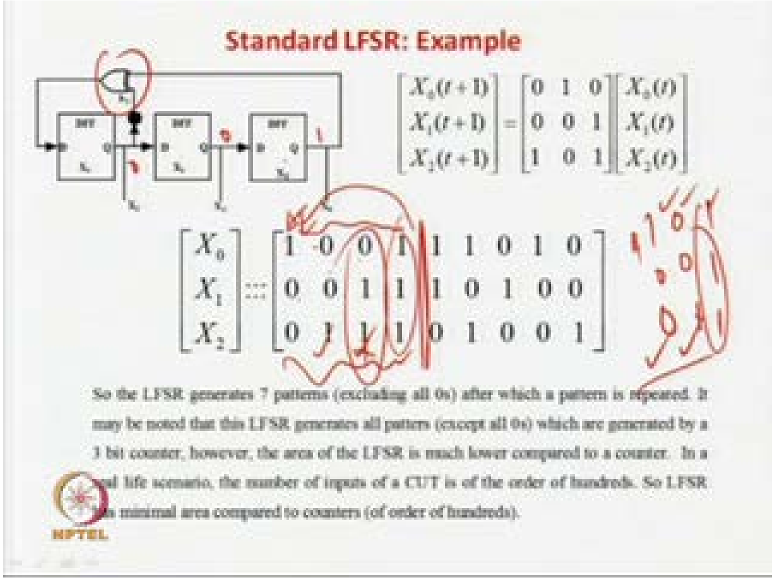
$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

It may be noted that output of flip-flop X_2 provides feedback to the XOR network, while flip-flop X_1 does not; so $h_1 = 0$ and $h_2 = 1$. The characteristic polynomial of the LFSR is $f(x) = 1 + x^2 + x^3$.

You also an plimer of an representation with this is the case, so you no that this thing is there so this with the X will will be their can . So, this 1 will be X which 1, X 1 plus h 2, X square is the X 1 is the 0 is the not their the only this one means the I have. So, this is the characteristic polynomial for the LFSR of this LFSR standard LFSR. Now, an important thing about that your theory is the if you study this with the find out what is this type of this characteristic which will tell with now, this is the characteristic polynomial which can generate all exhaustivec generate that theory will not and there are an list we look at the difference some standard already uploaded I mean sports we can see that the textbooks of the standard of the 5 stop the characteristic exhausted so these type of characteristic polymer.

So, the level of during that the generate the exhausted test patterns pattern in so, you can select the so, now will go to the example so in this case already their the at this 1 I the at this the 1 plus the X cube is already the no to be remaining the exhasuted and the already in this cause so that, an that is in the list of the exhaustive that same on your near exhaustive and in exception 0 and all pattern in 0 this characteristic which is in their list exhausted of the exhaustive that the pattern of the polynomial state. So, there is an list of a exhaust with y the characteristic polynomial in this the characteristic and it will generate all the possible pattern.

(Refer Slide Time: 45:49)



Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

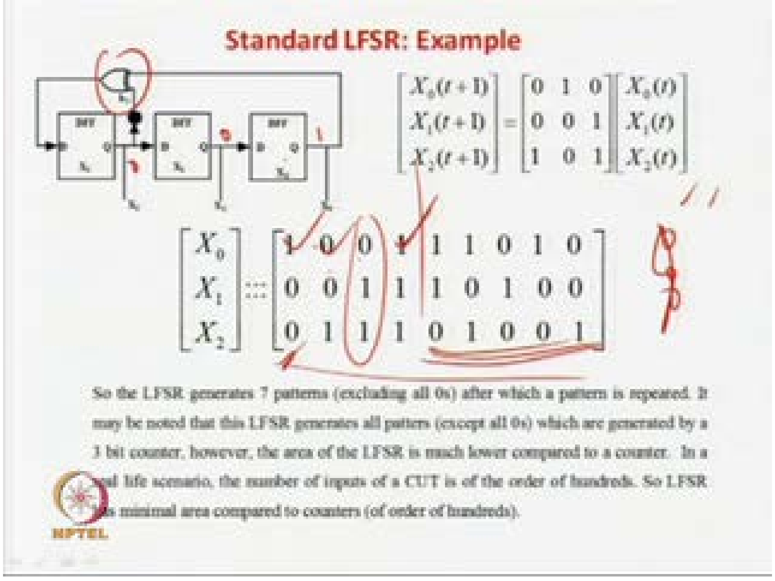
$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} ::: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all patters (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

So, this one . So, it will the expected to their also the well pattern the theory you can look over there.So, this is your matrix already have seen now, what is the requirements where is the see that it wants So what is this I have already have mentioned that you have to give a seat. Seat cannot be all 0 so you have to give a seat. Seat means it is a starting point so starting point is a starting point is we are giving a one over here you can see the X 0 is 1 X 1 is say we can say that s one is 0and it is a 1. So, we can said an incentive of an resource accordingly because by an by any meaning that So, you can also start with the 1, 1 and 1 that is not a problem you can apply. So, these see for a example as I told you so for example, the testing associates very important test pattern is so for th such a this 1 this is 1, 0 ,0 is important then you can say What, I will see is that so if I start with this, so these is start from,. So, these is start point these is see than first clock one partially testing my faults next I will get this very good. So, 2nd pattern is now again these 3rd metal is not of requirement not my requirement because I mean 1 1 1 this will not anything this will go blank. Now, what happened now this one one one is equal thin it is when ever you require again again you get certain come back so there are two type of once you can say you can say characteristic polynomial in such a why so the generate only this pattern it is possible it may be possible it will generate all these patterns in this sequence many of the do it for a combinational circuit.

So you can try out that we want to sequential circuit it will generate these and these these and these only these and these we will find out that it will take much more gates then a single it will be much much using the optimize state here also four then you go for this minimization of these is 1 these is 2 b these is 3 b so, 3 to optimize input for all those stand state machine designing you see you always know design is a state machine is insert this will be much more 0 single X . So, if you go back to that way if circuit can design and implement bring out this patter this pattern and this patter but, it will be take much more number of gates then huge no of gates these area awarded equal restriction.

(Refer Slide Time: 48:46)



Standard LFSR: Example

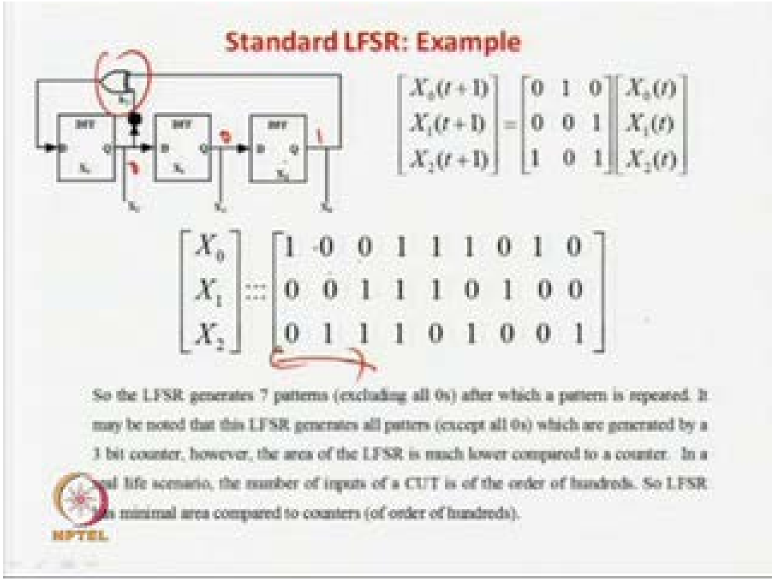
$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} ::: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all pattern (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

So, what we are doing using the random pattern thus it is various only then are there generating the exhaustive it in pattern excepting the case of 0 0 0 that cannot be generated because you we say you will hang only all these generated require on the 1 requirement is this 1 this 1 and this 1. So, only we start with this seat only with we get one will be pasted and all of us is very good for us. But, if we start with this seat problem because this is a important requirement you say the important pattern than again we have to give all the junk patterns if we go back to this.

(Refer Slide Time: 49:25)



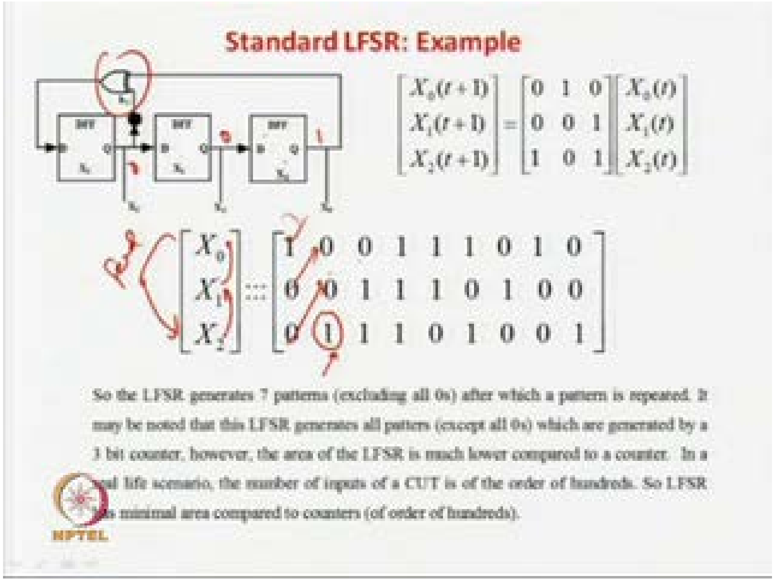
Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} ::: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all pattern (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

(Refer Slide Time: 49:25)



Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

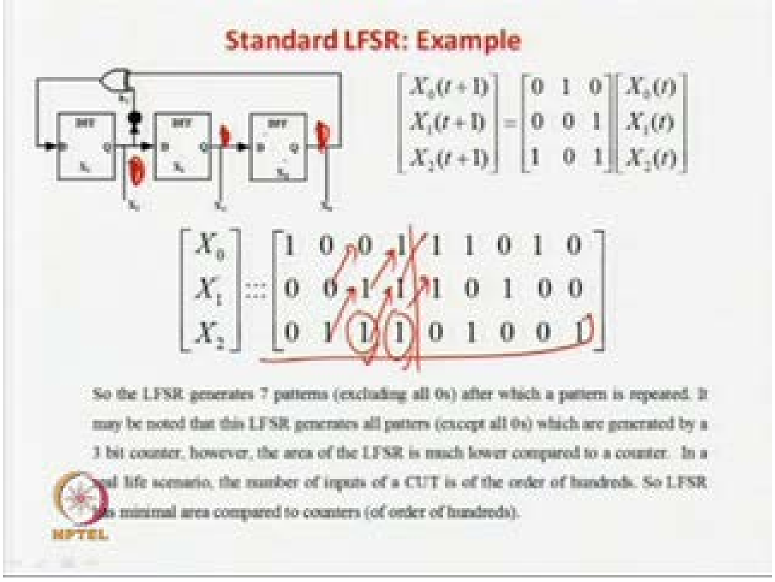
$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} ::: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all pattern (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

Theme selection and characteristics polynomial section is very very good and is very very important. So, this type of requirement easily find out characteristics polynomial which will generate always these patterns is always done that is very good but, some time it may not be possible that in that case what we will do is exhaustive case than we actually put this CD in such a way than the important patterns come first than the non important pattern. So, in this case we have put this has 0, 0, 1over so we see what what is going on so this X 0 input will control this so will control this next time you know that what is going to happen.

So, you know the what is going you have this X 1 will go to the value of this 1 and go to along with X so will control the X naught ot andx 2 will control the X 1 and X 0 with that it will control the exits. So, X 1 control with X 0 X 2 will control X 1 and this exceed X 0 and X 0 with feedback it will control the X and X 1 control with 0, X 2 control with X 1 and this X . with feedback and X 1will be characteristics.

(Refer Slide Time: 50:25)



Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} ::: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all patters (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

So, these are your matrices so this one go with way and this last will depend for the it will depend on this one and this one controller with got him this 1 x are0 nothing but, the 1. So, this1 are going to get by this X 0 with X 2 this way so this guy and this guy going the why these are the this pattern is generated. So, what is the X pattern now in next pattern is 0 X pattern is now, 0 0 0 and that is the 1 over here now. So, now what the next pattern over here you know that this 1 X will controller this case and now how to determine X 2 it will depend on the X are 0 and certain one nothing but, the 1 this is have give will be the 0, 1, 1.

So, now we are going as 0, 1 and 1 . So, now again this will be the case and now this last be again depending on 1 and this 1 are X are 0 this 1. So, now we are going to get so its a 1 X 1is x 1 and this is 0. So, you can find out the next pattern you will something like this 0 X are 0 something. This is how this pattern will go on and what is that find you can study that is will go by whatever if you go through find out that after 0, 0, 1 that is not X see though X 1 and X 1 this is 1.

So, you will get a repetition there repetition has to be there and this is here what is going to their this one going to there and 1 X are 0 these are 0 the after that it is so 1 X 0 is 0 X naught is 1 so this pattern what will be the case and this is now you know that it is will be 0 over here.

It will be a so what is the next pattern and to compute what is the next pattern of 0 0 1. So, X naught is 0 X 1 is the 0 and X 2 is 1 over here. So, what in the next pattern of this 1 so you know that X determine this 1 0 it will be 0 over here then, what is the next pattern here, so this is the case from the next pattern will be so will be one over here and this 1 you can determine see over here and X the and X are with this again a 1.

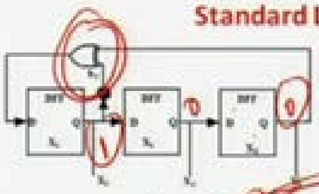
So, it is and we will getting a 1 over here. So, you can see that this pattern is repeated. Sso, that is why i am saying so again the start will be there and lot of repetations over here so, what is the next and then emphasize that it will start this one so what is going to happen and going to all those pattern and after that there is another pattern that which is actually repeat.

So, that means what is generated 7th pattern 1 of the case is we know that this pattern was an exhausting pattern so, what we can say polymer is a what you than call existing polymer kind of this the characteristic polymer such that will exhausting set of patterns so, which have generated all the say this pattern starting from this an after this redial these are the 0, 0, 1 is the repetation over here. So, that means what based on remove this general step and then another pattern will start out .

So, that is after generally exhausted the pattern one more pattern is began in this expression was what we call on existing paranormal because possible, so you can a according to this from a sheet and then finally, important and pattern will up to here what you can reset. So, basically what we have a chip we have achieve are you can also think that the important patterns are over here.

(Refer Slide Time: 54:13)

Standard LFSR: Example



$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

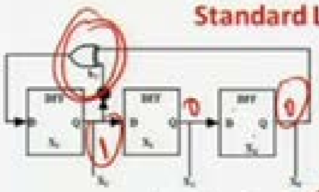
$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} ::: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all patters (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

So, we achieve start with this one see for we bounds in the bad situation are you can say the worst situation and you can have something like so this is an important so this is an important but, these are the pattern see go for what we have that they see how we achieve this pattern in a better way so that more more that will seen on the next class but, today what are the main idea was that to show them that just using an and changing the values of any X 1, X 2 and X 3 whatever you can so generate exhausting say the patterns.

(Refer Slide Time: 54:47)

Standard LFSR: Example



$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

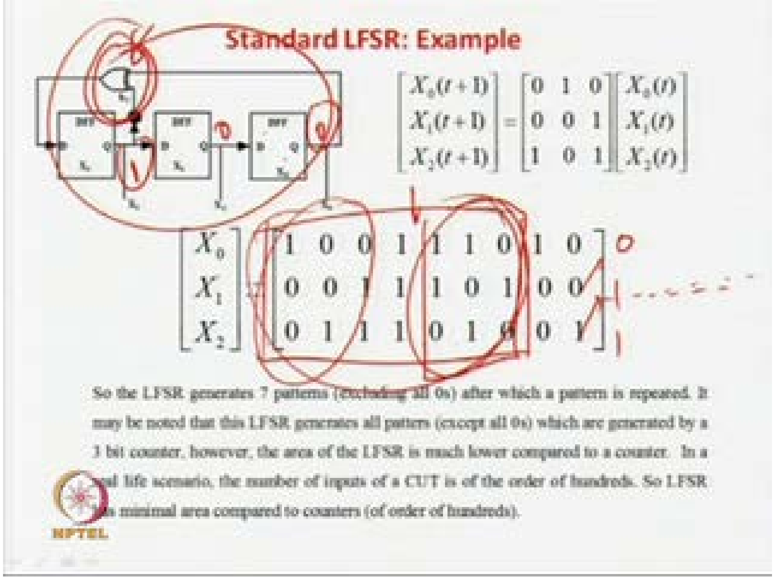
$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} ::: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all patters (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

So, area requirement is already 1 X 1 and X 2 target and by depending on this sheet implies that we can use the this one and also this one pattern that that will also we see trying the different see and your pattern will be different. But, again will be the cycle over the exhausted that is very important it will go an up is to all the exhausted that is but, see that if you see that sit Og pattern of important of you they are in X y this 1 also to will start with the sits the very quickly the we will get this coverage but, a say you can start the round this.

Similarly, if you have some other important the say your test pattern of an important the this one say your repetition you can the and see the repetition the so, up to these only the is this thing the So, after this t ,1 0, 0 this 1 is the so after that again the we will 1, 2, 3, 4, 5, 6, 7. So, after 7 the is the repetition the this guy 1, 0, 0, 1, 0, 0 so after save an it will always the always the replace that will gain the easily see.

(Refer Slide Time: 55:50)



Standard LFSR: Example

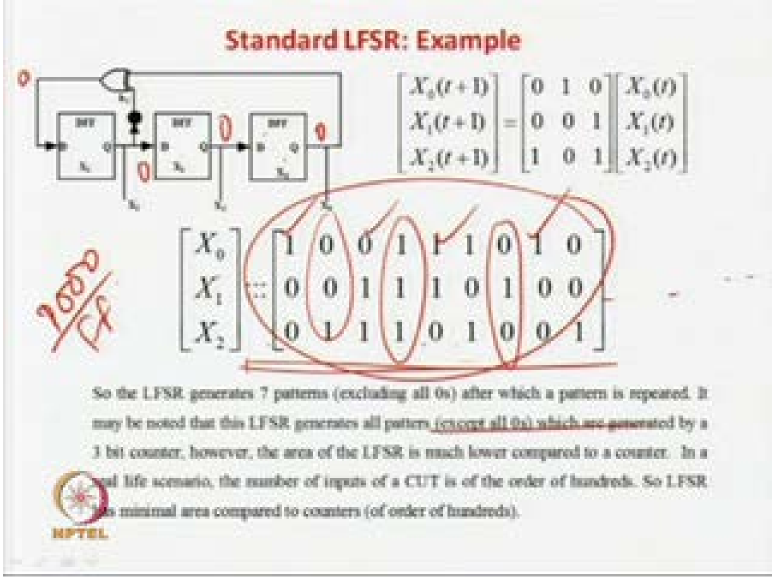
$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all patters (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

So, this is only the exhaustive part this part the 7 elements 1, 2, 3, 4, 6, 7 after that we will get some repeat them is a dot dot dot dot. So, after this one will say it was 0, 1, 1 this was the dot dot we are already and will be repeated. So, this whole guy will be again with the that already we have checked on this one will be, so again this will be periodical on this 1. So, this is the block of the so depending upon the whatever it is the importance it will be the importance lies over the there are importance of the lies all of the here pay choose accordingly and we can do so, another advantage with the is by choosing the seen properly and properly exams there is characteristic polynomial you can use the whole area of a to generate the pattern of the importance. But, if you want to do it the deterministic with the this I need a only these and not any other then will be the big problems because we have to go for final statement design fundamentals so, we will have to the optimize the number of the then also it will be much larger than as serial exhausted.

(Refer Slide Time: 57:00)



Standard LFSR: Example

$$\begin{bmatrix} X_0(t+1) \\ X_1(t+1) \\ X_2(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0(t) \\ X_1(t) \\ X_2(t) \end{bmatrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} :: \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So the LFSR generates 7 patterns (excluding all 0s) after which a pattern is repeated. It may be noted that this LFSR generates all pattern (except all 0s) which are generated by a 3 bit counter, however, the area of the LFSR is much lower compared to a counter. In a real life scenario, the number of inputs of a CUT is of the order of hundreds. So LFSR has minimal area compared to counters (of order of hundreds).

So, we do not have the that much area flexibility and the or that much area of the operator capability in best 2 have to this who you go for a little register and then by choosing the property sit properly wherever is all important see important test patterns are there which choose this there but, in the it is not the golden but, life is not the as been as the sometimes we may have patterns like this 1 gap then, what is the cause what they then we are a loss then what is the idea then I mean at already .So, we have to go for round type in the losses but, if you good have all generate the better the area would have been larger but, he would done the this faster but, any you do not have that area a flexibility but, there what is this solution find out some electristics polynomial and some see, so that it will generate only this patterns of the may be worn out to be more than then particular cannot guarantee but there will also a guarantee may be possible but, what does the compassing with the we are gene ring the all easiest pattern the based on the se we an generate the important pattern is and using the various these very less of the.

So, this was about the LFSR and what is the importance in patten generation there is an end of generation an area this one So, on the only all 0s cannot be generated that is there is obligation can find out that the put all 0s then what is the case that so, all 0s can this 1 is 0s of the again 0 this will be 0 cannot generate anything else. So, all 0, 0 is not allowed to genatare. Now, will must be an that if this is are 3 input that thing that inputs have the 1000. So, if you have to generate exhausting pattern like this, so in your this as you require 1000 t plus and some more steps is 1000 I will less than but, he will want to extra the pattern generation domestic patron generation like a digital circulation design is having months of design then if area of over head if you extremely high for each of the inputs of these if you generate of the different combination which will not patterns there will be occupation on at least of the very high in number but, in cause with you in cause of few number of we should do this.

So, with this we stop our lecture for this day and in the next class of this module we are going to see another type of standard LFSR was there so we will see the next type of LFSR and this impact which call the modular element so, what is the advantage and what is this advantage and then we are also see about the response compaction that is the why the response compaction and what is the advantages so that will be next lecture which will complete our discussion.