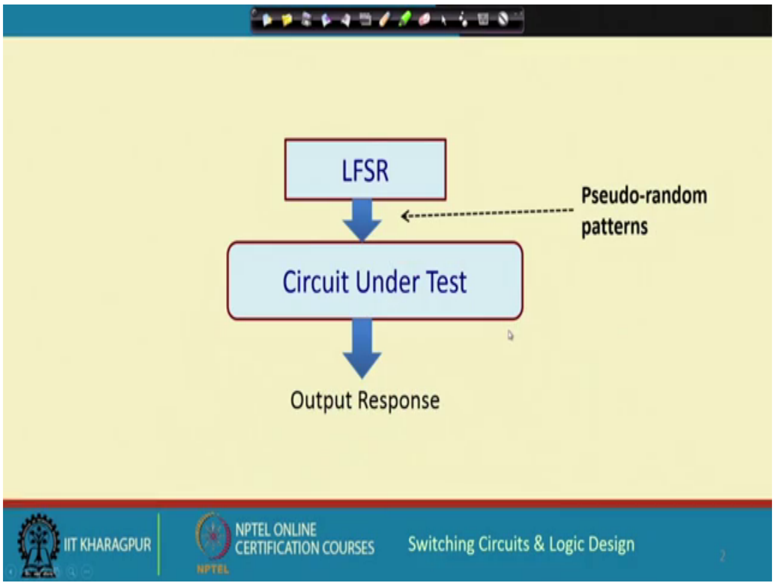
Switching Circuits and Logic Design

Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 60
Built-in Self-Test (Part II)

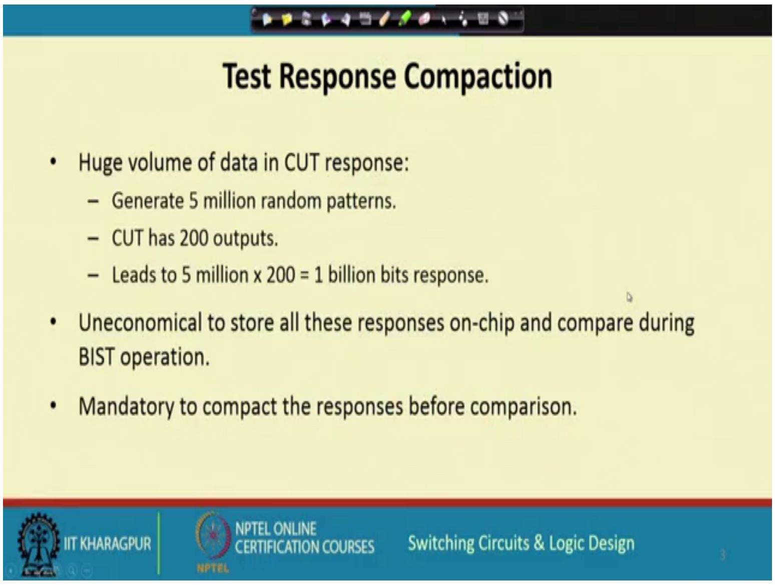
So, we continue with our discussion on Built in Self Test; in our last lecture we talked about how we can generate the test patterns inside the chip. Now in this second part of the lecture built in self test we shall be talking about the so called Response Compaction Part.

(Refer Slide Time: 00:36)



So, if you look into this picture we have already seen how using LFSR you can generate the test patterns the so, called pseudo random test patterns. Now let us say we are applying 10 to the power 4 or 10000 such patterns. So, at the output of the circuit we will be getting 10000 circuit output responses. What do you do with these responses? How do you take a decision whether this is good or bad? This is the question we are trying to answer now ok.

(Refer Slide Time: 01:19)



Test Response Compaction

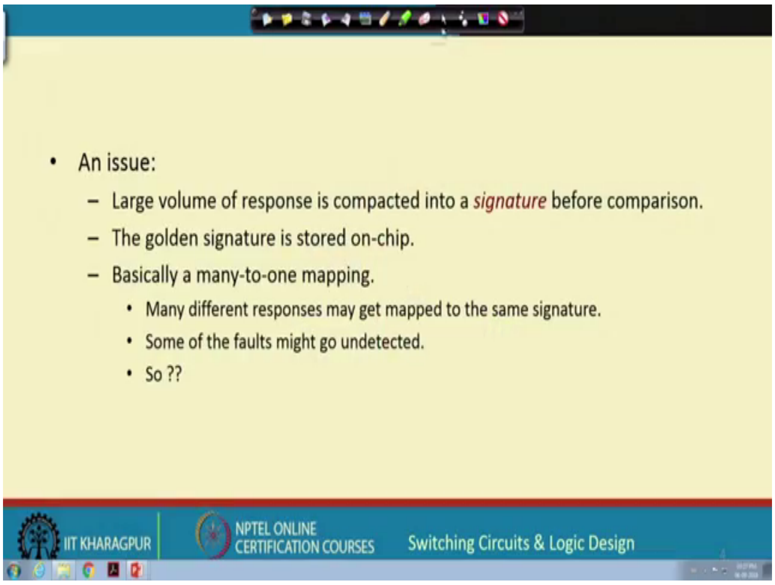
- Huge volume of data in CUT response:
  - Generate 5 million random patterns.
  - CUT has 200 outputs.
  - Leads to 5 million x 200 = 1 billion bits response.
- Uneconomical to store all these responses on-chip and compare during BIST operation.
- Mandatory to compact the responses before comparison.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Let us see, the first thing to note is that we required to do something called a compaction operation. Why is it required? Let us take an example just like I was saying that the circuit output can generate huge volume of data. Suppose realistic figure we are applying 5 million random patterns, the circuit can have 200 outputs which is pretty common in modern day circuits. So, how many bits of data is generated? Number of test patterns multiplied by number of outputs; so, it will be 5 million multiplied by 200 this translates to 1 billion bits. So, what do we do? Do we store all this 1 billion bits in memory and for every pattern the output generated we compare it; this means we require a 1 gigabyte size memory just to store the circuit responses well; obviously, this is impractical right.

So, we have to do something else. So, it is really uneconomical or impractical to store all these responses on chip. So, we have to do some kind of compaction before we compare the circuit responses ok. This motivates us why do we need compaction.

(Refer Slide Time: 03:00)



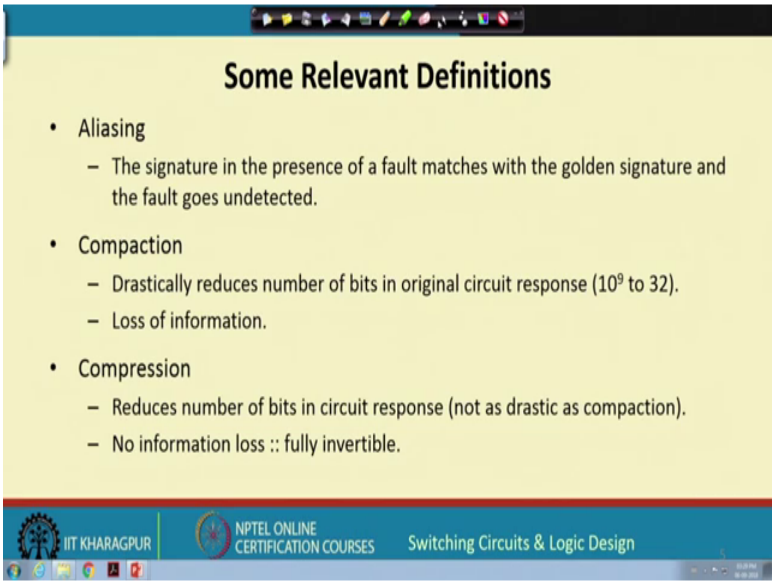
- An issue:
  - Large volume of response is compacted into a *signature* before comparison.
  - The golden signature is stored on-chip.
  - Basically a many-to-one mapping.
    - Many different responses may get mapped to the same signature.
    - Some of the faults might go undetected.
    - So ??

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

Now, the point to notice that we have huge volumes of data, we said the circuit is generating huge volume of data then we have a compacting operation let us say C. The compaction operation test is huge volume of data and generates a small compacted response and this small compact response is referred to as a signature. So, what we store in a ROM in a because this signature is relatively much smaller in size you can store this golden signature; that means, the signature in the absence of any faults.

So, through fault simulation you can do this experiment beforehand, you calculate the good signature, you store it in a ROM and you can compare this signature with the signature stored in a ROM whether they are equal or not right. But, you see here we are having a many to one mapping; a huge set of data is being compacted into very small signature typically, let us say the signature can be as small as 32 bits. So, there is always the possibility that two or more input patterns in the set can map to the same signature because of which some of the faults might go undetected. So, what do you do? What is the probability of that? That a fault is occurring, but it is going un detected let us try to make a simple calculation.

(Refer Slide Time: 04:52)



Some Relevant Definitions

- Aliasing
  - The signature in the presence of a fault matches with the golden signature and the fault goes undetected.
- Compaction
  - Drastically reduces number of bits in original circuit response (10^9 to 32).
  - Loss of information.
- Compression
  - Reduces number of bits in circuit response (not as drastic as compaction).
  - No information loss :: fully invertible.

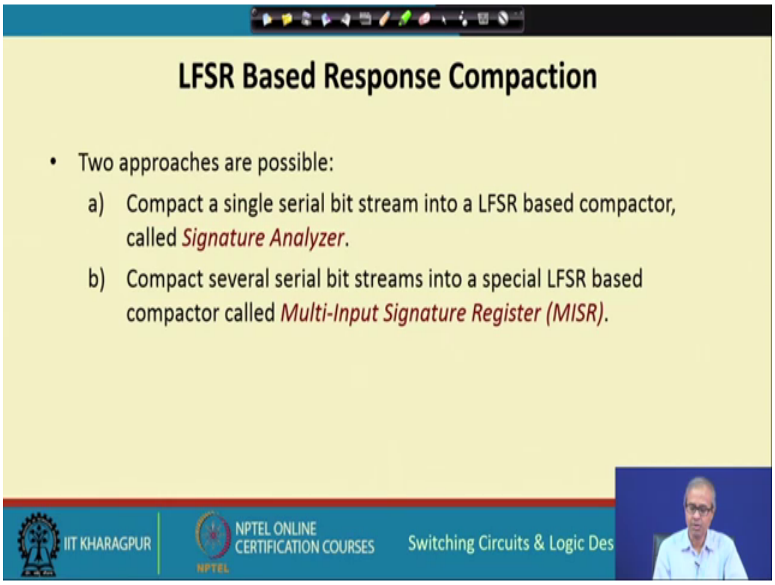
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

But before that let us talk about a few terminologies that is something called aliasing. Well, aliasing said just the point I have made that there is a fault, but the faulty signature by accident becomes the same as the good signature ok.

The signature in the presence of the fault matches with the golden signature and under such conditions fault goes undetected, we say that aliasing has taken place. Well to reduce the size of data we broadly use two techniques: one is called compaction other is called compact, this is called compression. Now in computers sometimes we compress files there are commands like tar, zip, rar many commands are there. They are used to reduce the size of a file, but this is a temporary process whenever we need the file again we can unzip it, we can expand the file again and get back the original. So, this compression is a reversible process, you can do compression you can again uncompress it ok.

Reduces number of bits, but there is no information loss it is fully invertible. But, in compaction you cannot go back, you can drastically reduce the number of bits like you can have a huge 1 gigabit data you can bring it down to as low as 32 let us say, but there is some loss of information. From this 32 bit you cannot go back to the original 10 to the power 9, this you need to remember. Now in circuit testing in best we go for compaction, we do not go for compression ok.

(Refer Slide Time: 07:00)



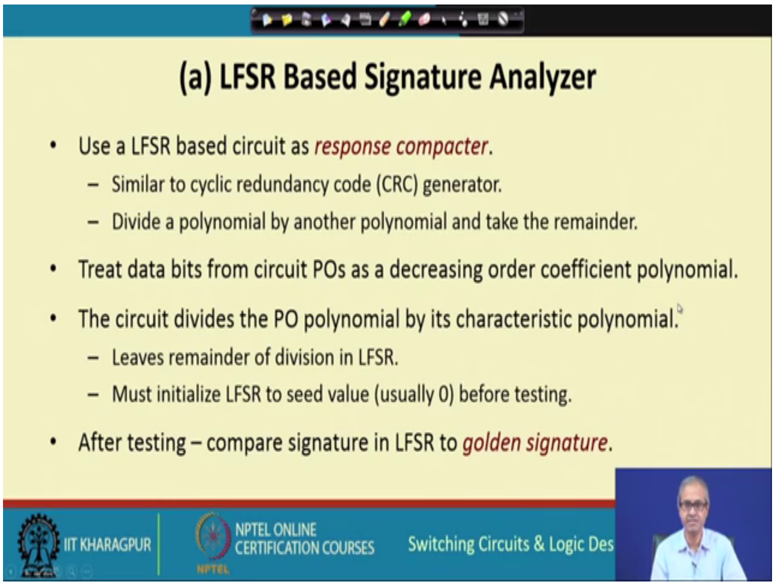
LFSR Based Response Compaction

- Two approaches are possible:
  a) Compact a single serial bit stream into a LFSR based compactor, called *Signature Analyzer*.
  b) Compact several serial bit streams into a special LFSR based compactor called *Multi-Input Signature Register (MISR)*.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

Now, broadly speaking this compaction and compression can be done in two ways: one using something what signature analyser, which can compress a single serial bit stream; that means, assuming the circuit has a single output. So, single bit stream is coming and second alternative is something called multi input signature register, where the circuit can have multiple outputs. So, we can compress all of them together ok, let us see how.

(Refer Slide Time: 07:40)



(a) LFSR Based Signature Analyzer

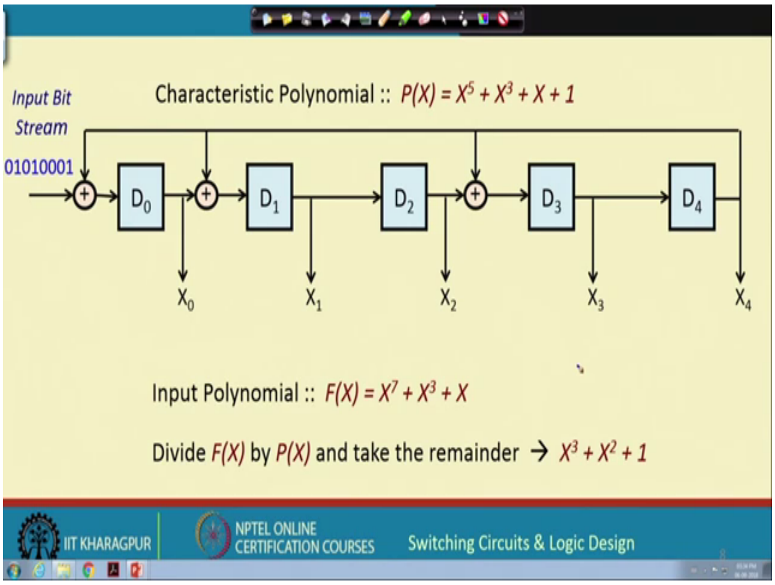
- Use a LFSR based circuit as *response compacter*.
  - Similar to cyclic redundancy code (CRC) generator.
  - Divide a polynomial by another polynomial and take the remainder.
- Treat data bits from circuit POs as a decreasing order coefficient polynomial.
- The circuit divides the PO polynomial by its characteristic polynomial.
  - Leaves remainder of division in LFSR.
  - Must initialize LFSR to seed value (usually 0) before testing.
- After testing – compare signature in LFSR to *golden signature*.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

First let us talk about signature analyser. So, how it works, we can use an LFSR again here Linear Feedback Shift Register again to use as a response compactor, the same technique is used for error detection using cyclic redundancy code check. The basic mathematical foundation is that we are dividing a polynomial by another polynomial and take the remainder this remainder is what we are calling as the signature mathematically this is the basic concept, but how it is implemented let us see. You see this circuit output is generating some data we are applying input patterns one by one and the output bits are generated.

Now, these bits are treated as coefficients of a polynomial in decreasing order we will take an example and we have a compaction circuit based on LFSR. What the compaction circuit will be doing it will be dividing this polynomial representing the output bit stream by the characteristic polynomial of the LFSR. So, LFSR we have seen that it there is a characteristic polynomial. So, polynomial division will take place and whatever remainder is will be there in the LFSR that will be the signature and one thing remember before testing starts we must initialise the LFSR to the all 0 pattern fine. Let us see how it works.

(Refer Slide Time: 09:37)



Characteristic Polynomial :: $P(X) = X^5 + X^3 + X + 1$

Input Polynomial :: $F(X) = X^7 + X^3 + X$

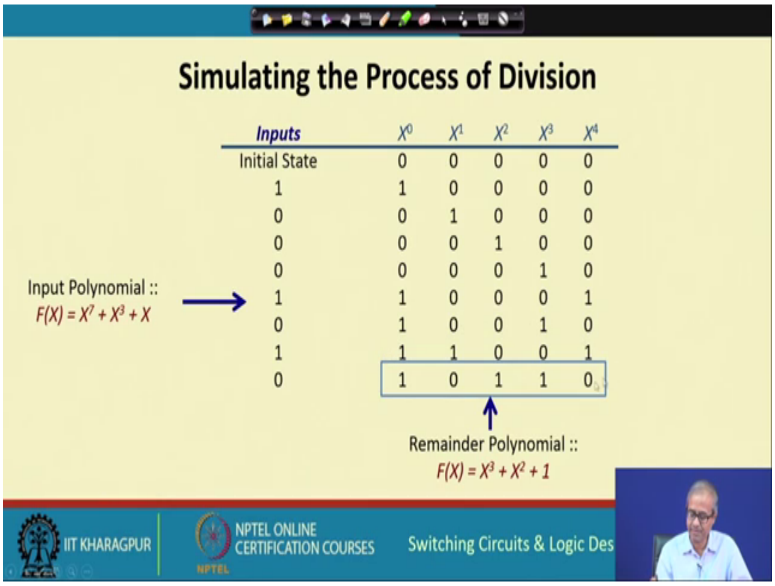
Divide $F(X)$ by $P(X)$ and take the remainder → $X^3 + X^2 + 1$

Let us take an example this is a 5 bit LFSR which implements a polynomial like this X to the power 5 X 3 X and 1. So, the feedback are coming like this X to the power 5 X 4 X then X X power X cube there is a feedback from X cube then X then 1. Well you can actually call them in a different in terms of the polynomial these are the outputs, but in terms of the coefficient of the polynomial you can call it x to the power 5 you can call this x to the power 4, you can call this x to the power 3 x square x and this is x to the power 0 or 1.

So, feedback is taken from x to the power 5 x to the power 3 and an x you see these terms are there and we have used the type two LFSR, where the XOR gates are in between the D flip flops right and the circuit is generating an input bit stream say 0 1 0 1 0 0 0 1 ok. So, the input polynomial you see if you consider the output side 0 1 2 3 4 5 6 7. So, this will be X to the power 7 7 6 5 4 3 X to the power 3 2 1, that is why X to the power 7 X 3 plus X. So, any bit pattern can be represented by a polynomial. So, input polynomial is this, your characteristic polynomial of the LFSR is this.

So, I am saying that there will be a division of this polynomial by the characteristic polynomial, this will happen automatically and here if you just work it out take the remainder. Remainder comes to X cube plus X square plus 1 which in terms of bit pattern X cube X square no X and 1 1 1 0 1 this should be the remainder right.

(Refer Slide Time: 12:05)



Simulating the Process of Division

Inputs	X^0	X^1	X^2	X^3	X^4
Initial State	0	0	0	0	0
1	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
1	1	0	0	0	1
0	1	0	0	1	0
1	1	1	0	0	1
0	1	0	1	1	0

Input Polynomial :: $F(X) = X^7 + X^3 + X$

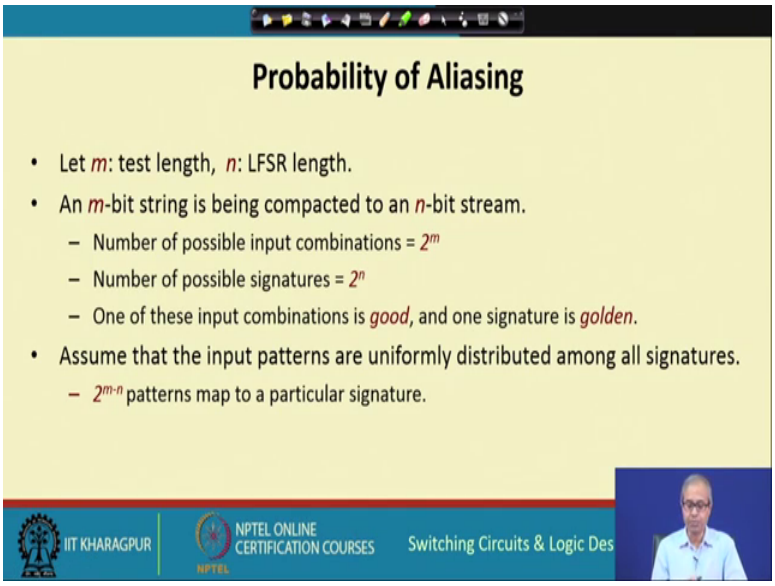
Remainder Polynomial :: $F(X) = X^3 + X^2 + 1$

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Switching Circuits & Logic Des

Now, if we just work out just initialise it to the all 0 pattern simulating the process of division the input bits are coming like this. And, if you work out I leave it an exercise for you leave it you just see it step by step this corresponds to the input polynomial and as this input bits are coming one by one the LFSR runs starting from the all 0 state you will see that the end whatever remains this is nothing, but this 1 1 0 1 you see this is this polynomial I said X cube plus X square plus 1.

Final value in the LFSR whatever it remains this will be the remainder polynomial; this is how polynomial division takes place or works ok. So, you need not have to know much more detail about this just remember that it works.

(Refer Slide Time: 12:58)



Probability of Aliasing

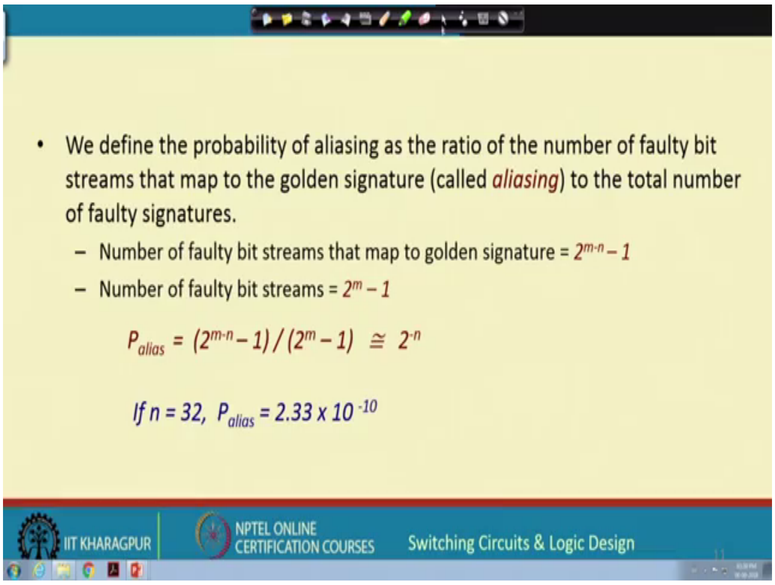
- Let *m*: test length, *n*: LFSR length.
- An *m*-bit string is being compacted to an *n*-bit stream.
  - Number of possible input combinations = 2^m
  - Number of possible signatures = 2^n
  - One of these input combinations is *good*, and one signature is *golden*.
- Assume that the input patterns are uniformly distributed among all signatures.
  - 2^{m-n} patterns map to a particular signature.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Switching Circuits & Logic Des

Talking about the probability of aliasing that what is the chance that error will some error will occur, but will not be able to test it Let us see. Suppose the number of bits in the input stream is m and LFSR is an n bit LFSR. So, in the input we have an m bit stream which naturally will be having 2 to the m combinations and finally, in the LFSR we are compressing it to an n bit stream. So, there can be 2 to the power n possible signatures and out of these 2 to the power n one of the signature will be good and out of this 2 to the you see there is 2 to the power m input combinations in m bit.

But out of them one of them is good, similarly in the output signature there can be 2 to the power n possible signature out of them one of them is good. So, there are 2 to the power m minus 1 faulty input streams and 2 to the power n minus 1 faulty signatures ok. Now assuming that the input pattern this is a much larger set are uniformly distributed among the signature. So, how many input patterns will map to every each signature? 2 to the power m divided by 2 to the power n which comes to 2 to the power m minus n, 2 to the power m minus n patterns map to a particular signature in the LFSR just remember this.

(Refer Slide Time: 14:51)



- We define the probability of aliasing as the ratio of the number of faulty bit streams that map to the golden signature (called *aliasing*) to the total number of faulty signatures.
  - Number of faulty bit streams that map to golden signature = $2^{m-n} - 1$
  - Number of faulty bit streams = $2^m - 1$

$$P_{alias} = (2^{m-n} - 1) / (2^m - 1) \cong 2^{-n}$$

$$If\ n = 32,\ P_{alias} = 2.33 \times 10^{-10}$$

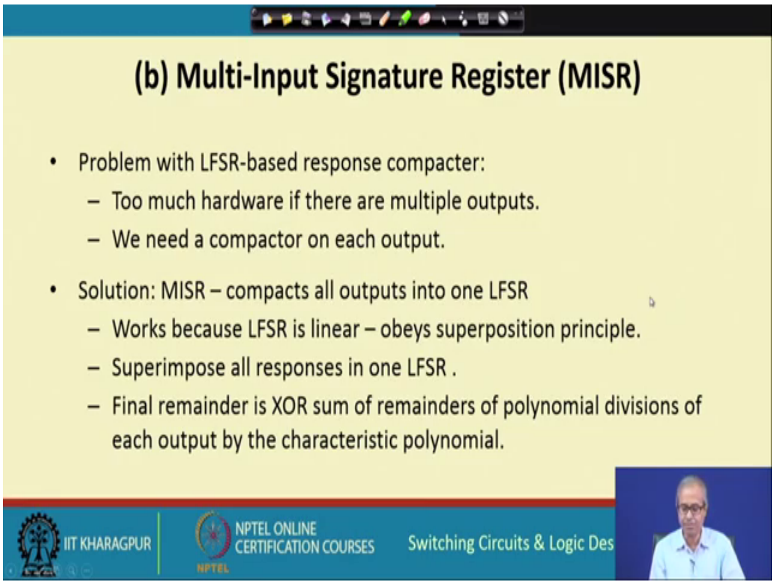
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Design

So, probability of aliasing I mentioned it is defined as the ratio of the number of faulty bit streams it is a probability. So, number of faulty bit streams that map to the golden signature to the number of total faulty signature.

Now, we have said already seen that 2 to the power m minus n input patterns can map to every signature and one of them will be the golden signature. So, 2 to the power m minus n input patterns will map to the golden signature and out of these 2 to the power m minus n one of them will be the good one. So, 2 to the power m minus n minus 1 will be faulty bit streams that map to the golden signature and these will lead to aliasing. So, probability of aliasing will be this number of cases divided by total number of cases, total number of faulty bits 2 to the power m out of them one is good. 2 to the power m minus 1 so, if m is very large.

So, this approximates to 1 by 2 to the power n and you see even if you choose n equal to 32 this is independent of m 1 by 2 to the power 32 is so small the probability something into 10 to the power minus 10. So, well yes there can be aliasing, but the probability of aliasing is very very small because, of this people do use this technique and a signature of size 32 or 64 is considered to be good enough alright ok.

(Refer Slide Time: 16:44)



(b) Multi-Input Signature Register (MISR)

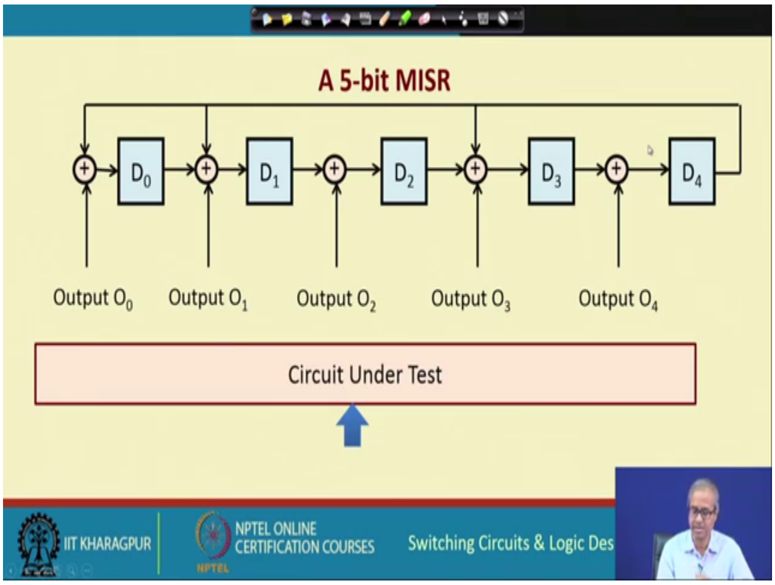
- Problem with LFSR-based response compacter:
  - Too much hardware if there are multiple outputs.
  - We need a compactor on each output.
- Solution: MISR – compacts all outputs into one LFSR
  - Works because LFSR is linear – obeys superposition principle.
  - Superimpose all responses in one LFSR .
  - Final remainder is XOR sum of remainders of polynomial divisions of each output by the characteristic polynomial.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

Now, talking about multiple input signature register let us briefly talk about it without going into the detail because, a circuit in general can have multiple outputs ok. So, when you apply a sequence of test patterns bits will be generated from all of these outputs in a stream right.

So, suppose if you use an LFSR based compressor with every output bit then this will require lot of too much hardware lot of hardware, if there are 100 outputs we will be needing 100 LFSR’s ok. The solution is to use multiple input signature register is called MISR in short, where you are using a single LFSR how I will show you. So, you need not have to remember all these details because, LFSR is linear you can compress all the output bit streams in a single LFSR the final response will remain something like same; as if you XOR these responses and the final response you will get they will be same ok. Let us see how it works? How the circuit will look like?

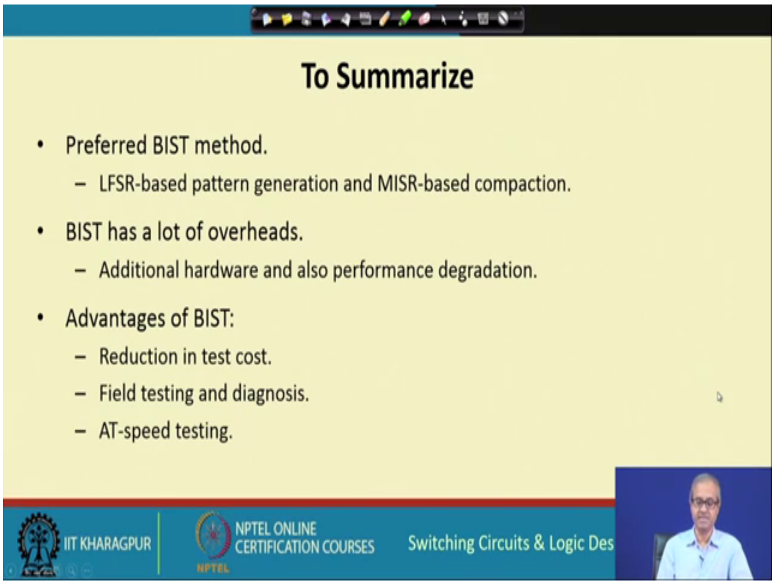
(Refer Slide Time: 18:01)



A 5-bit MISR

Well a MISR looks like this, you see this is a type two LFSR where not only I am feeding data in the first XOR, but I am adding an additional input to all other XORs. So, even in the places where there is no feedback or taping connection I am adding an XOR gate. So, that for the circuit so, all the outputs can be fed in parallel to all these XOR gate inputs. So, all the bit patterns suppose out here I am applying 10 to the power 6 patterns 1 million patterns I am applying. So, each of the circuit outputs should be generating 1 million bit streams, they will all be compacted together and finally, a single signature will remain stored in the register at the end right. This is how this circuit works.

(Refer Slide Time: 19:07)



To Summarize

- Preferred BIST method.
  - LFSR-based pattern generation and MISR-based compaction.
- BIST has a lot of overheads.
  - Additional hardware and also performance degradation.
- Advantages of BIST:
  - Reduction in test cost.
  - Field testing and diagnosis.
  - AT-speed testing.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Switching Circuits & Logic Des

So, to summarise here if you talk about built in self test the preferred method as I told is that you use pattern generation using LFSR based random patterns and on the output side you do compaction using multiple input signature register. But, you understand BIST has a lot of overheads you need to have this pattern generator, response compactor, the ROM to store the golden signature, the comparator this additional hardware also has to go inside the chip ok. These are the additional overheads and they also lead to performance degradation because now, the primary inputs that your feeding to the circuit that has to go through a multiplexer. The multiplexer will be having some delay so, the delay of your circuit increases. So, some degradation in performance also occurs.

But the advantages of BIST's are pretty significant test cost gets reduced drastically, you are able to do field testing and diagnosis, which chip is good or bad and you can test the circuits at its full speed these are some of the very big advantages. So, with this we come to the end of this lecture and also this course. So, over the last 12 weeks we have covered many of the topics, most of the topics are conventional in the sense that you will find them as part of standard curriculum in most of the courses in switching circuits or digital circuits. But I have also tried to cover a few topics, which are a little unconventional, which are little of the tracks, which are normally not a part of the syllabus, but I just wanted to give you a flavour of some of the techniques or technologies which have possibly greater impact that you can see in the future. So, with this and wishing you all the best let me say goodbye to you.

Thank you very much.