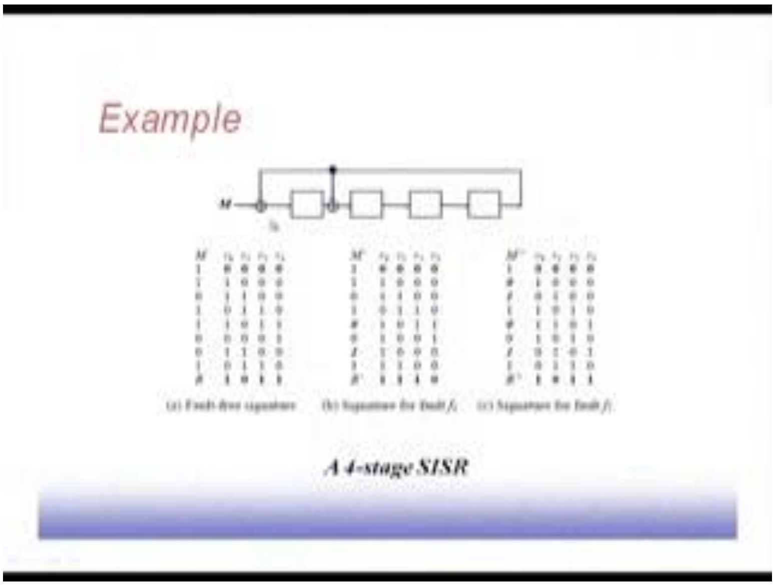
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 27
Logic BIST (Contd.)

So, for example, if we have got say this as the signature analyzer, now this M we are getting the input suppose the fault free signature is this one that is for the first pattern it generates 1, second pattern 1, third pattern 0, so like that it generates the thing.

(Refer Slide Time: 00:24)



Example

M	r_0	r_1	r_2	r_3
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
1	0	1	1	0
1	1	0	1	1
0	0	0	0	1
0	1	1	0	0
1	0	1	1	0
R	1	0	1	1

(a) Fault-free signature

M'	r_0	r_1	r_2	r_3
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
1	0	1	1	0
0	1	0	1	1
0	1	0	0	1
1	1	0	0	0
1	1	1	0	0
R'	1	1	1	0

(b) Signature for fault f_1

M''	r_0	r_1	r_2	r_3
1	0	0	0	0
0	1	0	0	0
1	0	1	0	0
1	1	0	1	0
0	1	1	0	1
0	1	0	1	0
1	0	1	0	1
1	0	1	1	0
R''	1	0	1	1

(c) Signature for fault f_2

A 4-stage SISR

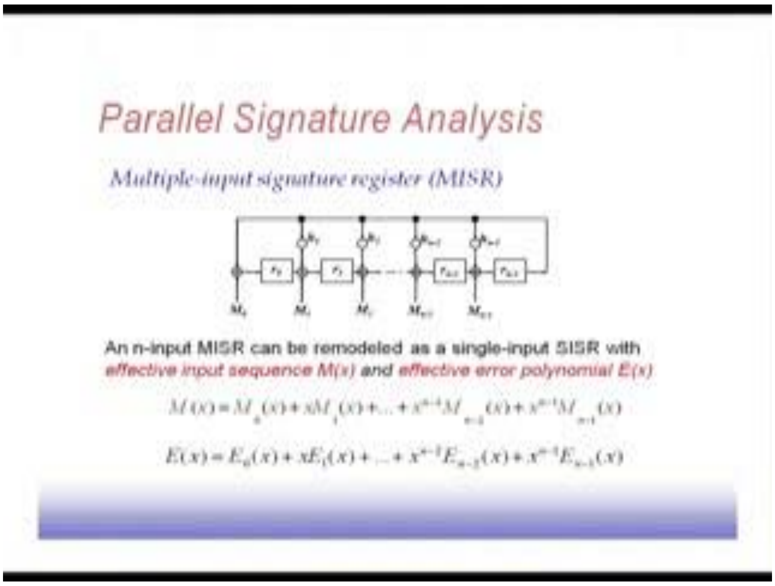
Now initially all these sales are having 0 they are the set and these if you follow these operation then the after the first cycle this 1 is XORed with 0. So, this 1 comes to the cell r 0 and these contents are gets shifted. Then in the second cycle again this is 1, this 1 will XORed with again 0. So, 1 comes here and 1 goes there. So, that way at the end of all the test pattern application. So, we have got this response as 1 0 1 1. So, this is the fault free signature or called also called the golden signature.

Now, if there is a fault like say there is a fault f 1 such that for this particular pattern the response instead of being 1 it is 0, then this, then if we follow the operation then at the end we find that the response the signature becomes 1 1 1 0 which is different from 1 0 1 1. So, if we do a fault simulation before running this section if we do a fault simulation we can check that we can identify that situation that if this is 0, let us say due to fault f 1.

So, this output has become 0 and this output has become 1, so these 2 are the changes. The rest of the patterns they did not have any effect on this particular fault, but this 2 patterns they their responses got changed.

So, ultimate response is 1 1 1 0. So, so fault simulation we may know that if the response is 1 1 1 0; that means, if is fault f 1. Similarly if the fault f 2 has occurred maybe this response and this response, all these have changed. So, this pattern, this pattern, this and this all these test patterns the responses are different. So, in that case we can say that there this signature is 1 0 1 1. So, in this case you see that this has aliased with the fault free response. So, fault f 2 could not be detected by this particular signature analyzer. So, this is the aliasing that has occurred fault free response and faulty response they are becoming same.

(Refer Slide Time: 02:55)



Parallel Signature Analysis

Multiple-input signature register (MISR)

An n-input MISR can be remodeled as a single-input SISR with *effective input sequence M(x)* and *effective error polynomial E(x)*

$$M(x) = M_0(x) + xM_1(x) + \ldots + x^{n-2}M_{n-2}(x) + x^{n-1}M_{n-1}(x)$$

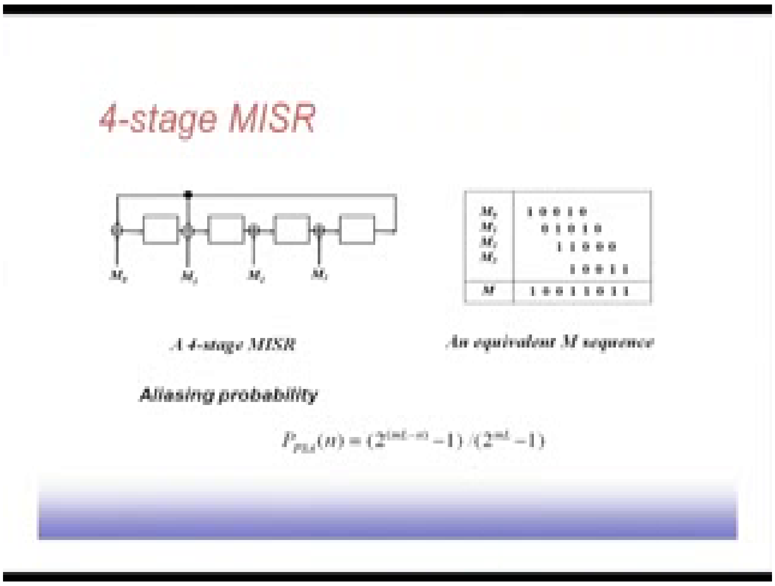
$$E(x) = E_0(x) + xE_1(x) + \ldots + x^{n-2}E_{n-2}(x) + x^{n-1}E_{n-1}(x)$$

Another possibility of this signature analysis is the multiple input signature register the problem with single input signature register is that we have got only 1 input to the signature register. So, if a circuit has got multiple outputs then we need some kind of parallel to serial converter to be to be used before that. So, that we can send all those responses serially and it is very difficult also because after applying 1 test pattern you have to shift in all those outputs into that chain and all that other option is that out of the say M outputs of the system you select 1 output and only use that output for signature analysis. So, in that case also we will miss many cases, many interesting situations that may occur.

So, other possibility is to have a multiple input signature register MISR. So, here for each of this, each XOR gates we have got another input here. So, in the previous case this individual cells we did not have the XORs. So, now, we have got individual cells as XORs. Now, this M 0, M 1, M 2, they are fed as patterns to this individual XOR cells.

Now an n input MISR it can be remodeled as a single input SISR with effective inputs sequence M x and effective error polynomial E x. So, this a theoretical result, we do not have time to go into that, but the effective input sequence M x is M 0 x plus x M 1 x plus x n minus 2 M minus 2 x like that and the error polynomial is this one. So, if this, if there is some error then this error polynomial will tell us what will be the signature at the output. So, again it follows that polynomial division philosophy and the remainder is the error polynomial.

(Refer Slide Time: 04:52)



4-stage MISR

A 4-stage MISR

M_0	1	0	0	1	0			
M_1		0	1	0	1	0		
M_2			1	1	0	0	0	
M_3				1	0	0	1	1
M	1	0	0	1	1	0	1	1

An equivalent M sequence

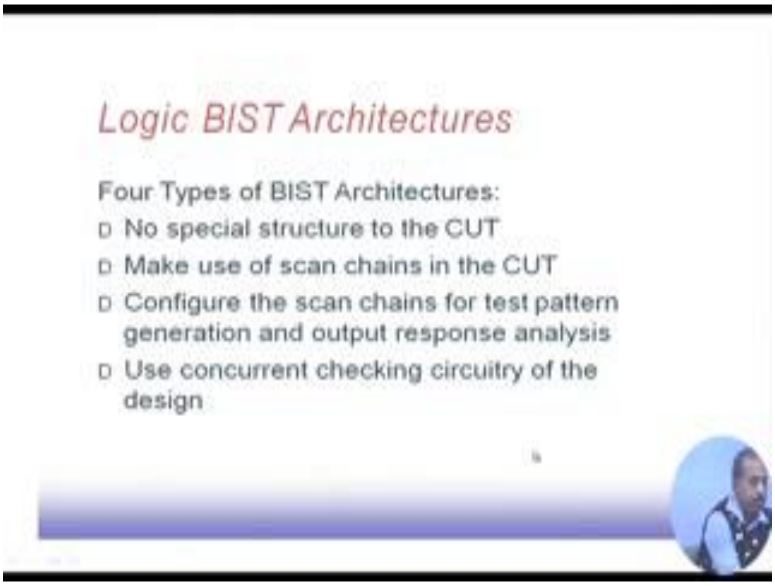
Aliasing probability

$$P_{PSA}(n) = (2^{(mL-n)} - 1)/(2^{mL} - 1)$$

Coming to the example, this is that four stage MISR we have. So, M 0, M 1, M 2 and M 3, they are fed here. So, M 0 is first this, this M 0 is coming here then after 1 cycle. So, this is this is a shifted version, we can also consider as if this M 0, M 1, M 2, M 3, they are getting shifted because this is 0. So, when this first line of M 1 is applied so that will not have any effect. So, that this will the M 1 will remain unaltered. So, that way it will go. So, this four stage MISR when we apply this M 0, M 1, M 2, M 3. So, M 0, M 1, M

2, M 3 are like this they are applied in a time shifted fashion then this is the resulted polynomial M. So, the previously that M x that we were talking about, these becomes the equivalent M sequence and aliasing probability is given by this expression.

(Refer Slide Time: 05:51)



Logic BIST Architectures

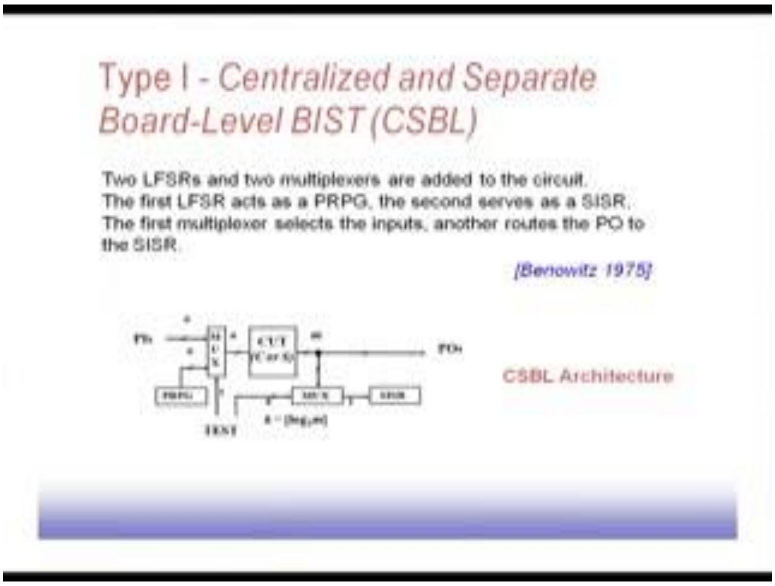
Four Types of BIST Architectures:

- No special structure to the CUT
- Make use of scan chains in the CUT
- Configure the scan chains for test pattern generation and output response analysis
- Use concurrent checking circuitry of the design

Now, that is about this test pattern generator and the response analyzer. Now there are several logic BIST architectures, there can be four types of logic BIST architectures, no special structures to the, in one case we have we do not have any special structure to the circuit. We use the there are scan chains in the circuits. So, we try to use those scan chains, we can configure the scan chains for test pattern generation and output response analysis because these test pattern generator, it is nothing, but a set of flip flops and XOR gates. So, if we can modify the scan chain itself so that by in some test pattern generation mode. So, it will be acting the XORs will get activated. So, that way it can act as the test pattern generator.

And the response analysis here also there is some summation. So, that is also acting as the some XOR gates there in the circuits, that is getting inserted. So, if we can modify the circuit so that we can modify the scan chain. So, that when required they can act as pattern generator or response analyzer. So, that can be done. So, if it is done also this is going to be useful. Or we can use concurrent checking circuitry of the design. So, that can also be done.

(Refer Slide Time: 07:12)



Type I - *Centralized and Separate Board-Level BIST (CSBL)*

Two LFSRs and two multiplexers are added to the circuit.
The first LFSR acts as a PRPG, the second serves as a SISR.
The first multiplexer selects the inputs, another routes the PO to the SISR.

[Benowitz 1975]

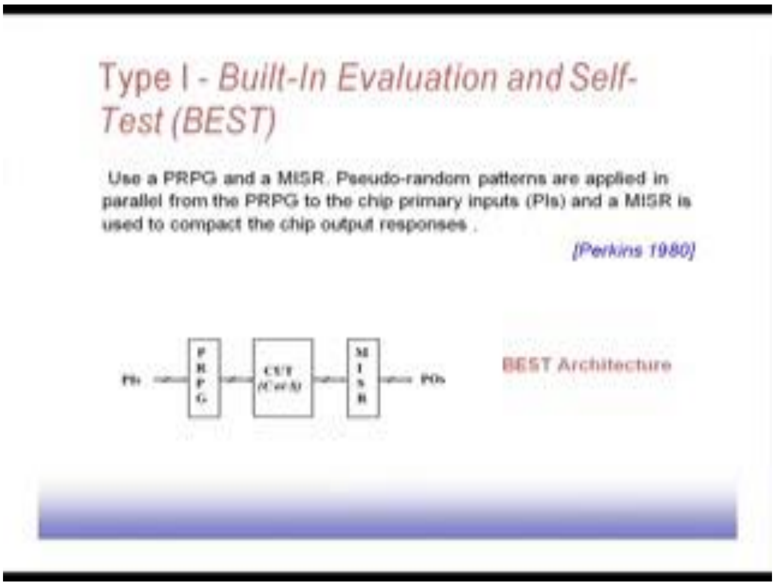
CSBL Architecture

So, the first one is the centralized and separate board level BIST or CSBL. Here 2 LFSRs and 2 multiplexers are added to the circuit, the first LFSR acts as pseudo-random pattern generator, second LFSR acts as the single input signature register.

The first multiplexer selects the input and another routes the primary output to the SISR. So, what is happening is that there is a pseudo-random pattern generator and there is one SISR. So, these 2 are added and then this primary inputs are there, but from the pseudo-random, in the test mode this multiplexer will select the primary input; this pseudo-random pattern that are generated to be applied to the circuit. And this k equal to log n, this basically if there are M lines going, there are M primary outputs going that are there then out of that, it will select which of them will go to the SISR.

So, basically this input the outputs will be fed serially to the SISR. So, after applying one pattern, after this pseudo-random pattern generator has generated 1 pattern and through this multiplexer it has been applied. So, there are M outputs, those M outputs will be selected serially and put into this SISR. So, this structure can be used as a for board level, so we can if there are circuits which are fed by the same PRPG. So, that also can be done, that also can be done by means of some other selection lines like the test signal given to the first module here. So, it can be given to some other module as well. So, we can select between the module to be tested or circuit to be tested. So, this is one type of board level BIST.

(Refer Slide Time: 09:17)



Type I - *Built-In Evaluation and Self-Test (BEST)*

Use a PRPG and a MISR. Pseudo-random patterns are applied in parallel from the PRPG to the chip primary inputs (PIs) and a MISR is used to compact the chip output responses .

[Perkins 1980]

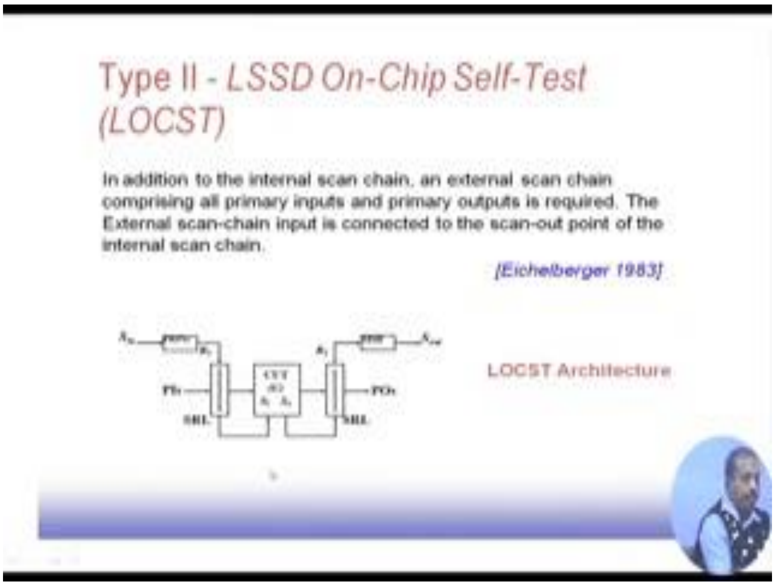
PIs
PRPG
CUT
(C or S)
MISR
POs

BEST Architecture

Another one is built in evaluation and self-test or BEST type of structure. So, here we use a pseudo-random pattern generator and a multiple input signature register then the pseudo-random patterns are applied in parallel from the pattern generator to the chip primary input. So, here it is very I should say some sort of very dedicated pattern generator and signature register that we have and then this is very comfortable because now if the circuit has got say n inputs then this PRPG is also n bit wide. So, this n bits are generated and fed parallely to the circuit and similarly this if the circuit has n outputs then n MISR is an n bit output. So, that will be fed to this thing.

So, this patterns the pseudo-random patterns are applied in parallel from the PRPG to the chip primary inputs and a MISR is used to compact the chip output responses. So, this output responses will be compacted by MISR. So, there may be some bypass mechanism in the circuit we do not want to test. So, in that case this primary input will be applied directly to the circuit or this outputs primary outputs will be available from this circuit, but in test mode. So, it will work like that.

(Refer Slide Time: 10:45)



Type II - *LSSD On-Chip Self-Test (LOCST)*

In addition to the internal scan chain, an external scan chain comprising all primary inputs and primary outputs is required. The External scan-chain input is connected to the scan-out point of the internal scan chain.

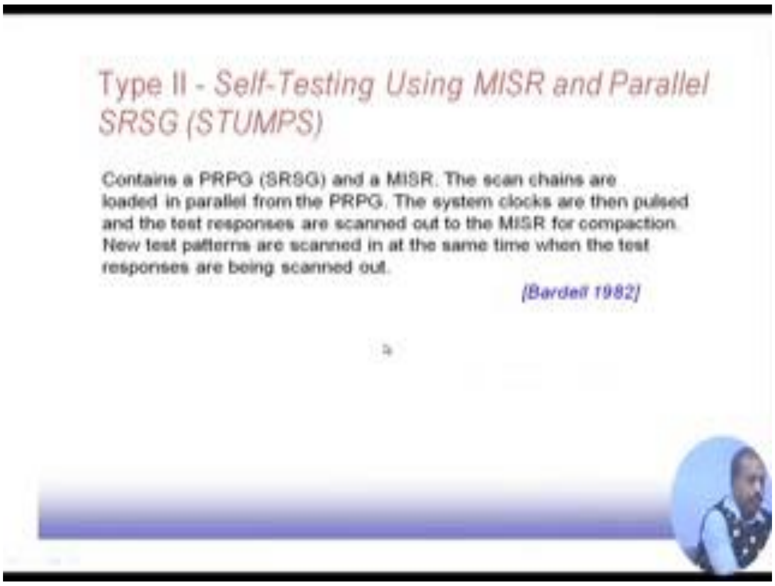
[Eichelberger 1983]

LOCST Architecture

Another approach that we can have is LSSD that is legal sensitive scan design that we have seen previously for on chip self test called LOCST. So, this what it does in addition to the internal scan chain an external scan chain compressing all primary inputs and primary outputs is required. So, this circuit it has got internal scan chain where S i is the scan input S o is the scan output. Apart from that we have got separate scan lines are there. So, this, what happens is from the pseudo-random pattern generator, this pattern is generated and then that goes into this shift register latch and this shift register latch. So, it can either pass this primary input to the circuit or it can serially shift the pattern that are coming through the r 1 line to these latches and then these values will be loaded into this scan chain as well.

So, the primary inputs and the scan chain inputs. So, they are fed from the PRPG the pseudo-random pattern generator and the response part, response part similarly the scan chain output and this primary output. So, they are all put into this serial latch then in the serial latch they are loaded and then they are going to SISR one after the other. So, this way this apart from this internal scan chain that we have externally 2 scan chains are connected one for the scanning part or this pseudo-random pattern scanning part and one for the pseudo this response, this scan chain output response for that is routed through this and coming to the SISR.

(Refer Slide Time: 12:39)



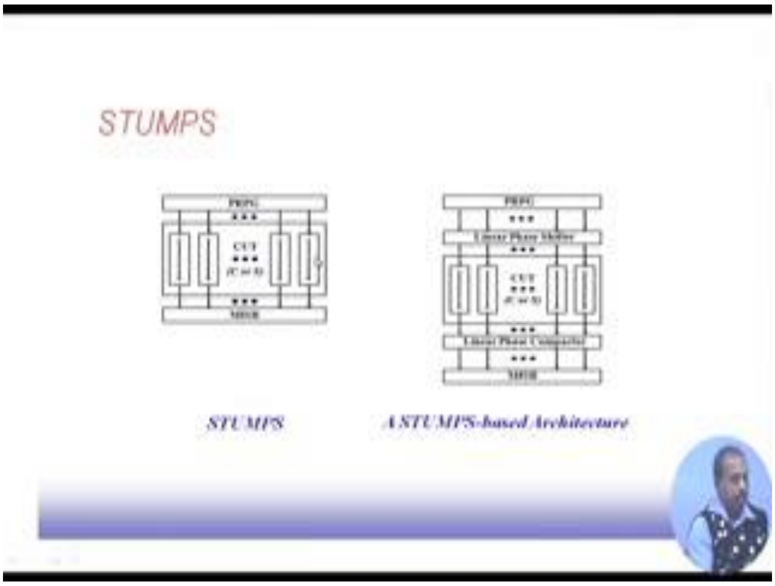
Type II - *Self-Testing Using MISR and Parallel SRSG (STUMPS)*

Contains a PRPG (SRSG) and a MISR. The scan chains are loaded in parallel from the PRPG. The system clocks are then pulsed and the test responses are scanned out to the MISR for compaction. New test patterns are scanned in at the same time when the test responses are being scanned out.

[Bardell 1982]

Then there is another very popular structure which is known as self testing using MISR and parallel shift register, SRSG. So, they are called that is called stumps. So, this contains a pseudo-random pattern generator and one MISR the scan chains are loaded in parallel from the pseudo-random pattern generator the system clocks are then pulsed and the test responses are scanned out to the MISR for compaction. So, let us see how this structure looks like.

(Refer Slide Time: 13:10)



STUMPS

STUMPS

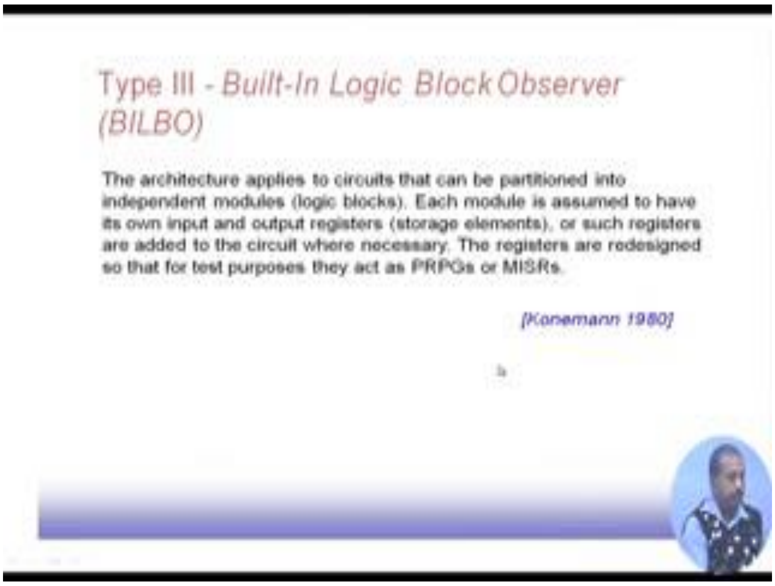
A STUMPS-based Architecture

So, this is the stumps structure, this circuit that we have. So, it has got a number or scan chains. So, these are the internal scan chains that I have.

Now, from the pseudo-random pattern generators, they are actually feeding this scan chains and this responses of are collected from the scan chain to the MISR. So, this is this is a typical stump architecture. Now it may so happen that this process requires. So, this number of scan chains is much larger than the PRPG the pseudo-random pattern generator. So, we cannot make it very large because that will increase the circuit area of power consumption and all that. So, what can be done is that we make this PRPG small, but then it goes through a linear phase shifter. So, this produces more number of shifted versions of these outputs.

Now, this and then they are actually used to feed the scan chains. So, if you have got large number of scan chains in the circuit then we can use this philosophy to feed the, we can use this philosophy to feed those without doing any, without having a large PRPG, Similarly at the response side as well we have got a linear phase compactor. So, what this linear phase compactor does is that it actually combines some of the outputs and then it is fed to the MISR for the signature analysis. So, for connections, these MISR is of smaller may be of smaller length and this linear phase compactor. So, it combines a number of outputs into one of them. So, that way this side maybe a more number of inputs may be coming to this phase shifter phase compactor and this output will be much less than that that will be (Refer Time: 15:04) put into the MISR.

(Refer Slide Time: 15:07)



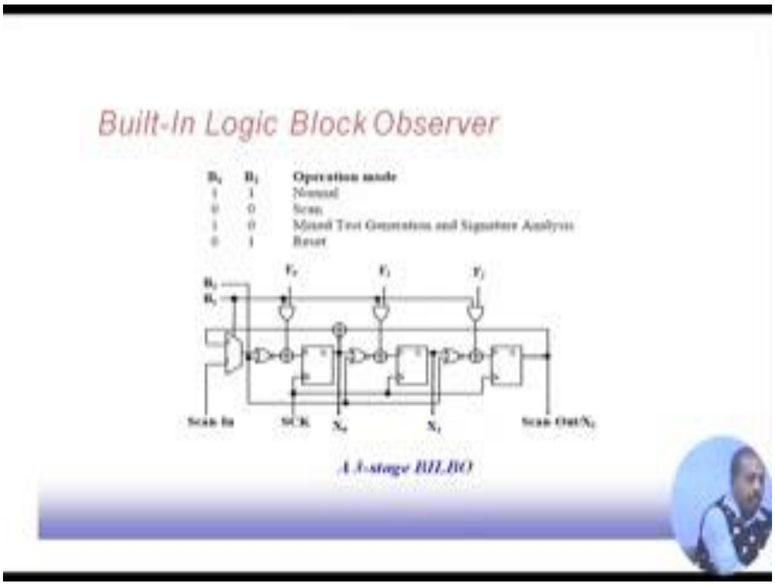
Type III - *Built-In Logic Block Observer (BILBO)*

The architecture applies to circuits that can be partitioned into independent modules (logic blocks). Each module is assumed to have its own input and output registers (storage elements), or such registers are added to the circuit where necessary. The registers are redesigned so that for test purposes they act as PRPGs or MISRs.

[Konemann 1980]

Then, there is another structure which is known as built in logic block observer or BILBO. So, BILBO is a circuit that combines all these operations into one. Now this architecture applies to circuits that can be partitioned into independent modules or logic blocks each module is assumed to have its own input and output registers or storage elements or such registers are added to the circuit where necessary. The registers are redesigned for, so that for test purposes they act as PRPGs or MISRs. So, this is very important. So, this extra registers that we are adding they themselves will act as pattern generator or signature register.

(Refer Slide Time: 15:48)



Built-In Logic Block Observer

B_1	B_2	Operation mode
1	1	Normal
0	0	Scan
1	0	Mixed Test Generation and Signature Analysis
0	1	Reset

A 3-stage BILBO

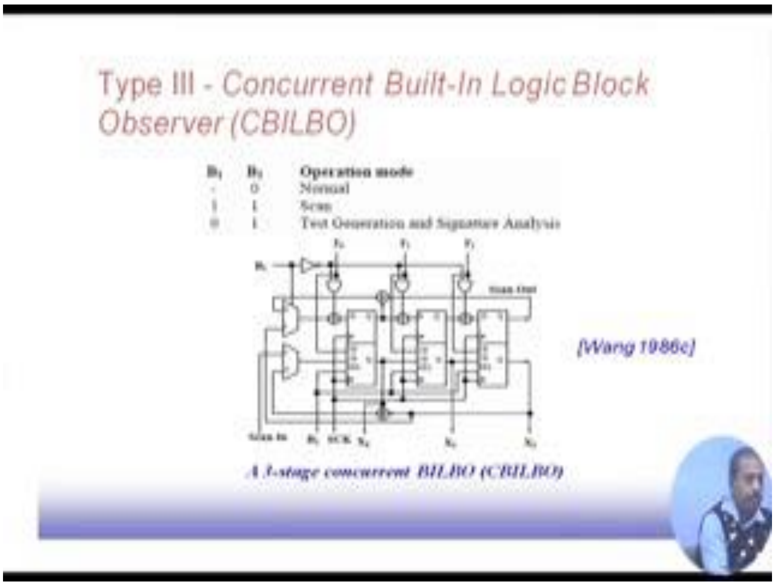
So, this is a typical BILBO structure you see that is has got several modes. So, there are 2 control signals B 1 and B 2 and this functional data when B 1 equal to 1 and B 2 equal to 1 what is happening is that Y 0 value is getting loaded into this clock. Similarly Y 1 value is getting loaded into this flip flop, Y 2 value is getting loaded into this flip flop. So, that way for both of them being equal to 1, this circuit operates in the normal mode Y 0, Y 1, Y 2 loaded into this latches.

Next, if we have this both of them as 0, B 0 B 1 both of them are 0. So, if both of them are then this will be selecting this line. So, this line will be selecting this scan out or X 2 line. So, that will be coming here and this will come here then since both are 0. So, this will be, if this whatever is the whatever is coming on this line. So, it will be appearing at this point and at the output of this NOR gate this that line the inverted version of that will come

So, inverted version of this 0 input is coming here and this part, since this is also this input is 0. So, whatever you are getting here will be coming into this point. So, that way the value that we are going to store that we are going to shift in, so that will be the scanning. So, that will be coming into this picture. Now after that in the next cycle, if we have got, so 1 0 so that is a mixed mode mixed test generation and signature analysis. So, in this case this signature analysis, this signature analysis will this mixed test generation and signature analysis. So, this will be the circuit will be operating in the test generation and the signature analysis mode. So, when this is 1 and this is 0. So, this B 1 equal to 1, B 1 equal to 1 will select this whatever this scanning input is coming. So, it will be loaded into this d flip flop and then this B 2 value, this B 1 value being equal to 1 so this value will be XOR. So, whatever is coming here in Y 0. So, that will be XORed with these value. So, as a result this will be acting as signature register now.

And also it works as test generator like this at this point X 0, X 1 and this X 2 they will act as the test pattern that is getting generated then they will be since this is shifting is there, so it will also act as shift register. So, I suggest that you go through this different modes B 0, B 1, B 2 settings and you can find that for all these case different modes. So, it is doing this operations as required. So, for 1 1 it is normal mode for 0 0, it is scan mode 1 0 is test generation and signature analysis. So, this is generally the BIST mode you can say and 0 1 is the reset mode where it is the content is getting reset.

(Refer Slide Time: 19:10)



Type III - *Concurrent Built-In Logic Block Observer (CBILBO)*

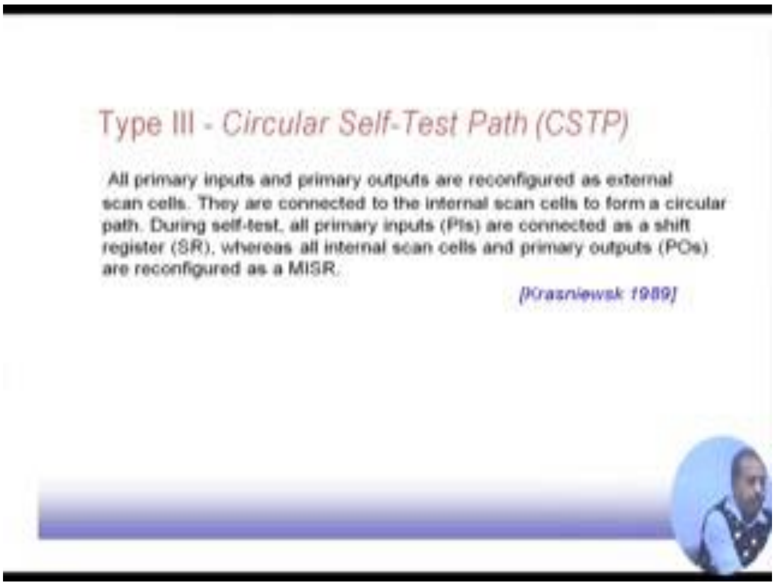
B_1	B_2	Operation mode
-	0	Normal
1	1	Scan
0	1	Test Generation and Signature Analysis

[Wang 1986c]

A 3-stage concurrent BILBO (CBILBO)

There is another structure which is known as CBILBO or concurrent BILBO. So, here actually this individual stages are there, but they can operate simultaneously. So, this, here we have got 3 modes like whenever B 2 is 0 so that is the normal mode of operation. So, B 2 means 0, this cells are same to, same as what we have seen in previously they will act as normal mode of operation when this is 1 1, it will act as scan mode and when it is 0 1 it will act as test generation signature analysis mode. So, this is otherwise it is similar to that BILBO structure, but it is concurrent BILBO because we have got 3 different BILBO modules and they can act for 3 different modules in the circuits. So, all of them are tested parallelly.

(Refer Slide Time: 20:03)

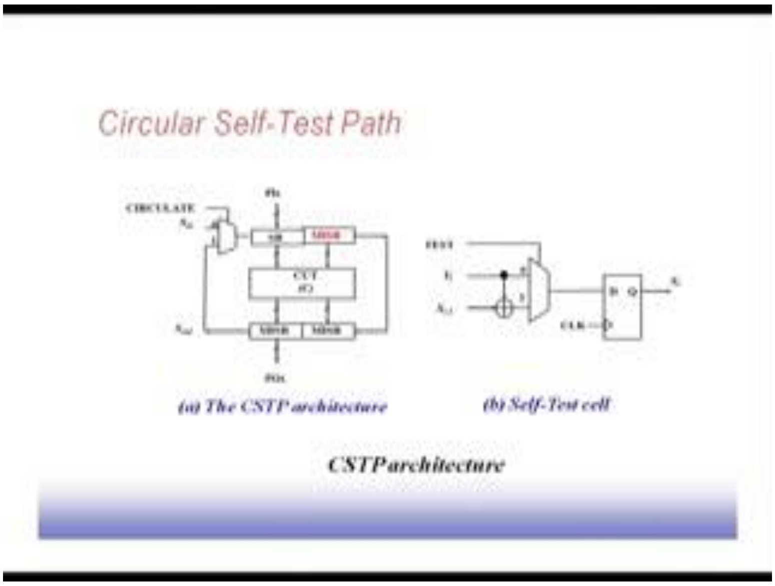


Type III - *Circular Self-Test Path (CSTP)*

All primary inputs and primary outputs are reconfigured as external scan cells. They are connected to the internal scan cells to form a circular path. During self-test, all primary inputs (PIs) are connected as a shift register (SR), whereas all internal scan cells and primary outputs (POs) are reconfigured as a MISR.

[Krasniewsk 1989]

(Refer Slide Time: 20:08)



Circular Self-Test Path

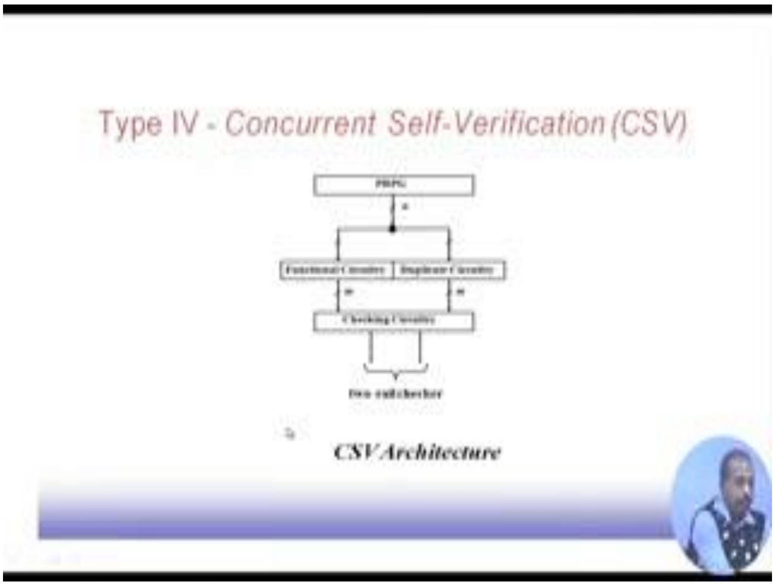
(a) The CSTP architecture

(b) Self-Test cell

CSTParchitecture

Then there is another structure which is circular self test path. So, here this is 1 shift register and MISR is there, this circuit under test. So, this pattern will be applied and this is going to be circulated in the sense that when the circulated input is 1 this MISRs response, MISRs response is again put into this shift register so that the same test pattern whatever response was generated so that is actually again fed as test pattern for the shift register. So, if it is circulate is 0. So, we can shift it in the pattern, new pattern can be shifted in and when this circulate bit is 1. So, it is basically the same the response that is generated by the circuit and MISR. So, that is fed as the pattern for the shift register. So, this way this self testing will work.

(Refer Slide Time: 20:58)

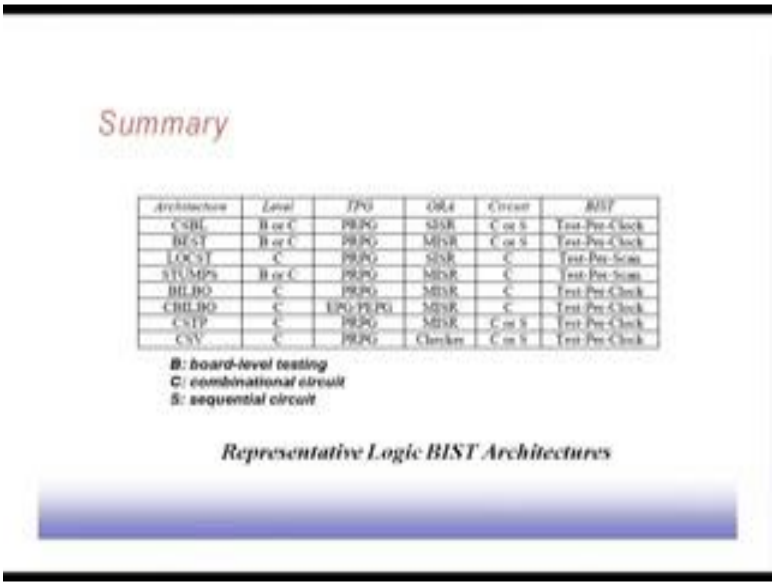


Type IV - Concurrent Self-Verification (CSV)

CSV Architecture

Then there are concurrent self verification approach, here this pseudo-random pattern generator that we have. So, we have got 2 circuits the functional circuit and the duplicate circuit and then we are have some checking circuit. So, basically it checks whether these 2 responses are correct or not. So, if these 2 responses are same then it will say it will check that whether if either of the circuit is faulty then this response will not be same. So, that way it is some sort of I should say confidence that circuit gets on its own operation whether it is working correctly or not. Because if the circuit is working correctly then both the responses will be correct if the response, if the checking circuitry is say is false then either of them is faulty. So, then you can say that the system is not working properly. So, we can say that a faulty situation has been detected. So, the advantage is that we do not have any MISR sort of thing. So, that way it is going to be helpful.

(Refer Slide Time: 22:06)



Summary

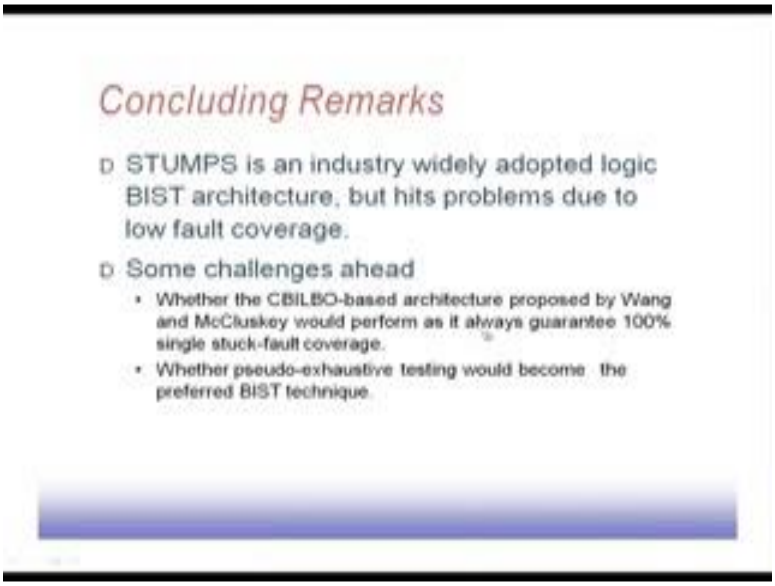
Architecture	Level	TPG	ORA	Circuit	BIST
CSBL	B or C	PRPG	SISR	C or S	Test-Per-Clock
BEST	B or C	PRPG	MISR	C or S	Test-Per-Clock
LOCST	C	PRPG	SISR	C	Test-Per-Scan
STUMPS	B or C	PRPG	MISR	C	Test-Per-Scan
BILBO	C	PRPG	MISR	C	Test-Per-Clock
CBILBO	C	EPG/PEPG	MISR	C	Test-Per-Clock
CSTP	C	PRPG	MISR	C or S	Test-Per-Clock
CSV	C	PRPG	Checker	C or S	Test-Per-Clock

B: board-level testing
C: combinational circuit
S: sequential circuit

Representative Logic BIST Architectures

So, to summarize we have got different types of structures, we discussed some them. So, out of all these structures that we have seen the stumps is 1 industry standard. So, that is accepted widely for this BIST architecture. But the problem are due to this low fault coverage.

(Refer Slide Time: 22:16)



Concluding Remarks

- STUMPS is an industry widely adopted logic BIST architecture, but hits problems due to low fault coverage.
- Some challenges ahead
  - Whether the CBILBO-based architecture proposed by Wang and McCluskey would perform as it always guarantee 100% single stuck-fault coverage.
  - Whether pseudo-exhaustive testing would become the preferred BIST technique

So, there is some proposal the CBILBO based architecture that would perform always, that would perform whether this will be 100 percent fault coverage or not. So, that is a question in the CBILBO it is not yet established and whether this pseudo exhausting testing would become the preferred BIST technique, instead of going for this pseudo-random testing should we go for pseudo exhaustive testing.