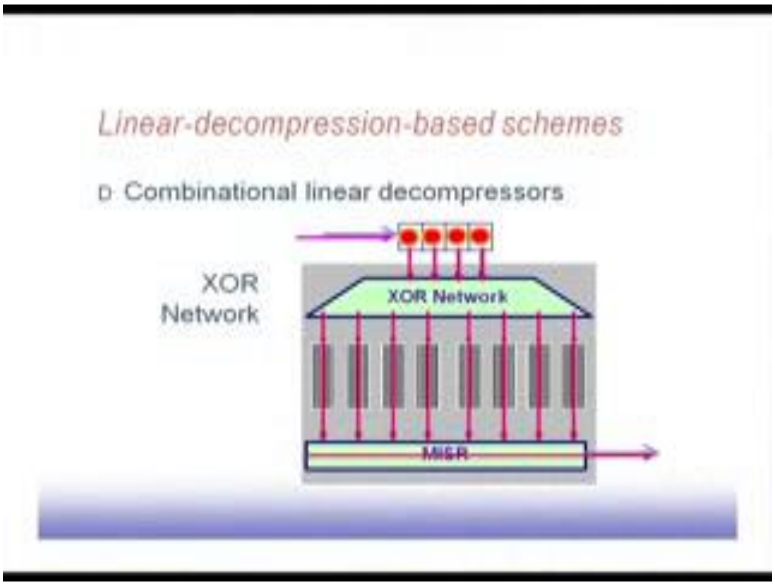
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 30
Test Compression (Contd.)

So, linear decompression based schemes; so, they actually use some linear network for a decompression purpose.

(Refer Slide Time: 00:21)

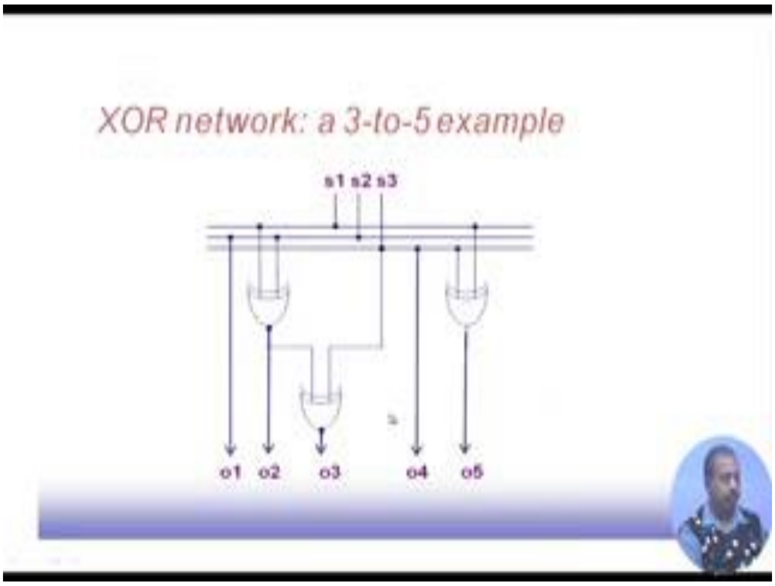


Linear-decompression-based schemes

- Combinational linear decompressors

So, from the ATE or from the, yes from the ATE, we get some coded pattern, and the coded pattern passes through a decompressor circuitry, to decompress into the base pattern that will be fed to different scan chains; like in this case. So, may be from the ATE we are getting a 4 bit in to a code, and that is passing through this XOR network and this XOR network is generating a actual pattern that will be fed to different scan chains. So, after the scan chains have been filled, then it will be applied, that the circuit will operating the normal mode, and the response will be captured, and they actually when this successive pattern are coming. So, their responses are getting added up in this is MISR, multiple input signature register, and then when all the inputs when this, and it is serially taken out. So, from the tester serially the base data is coming in, and the compacted response is going out. So, this type of situation gives rise to this linear decompression schemes.

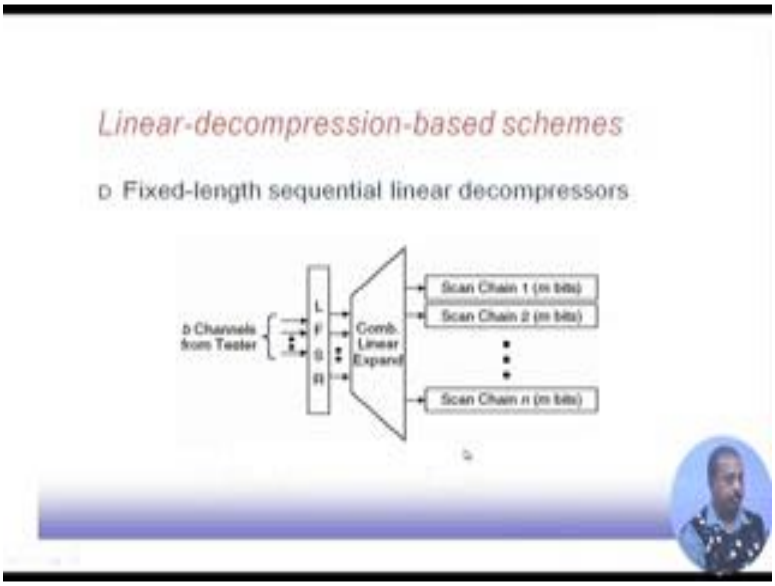
(Refer Slide Time: 01:36)



XOR network: a 3-to-5 example

Typically example may be like this, may be we have got 3 bits coming from ATE s 1, s 2 and s 3, and then we have got these de-compressor, made up of 3 XOR gate, and some of the lines are direct like o 1 then this o 4. So, these lines directly taken from say s 2, s 3, etcetera, and rest of the lines are generated by a combination of this. So, this is done actually by the means of that decompression, that Gaussian elimination process, to get a particular test pattern what should be the seat value. So, that is obtained, and then that seat value is paid from the ATE, and this de-compressor is actually calculating this response, this pattern that will be fed to the circuit for testing.

(Refer Slide Time: 02:26)



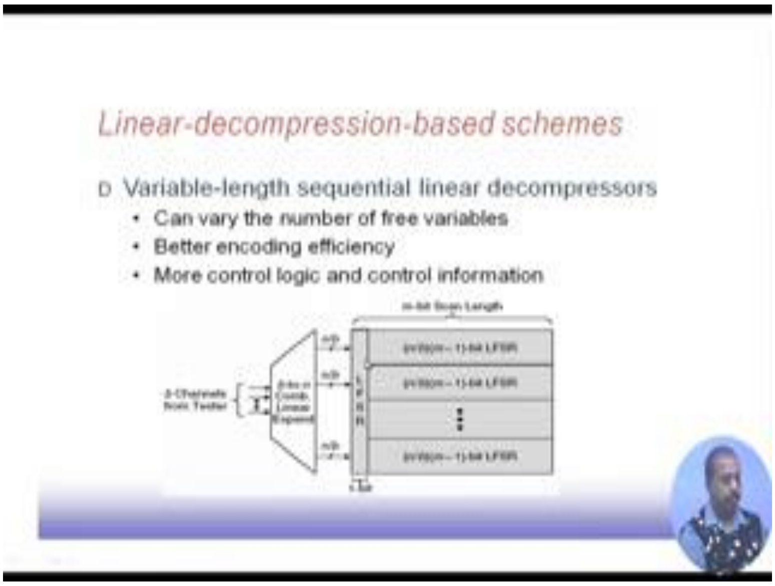
Linear-decompression-based schemes

- Fixed-length sequential linear decompressors

So, there can be fixed length sequential linear decompressors, here from the tester we get B channel, the B channel that feeds this LFSR, this LFSR generates some pattern and that pattern passes through a combinational linear expander. So, and then this linear expander, so this is a giving us N number of bits. So, each of this scan chains are N bit wise. So, different cycles this LFSR will generates pattern, and those pattern will be fed bit by bit in to this scan chains slides.

So, in the first clock cycle, the first slice will be loaded, then the next slice will be loaded. So, that way this scan chains are getting loaded. So, you see this LFSR is a B bit LFSR, because this parallel load sort of facility may be there, and then this combinational linear expander. So, this will be doing this expansion process, and then it will be put in to this circuit. So, may be from tester, we get a single pattern, and then it goes on during the thing that can also be done.

(Refer Slide Time: 03:35)



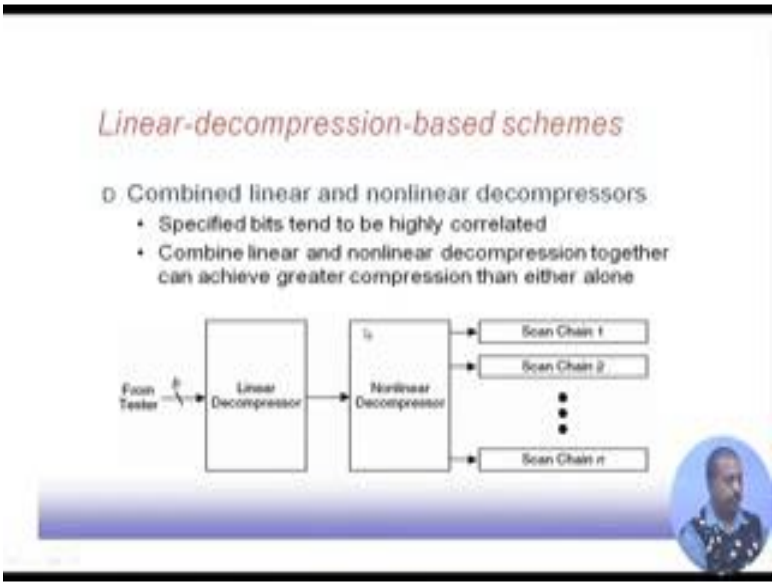
Linear-decompression-based schemes

- Variable-length sequential linear decompressors
  - Can vary the number of free variables
  - Better encoding efficiency
  - More control logic and control information

You can also have variable length sequential linear decompressors. So, here we can vary the number of free variables. So, basically what is happening is that, we have got B channel form the tester. Now there is a B to N combinational linear expander. So, after that these N bits are coming, out of these N bits so, it is divided into B number of channels. So, each of them you having N by B number of bits, and they are actually fed into this LFSR. So, this LFSR, it is 1 bit, 1 bit will be taken from the LFSR, and then its further feeding N by B to in to M minus 1 bit LFSR.

So, remaining N bit N minus 1 bit will take from this LFSR, in the 1 bit will be taken from this. So, that way different scan chains will be fed by doing this transfer. So, encoding efficiency its better, because the structure itself is quite complex, and you can have more control logic and control information that will be recovery; like how are we going to lose expansion then controlling this LFSR to generates the bits, and this extended LFSR to generate the extended pattern. So, that way the control becomes a bit complex.

(Refer Slide Time: 04:58)



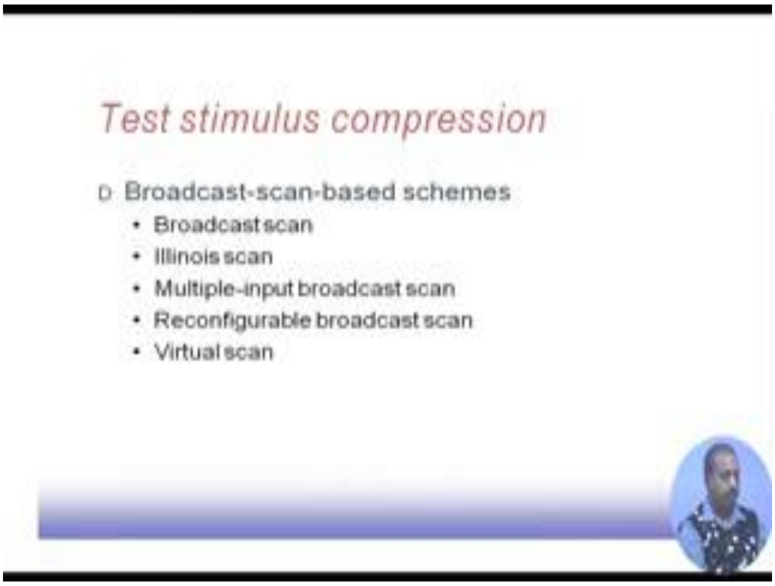
Linear-decompression-based schemes

- Combined linear and nonlinear decompressors
  - Specified bits tend to be highly correlated
  - Combine linear and nonlinear decompression together can achieve greater compression than either alone

There are combinations like linear and non-linear decompressions. So, linear decompression is that XOR gate or this, XOR network based strategy, or maybe you can augmentative, you can LFSR, and after that we have got this non-linear decompressions. So, non-linear decompression ah actually the dictionary based approaches that you have seen. Now if you combined these 2 together, then it can be give rise to further compression. So, the reason behind this is like this the specified bits tend be highly correlated. the reason is that, if the first test pattern has excited some fault, and second test pattern excite some fault in the, which is close to it then it is very much likely that due to this physical proximately of true or false, the many of the bits of the test pattern will be same. So, this specified bits of the test patterns they are going to be same.

So, if we are doing a linear decompression then so that may give rise to the, if you take say difference up 2 successive pattern and all that. So, that will give rise to 0, large number 0s. So, after this linear decompressing has been done. So, if you do a non-linear decompressing. So, may be by run length coding or may be dictionary based coding; like that. So, there is a chance that you will be achieving a further compression. So, this combined linear and non-linear decompression together, it can achieve greater compression, then either output.

(Refer Slide Time: 06:43)

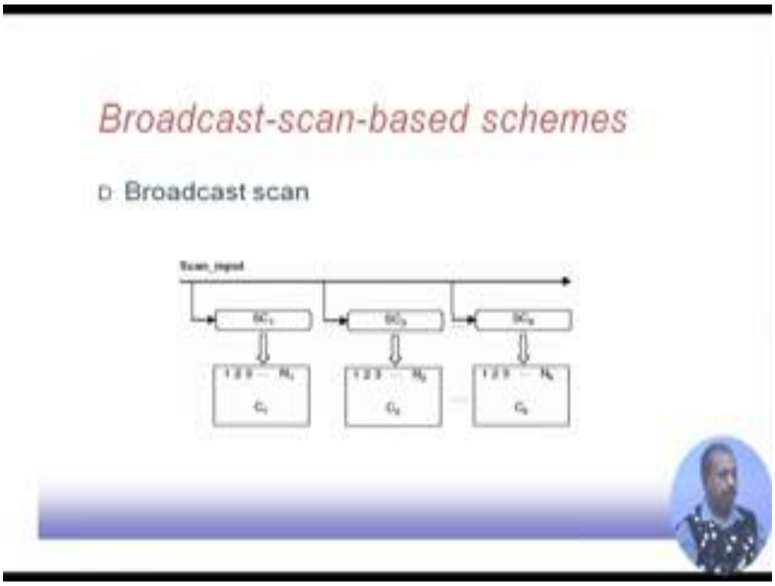


Test stimulus compression

- Broadcast-scan-based schemes
  - Broadcast scan
  - Illinois scan
  - Multiple-input broadcast scan
  - Reconfigurable broadcast scan
  - Virtual scan

So, that way it is going to be helpful. Next we look in to a test stimulus compression in that category; we will be looking into broadcast scan based scheme. So, broadcast scan based scheme means that there are multiple scan chain, and these multiples scan chain are fed simultaneously in a broadcast mode, and there are several such scan mechanism so we will look into some of them.

(Refer Slide Time: 07:06)



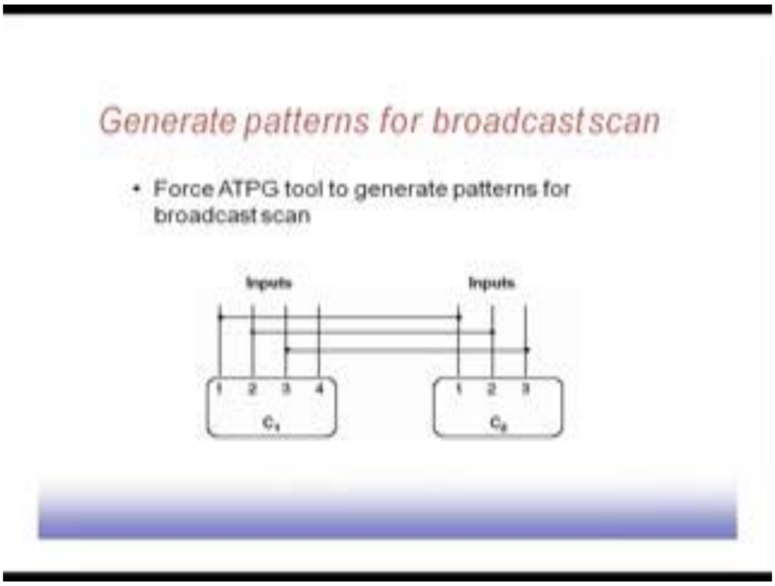
Broadcast-scan-based schemes

- Broadcast scan

So, this is the basic idea of broadcast scan chain. So, suppose we have got 3 different circuits C 1, C 2 and C 3. Now this C 1, C 2 and C 3 they have their corresponding scan chains SC 1 SC 2 and up to say SC 3.

So, if you have K number of circuit you have got SC 2. Now if the same scan input drives all of them, then the same pattern will be will be getting loaded into all this scan chain. Now it may so happened that the one test pattern which is good for C 1, it can detects some fault in C 1, it can also detects some fault in C 2, it can also detects some fault in C 2. So, the same pattern may be broadcasted to all the scan chain. So, that is the basic idea your broadcast scan, it is same pattern is loaded into the scan chains of all the circuit. So, in general the circuits may be different.

(Refer Slide Time: 08:08)



Generate patterns for broadcast scan

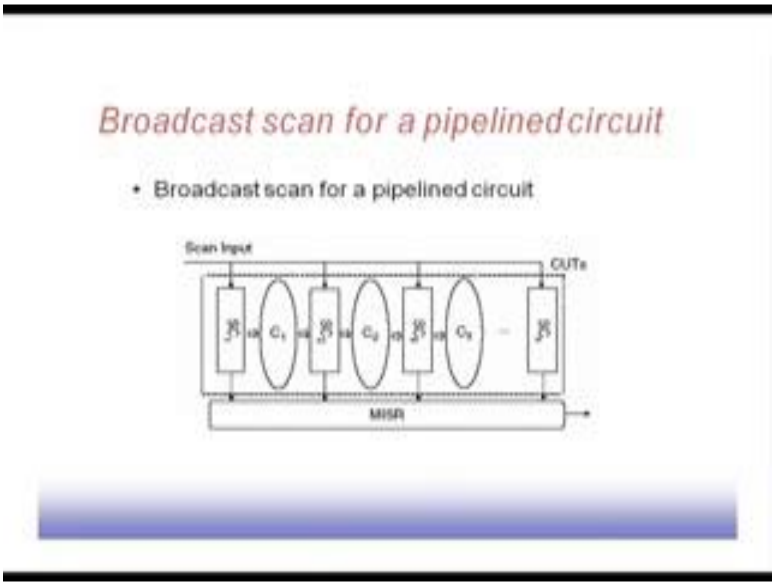
- Force ATPG tool to generate patterns for broadcast scan

But in reality what will happen is that, I have got only one circuit which may be partition and all that. So, how can we do this thing? How can we tell the ATPG to generate patterns which are same for both the circuit? So, what is done here is that this ATPG for the circuit that is GTATPG is a bit modified one, what like this C 1 and C 2, these are 2 circuits. So, if you give APTG separately the circuit C 1 and C 2. So, it will be generating pattern specific for C 1 and specific for C 2. So, it is very much possible that their patterns are not similar; however, if we modify the circuit like these 2 inputs are sorted. So, whatever you give here, it will also appear here, whatever you give here will also appear here. So, like that if you do this thing, then we can say that, if this modified circuit it is fed to the ATPG, then it will be it will generate pattern. So, this bits, bit number 1 or C 1, and bit number 1 C 2 they are going to be same and. So, that way if we modify the circuit and give it to the ATPG, so it can generate test patterns which are similar for C 1 and C 2. Now a in some cases; some ATPG tools, they allow you to do this do this operations

So, it is, even if you do not have 2 different circuit. So, you can constrain the ATPG to tell, that certain primary input will get the same values. So, that also serves our purpose. So, if you know that these are the various scan chains in my circuit, and if I want that scan chain 1 and scan chain 2, they should get, always get the same value, then we can we can tell the APTG that this primary input they should always have the same values.

So, instead of constructing another copy of the circuit as it is shown in the diagram. So, you can do it by telling the ATPG that these 2 bits must be same and things like that.

(Refer Slide Time: 10:07)



Broadcast scan for a pipelined circuit

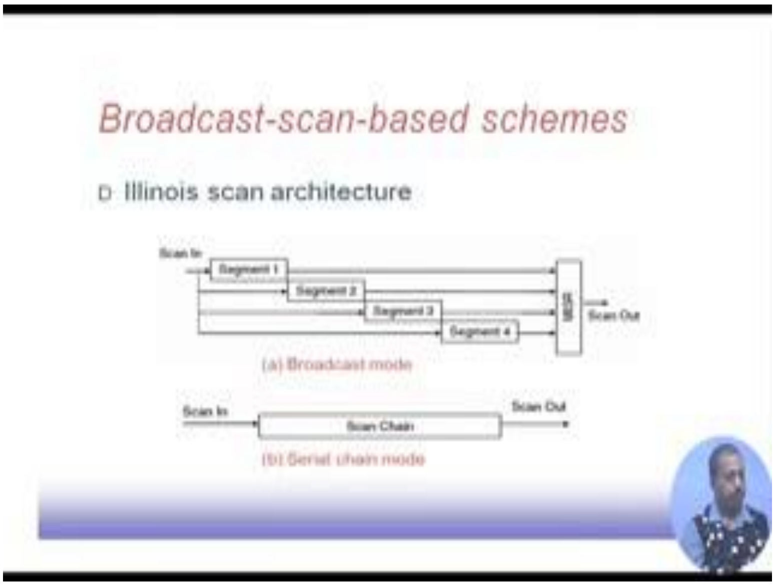
- Broadcast scan for a pipelined circuit

So, it can also be used for a pipeline circuit, like in a pipeline circuit what happens is that, we have got this pipeline stages C 1, C 2, C 3. So, there are pipeline stages, and in between the pipeline stages, we have got registers which hold the values of inputs and outputs of the pipeline stages. So, this SC 1, this is consisting of the register, this is actually the register with feeds values to C 1. Similarly output of C 1 is stored in SC 2 to this, this register SC 2, and that way it goes. Now for the scan mode of operation, these registers, they are converted into scan register by converting the normal flip flop into scan flip flop.

Now again the same thing now, I have got multiple circuit. Now it may so happen that called C 1, C 2, C 3 that test patterns are quite similar to each other. So, in that case, we can give the same scan input to all of them. So, this is scan input is coming to SC 1, SC 2, SC 3 and all of them. So, it is, if again that APTG tool it can be instructed that, the first sale of a SC 1, SC 2, SC 3 they should get the same value.

Now, the point is, that in this way it will be detecting many of the fault, it will be detecting many of the faults, but the response compactions becomes another issue. So, for the response compaction part, we can have one multiple input signature register. So, it is basically, it will be compacting all these responses into one signature, and it will be calculated here, and that will be for serially outputted from the circuit.

(Refer Slide Time: 11:59)



Broadcast-scan-based schemes

- Illinois scan architecture

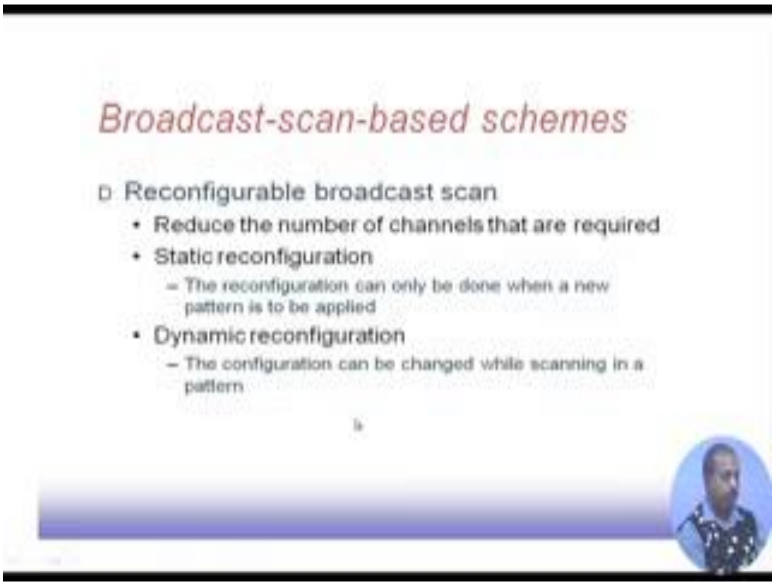
(a) Broadcast mode

(b) Serial chain mode

This way pipeline circuit also you can use a broadcast scan based approach. There is another architecture which is known as Illinois scan architecture. So, this has got the name, because the people they were from the University of Illinois. So, there we have. So, what happens is that some of the fault, that can be detected by having similar contented all the scan chains, whereas some other fault they will require some exclusive patterns. So, at the scan chains are to be loaded in a different way. So, what is the done is that, say this this scan chains segments; segment 1, 2, 3, 4. So, they may constitute 4 multiple scan chains, four parallel scan chains. So, what is done, the same scan in input is going to all of them. So, in the broadcast mode, the same scan in input, scan input will go to all the segment, they will get the same pattern. So, that way the tasting will be fast, because loading time will be less. Now after that certain fault will remain, it will not be tested by this broadcast scan based method. So, then all this scan chains, they will be joined together, or stitched together into a serial scan chain.

And then the serial mode the scan in comes to the segment one, output of segment 1 is connected to input of segment 2 like that. So, this constitutes say serials scan chain. So, whatever scan in pattern is coming so that will fill up this san chain, and then that pattern will be applied. So, for herd faults, we need to go to the serial a chain mode and the patterns had to be applied in the serial channel, patterns have to be applied in the serial chain mode.

(Refer Slide Time: 13:47)

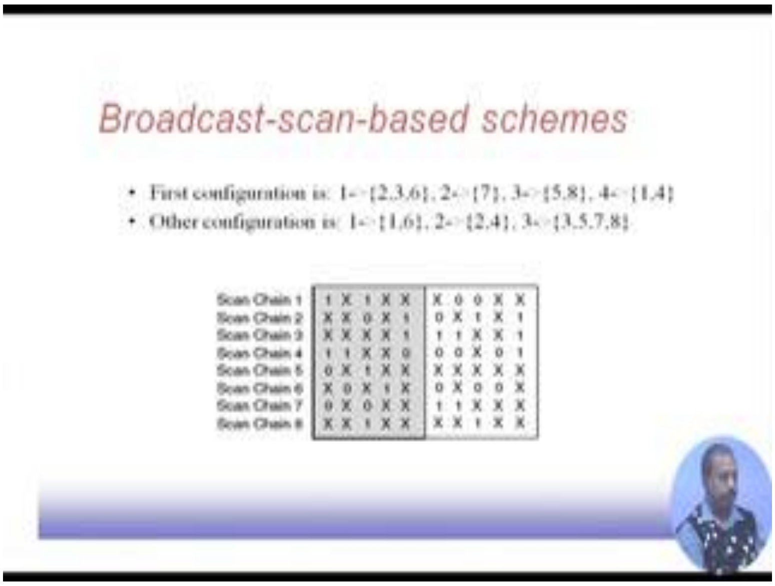


Broadcast-scan-based schemes

- Reconfigurable broadcast scan
  - Reduce the number of channels that are required
  - Static reconfiguration
    - The reconfiguration can only be done when a new pattern is to be applied
  - Dynamic reconfiguration
    - The configuration can be changed while scanning in a pattern

Sometimes we can reconfigure this broadcast scanning. So, reduce the number of channels that are required. So, we can reconfigure the broadcast scan architecture; like this segment 1 was connected to segment 2. So, maybe we can do it in a different way. So, there can be 2 possible strategies; one is static reconfiguration, one is dynamic reconfiguration. In static reconfiguration, so when a new pattern is applied, we can change the structure. So, we can change the configuration of the scan chains. On the other hand dynamic configuration, while the pattern is being scanned in then also the configuration may change.

(Refer Slide Time: 14:30)



Broadcast-scan-based schemes

- First configuration is: 1<->{2,3,6}, 2<->{7}, 3<->{5,8}, 4<->{1,4}
- Other configuration is: 1<->{1,6}, 2<->{2,4}, 3<->{3,5,7,8}

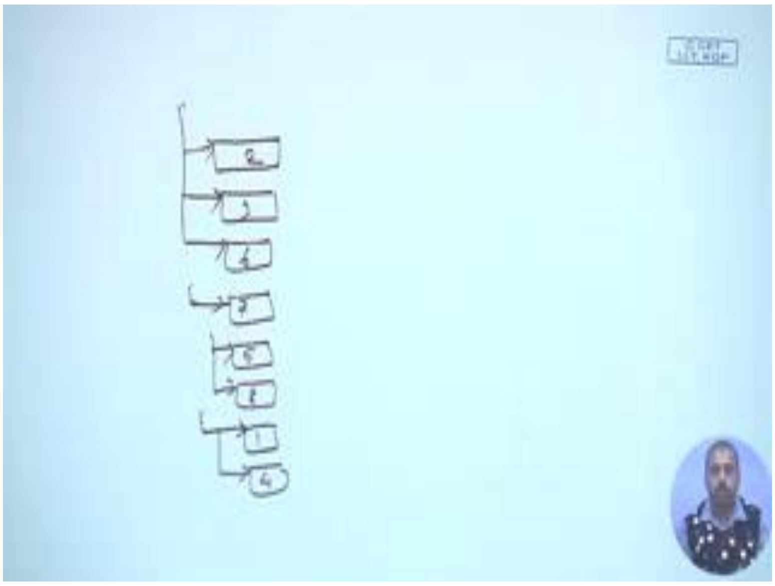
Scan Chain 1	1	X	1	X	X	X	0	0	X	X
Scan Chain 2	X	X	0	X	1	0	X	1	X	1
Scan Chain 3	X	X	X	X	1	1	1	X	X	1
Scan Chain 4	1	1	X	X	0	0	0	X	0	1
Scan Chain 5	0	X	1	X	X	X	X	X	X	X
Scan Chain 6	X	0	X	1	X	0	X	0	0	X
Scan Chain 7	0	X	0	X	X	1	1	X	X	X
Scan Chain 8	X	X	1	X	X	X	X	1	X	X

So, this is a typical example like you see, if you look into this particular structure, you see that. So, supposes scan chain one will get this particular pattern, scan chain 2 is supposed to get this pattern. So, this is the thing.

Now, you see that if you look into this, this scan chain 1 and scan chain 2 they are compatible to each other; like. So, scan chain 1, this column 1 column, this column 1 and column 2, so they are computable to each other. So, this is 1 and this is X, and this is X. So, they are all similar like, this is X and this is 0. So, I can put a 0 here. So, that way this column 1 and 2, they are compatible. In fact, this 2, 3 and 6, so this 2, 3, so they are compatible, similarly 2, 3 and 6, so this 1 also this 3 are 2, 3 and 6, these are all compatibles. So, that way I can, I can do this thing that is, I can combine then into one particular sequence. So, I can combine them into, sorry this 2 3 6, this scan chain 2 3 and 6. So, this scan chain 2 3 and 6, they are compactable like this is X X 0 X 1. I am sorry for the previous explanation by column. So this is actually by the scan chain. So, 2, 3, scan chain 2, scan chain 3 and scan chain 6, they have compatible to each other.

Then scan chain 7; scan chain 7 is not compatible to anybody. So, that way this is this maybe it is taken separately. So, it is compatible with scan chain 2, but 2 is already grouped with, it is already grouped, so 2, 3, 6. So, I cannot group it together with 2 again. So, then 5 and 8, 5 and 8, so they are compatible, 5 and 8, scan chain 5 and scan chain 8 compatible so that way it seems that I can, whenever broadcasting pattern. So, I can broadcast the same pattern to scan chain 2 3 and 6. I can broadcast the same pattern to scan chain 5 and 8, and again to scan chains 1 and 4. So, I can have this type of configuration, where this scan chain 2, 3, 6, they are getting the same pattern.

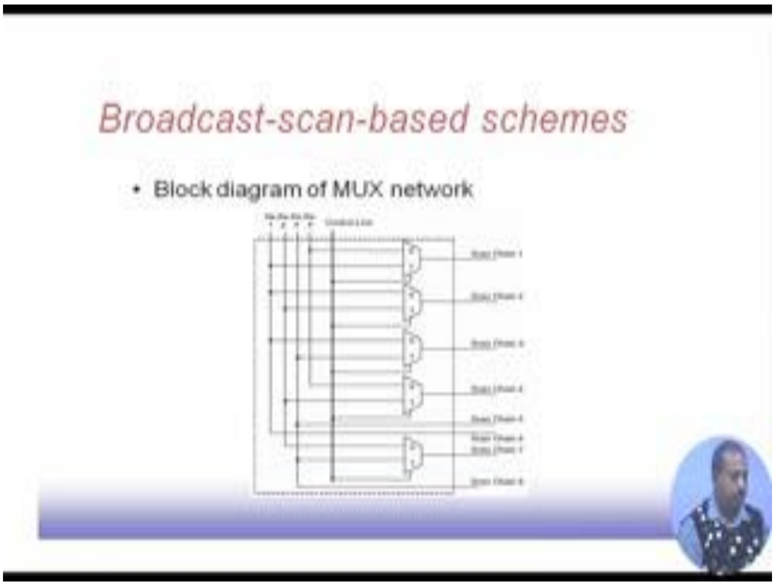
(Refer Slide Time: 17:13)



So, this is 2, this is 3, and this is 6, they are getting the same pattern, same scan input will be coming there. Then this scan chain 7 is having one scan input, then this 5 and 8, they are having another scan input, and then this 1 and 4 they will be having another input. So, ultimately if I have got this 3, I need this 3 separate input to come, and then in the broadcast scan mode, this few test pattern. So, this is the first test pattern, this test pattern can be loaded.

So, after that what happens is that, after this part has been done. For the second part you can see that 1 and 6, so they are compatible. So, this is X, this is 0, this is 0 0 either X or 0. So, that way 1 and 6 are compatible, then 2 and 4 are compatible. So, 2 and 4, so they are compatible. So, this way you can see that after we have shifted this many bits, this 5 bits. So, it is possible, it is required that different type of configuration will exit. So, at that time I need to change the configuration. So, for the second part I need to change the configuration. So, if the scan chains this connection, this scan chain connection can be reconfigured, then it can help us in the shifting process.

(Refer Slide Time: 18:55)



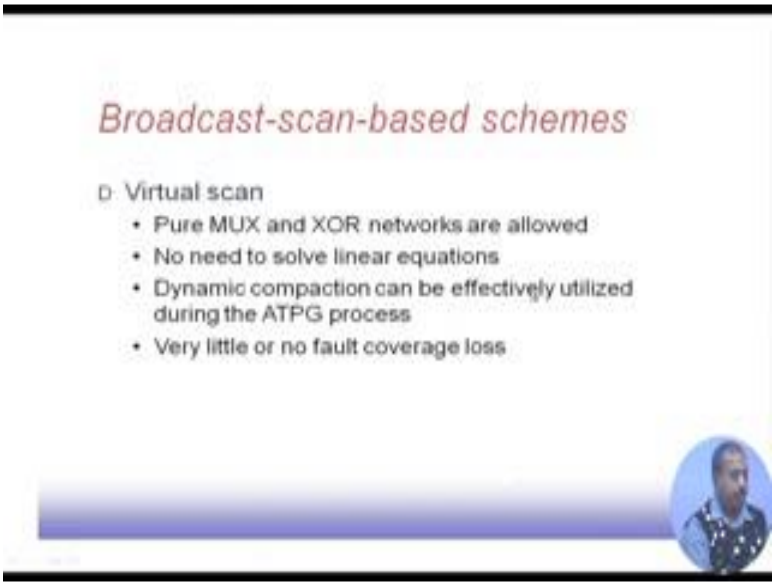
Broadcast-scan-based schemes

- Block diagram of MUX network

So, from the control line, it can select this scan chain 1, in the first session I can say that this scan chain 1 is getting the value from 4, and scan chain 1 is getting the value 4. So, if you see here, scan chain 1 should get the input from pin 4.

Then in the second part, scan chain 1 should get the input from pin 1. So, in the second part you see the scan chain 1 should get input from the pin 1. Similarly scan chain 4 in part, the first part it should get from pin 4, the second part it should get input from pin 2. You see that 4, in the first one it should get from input 4, the second one it should get from pin 2. So, this way, this network can be configured, this network can be designed, and by through this control line. So, we can have this decompressor realize. So, this is from; we are getting only 4 bit input, but they are feeding 8 different scan chains.

(Refer Slide Time: 20:09)



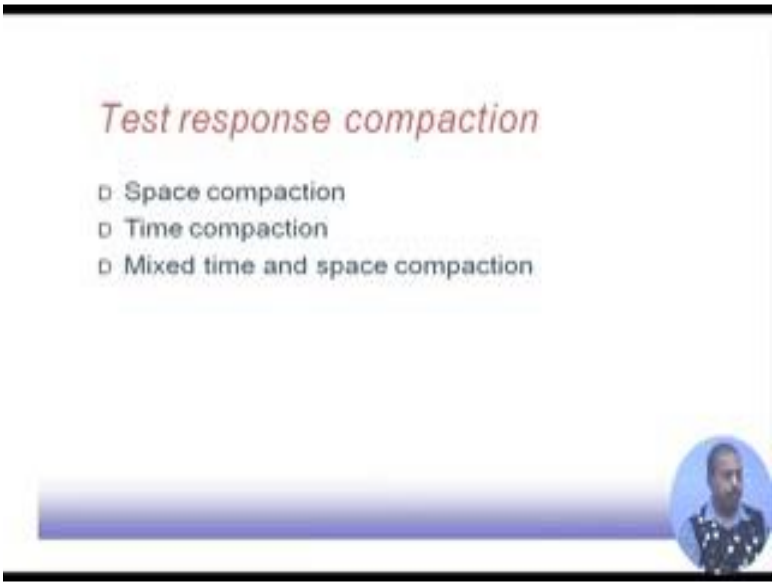
Broadcast-scan-based schemes

- Virtual scan
  - Pure MUX and XOR networks are allowed
  - No need to solve linear equations
  - Dynamic compaction can be effectively utilized during the ATPG process
  - Very little or no fault coverage loss

Sometimes we have got another type of structure which is known as virtual scan. So, virtual scan what happens is that, instead of only multiplexers XOR network can also be allowed, and in fact, you can also allow any other gate network and dot NAND that type of inward you gates, can also be possible. So, now, the advantages that since we are allowing other logic, it is no more linear. So, we do not need to solve linear equations, but we can try to configure those decompressor using those extra gates. So, that it gives us good amount of compression, then dynamic compaction it can be effectively utilized during the ATPG process, because dynamic compaction. So, it tells that after sending.

A pattern up to this much, so the remaining part of the pattern. So, these 2 bits are same. So, if this information is available, then the ATPG can generate pattern like that. So, that goes to bits are same for the remaining part, so these dynamic compaction; so this will result in a very little or no fault coverage loss, because ATPG itself is doing the manipulation of those bits.

(Refer Slide Time: 21:30)



Test response compaction

- Space compaction
- Time compaction
- Mixed time and space compaction

Next we go to the response compaction part. So, far we have talked about the stimulus compaction; that is the test pattern that we are going to apply. So, that will be applied to the, coded pattern will come from the ATE, and it is decompressed by the decoder present at the beginning in circuit, and then it is going to do the stimulus, then it is doing to decode the stimulus, and apply the stimulus to the circuit.

In the response that is obtained, again it has to be compacted. So, this response compaction can be done in 3 different modes. Like we can have space compaction, we can have time compaction, or we can have some mixed mode of compaction. So, space compaction, because out of, say if the circuit has got 100 bits as output, then there is no point keeping all the 100 responses in the signature part. So, we need to store only a few of them. So, that maybe, we can do some reduction, so that is the space compaction. Then we can have time compaction. So, time compaction is maybe I am applying ten thousand patterns. Now we do not want to remember all the 10000 responses. So, that way we want to remember some summary of the whole session. So, that way the time compaction can be done, and there can be naturally a mix of this 2. So, we will continue in the next class.