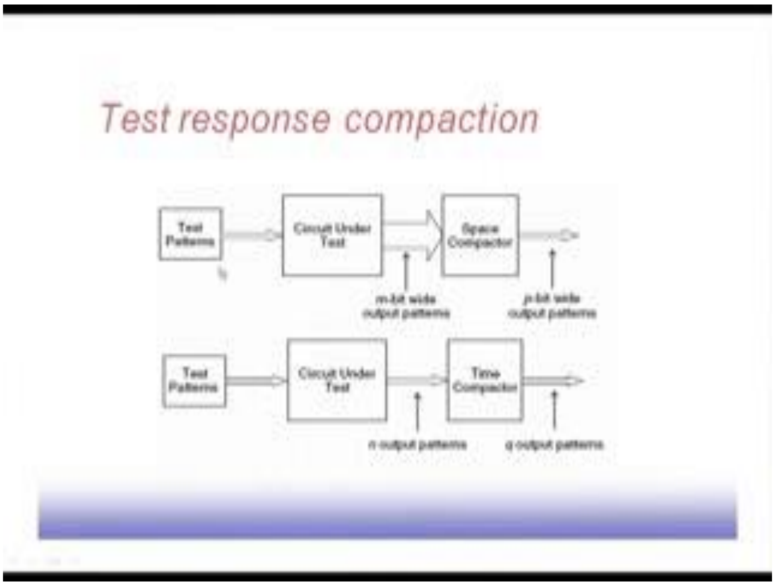
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 31
Test Compression (Contd.)

Test response compression; so, it comes into the overall diagram can be shown like this that the test patterns that are coming from the AT, so that is after the; they are may be a decompressor here that is not shown explicitly.

(Refer Slide Time: 00:25)



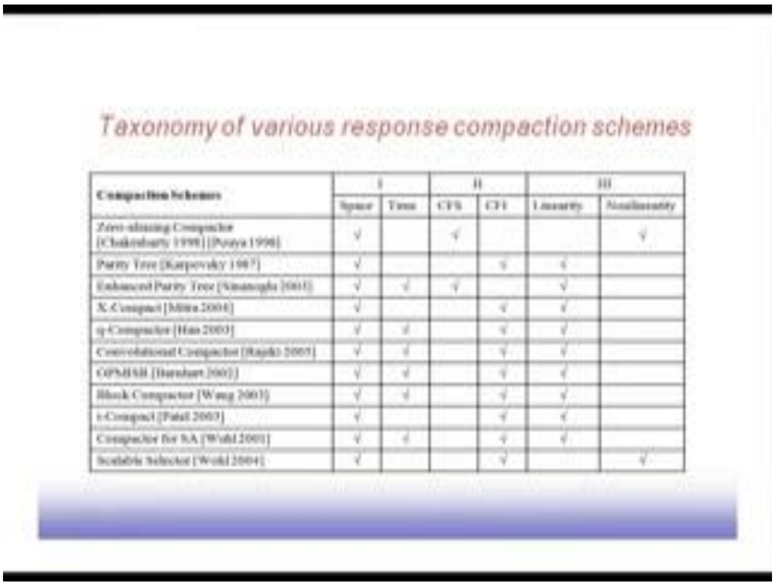
Test response compaction

So, after the decompressing the test patterns are applied to the circuit under test. Now circuit generates m-bit wide output pattern, now out of that m-bit this space compactor will generate it p-bit wide output response, so output pattern. So, it may a typical case can be that it just selects p of the outputs out of M and then it is coming as this p-bit part pattern. So, that is one possibility, but naturally if we just select p out of m-bits then p out of M outputs then naturally many of the outputs is not contributing in the signature part. So, normally we put some network there the space compactor is a network which will be combining the responses and accordingly these p and it is an M to p circuit. So, M input p output circuit, so what network we put inside so that will decide the quality of the space compactor.

On the other hand for time compaction, test patterns are being applied to the circuit under tests. So, there are n also, if there are n input patterns so there will be n output patterns. So, out of this n output patterns we do not want to remember all the outputs, but maybe we remember few such sequences, now Q output patterns, in the ideal case Q is equal to 1 where we remember only the final value that we have got. Naturally what we want is we want to have the impact of every response loaded, have the impact of every response available in the final signature that we have. So, that way this time compacter is going to be useful in reducing the responses in the time domain.

So, again there will be some summation that will be done, some signature computation that will be done. So, some network will be put there, so that it does the compaction in time. So, both this space compactor and the time compacter, they are going to be used in such a fashion that we can store only may be 1 or 2 signatures from the for the entire test session.

(Refer Slide Time: 03:01)

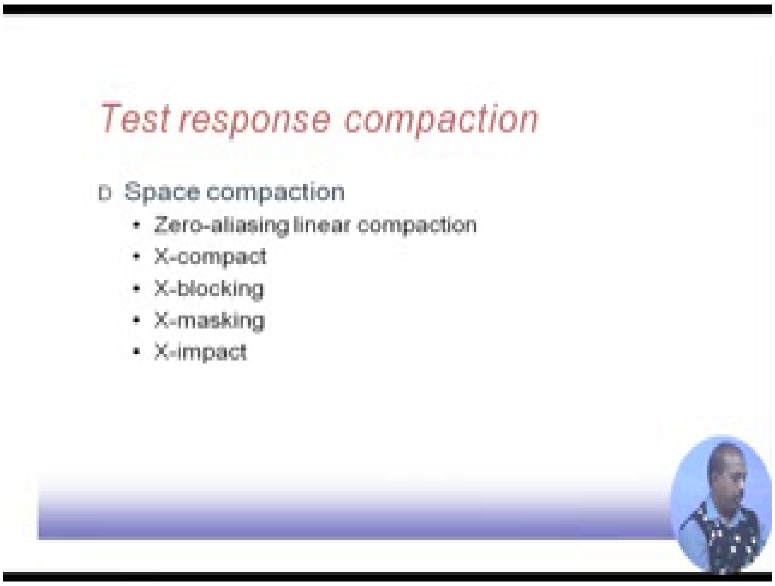


Taxonomy of various response compaction schemes

Compaction Schemes	I		II		III	
	Space	Time	CFS	CFI	Linearity	Nonlinearity
Zero-aliasing Compactor [Chakrabarty 1998] [Pouya 1998]	√		√			√
Parity Tree [Karpovsky 1987]	√			√	√	
Enhanced Parity Tree [Sinanoglu 2003]	√	√	√		√	
X-Compact [Mitra 2004]	√			√	√	
q-Compactor [Han 2003]	√	√		√	√	
Convolutional Compactor [Rajski 2005]	√	√		√	√	
OPMISR [Barnhart 2002]	√	√		√	√	
Block Compactor [Wang 2003]	√	√		√	√	
i-Compact [Patel 2003]	√			√	√	
Compactor for SA [Wohl 2001]	√	√		√	√	
Scalable Selector [Wohl 2004]	√			√		√

Now, there are many schemes that can be, that has been deported some of them are linear, some of them are non-linear, some of them do space compaction and time compaction both some of them do only space compaction. So, we will look into some of these strategies.

(Refer Slide Time: 03:18)

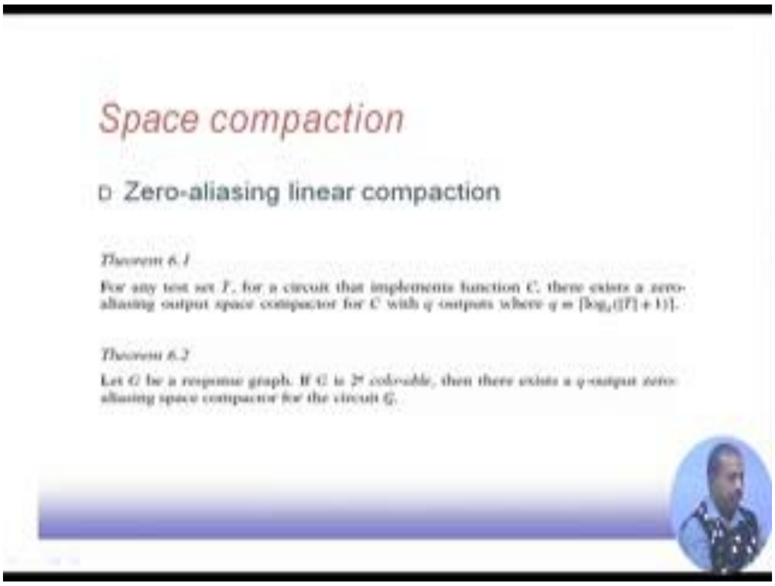


Test response compaction

- Space compaction
  - Zero-aliasing linear compaction
  - X-compact
  - X-blocking
  - X-masking
  - X-impact

Now, space compaction, so first one is 0 value as linear compaction. So, we want that due to this compaction these responses are not mixed up. So, it means that the one faulty response that should lead to a faulty signature. So, it should not get current same as the correct response. On the other hand if I have got say 2 faulty responses then it is expected that those 2 faulty responses they should give rise to 2 different signatures. So, in that case I will have a zero-aliasing linear compaction. Then there are certain other strategies like compact X-compact, X-blocking, masking, X-impact etcetera that are used for this compaction process.

(Refer Slide Time: 04:09)



Space compaction

- Zero-aliasing linear compaction

Theorem 6.1

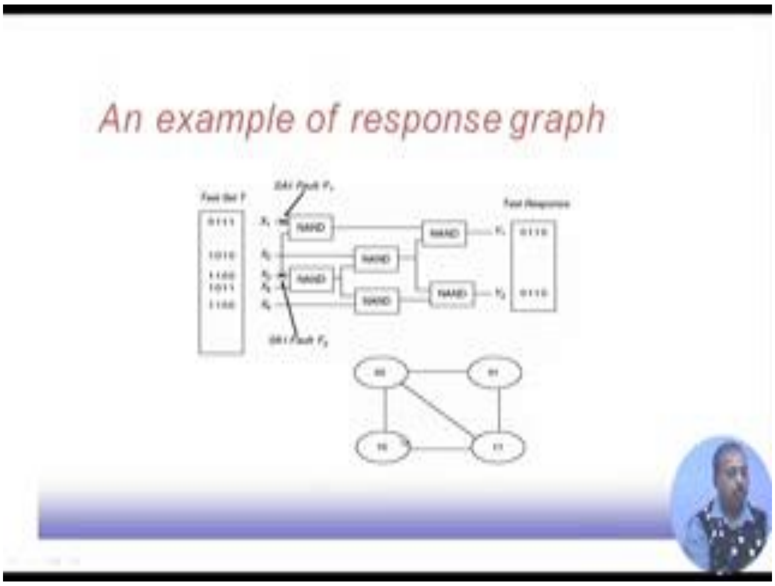
For any test set T, for a circuit that implements function C, there exists a zero-aliasing output space compactor for C with q outputs where $q = \lceil \log_2(|T| + 1) \rceil$.

Theorem 6.2

Let G be a response graph. If G is 2^q colorable, then there exists a q-output zero-aliasing space compactor for the circuit G.

So, zero-aliasing space compaction, linear compaction it says there are 2 results it says that for any test T for a circuit that implements function C there exists a zero-aliasing output space compactor C with Q output where Q is given by log 2 T plus 1. So, this is a result. So, for any circuit, I can have, if it has got. If I have got T as the test set size then this mod of T is actually the size of the test set. So, we can always have a; you can always come up with a circuit C that has got Q outputs and this Q is equal to this and there will be 0 value aliasing and if another results says that if G be a response graph. So, what is the response graph we will see and if G is 2 power Q colorable, then there exists a Q output zero-aliasing space compactor for the circuit.

(Refer Slide Time: 05:10)



An example of response graph

So, what is a response graph that we will see, suppose this is a test set that we have and this is a circuit that we have and these are the patterns that are applied 0 1 1 1and if we apply the first pattern then the response that we get is 0 then if we apply the next pattern we get the response as 1. So, like that it is it different responses are happening in Y 1 line and Y 2 line. So, when the fault F 1 is present when the fault F 1 is present then this response changes from 0 0 to say 1 1 or 0 0 to 0 1. So, what happens is that when this when a particular fault is present then from the fault free response the response changes to some other patterns. So, this is actually the response graph that we are talking about.

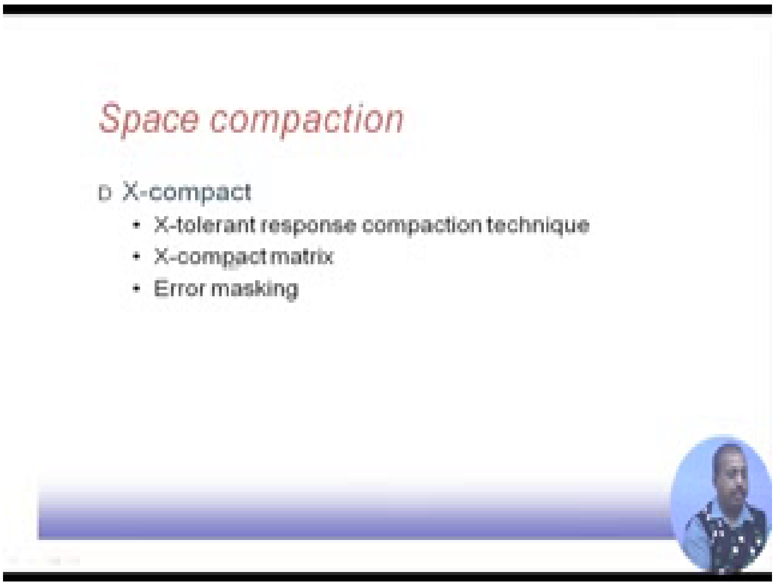
So, under the presence of a particular fault if the response changes from the fault free response to something else then we have an age between the 2 so; that means, under the presence of some fault response changes from 0 0 to 0 1, but there is no fault in the circuit that can change this response 0 1 to 1 0, so that is not possible. However, for if 0 0 is the good response. So, we see that due to the presence of different faults. So, it can give rise to different response they are different other responses and we add ages in those cases.

So, this is actually the response graph and this result says that if the graph G is 2 power Q colorable then there exist a Q output zero-aliasing space compactors. So, coloring of a graph means if I have a graph then I want to put, I want to give some color to the nodes such that if 2 nodes are having same having an age connecting them they should not have the same color. So, if you want to color this graph then suppose this node I color has red then none of these three nodes can be colored as red. So, maybe I color it with a green, then since this red and green you already used, so I cannot use those colors for this one. So, I have to use a third color here. So, this is say blue.

Now you see that this node, for this node I do not need a new colors, because this green color can be used to color this one also. So, you see that 3 colors are sufficient for coloring this particular graph. So, that is the result it says that if it is 2 power Q colorable, so naturally the graph is 4 colorable. So, taking it a power of, there will exist a 4 output zero-aliasing space compacter for this circuit.

So, since this graph four colorable, you can have a compactor that will have 4 outputs. Now in this particular case of course, it is not very important because this original circuit has 2 outputs, now I am telling that I will have a compactor that will have 4 outputs. So, it is not acceptable, but in general you can say that if the number of outputs are large then we can I have, we can construct these graph and taking help of that theorem. So, we can come to a network which will give us the space compaction without aliasing.

(Refer Slide Time: 08:37)

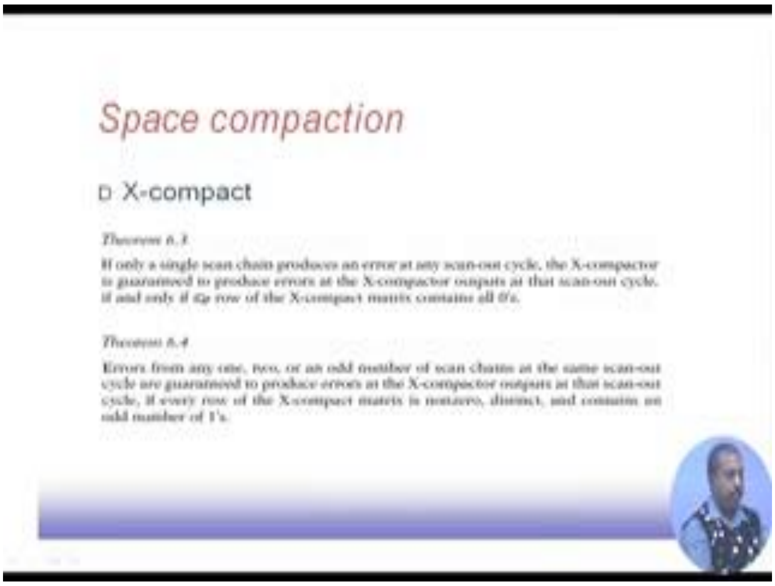


Space compaction

- X-compact
  - X-tolerant response compaction technique
  - X-compact matrix
  - Error masking

So, next we will look into this space compaction techniques, first one in that category is the X-compact. So, this is X-tolerant response compaction technique. So, what happens is that it will tolerate the excess that the unknowns that come in the response. So, it will tolerate that. So, will have a matrix which is called X-compact matrix and then this error masking will be utilized.

(Refer Slide Time: 09:02)



Space compaction

- X-compact

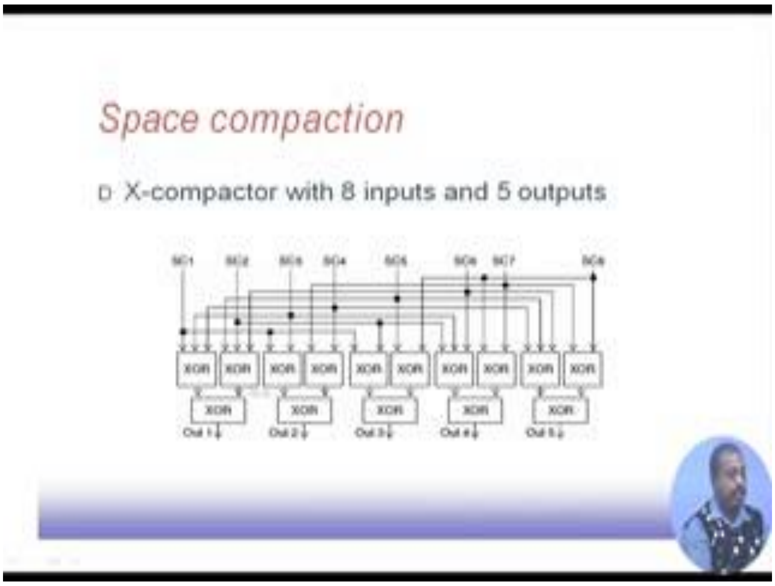
Theorem 6.3

If only a single scan chain produces an error at any scan-out cycle, the X-compactor is guaranteed to produce errors at the X-compactor outputs at that scan-out cycle, if and only if no row of the X-compact matrix contains all 0's.

Theorem 6.4

Errors from any one, two, or an odd number of scan chains at the same scan-out cycle are guaranteed to produce errors at the X-compactor outputs at that scan-out cycle, if every row of the X-compact matrix is nonzero, distinct, and contains an odd number of 1's.

(Refer Slide Time: 09:08)



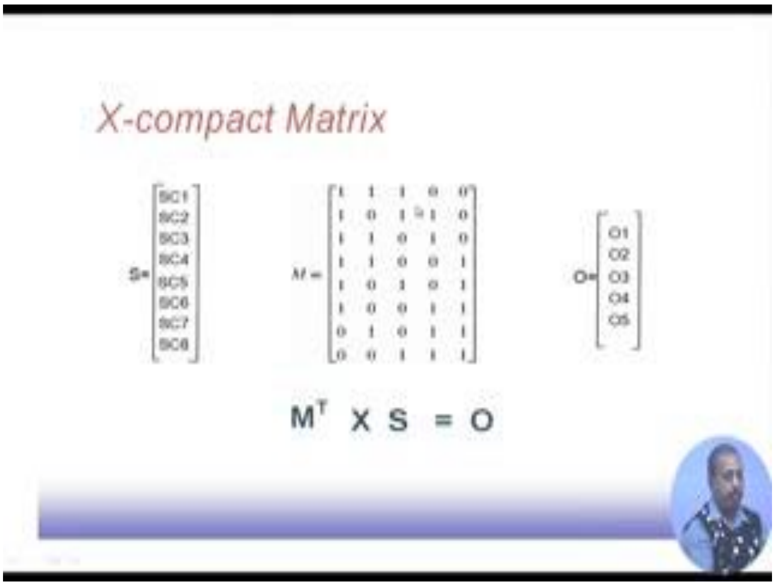
Space compaction

- X-compactor with 8 inputs and 5 outputs

Now, this X-compact, it has the basic idea is like this. So, it will frame one XOR network. So, in this XOR network this scan change. So, it has got say 8 inputs, so this scan change this 8 inputs are coming and then this 5 outputs are, 5 outputs are there, 8 input 5 outputs. Now this network is constructed in some fashions, so there this scan change values are added using this XOR gates, but how this compactor is fabricated, that is actually given by this 2 theorems. So, it says that if only a single chain produces an error at any scan-out cycle the X-compactor is guaranteed to produce errors at the X-compactor outputs at that scan-out cycle if and only if more over of X-compactor matrix contains all 0s.

So, it says that, X-compact matrix is this one. So, this is the X-compact matrix that gives the direct connection that we have here.

(Refer Slide Time: 10:00)



X-compact Matrix

$$S=\begin{bmatrix} SC1 \\ SC2 \\ SC3 \\ SC4 \\ SC5 \\ SC6 \\ SC7 \\ SC8 \end{bmatrix} \qquad M=\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad O=\begin{bmatrix} O1 \\ O2 \\ O3 \\ O4 \\ O5 \end{bmatrix}$$

$$M^T \text{ X } S = O$$

So, this connection say these output one it is basically a sum of these scan chain 1 then this is basically a sum of this scan chain, M transpose into C. So, M transpose if we take. So, it is basically the sum of all these scan chain, so SC 1, SC 2, SC 3, SC 4, SC 5, SC 6. So, if you look into this diagram you see that here I am taking SC 1, so here I am getting SC 3, here I am getting SC 4. So, these three are coming to the first XOR gate and for the second XOR gate I am getting SC 2, then this one that is SC 5 and this one that is SC 6 and then at the output of this XOR gate. So, all these up to SC 6 all these XOR all these are scan chain values are going to be added to produce this output one. So, this matrix also if you take M transpose into S, what you are getting is that these particular columns 1 1 1, they get multiplied by this S and that will be defining the O 1. So, that is the structure of the X-compact matrix.

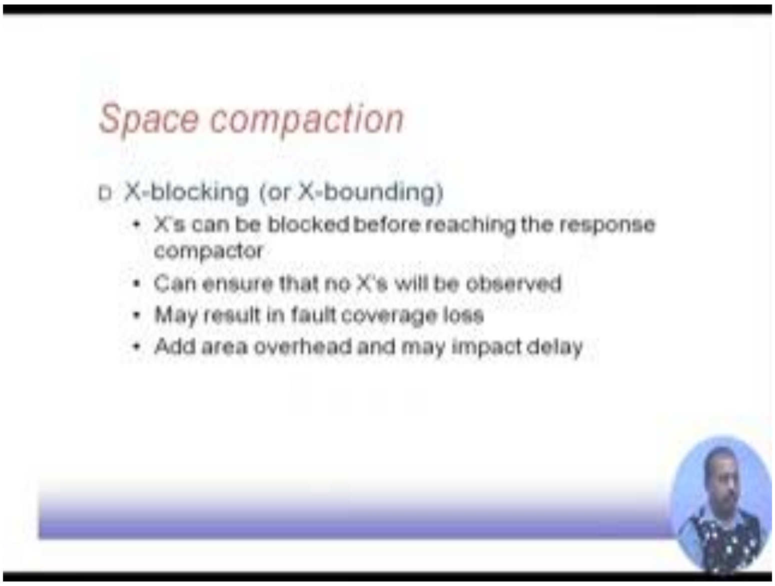
Now, it says that the if the X-compact matrix has got this property that is if it has got no row of X-compact matrix contains all 0s, if this condition is ensured that in that case any error that is produced in some scan out cycle. So, if the error comes only in a single scan chain if only a single chain produces an error at any scan out cycle then this X-compact matrix will be able to catch that error at that scan out cycle. So, the compacted output also will have a different value at that particular scan out cycle.

So, it is true if only a single chain produces an error at any scan cycle. Now if there are multiple chains that produce error. So, errors from any 1, 2 or an odd number of scan chains at the same scan-out cycle are guaranteed to produce errors at the X-compactor outputs at that scan out cycle, if every row of the X-compact matrix is non-zero, distinct, and contains odd number of 1’s. So, this is another constraint that is put on the construction of the X-compact matrix that the rows they should be non0, the rows should be distinct and each of them should contain an odd number of 1’s. So, if we look in to this X-compact matrix that has been used here, this contains odd number of 1’s this is all of them are containing odd number of 1’s. So, and none of them are 0 all of them are distinct, none of them are same. So, all the three properties are satisfied.

In that case, it says that this particular property will be satisfied that is error from any 1, 2 or odd number of scan chains at any scan-out cycle will produce some erroneous output at the X-compactor outputs. So, actually, if any of the say odd number of chains are giving faulty result like SC 1, SC 3 and say SC 6; they are giving faulty result then out of this out 1, out 2, out 3, out 4, out 5. So, at least one of them will produce the output which is different from the fault free output. So, that way it ensures that whatever is; if there is some fault occurring and it is coming to the scan chain. So, the circuit is producing a different output and it is coming to the scan chain the test corresponding to this pattern to detect that particular fault then this compactor circuit is not going to lose the particular detection. So, it will pass the detection to one of the outputs and that way it is guaranteed that this the aliasing will not occur, the fault free faulty response will not match with the fault free response for that particular cycle.

So, this is the beauty of this X-compact scheme. So, we can construct this network. Of course, there are catches like say if there are even number of chains fault then this none of these results tell anything. So, if there are even number of faults then of course, we cannot do anything so, but it is very much likely that this even number of faults the, it is if some test patterns create even number of errors then of course, there may be some other test patterns which will detect it at a odd number of scans, odd number of output for the prime circuit under test, as a result it will produce faulty outputs at odd number of scan chains. So, as a result it will get detected by the second pattern. So, that way we are from. So, this is a very good approach for doing the space compaction where we can without any aliasing occurring.

(Refer Slide Time: 15:16)

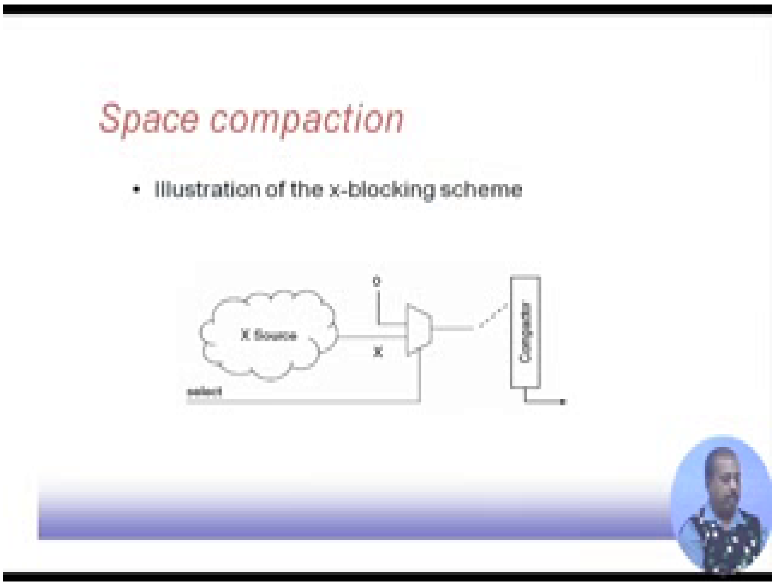


Space compaction

- X-blocking (or X-bounding)
  - X's can be blocked before reaching the response compactor
  - Can ensure that no X's will be observed
  - May result in fault coverage loss
  - Add area overhead and may impact delay

Another possibility is space compaction is known as X-blocking or X-bounding. So, Xs can be blocked before reaching the response compactor. So, we can say that if there is some unknown value coming. So, we can block that value. So, we can ensure that the Xs no Xs will be observed. So, if this unknown, you can stop it before it reaches the compactor.

(Refer Slide Time: 15:43)

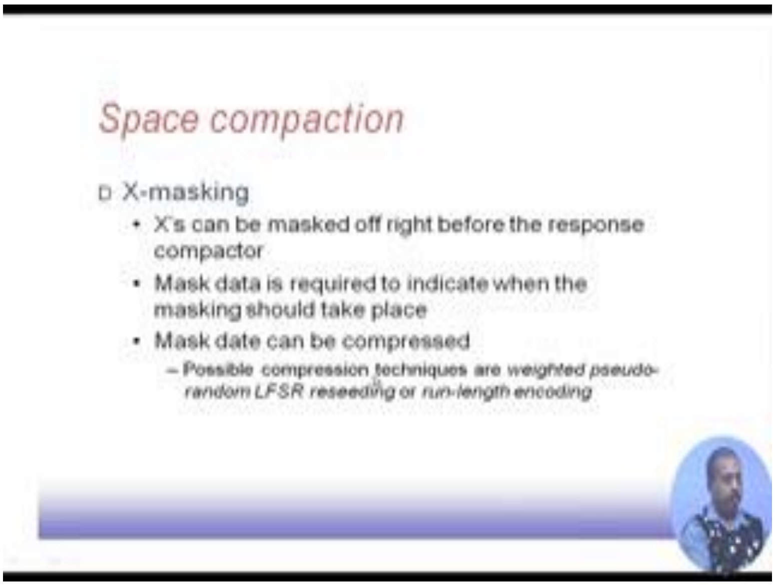


Space compaction

- Illustration of the x-blocking scheme

So, basically the idea is like this. So, if we know that a particular scan chain can produce some unknown value at a particular point. So, we pass it through a multiplexer and this multiplexer other input is 0. So, if we know that at this particular test pattern the output of this scan chain is going to be undefined then we can say; we can put this select line. So, that it passes 0 to the compactor. So, the effect of this particular scan chain will not affect the compactor operation. So, that is one X-blocking operation in the space compaction.

(Refer Slide Time: 16:20)

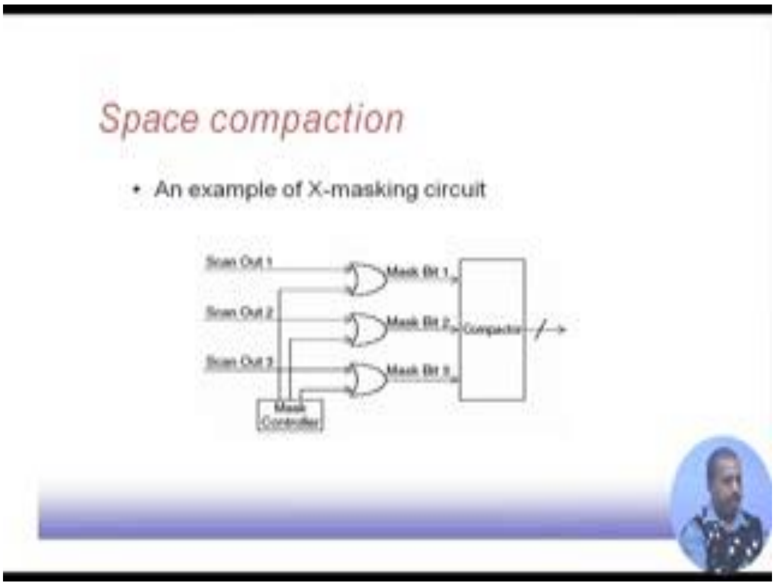


Space compaction

- X-masking
  - X's can be masked off right before the response compactor
  - Mask data is required to indicate when the masking should take place
  - Mask date can be compressed
    - Possible compression techniques are *weighted pseudo-random LFSR reseeding* or *run-length encoding*

Another possibility is X masking. So, the Xs can be masked off right before the response compactor. So, they can also be masked off, masking I think, this is the thing.

(Refer Slide Time: 16:30)

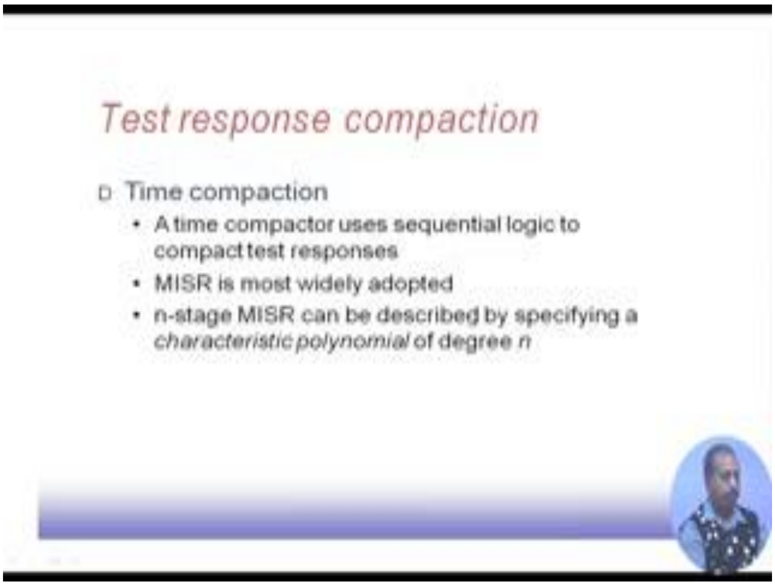


Space compaction

- An example of X-masking circuit

So, we have got a mask controller. So, again the same thing, these are scan-out, scan chain outputs are coming. So, we know that this particular line has to be masked out, we can put the mask controller can put a one here, so that this one gets masked off. So, this mask controller it continually generates masks. So, as this scan shifting operation is going on. So, this will continually generate the masks so that whichever pattern whichever mask which ever scan chain has to be allowed so that will go and in that case this mask bit is 0. On the other hand if the mask bit is 1 then that particular scan out bit will not go, it will get a 1 at the compactor input. So, this way we can have this X-masking circuitry. So, that this X bit, this unknown values do not reach the space compactor.

(Refer Slide Time: 17:25)



Test response compaction

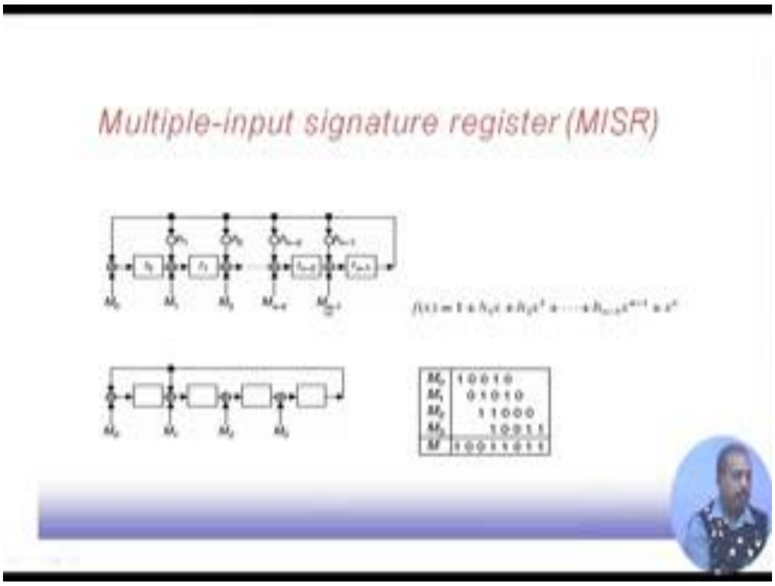
- Time compaction
  - A time compactor uses sequential logic to compact test responses
  - MISR is most widely adopted
  - n-stage MISR can be described by specifying a *characteristic polynomial* of degree *n*

Next we look into the time compaction. So, space compaction what it has done? It has ensured that out of say M outputs from the circuit or the M scan chains that we have. So, it has selected it has combined those M outputs in some fashion. So, that it gives say p outputs, but for each test pattern, it is for each scan cycle it is giving us one of one such p-bit pattern. So, because this continually this scan out operations are going on when the next pattern is being shifted in the previous responses being shifted out. So, over the entire scanning operation, over the entire scan operation this Q bit responses will continue to come. But if we store all those p-bit responses then again there will be huge amount of information that needs to be stored. So, what is done is we compact these responses in the time domain as well. So, this is known as the time compaction.

So, time compactor it normally uses sequential logic to compact test responses, so time compact we cannot do with combinational logic because combinational logic does not have any memory. So, that ways, it will lose its content of the previous cycle whatever that had happened in the previous cycle.

So, this combinational logic cannot be used. So, we go for a sequential logic of the compact test responses the most widely used structure is the MISR. So, this MISR, this can continually receive the responses and they can be combined into one value. So, this actually you know that if MISR has got a polynomial its structure is represented by a polynomial and the inputs are being fed. So, whatever remains in the MISR after running for some number of cycles this is basically a division operation that is the remainder of that input stream getting divided by the polynomial so that will come as the signature. So, any n-stage MISR can be described by specifying a characteristic polynomial of degree n. So, this we have already seen in this chapter.

(Refer Slide Time: 19:40)



Multiple-input signature register (MISR)

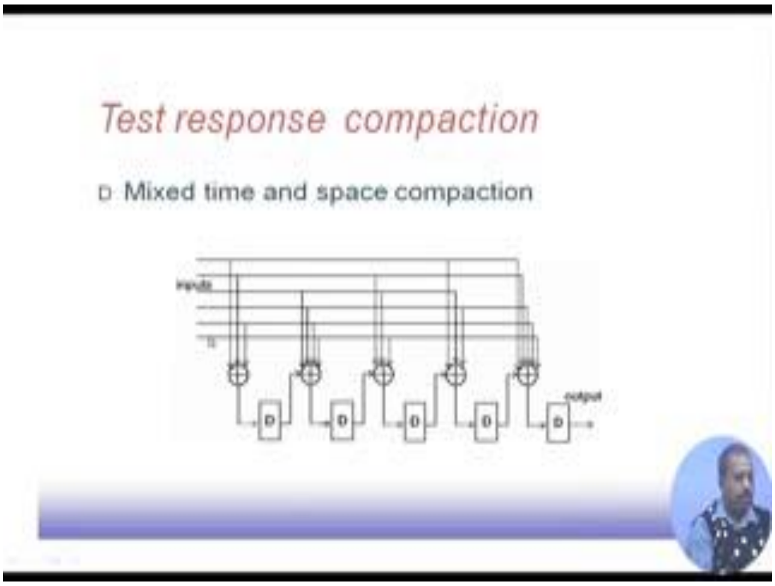
$f(x) = 1 + h_1x + h_2x^2 + \cdots + h_{n-1}x^{n-1} + x^n$

M_0	10010
M_1	01010
M_2	11000
M_3	10011
M	10011011

So, if this is one multiple input signature register. So, we have got this r 0, r 1 up to r n minus 1. So, this is represented by this characteristic polynomial 1 plus h 1 x plus h 2 X square h n minus 1 x n minus 1 plus x to the power n. So, depending upon the values of this h s, this characteristic polynomial is getting defined.

Now, this responses that are coming from the scan chain, they are connected to this line this M lines - M 0, M 1, M 2, M n minus 2 up to M n minus 1. Or say this is a typical example where we have got these typical, one particular connection and this M 0 M 1 line they are connected like this. Now you see if you trace through this operation like in the first cycle, initially all these contents are 0, now this M 0 is 1. So, this cell will get a one after some, after the first cycle then after the second cycle this M 0 is 0 and it will depend on what value it comes at this point. So, it is basically at this point this is 0. So, 0 is coming here so this will again get a 0. So, this way this division will take place. Ultimately it will produce this particular pattern 1 0 0 1 1 0 1 1. So, that is a pattern that will be, that is the actual the representation. The same message bit that we have over number of cycles can be represented by a single message bit like this. So, what the remainder will be coming is nothing but division of this message bit by these effects by the characteristic polynomial effects.

(Refer Slide Time: 21:20)

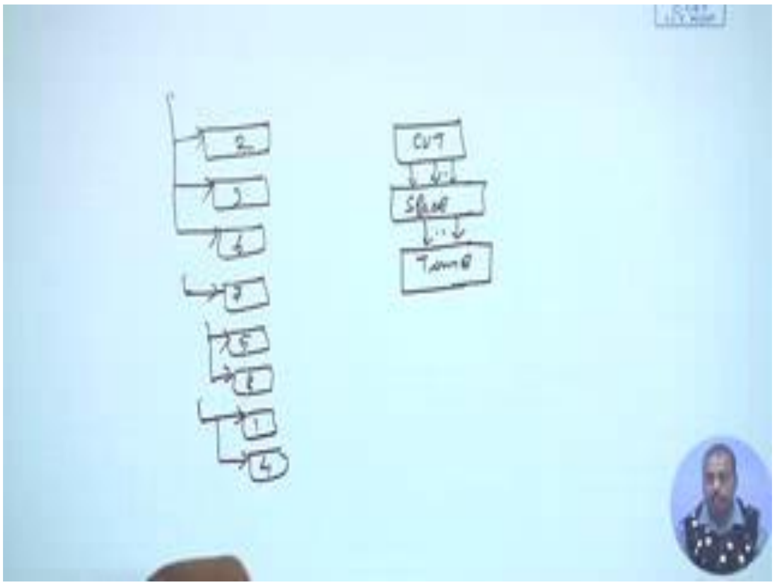


Test response compaction

□ Mixed time and space compaction

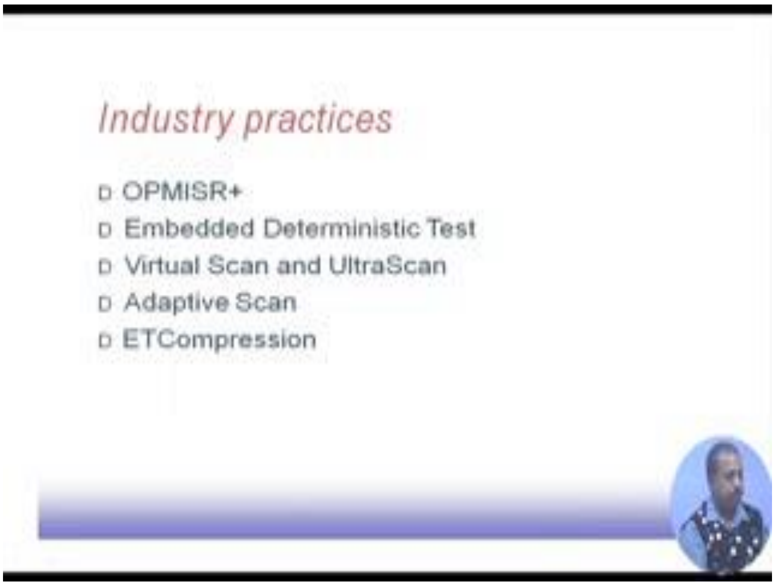
So, we can also have mixed time and space compaction mechanism. So, you see what has happened is that this circuit, this is coming from the scan chain side. So, it has got 1, 2, 3, 4, 5, 6, there are 6 inputs; there are 6 scan chains that are coming and I have got, so this x n network, this is forming, it is some sort of space cum time compactor. So, see instead of feeding a single value what we have done? We have fed here four values. So, here I have fed this 3 value. So, here I have fed these 4 values. So, this is basically the space compaction part plus this flip flop connection is going there. So, this is also XOR. So, this is basically the time compaction part. Ideally we should do first the space compaction. So, this is basically the circuit. So, here I have got this response. So, these are the responses coming, so this is passing through the space compactor.

(Refer Slide Time: 22:30)



And then it produces some less number of inputs, less number of outputs that passes through a time compactor and, at the end of this time compactor. So, this will have the golden response stored here. So, golden signature at the end this time compactor will have the golden signature in it. So, the circuit that is shown in this particular diagram, here it is done combinely, so this space compactor is nothing but some XOR network and this time compactor is nothing, but some MISR. So, this XOR part can be merged together and we can come up with a structure like this which combines the space and time compaction jobs together. So, that way it is a mixed time and space compactor.

(Refer Slide Time: 23:17)

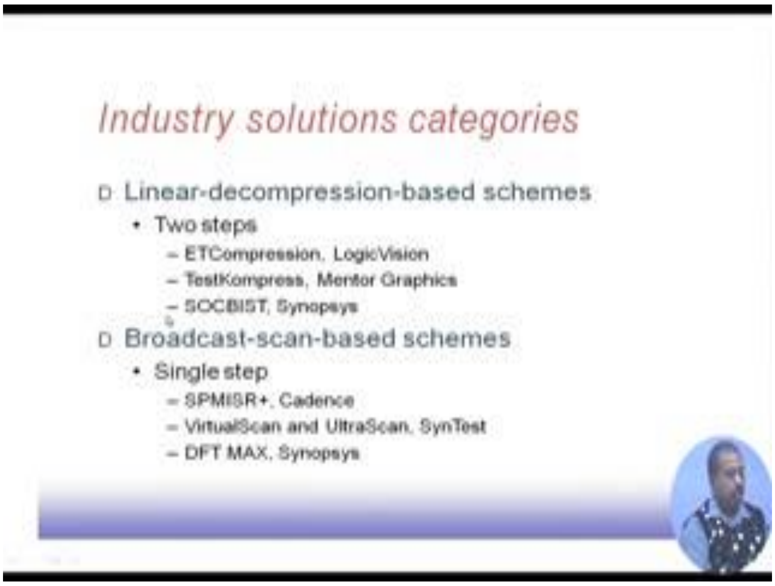


Industry practices

- OPMISR+
- Embedded Deterministic Test
- Virtual Scan and UltraScan
- Adaptive Scan
- ETCompression

There are several standards industries like OPMISR, then embedded deterministic test, virtual scan, adaptive scan, etcetera.

(Refer Slide Time: 23:28)

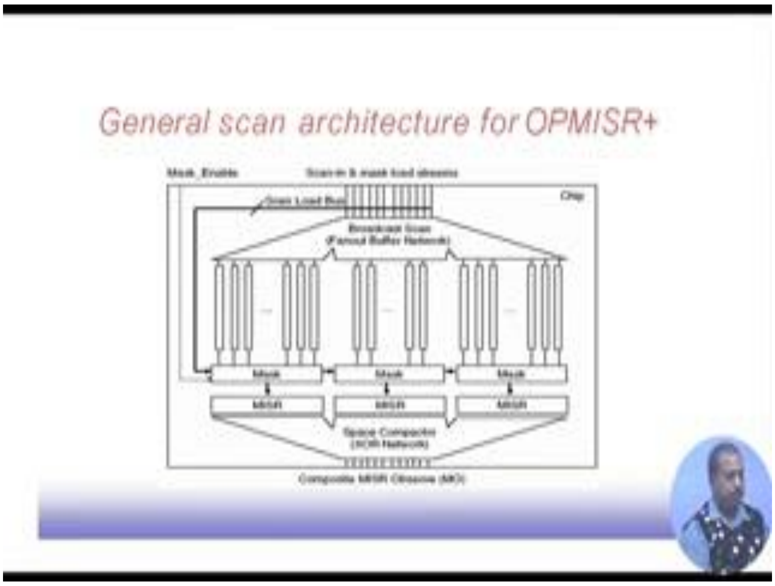


Industry solutions categories

- Linear-decompression-based schemes
  - Two steps
    - ETCompression, LogicVision
    - TestKompress, Mentor Graphics
    - SOCBIST, Synopsys
- Broadcast-scan-based schemes
  - Single step
    - SPMISR+, Cadence
    - VirtualScan and UltraScan, SynTest
    - DFT MAX, Synopsys

Now, this linear decompression based schemes. So, there are some schemes which are based on linear decompression. So, these are the schemes like ETCompression, LogicVison, TestKompress, Mentor Graphics tool then SOCBIST, synopsys tool. So, they are based on linear decompression and broadcast-scan-based schemes we have got SPMISR plus then the tool cadence, virtualscan, ultrascan, DFTMAX from synopses. So, they have got this type of solution, broadcast scan based solutions.

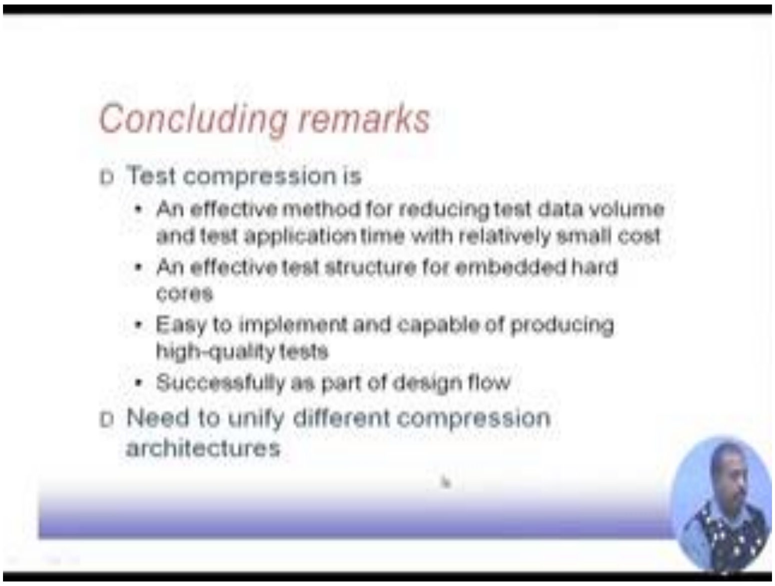
(Refer Slide Time: 24:03)



General scan architecture for OPMISR+

So, we look into this OPMISR plus structure. So, here what happens is that we have got this scan chain; they are going in to this broadcast scan mode, in the broadcast scan mode this scan chains are loaded. Now there is a mask, this scan whenever the patterns are getting loaded then this mask control is also loaded. So, as a result when these responses are coming from this scan chains, depending upon the mask bit some of the responses will be allowed to come to this MISR and others will not be there. So, that way it will be doing this, it will be coming to this MISR as a result it will do a space compaction and this finally, we get this result. So, here the time compression is done first and then followed by the space compression by this space compactor XOR network. So, that way it is composite MISR observed, so that is output part.

(Refer Slide Time: 25:08)



Concluding remarks

- Test compression is
  - An effective method for reducing test data volume and test application time with relatively small cost
  - An effective test structure for embedded hard cores
  - Easy to implement and capable of producing high-quality tests
  - Successfully as part of design flow
- Need to unify different compression architectures

So, there are many other standards, you can just go through some literature on that. In conclusion, like test compression it is an effective method for reducing test data volume and test application time with relatively small cost because it reduces the cost because we do not need to use more advanced AT to store huge amount of test data or apply the test at a higher frequency. At low frequency also we can just transfer the patterns say and then we can using a decompressor we can just decompress the test patterns and apply to the circuit and then we can also do some response compaction. By response compaction we can combine the responses in to 1 or 2 signatures and that way this amount of information to be stored for response will also be less. So, we can do this operation. So, we can do the compression part easily.

Now, it is easy to implement a capable of producing high quality tests, we have seen that there is zero-aliasing space compactors are available then for this time compaction also this LFSRs they show very high aliasing rate. So, as a result we can say that they are the false will not get alias the responses are not getting aliased. So, that way we can have very good scheme when this test compression is put into operation. Of course, you need to unify different compression architectures. So, there are different architecture we have seen only OPMISR plus, there are other architecture as well, but as a result what has happened is that standardization is a question. So, which one can be a standard architecture, that has to be decided.