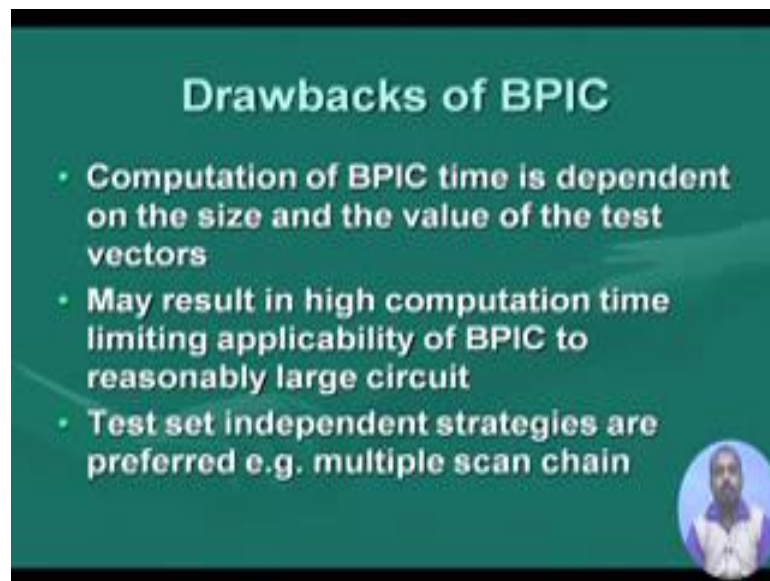


Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 36
Low Power Testing (Contd.)

(Refer Slide Time: 00:26)

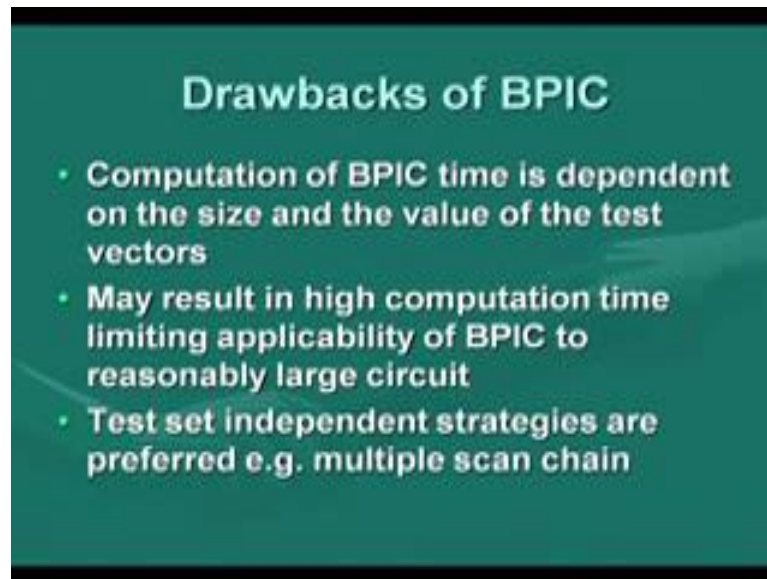


Drawbacks of BPIC

- Computation of BPIC time is dependent on the size and the value of the test vectors
- May result in high computation time limiting applicability of BPIC to reasonably large circuit
- Test set independent strategies are preferred e.g. multiple scan chain

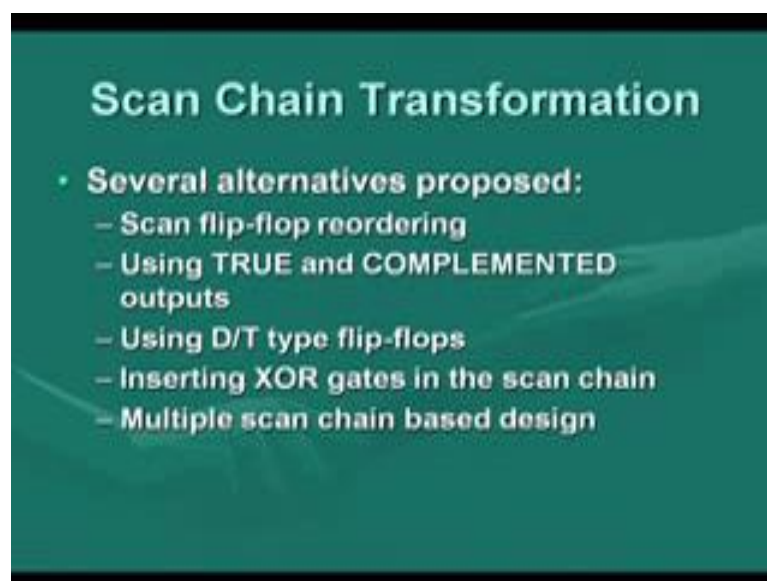
So, BPIC has got some drawbacks as well. Like computation of the BPIC time is dependent on the size and value of the test vectors. So, the base primary inputs change, so when this will be done. So, as I have said that it is not there is no straightforward way to tell it, so it is very much circuit dependent and of course, the pattern dependent as well. So, maybe for the most difficult part is that for different patterns, the best input change time maybe different. So, as a result, it may be difficult to come up with a very some fixed input change time for all the test patterns, so that creates a difficulty. And this may result in high computation time limiting applicability of BPIC to reasonably large circuit. So, that I have told that ok this will take lot time to find out it, but definitely it can improve these power significantly. So, since it is an one time effort for a given test pattern set, so it may be the case that we go for this.

(Refer Slide Time: 01:22)



And test set independent strategies are preferred like we should have multiple scans chains. So, one possible, so if multiple scans chains are there then this individuals scan chains are of shorter length. So, as a result this number of shifts cycles will be less in this case. So, naturally we can have these multiple scan chain based strategy and BPIC can be computed there, because the number of shift cycles is less.

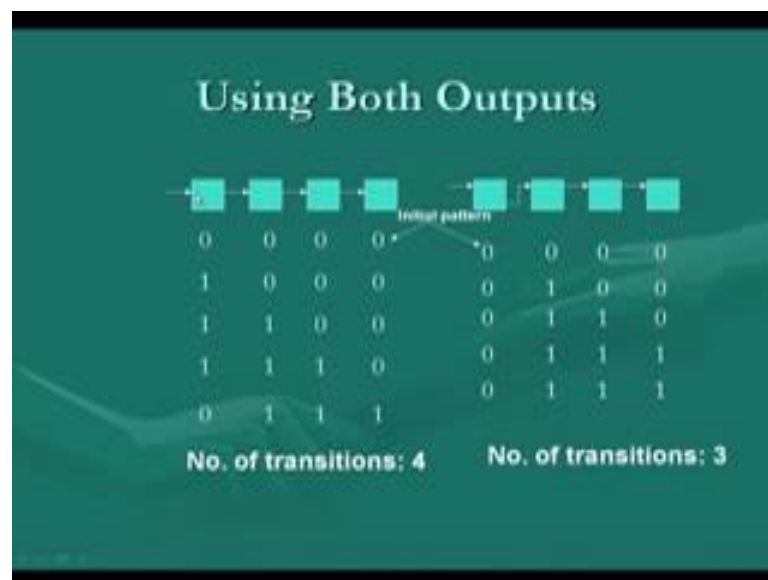
(Refer Slide Time: 01:45)



Another class of techniques by which we can reduce the scan power is known as scan chain transformation technique. So, one possibility is the scan flip-flop reordering that we have already seen like that scan cell reordering basically. So, we change the scan we can do scan flip-flop reordering. Then we can traditionally these scan chains are made of D flip-flop, where this Q output of previously flip-flop is connected to the D input of the next flip-flop, but what about the Q bar input. So, Q bar input can also be used. So, if we do that maybe we can reduce some transitions. Sometimes we can change the type of flip-flop that we are going to use in the scan chain.

So, instead of you normally we are suggesting D flip-flop, but we can go for some other types of flip-flops as well like say T flip-flop, etcetera. Sometimes we modify the scan chain by putting some additional gates into the circuit; and may most of the time when XOR gate is put, because XOR gate has the possibility of doing complementing the input and that way XOR gates are used in many cases. And we have got multiple scan chain based designs, so that is also there in many situations.

(Refer Slide Time: 03:02)



So, we will start with an example when we are using both the outputs of flip-flop. So, this is the first - the left side, so this is the standard scan chain where the D input of a flip-flop is connected from the Q output of the previous flip-flops conforming the scan

chain. So, if the final pattern is 0 1 1 1, so this is what we want to achieve then we have to do the shifting like this. Assuming that initially the flip-flops are all 0, then this 1 has to be shifted then again this 1, then again this 1, then this 0, as a result finally you get this thing. So, number of transitions, so here I have one transition, 2 transitions, 3 transportation, 4 transition.

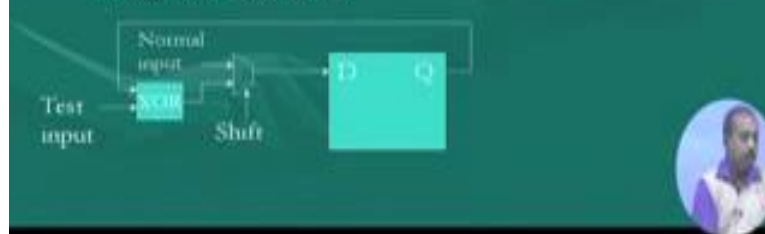
On the other hand, if suppose for the second flip-flop, the D input is connected from the Q bar output of the previous flip-flop. Now, here in this case since this Q output was 0, Q bar was 1. So, when we are shifting we are applying a shift pulse then this one that we have at this point get shifted to the D to this flip-flop as a result this becomes 1. So, this way the pattern that you need to shift will be different of the input that you have to apply there will be different.

For example, in this case we have to always keep it at 0, so that finally, we get 0 1 1 1 as the content of the scan chain. But number of transition gets reduced, this the first cell does not see any transition, this cell is one transition, one transition and one transition total 3 transitions can be obtained. So, this way by modifying some of the D cell some of the flip-flops from Q to Q bar connection, so we can go for scan transition reduction. So, this may be one possibility of reducing the scan transition.

(Refer Slide Time: 04:55)

Using T and D Flip-flops

- D flip-flop can be changed to T flip-flop during scan-in operation
- To capture the response, it remains D flip-flop
- Response part will get modified while being transmitted through the scan chain



Another possibility is by using a mix of flip-flops. So, D type and T type flip-flop. So, in fact, the traditionally we are using D flip-flop for this scan chain formation, but we know that there are many other types of flip-flop and T flip-flop is one such thing. So, for normal operation of the circuit, so the structure is modified like this. For the normal operation of the circuit, so these shift pulses, so the shift is this scan enable or shift, so that is equal to 0. So, the normal input goes to this multiplexer, it comes to this D flip-flop, so there is no problem. But for the scan mode, this input will come to this XOR gate. So, what we are doing this Q output is XORed with the next input, so that this becomes this entire structure XOR followed by D, it behaves like a T flip-flop. So, the flip-flop is not changed. So, flip-flop type is not changed. So, for the normal circuit operation, there is no problem that still remains as a D flip-flop, but in the test mode this flip-flops are converted to T flip-flop.

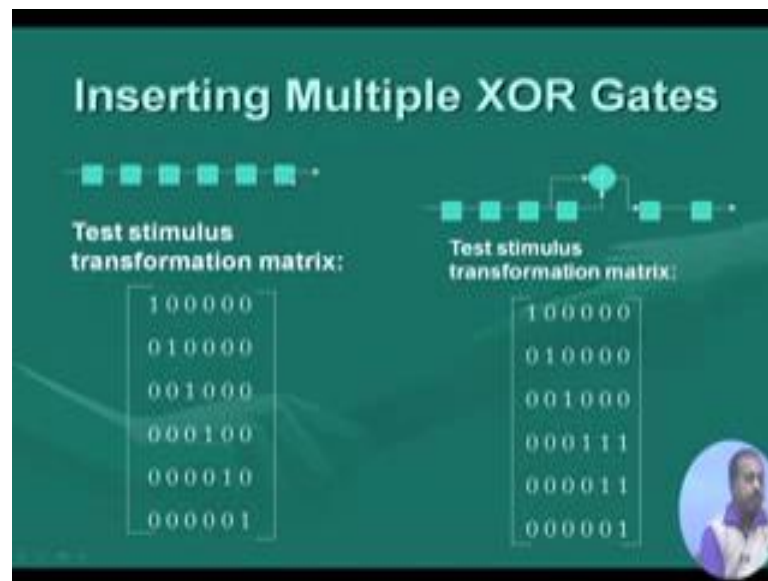
Now, it may be the case that this D flip-flop, it may be the case that modifying all D flip-flop to T may not be beneficial, maybe for the best result we need to modify only some of them. Of course, there is a catch because some of the flip-flops are now modified to T flip-flop, so the response part will also change, because when this input is applied to T flip-flop it behaves differently compared to a D flip-flop. So, whatever response the ATPG has told us, these are the responses, so that will not be valid. So, we have to re-compute them based on the scan chain architecture. So, again another fault simulation has to be done. So, those are there, but if it is beneficial, we can modify some of the D flip-flop to T flip-flop.

(Refer Slide Time: 06:54)

D-T Flip-flop - example							
D	D	D	D	D	T	T	T
0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0
1	0	1	0	0	0	1	1
0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1
No. of transitions: 9				No. of transitions: 6			

So, this is typical example. Like say if I have got all the flip-flops as D flip-flops may then to get this final pattern 0 1 0 1 onto the scan chain, we have to apply shift in the bits like this, and that creates a total of 9 transitions. On the other hand, suppose we modify these 3 flip-flops from D they are modified to T. So, what will happen, so this will be doing transitions the transitions will come like this and if you complete the number of transitions here there are 6 transitions. So, number of transitions is less. So, this gives rise to a better way of getting that power deduction. So, we can get less power, because the transitions are less, so circuit is also as you expected to generate less transition.

(Refer Slide Time: 07:42)



Also another popular strategy is to include this some XOR gate into the scan chain itself. Like you see in normal case what is happening is these D inputs are feeding this successive stages, and it is going like this the scan chain is going like this. Now, if we put a XOR gate, so that this input that we feed to this stage is a sum of these 2, so the sum of this input and this input. So, in normal matrix, so test stimulus matrix, so it is like this. So, whatever be the inputs, so that will get modified there. But in this case, so this matrix if you multiply by the given test stimulus, so you can find out what is the new stimulus that you should apply for getting the particular pattern onto the set. Now, this can reduce the transition though it is not very obvious that whether it is always reduce transitions or not.

(Refer Slide Time: 08:43)

Multiple XORs (contd.)

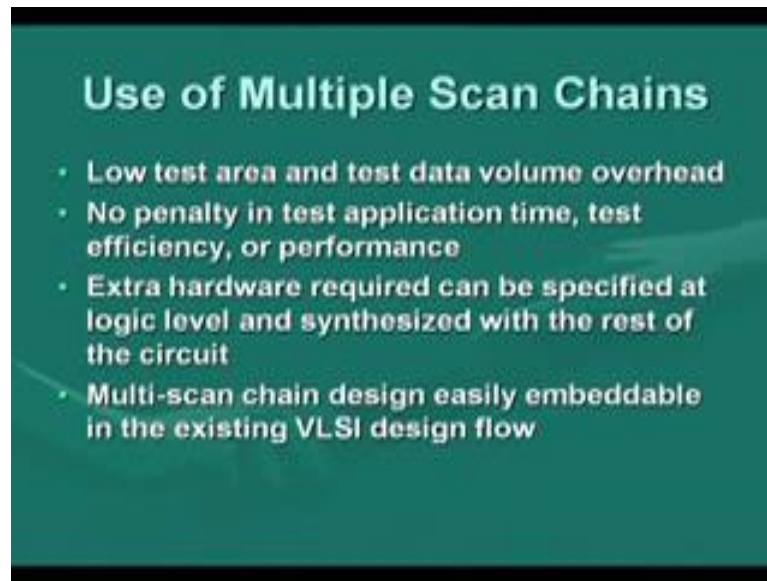
- Reduces number of transitions in the overall test set.
- For example,

<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table> 14 transitions	1	0	0	1	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	×	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	1	=	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> 10 transitions	1	0	0	1	1	1	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1	1	1	1
1	0	0	1	0	0																																																																																			
1	0	0	1	0	1																																																																																			
1	0	1	1	0	1																																																																																			
1	0	1	1	0	0																																																																																			
1	0	0	0	0	0																																																																																			
0	1	0	0	0	0																																																																																			
0	0	1	0	0	0																																																																																			
0	0	0	1	1	1																																																																																			
0	0	0	0	1	1																																																																																			
0	0	0	0	0	1																																																																																			
1	0	0	1	1	1																																																																																			
1	0	0	1	1	0																																																																																			
1	0	1	1	1	0																																																																																			
1	0	1	1	1	1																																																																																			

Caution: These are not weighted transitions

But this can be this can happen. Like say it is original test pattern set may have 14 transitions, so when you are post multiplying it by this, so we are getting 10 transitions of course, this calculation is this is not weighted transition. So, this is just assuming that these patterns will be applied directly to the test set to the circuit. So, there are some 10 transitions in this case. Like here, I have got one transition, here I have got, so this way actually this is one transition this is one transition this in another this is another, so that way if the patterns are going sequentially through these scan chains, so it creates 10 transitions. However, these are not weighted transitions, this calculation 14 or 10, but the work that is reported in the literature, it is reported like this. So, of course, this weighted transition values will be different, so that needs to be computed.

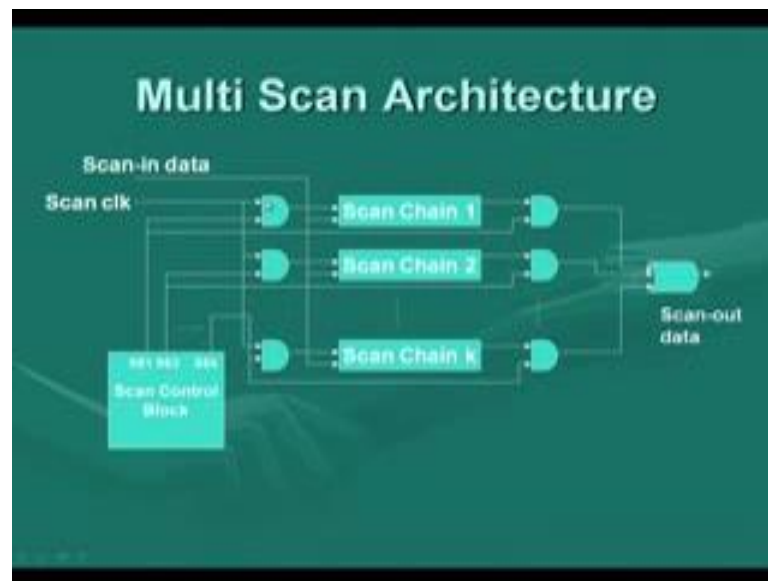
(Refer Slide Time: 09:37)



We can also use multiple scan chain. So, multiple scan chain, we have the concept we have already seen. So, because a single scan chain, since there are a large number of scan flip-flops. So, this maybe the scan chain may be large, so it will require large number of shifts scan shifts. So, we can take help of this multiple scan chains to reduce this. So, the test area will be reduced, because I do not need to have see this long inter connects connecting all these scan cells. And this test data volume overhead is also reduced in some cases because we can do because of this multiple scan chain. So, we can form test slices and many of these compression mechanism like Huffman code and that, so they rely on this scan slice, so that way we can have this test volume can be reduced by doing the compression.

So, test application time is can also get improved. So, there is no penalty, but rather it can improve because this shifting can be done faster; efficiency, performance they do not affect. However, so extra hardware is required and that can be specified at logical level synthesized with the rest of the circuit. So, we need some separate scan control hardware, so the multiple scan chains are there. So, scan control hardware is there, but of course, we can we can do in the synthesis, so that they get some of some part of it get merged with the design itself; as a result this overall area requirement may not be that high. And they are embedded in VLSI design for most of the cad to allow it that we have seen.

(Refer Slide Time: 11:14)



So, this is the typical structure of multiple scan chains that we seen here. What is happening is that this scan control block, so it actually selects one of the scan chain and then this scan chain data is loaded into it. Now, it can be a broadcast scan or serial scan. So, for broadcast scan, so all these select inputs will be made equal to one. So, whatever scanning data is coming that will get loaded to all the scan chain or it may be a serial one. So, serial one that will be loaded one after the other, first this chain is selected, then this chain is selected, then this chain is selected etcetera. And for the scan out purpose I have to do it serially. So, there these scan; this AND gate has to be enabled as a result this chain will come, this chain output will come or this will come or that will come. So, of course, there are many other component that can be there, they are like scan chain masking and all that, so that is not shown here.

(Refer Slide Time: 12:11)

Compatible, Incompatible, Independent scan cells

- To partition the scan cells into multiple ones, they are classified as:
 - Compatible
 - Incompatible
 - Self-incompatible
 - Independent

Now, when we are doing this multiple scan chain formation, we need to partition that scan cells for doing this partitioning of scan cells in a multiple chain; we first classify them into compatible, incompatible, and self-incompatible and independent, these 4 classes.

(Refer Slide Time: 12:28)

Example

- S_0 and S_1 are *incompatible* as different values are needed at x_0 to suppress spurious transitions originating from them
- S_2 and S_3 are *compatible* as $x_1=0$ suppresses spurious transitions from both of them
- Similarly, S_4 and S_5 are *compatible*
- Scan chains: $\{S_0, S_1, S_2, S_3\}, \{S_4, S_5\}$

The diagram illustrates a scan chain partitioning example. It shows three input lines, x_0 , x_1 , and x_2 , each connected to a pair of scan cells. x_0 is connected to S_0 and S_1 , which are connected to outputs Z_0 and Z_1 respectively. x_1 is connected to S_2 and S_3 , which are connected to outputs Z_2 and Z_3 respectively. x_2 is connected to S_4 and S_5 , which are connected to outputs Z_4 and Z_5 respectively. A 'Selection logic' block is shown with a 'Scan out' output.

So, compatible S 0 and S 1 are said to be incompatible as different values are needed at x 0 to suppress 4ier transitions originating from there. So, this S 0 is feeding this AND gate and this S 1 is feeding this OR gate. So, if the situation is like this now if you are having a control to so that we can mask of this scan cell result going to the circuit. Now since this is AND then this is OR, so I cannot use a single value of x naught to suppress both the transitions at z 0 and z 1, so one of them will cannot be suppressed by setting z 0, so that way this S 0 and S 1, they are incompatible flip-flops.

Similarly, S 2 and S 3 are compatible. S 2 and S 3 are compatible because by setting this x 1 to 0, I can suppress both z 2 and z 3 transition in both z 2 and z 3. Similarly, S 4 and S 5 they are compatible. So, scan chains that will make I should make it with the compatible set, so S 0, S 2, S 3, so they can be made into one cell one set; S 0, S 2 and S 3 they could be made into one cell - chain; and S 1, S 4, S 5, they can be made another chain. So, we divide it in such a fashion that by setting a single mask. So, I can suppress all the transitions that are occurring from that chain.

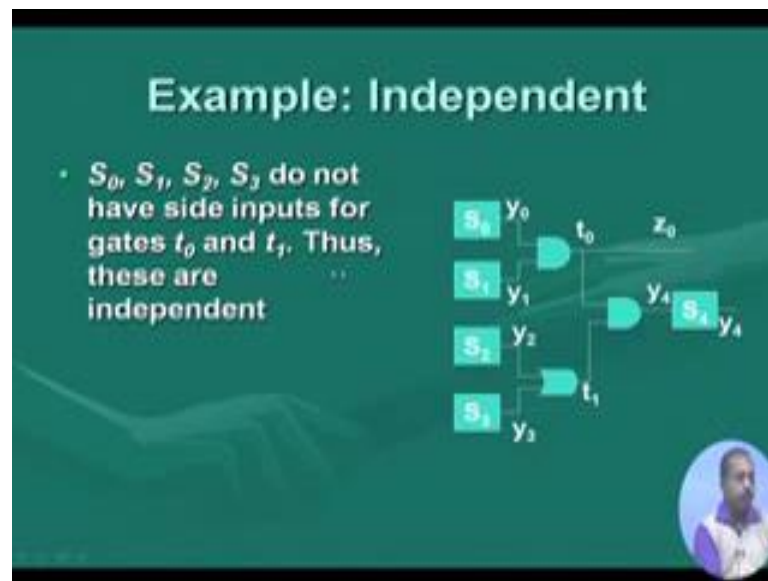
So, when I have got this S 0, S 2, S 3 into a into a state, so we can S 0, S 2, S 3 in a set, so we can control them by setting a x 0 and x 1 values to a particular values, so that they will not transit. Similarly, these S 1, S 4 and S 5 being in one set, so I can save this x 0 and x 2 to some value, so that 4ier transitions do not go. So, there is no conflict in the setting of x 0, x 1 and x 2. So, from power minimization point of view, so we can do this type of grouping though your scan chain this interconnect length requirement may require, this scan stitching in a different order.

(Refer Slide Time: 14:42)



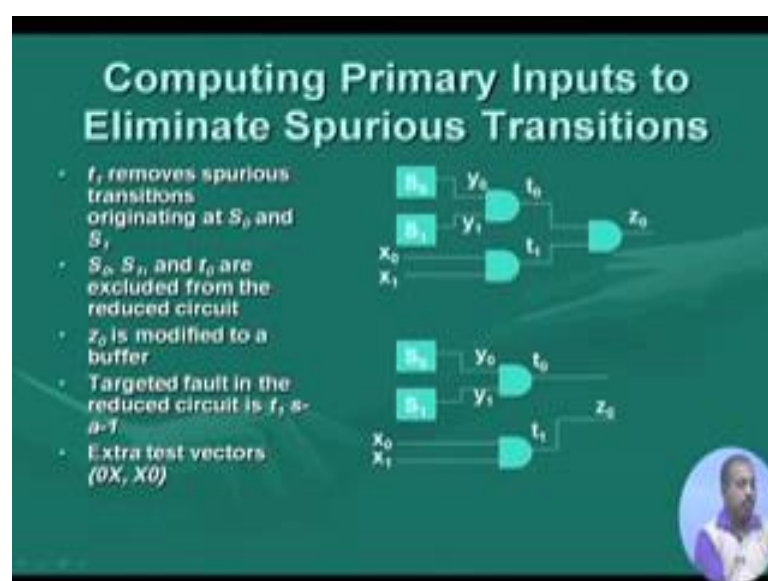
Other, the self-incompatible, so self-incompatible is like this. So, S_0 is a self-incompatible, because any transition for any setting of this x_0 , the transition proceeds either via t_0 or via t_1 because x_0 is 0 or 1; if it is 0, this y_0 will go to t_0 ; x_0 is 1 then it will go to t_1 . So, that way this S_0 becomes self-incompatible. So, it cannot be controlled anyway. So, they can be put into any arbitrary group because they are not compatible with itself.

(Refer Slide Time: 15:19)



And there are independent cells like S_0, S_1, S_2, S_3 they do not have any side inputs for gates t_0 and t_1 . So, naturally I cannot have any control over them. So, it depends explicitly on the value that S_0 and S_1 . So, this S_0, S_1, S_2, S_3 they are all independent.

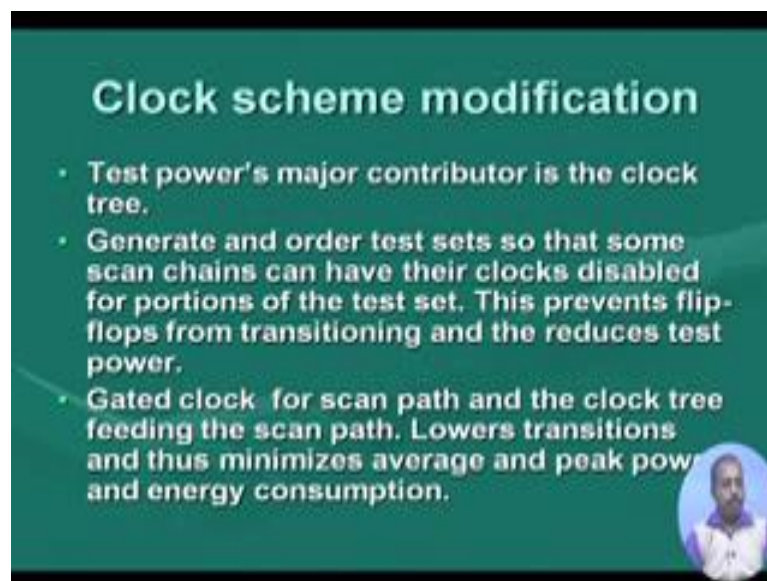
(Refer Slide Time: 15:42)



So, primary in; so we can add some primary compute some primary inputs to eliminate spurious transitions, so how can I do this. So, t 1 removes the spurious transition originating at S 0 and S 1. So, t 1 if I can put this t 1 to some value then it can avoid transitions going to z naught. Similarly, is S 0, S 1, t naught are excluded from the reduced circuit. So, we do not consider this S 0, S 1 and T 1, so for this spurious reduction, because t 1 can take care of this S 0 and S 1.


So, z 0 is modified to a buffer, so z 0 is just as if this x 0, x 1 is coming and it is going there. And targeted fault in the reduced circuit is t 1 stack at 1. So, if I can generate this t 1 stack at one that for that purpose, so this is the I have to set this x 0, x 1 to such a values, so that I get a 0 at this point. Because if I get a 0 at this point, this one get will this t 0 will get suppressed. So, this t 1 so should be stack at 0 stack at 1. So, that is the; so fault generator a TPG can be used for this purpose. So, we have to have this test vectors generated like 0 X and X 0 either of X 0 or X 1 will be set to 0. So, over a TPG can help us in telling the setting of these extra inputs, so that this z naught not does not see any transition when S 0, S 1 they are shifting. So, this way we can go eliminate the spurious transitions from of scan chain reaching the circuit.

(Refer Slide Time: 17:26)



Clock scheme modification

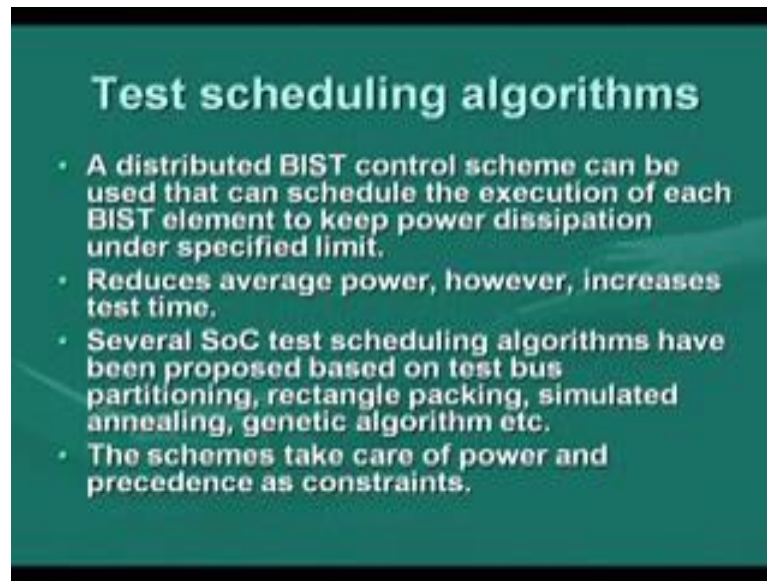
- Test power's major contributor is the clock tree.
- Generate and order test sets so that some scan chains can have their clocks disabled for portions of the test set. This prevents flip-flops from transitioning and thus reduces test power.
- Gated clock for scan path and the clock tree feeding the scan path. Lowers transitions and thus minimizes average and peak power and energy consumption.



There are some clock scheme modification techniques; so this test powers another source major source of power consumption is the clock tree. So, clock distribution network, so that goes through goes to the entire the chip; and when it goes to the entire chip, so it affects the almost all the components that we have in the circuit because almost all parts of the circuit, they will require the clock signal. So, when the transitions occur, so due to this long wire of this clock, so it will consume power plus it will create transition in all the most of the parts in the circuit. So, what we do is that we generate an ordered test sets, so that some scan chains can have their clocks disabled for portions of the test set. So, the scan chain for which we do not want to apply the shifting process, so we can stop their clock signal.

So, the flip-flops will not transit as a result it will reduce test process. If you have a multiple scan chains then the portions of the circuit that portions those scan chains which are not shifting the patterns or responses, so we can stop their clock, so that this clock is not there. So, gated clock for scan path and clock tree feeding the scan path. So, this is the gated clock can be used that is what I was telling. It lowers transitions and minimizes average and peak power, and also the energy consumption. So, this average and peak power minimization, so this will save the circuit from getting damaged; and this the peak power reduction will ensure that it is not crossing the power limit of the circuit of the circuit design, so it will protect the circuit. And this average power minimization, so this will improve the energy efficiency, so battery life will get extended.

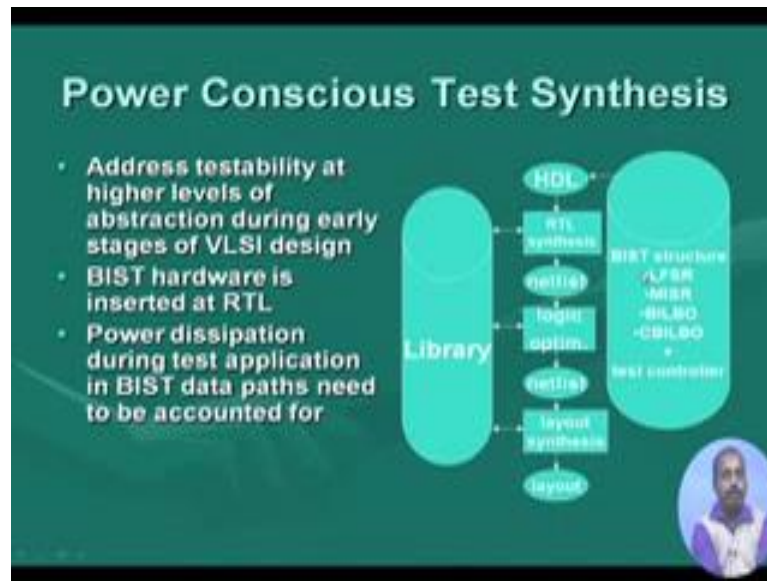
(Refer Slide Time: 19:19)



Another possibility of power reduction is via the test scheduling algorithms. So, we can have a distributed BIST control scheme that can be used which will schedule the execution of each BIST component BIST element to keep power reduction, the power dissipation under some specified limit. So if you have got several circuit modules they are BITSED, now I can have a control BIST control mechanism by which we do not enable all these BISTs simultaneously. So, they are done in some proper sequence as a result it will reduce power. So, this will reduce average power, because all of them are not tested simultaneously, but test time will increase.

There are several associate test scheduling algorithm that will see later classes. So, for this test power reduction based on test bus partitioning, rectangle packing, simulated annealing, genetic algorithms, so there are many approaches for that. This schemes that take care of power and precedence as a constant, so there are several scheduling policies that takes cares of this power as a constant and also precedence like which part can be tested after which one. So, there may be some precedence constraint, so they are also taken care of in the test scheduling algorithms.

(Refer Slide Time: 20:39)



Power conscious test synthesis, what is done is that? So, this is the normal flow of any design, so HDL to RTL to net list to logic optimization, net list, layout and layout. So, what is to be done is that in the HD l itself we put the description of the BIST, if it is a BIST based design, so LFSR, MISR, BILBO, CBILBO, test controllers all these are added into the description of the system itself. And then as the system is getting synthesized, these test hardware parts also get synthesized. So, this addresses test ability at higher levels of abstraction during early stage of VLSI design. BIST hardware it will be inserted at the RTL level itself. And power dissipation during test application in BIST data path need to be accounted for. So, this power estimator it will also figured out that well in the BIST part is activated then what is the power requirement. So, this power dissipation is computed that way.

(Refer Slide Time: 21:34)

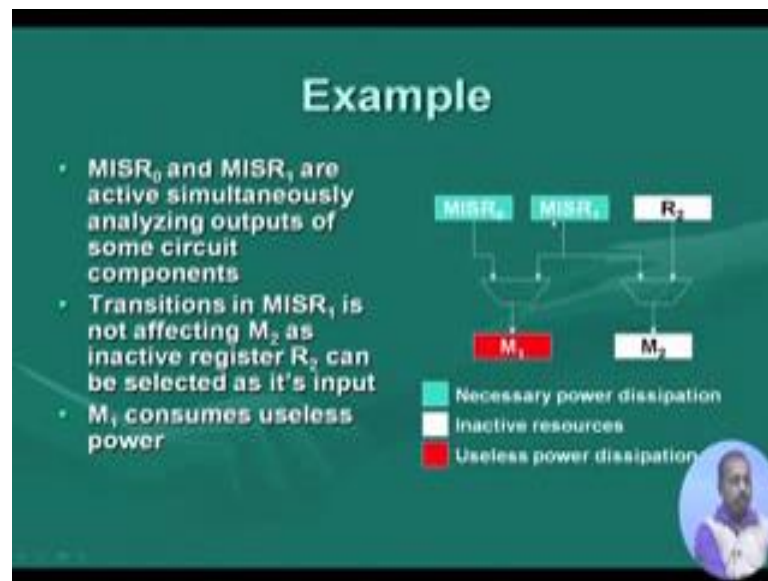


Power Dissipation in BIST Data Paths

- Existing power constrained test scheduling approaches are optimistic for BIST data paths, since,
 - Test scheduling assumes fixed amount of power dissipation associated with each test which is not the case for BIST data paths – useless power dissipation in untested modules
 - Test scheduling is performed on a fixed test resource allocation without considering the strong interrelation between test synthesis and test scheduling

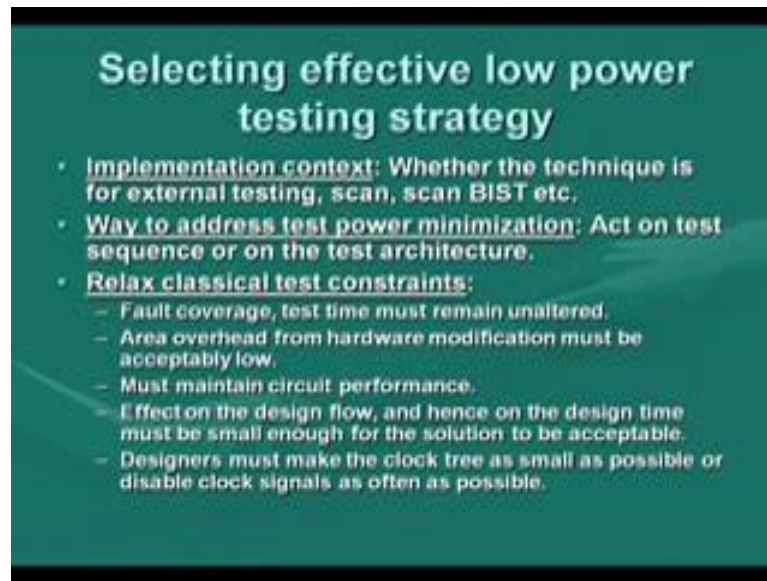
So, existing power constrained test data schedules approaches are optimistic for these data paths, because test scheduling assumes fixed number of a fixed amount of power dissipation associated with each test, which is not the case for BIST data paths unless power dissipation is unless; useless power is dissipated in the untested modules. Because all the parts are being under BIST sessions, so if all of them are getting power then maybe at present I am not testing a particular module, but that is also getting power. So, that way there is useless power dissipation. So, we have to do something, so that this useless power is reduced.

(Refer Slide Time: 22:18)



So, this is a typical example. Suppose, we are going to test the module M₂, so this MISR₀ and MISR₁, so they are actually generating the either the test pattern, they are analyzing the output of the circuit. So, they need to be activated. And this transition in MISR₁ is not affecting M₂ as inactive register R₂ can be selected as its input. So, what we do this is not MISR once result need not come to M₂. So, we put this R₂ in some value, so that this M₂ component does not do see any transition and this multiplexes in this select, so that is R₂ signal is send to M₂. So, MISR₁ thought it is connected to M₂ it does not affect this one. So, M₁ consumes useless power. So, M₁ has to be somehow taken care of. So, M₁ useless power dissipation has to be reduced somehow.

(Refer Slide Time: 23:26)



So, selective an effective power testing strategy becomes a challenge. So, there may be several angles from which we need to consider it. For example, the implementation context, like whether the technique is over external testing scan, scan BIST etcetera. So, depending on that we may have different policy that for example, for the BIST, we have seen that there are number of strategies for external testing based on ATE we have seen number of strategies. For scan environment and non-scan environment, so we have got this a test vector ordering problem changed, so that way we have got this cost function for this power minimization, so that will get affected. So, this way you see that there are a number of ways by which we can take you can see this, this external testing or this test environment that creates problem.

Second thing is that how do we address this test problem minimization? Whether we act on test sequence or we act on the test architecture. So, if you are allowed to modify the test architecture like say this scan chain modification using mixed D flip-flop or scan cell reordering or this putting a mask between the scan chain and the circuit. So, this way if we have some strategies by which we can change this test architecture, so that is one possibility. And if those things cannot be touched then we have to work on the test vectors, and there we have got the options like test vector reordering we have got the

option of completion, we have got the option of compaction, so these are the various avenues that we can try out.

Now, as a test engineer, so we would like to have some relaxation in the classical test constants like fault coverage, test time. So, fault coverage actually nobody will accept the situation where fault coverage get reduced. So, if some of the test pattern we see that they are consuming lot of power then possibly we can say that ok, we do not apply those patterns. So, if you do not apply those patterns, certain faults are not covered. So, fault coverage will get affected, but still it may be acceptable because the power consumption is less. So, we can have this thing this test type that the test time may change, but this fault coverage normally it is not traded off. Then area over head from hardware modification must be acceptably low. So, this is another burden put on the test engineer that whatever modification we are suggesting, the overhead should be low. But if we are allowed to take some more overhead it maybe we can do a better test hardware design and as a result we can come up with a better power mechanism.

So, this circuit performance must be maintained. So, we need to operate at certain frequency only. So, this that frequency cannot be spared. So, it cannot be traded off. So, if we if we can test at a lower frequency then power consumption will be low, but that will affect this delay test all that. Then effect on the design flow and hence on the design time must be small enough for the solution to be acceptable. This is another pressure because we do not want to put lot of time on testing, though it takes a good amount of time, but it is still not sufficient. So, we need more time, in the design flow, so if we are allowed to put some more time on testing then it will be better.

And designers must make the clock tree as small as possible or disable clock signals as often as possible. So, this is another desired requirement from the test engineer viewpoint. So, clock tree, if it is small then this clock skew will not occur, so delay test and all that, so that they will they can be done properly, and this power consumption will also be low due to this clock tree, so that way disable clock signals as often as possible. So, if we can disable the clock then this power consumption will be low, so that way this test engineer may have some relaxation. So, these classical test constraints if they are relaxed a bit, then this power consumption during testing can be reduced significantly.