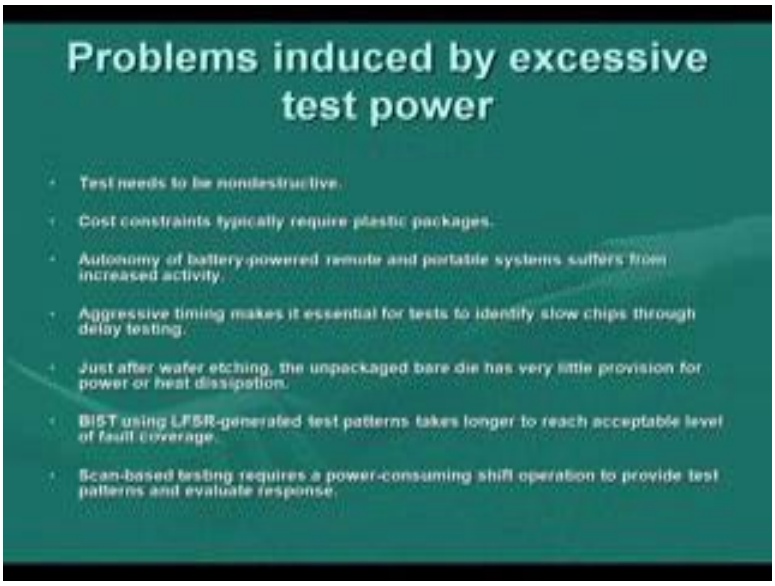
Digital VLSI Testing

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

Lecture- 33
Low Power Testing (Contd.)

The problems that get induced by excessive test power are the following. First of all the test needs to be non destructive.

(Refer Slide Time: 00:22)



Problems induced by excessive test power

- Test needs to be nondestructive.
- Cost constraints typically require plastic packages.
- Autonomy of battery-powered remote and portable systems suffers from increased activity.
- Aggressive timing makes it essential for tests to identify slow chips through delay testing.
- Just after wafer etching, the unpackaged bare die has very little provision for power or heat dissipation.
- BIST using LFSR-generated test patterns takes longer to reach acceptable level of fault coverage.
- Scan-based testing requires a power-consuming shift operation to provide test patterns and evaluate response.

So, when we are testing a chip. So, if the power consumption of the chip exceeds its power budget maybe the chip will be permanently damaged. So, this has to be avoided. So, if the excessive test power is high then there is a possibility that the chip will get destructed then cost constraints typically require plastic packages. So, we have seen the ad hoc, ad hoc solution we said that we can have some better packaging. So, if it is a metal casing then may be the heat will be dissipated fast. So, as a result the leakage power increase will not be that high, but due to cost constraints, we will have to go for plastic packaging and when you do that then the heat dispersed, heat will be taking the heat out of the system easily is a problem.

Then autonomy or battery operated remote and portable system suffers from increased activity now. So, this is the other issue that we have discussed that is battery operated systems remote systems, they cannot be accessed easily, but those systems may need periodic testing like we have got say this medical appliances fitted within the human body or say radio collar for this wild life and all that. So, they are remote system, but they need to be tested periodically. So, if the power consumption for testing is high then the battery will not last long. So, as a result this will be creating problem with the system performance.

Aggressive timing makes it is essential to test for test identify slow chips through relay testing. So, we have got this thing that this timing is a very important issue. So, this timing has to be met so, but, we are using the frequency operation. So, this delay test is a very important part that has come up. So, we need to identify the slow chips, if say if something is not able to operate at a high frequency the desired frequency then they need to be separated out. So, this aggressive timing so that will require that this slow chip has to be separated through delay testing, so delay testing has to be done at the operating frequency you cannot say that it will be met in a lower frequency.

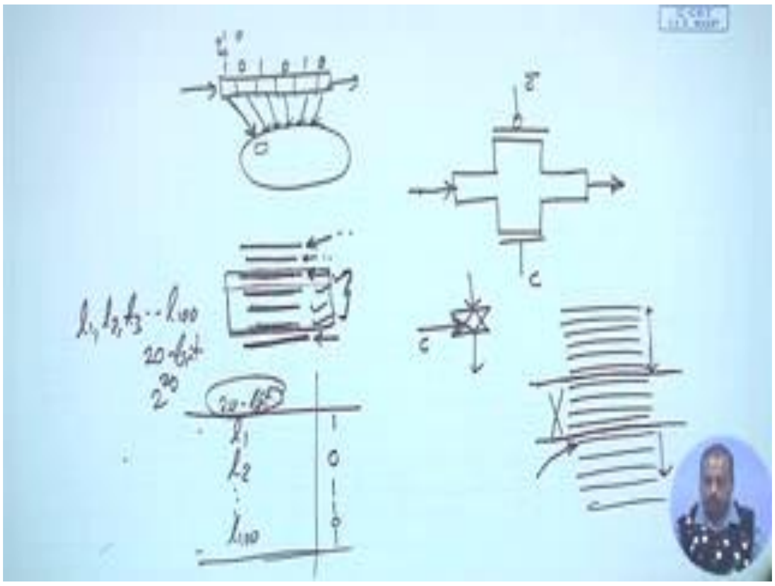
Then just after wafer etching the unpackaged bare die has very little provision for power or heat dissipation. So, this is the situation where the wafer has just been etched. So, it is the packaging has not yet been done. So, the connections are not very stable. So, as a result, they do not have provision for power or heat dissipation that it the individual die bare die they do not have this particular, this type of capabilities. So, we have to be careful and we can test the test power consumption at wafer level testing they should have very low power consumption.

Then if we look into the test techniques that we have seen so far like the BIST, in case of built in self test. So, we generally use some LFSR to generate large number of patterns and in fault simulation process what we can do - we can run the LFSR see the patterns that are getting generated and try to see what faults are getting covered by individual patterns. Then after running the LFSR for a good amount of time we reach we may reach the sufficient level of fault coverage that we are looking for and then we stop the LFSR. So, the LFSR has to be run for so many cycles. So, in the real operation whatever is the fault simulation result say for getting 95 percent coverage; it shows that the LFSR has to be run for 10000 cycles.

So, that in the actual run in the LFSR has to run for 10000 cycle, but out of this 10000 cycle, what has happened is that many times it generated patterns which are not detecting any new faults, but still since we cannot skip over the LFSRs output. So, we have to generate those patterns and to be; so that will have consuming power and those patterns are to be applied to the circuit so that will consume power. So, this is one issue with the BIST testing.

On the other hand, if we have got this scan based testing both for this ATE based approach and BIST based approach. So, there will be a shift operation.

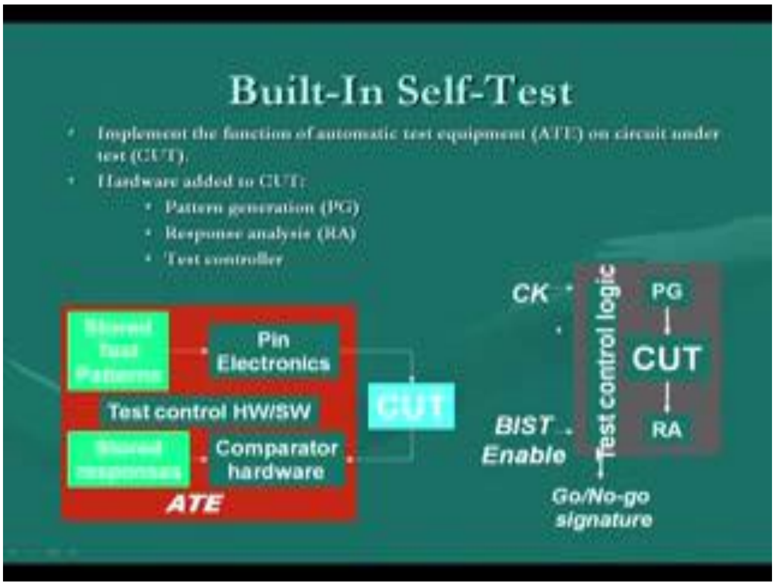
(Refer Slide Time: 04:53)



Now when this shifting is taking place, so what is happening is that the circuit that we have, here I have got that scan chain. So, through this scan chain the patterns are going, so they are actually feeding this scan scale outputs are feeding this circuit input. Now maybe I need a pattern to be loaded here as 1 0 1 0 1 0. So, for the shifting purpose first this cell will get a 0, first this cell will get a 0 then in the first cycle which corresponds to this 1; in the next cycle this cell will get a 1 and this cell will get that 1 transferred, it will go like this. So, you see, there is a transition at this particular bit from 0 to 1. So, that transition will be fed to the circuit, whatever gate is here there is a chance that this gate will do some transition due to this change in the input.

So, which is unnecessary, it is not required, but due to this scan shifting. So, circuit inputs will also change during the entire shift cycle this circuit inputs will also change so that will consume power which is unnecessary. So, scan based testing requires a power consuming shift operation to provide test patterns and evaluate response. So, these are the problems that we have with excessive test power. So, with that idea, we will see how this test power can be handled this test power consumption can be handled in internal testing techniques. So, internal testing techniques mean where we have got mainly this BIST based approach. So, this is the basic built in self test approach. So, this in BIST what happens is that it implements the function of this automatic test equipment on circuit under test then the circuit itself we have got this test pattern generated.

(Refer Slide Time: 06:20)



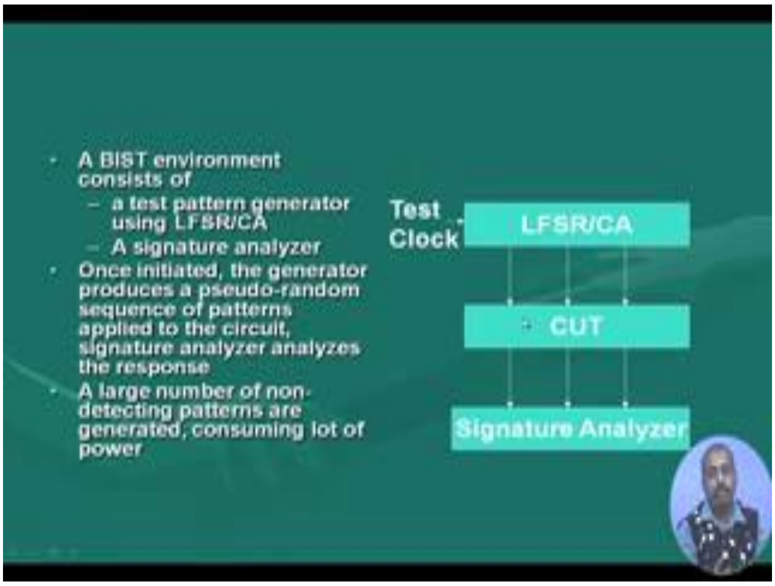
Built-In Self-Test

- Implement the function of automatic test equipment (ATE) on circuit under test (CUT).
- Hardware added to CUT:
    - Pattern generation (PG)
    - Response analysis (RA)
    - Test controller

So, this is the situation when we have got this ATE based approach where this stored test patterns are there and then there is this pin. So, there is a pin electronics which will tell like, which pattern will feed to which pin of this cut. So, that will go like this and then these circuits will the response will come. So, that will be compacted and it will be compared that will be compared with the stored response. So, that is the ATE, ATE based external testing.

For BIST what happens? We know that with the circuit itself we have got the pattern generator. So, from the outside when we give this BIST enable mode. So, put it in the BIST enable mode. So, this pattern generator will generate the patterns and that will be put into the circuit and the responses will be analyzed by this response analyzer and ultimately the test controller logic. So, it will say that whether the session was true or not. So, it is go no go type of signature. So, this is the idea of built in self test.

(Refer Slide Time: 07:36)



- A BIST environment consists of
  - a test pattern generator using LFSR/CA
  - A signature analyzer
- Once initiated, the generator produces a pseudo-random sequence of patterns applied to the circuit, signature analyzer analyzes the response
- A large number of non-detecting patterns are generated, consuming lot of power

Test Clock

LFSR/CA

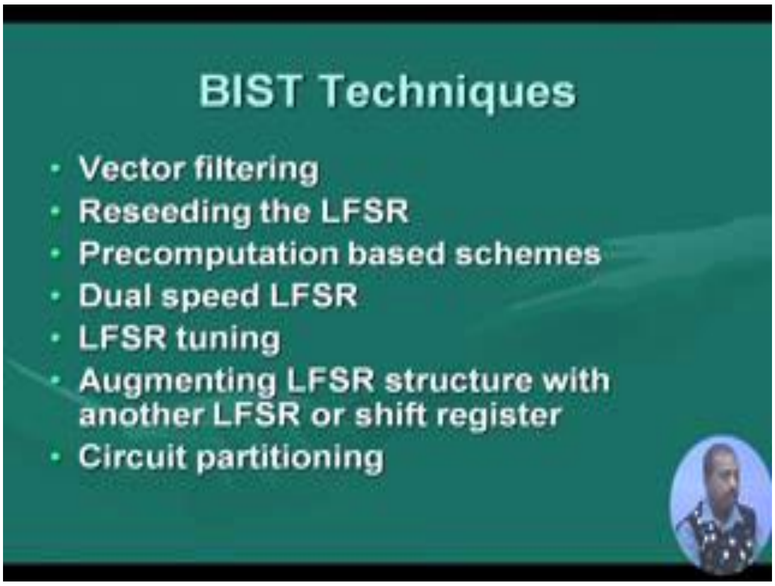
CUT

Signature Analyzer

Now in built in self test, so, what we have is we have got a test pattern generator which may be consisting of one linear feedback shift register or cellular automator or some counter based architecture whatever it is some test pattern generator is used at this point. Then we have got the circuit and there is a signature analyzer once we initiate this testing process the generator will produce a pseudo random sequence of patterns. So, those pseudo random patterns are generated and they are applied to the circuit. So, circuit will come up with the response and this response will be analyzed by this signature analyzer. So, that is the whole idea of this BIST based approach.

But the difficulty is that maybe we apply the first pattern, it detects number of faults. Apply the second pattern during fault simulation we have found that the applying the second pattern, but it does not detect any new faults. So, in case of ATE based approach what we can do is that we do not apply the second pattern to the circuit, but here that situation is that facility is not there because I cannot stop the LFSR from generating the second pattern. So, the second patter will be generated, it will be applied to the circuit and it will have no, it has got no contribution in improving the fault coverage then the third pattern will be coming. So, that maybe after generating the first pattern the 10th pattern that is generated by the LFSR is detecting some new faults. So, in between the patterns 2 to 9s, they are extraneous, they somehow they are actually causing the increase in power consumption.

(Refer Slide Time: 09:15)

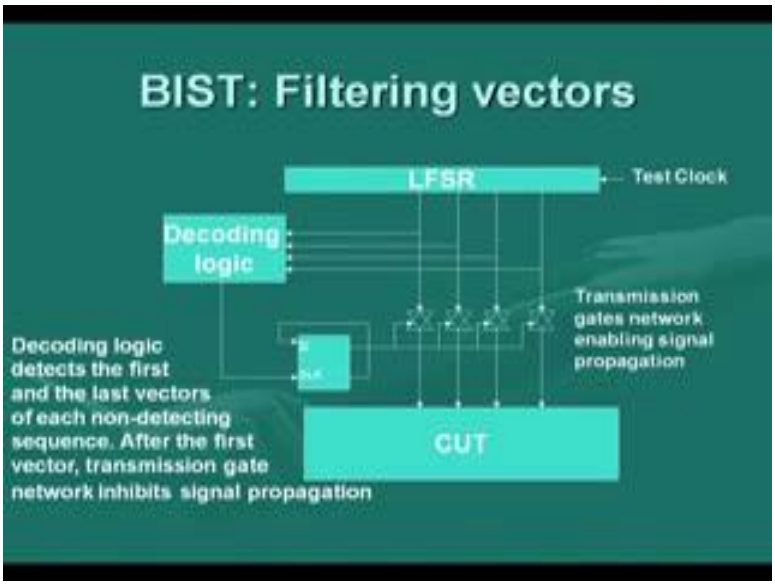


BIST Techniques

- Vector filtering
- Reseeding the LFSR
- Precomputation based schemes
- Dual speed LFSR
- LFSR tuning
- Augmenting LFSR structure with another LFSR or shift register
- Circuit partitioning

So, how can we control this extra power consumption? So, there are several techniques one is known as vector filtering we can do a reseeding of the LFSR, we can have some pre-computation based schemes, we can have dual speed LFSR we can do some tuning of LFSR then we can do some augmentation to the LFSR structure with LFSR and shift register or we can do a circuit partitioning. So, these are the various techniques that has been suggested in the literature for power reduction in BIST.

(Refer Slide Time: 09:49)



BIST: Filtering vectors

Decoding logic detects the first and the last vectors of each non-detecting sequence. After the first vector, transmission gate network inhibits signal propagation

So, vector filtering is like this. So, this test clock is applied to the LFSR. So, LFSR is continually generating the pattern. Now this decoding logic is it will detect the first and the last vectors of each non detecting sequence. So, suppose this LFSR is generating this sequence of patterns. So, this is out of that maybe we find that this is the first pattern which has detected this has detected some fault, this has also detected some new fault, but this pattern it does not detect any new fault and for that matter all these patterns. So, this, this, this and this they do not generate they do not detect any new fault only this pattern detect the next new fault.

So, somehow if I can stop these patterns to be applied to the circuit, this patterns to be applied to the circuit then at least the circuit power consumption will go though the LFSR will run an LFSR will consume power, but the circuit power consumption will go. So, that is what is done here. So, this LFSR output instead of being fed direct to the circuit. So, they are fed, they are passed through a transmission gate network transmission gate is an architecture where if this control is 1 then whatever is input here will come to the output and if this input is 0, so it will not come to the output. So, a transmission gate is manufactured like by 2 transistors in CMOS mode.

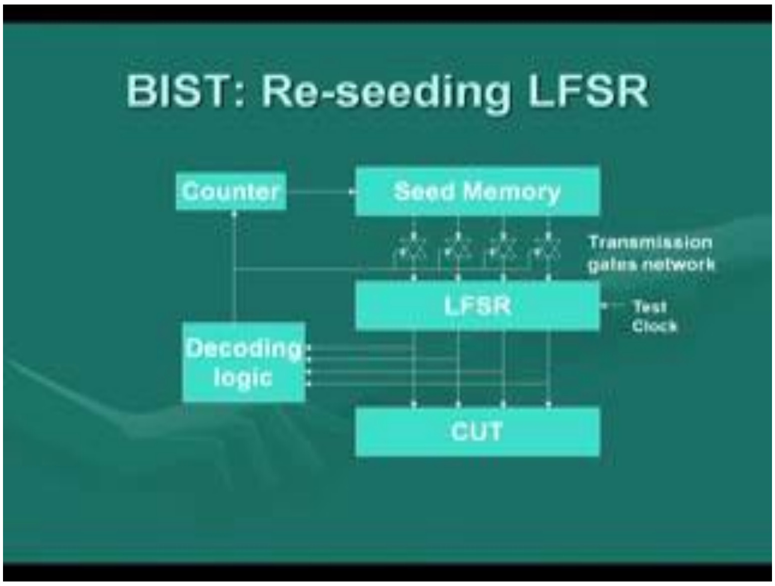
So, this is a p transistor this is an n transistor. So, this side we have got the input and this side we have got the output. Now this, there is control, so c and c bar. So, they are applied to this gate now if this c is equal to one then whatever value you give here will be available at this point similarly whatever value is here. So, that will be available on this point. So, that is why it is called a transmission gate, but if c is 0 then these 2 sides are not connected to each other. So, this is symbolically represented by this one. So, this is represented like this the transmission gate. So, this is the input this is the output and, here if I say this is the input and this is the output. So, this is like this and here I have got the controls the control c. So, this is the transmission gate. So, transmission gate network is there.

Now this decoding logic is designed in such a fashion that it will detect the first and the last vectors of each non detecting sequence after the first vector transmission gate network inhibits the signal propagation. So, initially this flip flop is set to 1 as a result this transmission gate network will be wrong. So, whatever pattern is generated by the LFSR they will come to the circuit, but as soon as this decoding logic is also continually looking into the patterns generated by the LFSR. So, when it sees that next few patterns will not detect any new LFSR; any new fault what it will do? It will give a clock pulse to this flip flop. So, since this q was equal to 1, q bar was equal to 0. So, this 0 will be loaded into the flip flop. So, this transmission, this output of the flip flop will become 0 as a result this transmission gate network will get disabled.

So this way this network will once network gets disabled. So, the patterns that are coming they will not be going to the circuit. So, decoding logic as we were looking into this example from the second pattern onwards, from the second pattern onwards, it has made this thing; it has made this transmission gate network inhibited. Now after it comes; after it comes to this particular pattern it finds that now this can detect new fault. So, the transmission network has to be turned on.

So, then this decoding logic will give another pulse to this clock line and since this q bar line was equal to 1. So, that line will be loaded into the D flip flop. So, this will become enabled. So, whatever is the LFSRs output. So, that will be coming to the circuit under test. So, after the first vector, transmission network, transmission gate inhibits the signal propagation and then after the last pattern again this will be restored. By this way by using this BIST filtering technique we can reduce the power consumption in the BIST session.

(Refer Slide Time: 14:34)



BIST: Re-seeding LFSR

Another possibility that we have is the BIST re-seeding approach. So, in case of this BIST session, what is happening is that we are generating, this LFSR is made to run in a in an autonomous mode it goes on generating patterns like this. Now out of that suppose after generate after say this particular pattern we see that, after say this particular pattern, it does not generate any new pattern, any new it does not generate pattern to detect new faults and we see that the next it occurs at this point only.

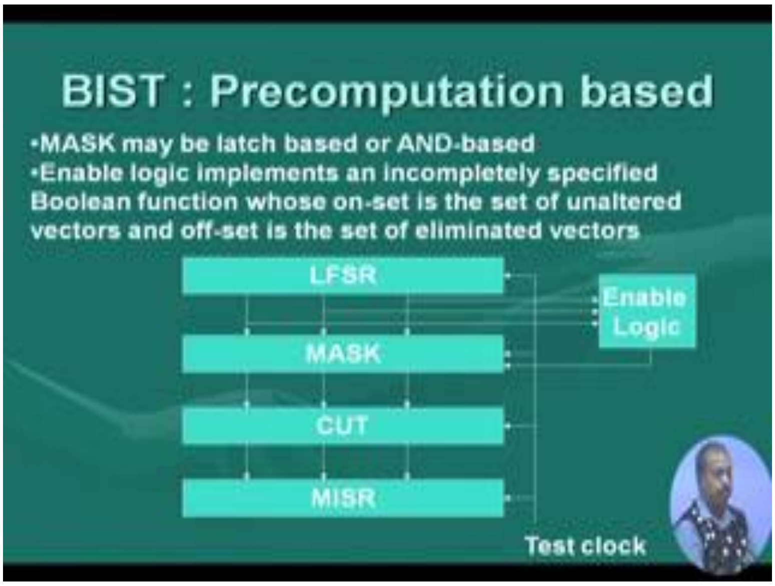
So in the previous case what we did in the BIST filtering case was that this filtered patterns they were not allowed to appear at the circuit input, but the LFSR was generating this patterns. Now in this reseeding approach, what we do we do not generate this patterns as well rather after this the LFSR is reseeded with this particular value. So, this particular value will be loaded into the LFSR. So, that it starts generating pattern from this point onward. So, in this way, whenever we find that this LFSR is not generating any useful pattern we change the seed value of the LFSR. So, that it generates pattern in the region where it can really use, where it can detect the faults.

So, here you see what is done this decoding logic. So, there is a counter and this counter is addressing the seed memory. So, this seed memory actually holds all the seed values that are needed for the entire test session. So, initially this counter value is 0 as a result the first seed value the location at memory location 0 that will be loaded into the LFSR.

So, this LFSR will, this decoding logic will put it as a 1 and this network transmission get networks true and the seed will be loaded into this LFSR. So, LFSR now operates and decoding logic continually monitor the output of the LFSR, after some time it finds that from this pattern it is non productive pattern. So, what it will do? It will stop this, it will inhibit this transmission gate for some time and it will tell this counter to it will give a up count pulse to its counter so that it will the counter value will become 1.

So, from the seed memory the next seed value that will now be loaded, now this decoding logic will enable this seed this transmission gate and the value will be loaded into the LFSR and then again the LFSR starts with the new pattern and it continues in there under similar fashion in a similar fashion it goes on generating the patterns. So, this way you see that this re-seeding LFSR, it is avoiding the generation of extra patterns. So, in case of this vector filtering approach we could filter the vector, but the test time was not reduced. So, BIST the LFSR was made to go through those extraneous patterns and generate the patterns there and generate the patterns and see, but in case of this reseeding based approach. So, it is not doing that, it is not generating those patterns at all. So, this reseeding LFSR is helpful from that side.

(Refer Slide Time: 18:03)



BIST : Precomputation based

- MASK may be latch based or AND-based
- Enable logic implements an incompletely specified Boolean function whose on-set is the set of unaltered vectors and off-set is the set of eliminated vectors

Another approach is known as pre-computation based approach. So, this pre-computation based approach what it does we have the LFSR. So, LFSR is continually generating the pattern, but there is a mask. So, this mask will tell whether the pattern that is generated by the LFSR will go to the circuit under test or not, and this enable logic, this enable logic that is generated so that will control this mask and that will either pass this pattern to the circuit or it will not do so.

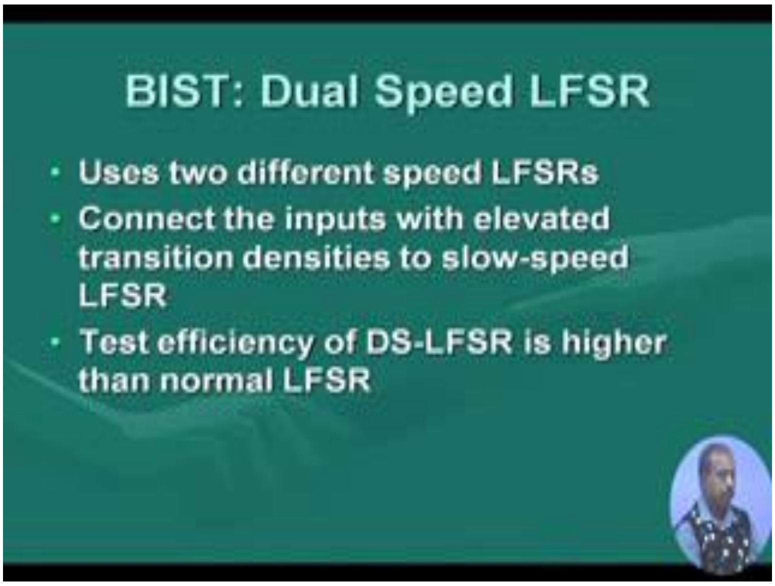
So it is similar to the filtering approach, but the mask can be a latch based mask or an AND gate based mask. So, o if it is and based mask then what will happen this whenever this enable signal is 0 then the output of this mask is all 0. So, circuit will see all 0 output all zero has coming as an input. So, it is expected that this circuit power consumption will be low because it is all 0. Another possibility is that we can make it a latch based mask. So, in case of latch based mask what will happen? This latches they will hold the last pattern that was generated by the LFSR and it is that was applied to the circuit.

Now even if the LFSR generates new patterns this mask will not allow that new pattern to come this side, circuit will see the previous pattern, the last pattern the last productive pattern, it will see that one only. So, as a result this there is no change in the input to the circuit. So, the circuit does not make any transition, there is no power consumption in this region. So, this enable logic, this will implement an incompletely specified Boolean function whose on set is the set of unaltered vector and off set is the set of eliminated vectors.

So, this on set and off set, they are specified, but for other patterns it does not tell anything. So, since those patterns will never be generated by LFSR. So, this enable logic truth table, it need not consider those patterns. So, that way say if I say that if I have got say 20 bit LFSR then it can generate 2 power 20 pattern. Now when I am specifying this enable logic for some of the cases, so say the sequence generated by LFSR is l 1, l 2, l 3 etcetera and say l 100 suppose this 100 cases only.

Now for this l 1, l 2 up to l 100, I have to tell that you whether it has to be a masked or not may be the first one will be true, second one needs to be masked, the third one is true, fourth one is also true, similarly this l 100 is also true in between there may be some 0s. So, this is the enable logic. So, enable logic truth table will be like this. So, this is a 20 bit value. 20 bit value of l 1, l 2, l 100 and, out of that only these bits are to be specified the remaining bits are all do not care. So, since the LFSR is not going to generate those patterns in the 100 cycles, so we do not need to specify those patterns. So, this logic can be designed efficiently. So, if you have got large number of do not care in the combinational logic. So, it can be minimized significantly so that we can get a very compact enable logic and, so rest of the thing is same as this BIST filtering technique.

(Refer Slide Time: 21:35)

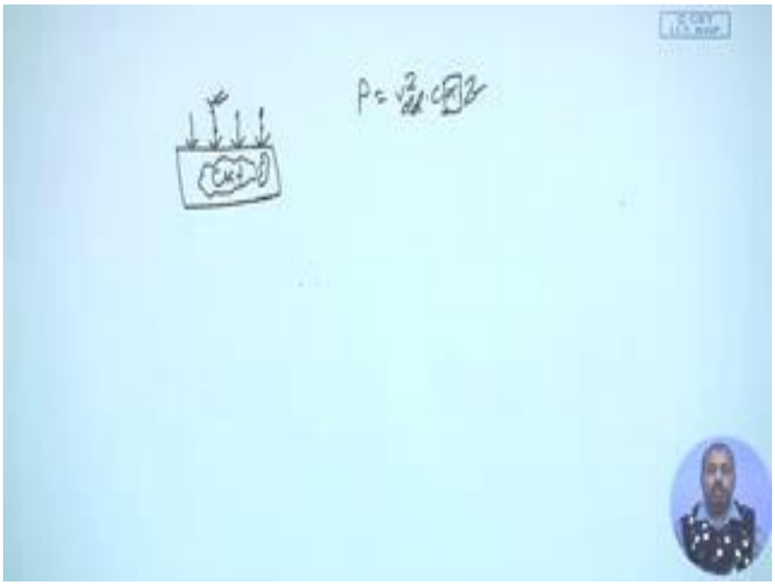


BIST: Dual Speed LFSR

- Uses two different speed LFSRs
- Connect the inputs with elevated transition densities to slow-speed LFSR
- Test efficiency of DS-LFSR is higher than normal LFSR

We can have dual speed LFSR. So, we have got 2 different speed LFSR. So, one LFSR is operating at a higher speed and another LFSR is operating at a lower speed.

(Refer Slide Time: 21:59)



So, one LFSR which is 2 inputs with elevated transmission densities to slow speed LFSR, so what happens is that if this is a circuit, it is not that all inputs of this circuit are equally going to change and they are going to affect the equal amount of transitions in the gate. So, normally if we, it may so happen that out of these four inputs we may see that these input if you change the value, it affects a larger region of the circuit whereas if you change this value it affects only a small part of the circuit. So, it is, while from that, from the power angle I will be happy if my test patterns they do not change this input much, but they change this input more. So, I am ready to have more number of troubles on this line, but I will have less troubles on this line. So, that is if the switching is high.

So, what is done is that in this dual speed LFSR. So, we have got two LFSR. So, thus inputs that have got elevated transition density that is they are going to affect more number of more amount of gates in the circuit. So, they will be connected to the slow speed LFSR. So, that slow speed LFSR means path you need time. So, it will see the less number of changes. So, it will be its transition will be low. So, it will see less number of transitions. So, that formula of the dynamic power consumption it was v d t square into c into alpha into f. So, this alpha factor will reduce, the switching activity on that particular line will reduce. So, as a result the power consumption will be less.

So, if elevated transition density inputs, they will be connected to slow speed LFSR and test efficiency of this DS-LFSR has been found to be higher than normal LFSR. So, that is another plus point that we have because there are two LFSRs. So, they are going to be helpful.

(Refer Slide Time: 23:58)

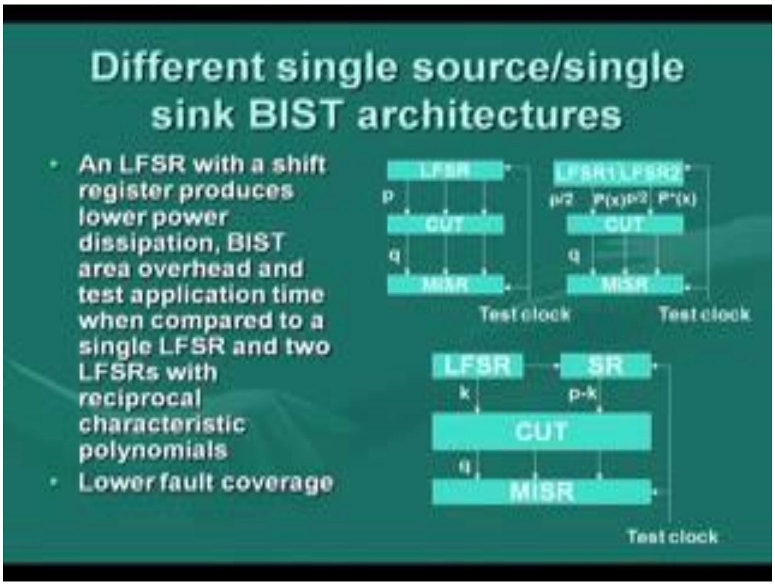
BIST: LFSR Tuning

- Parameters of LFSR:
  - Polynomial
  - Seed value
- Polynomial does not affect the power consumption significantly
- Seed selection plays a vital role in determining the power consumption

Another approach for reducing this power consumption in BIST environment is by LFSR tuning. So, in an LFSR design, there are 2 parameters that we can control one is the polynomial that defines the structure of the LFSR and the other one is the seed value. Now it has been seen that the polynomial does not affect the power consumption significantly because any polynomial you give, if it is most of the time will be using maximum length polynomials and it will generate all two power n minus 1 patterns and so that way it will continue.

So, it is very much unlikely that they will be giving different power consumption because of a very simple reason if you look into the individual bits of an LFSR. So, if it is the maximum length LFSR then all bits will have equal number of transitions in them. So, in terms of circuit power they often translate to same power value. So, this polynomial does not affect this power consumption effectively, but seed selection plays a vital role. So, which seed we allow which seed we put into the system, so that will be put into the LFSR that affects the power consumption because out of this 2 power n minus 1 patterns in the maximum length LFSR in its transition graph. So, putting different seeds will explore different parts of the cycle. So, as a result, we can, the switching activities for different parts are going to be different. So, if are applying only a 1000 patterns out of 2 power 20 we are applying only 1000s patterns. So, selecting those 1000 patterns from different regions naturally the power consumption will be less. So this way we can go for this LFSR tuning that can help us in reducing the power consumption.

(Refer Slide Time: 25:54)



Different single source/single sink BIST architectures

- An LFSR with a shift register produces lower power dissipation, BIST area overhead and test application time when compared to a single LFSR and two LFSRs with reciprocal characteristic polynomials
- Lower fault coverage

Another approach is difference single source, single sink based architectures can be possible like we can have an LFSR with a shift register. So, this is the normal LFSR. So, we have got the LFSR and the circuit and then it is signature analyzer, so the normal LFSR. So, what we can do is instead of having a single LFSR. So, we can have 2 LFSRs, LFSR 1 and LFSR 2 and if they are of length p by 2 original LFSR is of length p. So, this is p and this is p by 2 and p by 2 and it is done in such a fashion that the second LFSR the characteristic polynomial is the reciprocal of the first LFSR. So, in that case it has been seen that LFSR with a shift register like this. So, this is one configuration. Another configuration is like this we have got one LFSR followed by a shift register. So, in the phase shifting part we have seen that we can use this type of shift registers to generate that this phase shifted versions.

So, here also it is same things we have got k bit LFSR and p minus k bit shift register and that way this patterns are shifted and they are coming to this circuit under test and then it will be occurring. The general observation is like this that an LFSR with a shift register it produces lower power dissipation, BIST area over head and test application time when compared to a single LFSR and 2 LFSR with reciprocal characteristic polynomial. So, this structure is better compared to this structure as far as power consumption is concerned area over it is concerned. But the fault coverage is going to be less that is obvious because LFSR itself is of lesser length. So, the variety will be less, so the fault coverage is going to be less. We will continue in the next class.