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

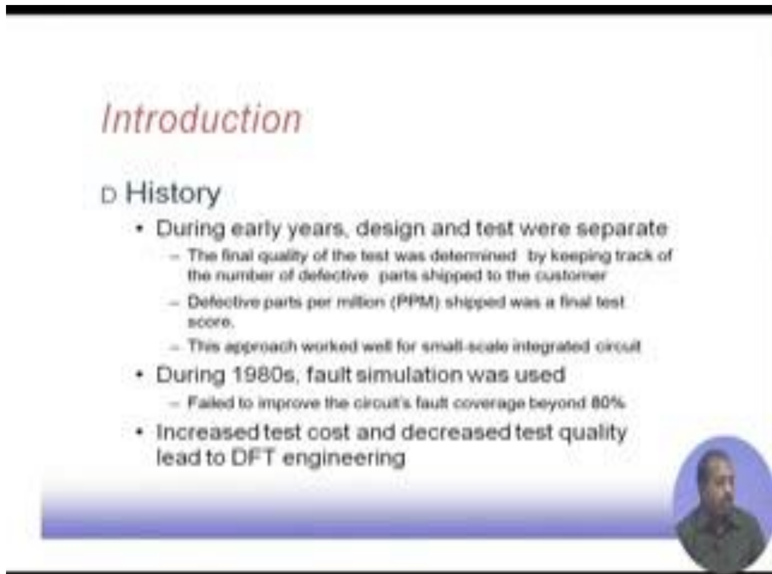
Lecture – 05
DFT

Next we will look into the topic design for testability, and the issues related with it.

(Refer Slide Time: 00:27)

So, as we have seen in our last lecture that design for testability is a very important part, as it enables a easy testing of the system. So, if you want to enhance the testability of a system so a circuit. So, we have to have some extra hardware put into it, so that this testing can be done efficiently. So, this is the contents first we will have a basic introduction then the testability analysis, then testability basics, scan cells designs, scan architecture and scan design rules, design flow and slowly will go to special purpose scans and RTL design etcetera.

(Refer Slide Time: 01:04)



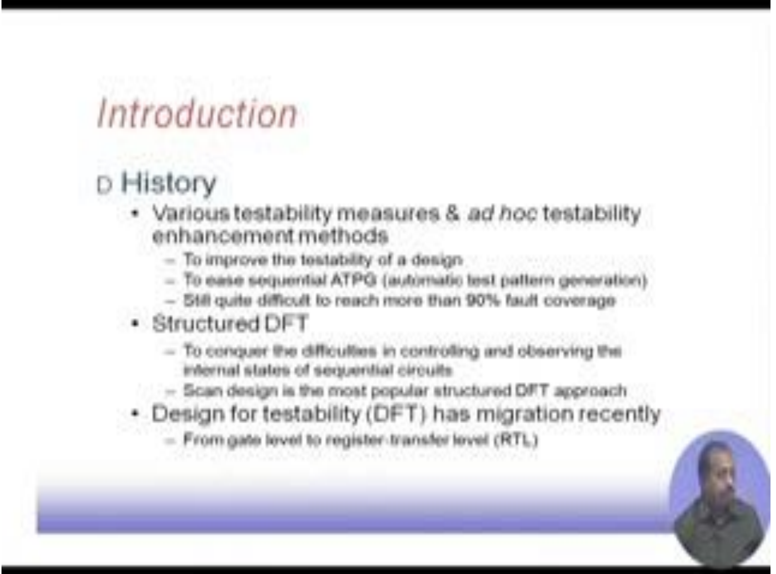
So, if you look back during early years design and test were separate. So, testing was not getting that much importance in the sense that the designer once they complete their design, and then they tell this is the design now you have the testing done on it.

So, you design your test vectors, so that this design can be tested; or when this chip comes up it can be tested. So, the final quality of test was determined by keeping track of number of defective parts shipped to the consumer to the customer. So, defective parts per million. So, since the test may not be very exhausted in that case. So, it may so happened that many of the good chips they are marked as bad, and many of the bad chips are marked as good. So, if it is bad chips marked as good then when it is shipped to the customer, then they will be detected on the field that they are not working properly; as a result they will get rejected. So, this defective parts per million that was a measure for doing this thing. So, for small scale design it is, because if I have got safe SSI design have has got less than 10 gates or so, so then it is easy because the design is simple. So, in most of the cases we can design this test vector even if the test design has been done before hand; then came fault simulation.

So, fault simulation is actually given the design, so we can try and the test vectors. So, we try to see how many faults are getting covered. So, that way we get some idea about the quality of the test vectors that are applied, but still by and we try to do some modification to the test vectors, so that this coverage improves, but getting coverage

more than 80 percentages was difficult. So, in our introductory classes we have seen that even with 90 percentage fault coverage, so there is a huge loss in number of defective parts that are send. So, 80 percentages is so poor a number. So, this increases the test cost and this decrease test quality. So, this become a concern like test cost is increasing and these quality of test is a decreasing. So, that leads to this design for testability are DFT engineering.

(Refer Slide Time: 03:30)



There are various testability measures and ad hoc test ability enhancement methods that have been used. So, for let we improve the testability of a design. As I said that for a design it maybe to complex, so we cannot access one particular line from primary input, we add a multiplexer and put it somewhere so that the line becomes controllable.

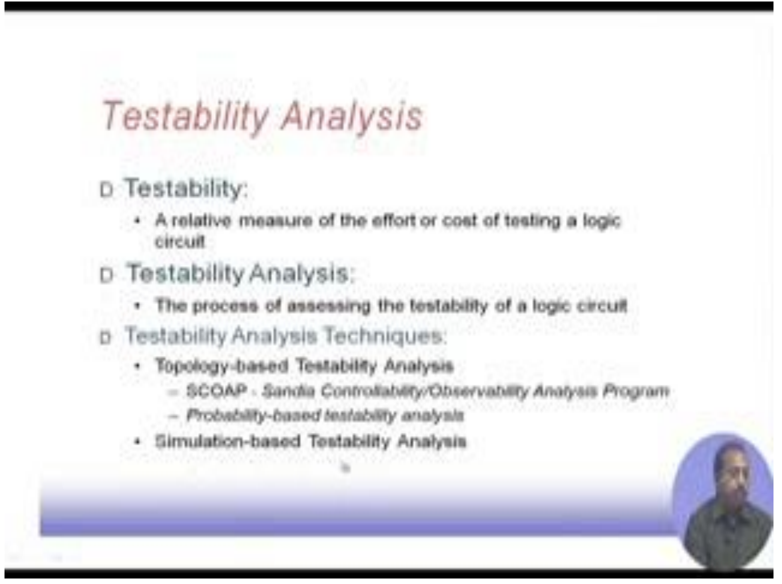
So, that way so that tries to improve the testability there is some ad hoc fashion. So, this is one problem; another problem that another issue was for sequential ATPG. So, if my circuits has got sequential elements like flip flops then there are some sequential ATPG algorithm that came up, that can so that generate these pattern for these sequential circuits. But still getting more than 90 percentage fault coverage was difficult; because of the same reason that many of the points may not be accessible, and the designers will not allow much modification to their design by introducing this extra multiplexers into the system because that will affect the critical path delay of the design. So, as a result so all these ad hoc multiplexer insertion may not be acceptable. So, that you give rise to

structure DFT. So, we want to conquer the difficulties in controlling and observing internal states of sequential flip flop, sequential circuits.

So, sequential circuit is a major concern because they are for testing a particular fault we need to puts some pattern to the internal flip flops, so that makes it more difficult. So, this putting some value to the internal flip flops becomes difficult. So, you want to observe and control the internal state. So, that becomes difficult. So, what came up is the scan design. So, scan design is the most popular structure DFT approach, because that will put all the flip flops on to a shift register, so that we can shift in the test patterns and shipped out the responses from them. And design for testability has migration; because what happened is that. So, even at this point so it was not sufficient.

So, it was observed that we should start the testing process test generation process from the same point at which we generate start the design process. So, from gate level to RTL level this testing process has shifted which has migrated, so that this design for testability measures can be incorporated easily into the system.

(Refer Slide Time: 06:07)



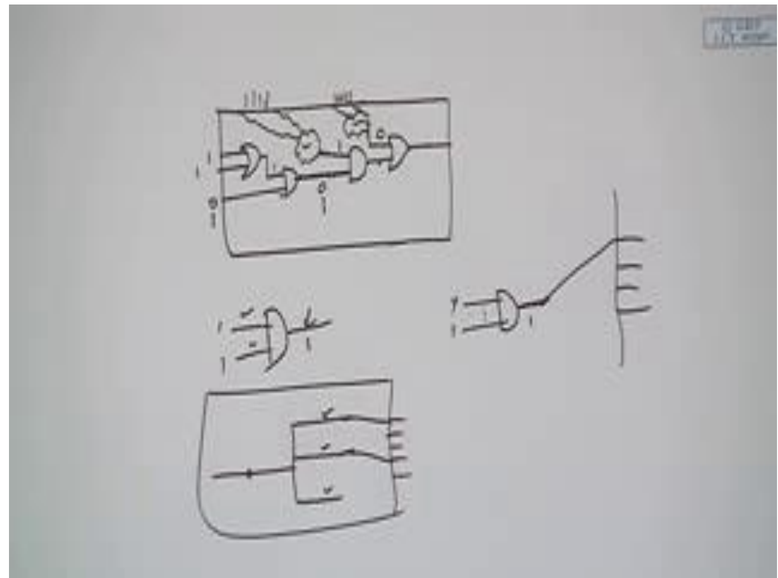
We will start with testability analysis. So, testability is a relative measure of the effort or cost of testing a logic circuit. So, how much effort we are putting, we need to put for testing logic circuits. So, testability analysis is process of accessing testability. So, there are various ways by which we can do this testability analysis; one is known as SCOAP approach which is Sandia controllability observability analysis program; there is another

one probability based testability analysis of which is more popular and of course, we have got simulation based analysis.

(Refer Slide Time: 06:46)

So, all these testability analysis technique so they try to measure two values; one is called controllability, another is called observability. So, controllability it reflects the difficulty of setting a signal line to a required logic value from primary input. So, what happens is that if I have got some circuit, then if I want to say put this particular line on to some logic value.

(Refer Slide Time: 07:07)



Now, I have to see this line is not directly accessible. So, this may be output of some gate AND gate and maybe this is getting there is an o r gate here which is feeding this AND gate. Now if this is the circuit and if I want that I want to put a value 0 here, there it is very simple that this particular primary input if I put a 0 then we are true. So this line will do this point will get a value 0 in a fault free condition.

On the other hand if I want that this line be set to 1 then I have to do many things, because I have to ensure that this line is one. So, I have to apply one here, plus I have to ensure that this line is also one; now for ensuring that this line is 1 I have to make either this line one or I have to make the other input of the OR gate o. So, these requirements are coming so, if it is deep inside. So, this line is only two level away from the primary input so, but still in that case you see that there is some complexity involved in the setting of that line. So, if it is deep inside the circuit then it becomes very difficult, it may be almost impossible to set some value for some circuit points.

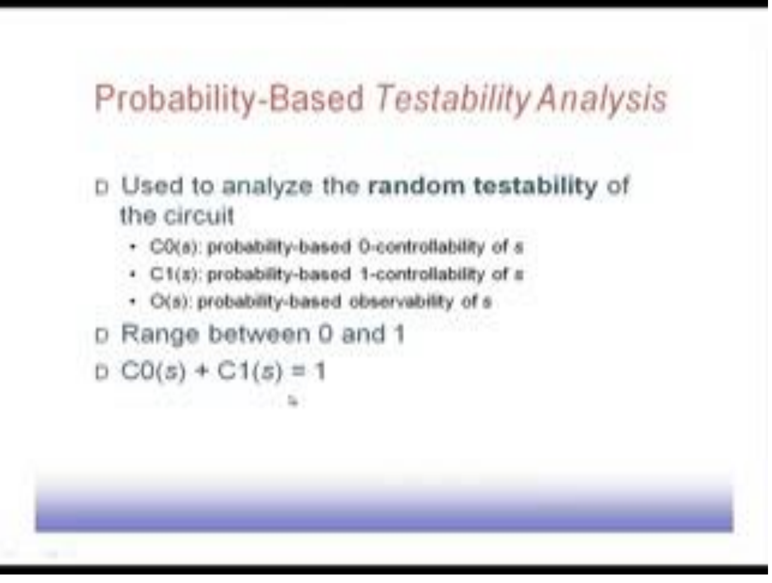
So, if that happens if we find that many of the points in the circuit they cannot be set to some desired value by playing with only some of the primary input. So, we say that the controllability is difficult for that I cannot control that point easily. Similarly if I want to say observe this value that we have been this line, then and if it goes through some gates like there is one AND gate here there is one OR gate here ok.

Now, you see this and get it can have some input from some other place. So, there is some logic here which actually feeds these AND gate, similarly there is some logic here with feed these OR gate. Now if you want to propagate this line to the output then first of all since this is an AND gate. So, I must somehow have this point to one because if this point is one, then only whatever be the value here will get propagated to this point; and since this is an OR gate. So, I must have a value 0 here, so that whatever is value coming here that is propagated to the output. So, if I can. So, again this individual circuit logic part that I have. So, they will have their own input and output set input set. So, whatever inputs are here, I should be able to set those inputs in such a fashion that at this point I get a one.

Similarly, whatever inputs here I must be able to set them at proper values. So, that I get is 0 here. So, this process may not be very simple. So, that depends on the number of input that we have for the circuit and for the complexity of this individual logic parts that

we have like say this part and this part, what is the complex complexity of this logic part. So, based on that only I will be able to decide like whether we can get this value communicated to the output or not. So, that is the observability. So, it reflects the difficulty of propagating the logic value of the signal line to the primary output. So, these are the two important measures that we have controllability of a point and observability of a point. So, when we have got a circuit with large number of lines for each of them we need to have this controllability and observability calculated.

(Refer Slide Time: 10:58)



In a probability based analysis. So, we used to analyze the random testability of the circuit. So, it is. So, C 0 is. So, this is called the probability based 0 controllability of s.

So, how what is the probability that I can put a 0 value at S. Similarly C 1 is the one controllability of S. So, what is the probability that I can put a one value at S? And O of s is the observability probability of s, what is the probability that I will be able to observe this value. So, range vales. So, this C 0, C 1and O, they range between 0 and 1 and this condition is true C 0 s plus C 1 is equal to 1. So, probability of, if it is difficult to control make a line 0, so, these 0 controllability will be high, and then this one will be low. So, 1 controllability will be low.

(Refer Slide Time: 11:57)

Probability-based controllability calculation rules

	0-controllability (Primary input, output, branch)	1-controllability (Primary input, output, branch)
Primary Input	p_0	$p_1 = 1 - p_0$
AND	1 – (output 1-controllability)	Π (input 1-controllabilities)
OR	Π (input 0-controllabilities)	1 – (output 0-controllability)
NOT	Input 1-controllability	Input 0-controllability
NAND	Π (input 1-controllabilities)	1 – (output 0-controllability)
NOR	1 – (output 1-controllability)	Π (input 0-controllabilities)
BUFFER	Input 0-controllability	Input 1-controllability
XOR	1 – 1-controllability	$\Sigma(C1(a) \times C0(b), C0(a) \times C1(b))$
XNOR	1 – 1-controllability	$\Sigma(C0(a) \times C0(b), C1(a) \times C1(b))$
Branch	Stem 0-controllability	Stem 1-controllability

So, let us see how can we calculate this for a probability based analysis. So, for primary input, so 0 controllability is P 0, because I can very easily make that primary input line 0 by applying a 0 at the input. And one controllability P 1 is 1 minus P 0. So, that. So, if it is coming from some from the environment. So, environment will tell us how easy or a how difficult it is to set a line to 0. So, that way P 0 probability will be coming from there and P 1 probability is 1 minus P 0. So, e if we assume that this is equiprobable, then this P 0 and P 1 both of them maybe 0.5, but it if it comes from some other system as input than this P 0, P 1 may not be 0.5.

Now, for the AND operation. So, and operation this 0 controllability is the 1 minus output one controllability. So, 0 controllability of an AND gate. So, let us see first see the 1controllability. So, how can make a one here? So, if I want to get a 1 at this point I have to make both of them as 1. So, this one controllability of an AND gate is basically product of input one controllability. So, if we have got these input 1 controllability. So, take a product that will be the input, that will be the one controllability for the output and AND gates 0 controllability is naturally, 1 minus output 1 controllability. Similarly for the OR gate. So, 0 controllability is product of input 0 controllability, because all the input must be 0 for the output to be 0 and 1 controllability is 1 minus output 0 controllability.

So, in this way for each type of gate I can say I can define some formula by which we can compute these 0 controllability and 1 controllability of individual get outputs. Similarly these buffer so this is there are input 0 P o is 0 controllability input 0 controllability buffer 1 is whatever buffer does not do any change in the logic. So, input controllability and values so they get transferred to the output controllability. Now these XOR gate, so for XOR gate what happens is that, if the output has to be 1 then I will the input a has to be 1, b has to be 0, or a has to be 0, b has to be 1. So, these two cases these two probabilities that to be summed up and that gives me 1 controllability, and 1 mines this that gives us the 0 controllability; similarly we have got XNOR.

(Refer Slide Time: 14:52)

Probability-based observability calculation rules

	Observability (Primary output, input, stem)
Primary Output	1
AND / NAND	Π (output observability, 1-controllabilities of other inputs)
OR / NOR	Π (output observability, 0-controllabilities of other inputs)
NOT / BUFFER	Output observability
XOR / XNOR	a: Π (output observability, max {0-controllability of b, 1-controllability of b}) b: Π (output observability, max {0-controllability of a, 1-controllability of a})
Stem	max {branch observabilities}

a, b: inputs of an XOR or XNOR gate

Now what the observability values so primary output it is definitely observable. So, it is taken as 1. So, this AND gate on NAND gate. So, output observability is equal to product of output observability and 1 controllability's of other input. So, what happens is that if I have got an AND gate if I have gotten got one AND gate.

So, it has got a number of inputs in it, first of all I should get a what to get a one here I have to set all these values to one. And the second thing is that this output. So, this output has to be propagated to some primary output. So, if this chip if this system has got a number of primary outputs, than this output has to be propagated to some primary output for observation.

So, naturally so we have to have this output observability, and we have to have this the inputs are to be control so that we get ones at this point. So, as a result we get the formula it is product of output observability and one controllabilities of other inputs. Similar OR or NOR gate it is again the same thing product of output observability and 0 observability 0 controllabilities of other inputs for not gate or buffer. So, it is output observability itself is sufficient. So, that is the observability value, because if we can get the output observable then we can just make 1 minus 2 get the input. So, similarly x o r x n o r can be done and for the stem part. So, if there is a stem this there is a fan out like this then if this value. So, if you want to observe this value.

So, we can go by any of these branches. So, if again the same thing if this is the cheap and these are the primary output. So, this stem may go here. So, the observability value of the stem is the cost of taking this stem to the output. Similarly these stems will maybe it is going to this outputs as a result. So, this is the cost of taking this output this stem output to this primary output. So, you take the maximum of all this branch observability.

(Refer Slide Time: 17:27)

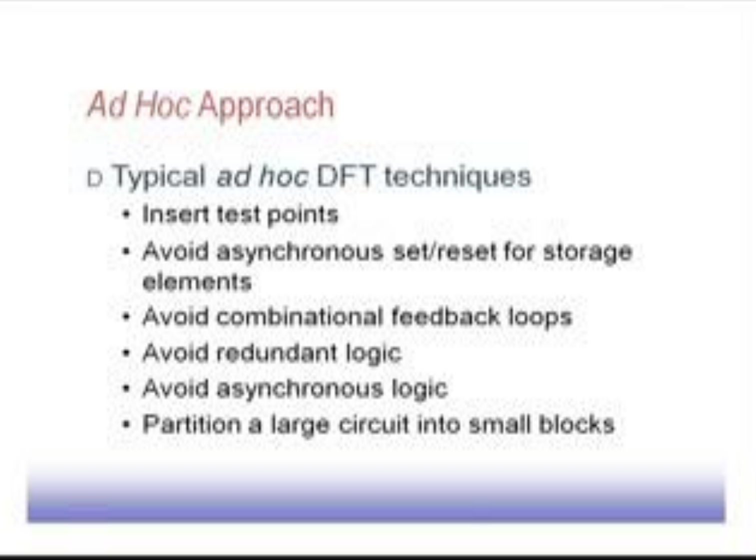
Design for Testability Basics

- *Ad hoc* DFT
  - Effects are local and not systematic
  - Not methodical
  - Difficult to predict
- A structured DFT
  - Easily incorporated and budgeted
  - Yield the desired results
  - Easy to automate

So, whichever the branch gives me the maximum observability. So, that is that easily the stem can be observed. Now for this DFT design, so there can be ad hoc design. So, effects are local. So, effects are local and not systematic. So, we just observe something an accordingly we do some local modification, so that this testability becomes better it is not methodical as I said that this is a totally ad hoc.

So, it is basically these are the test engineer. So, consults with the designer and C is can I introduce a line here or can I introduce a multiplexer here, and if it is possible then that that is introduced. It is difficult to predict like how much will be the benefit it is difficult to say, because some points we may not be able to modify. So, that is the problem with ad hoc DFT. On the other hand if we have structured DFT, so along with the design itself. So, we incorporate this testability hardware and it is budgeted at that point. So, it is the extra for extra testability. So, I will need this much of extra logic gates, this much of flip flop or this much of area so like that. So, that is put into the budget itself. So, the test engineer can be sure that I will have this much extra things that I can incorporate into the design yield the desired result. So, naturally if all those features are incorporated then possibly I will be able to reach my desired level of yield.

(Refer Slide Time: 19:14)



An automating the whole process becomes easy for example if I have the scan change, then I can very easily put the desired values at the scan flip flop, so that way automation becomes easy. So, will start looking with the ad hoc approaches; so typical ad hoc techniques are like this that insert some test point, then we avoid asynchronous set reset for storage elements. So, if there is asynchronous set reset then when we are trying to design a test pattern. So, all only sudden if we reset pulse comes to the chip to the flip flop and the flip flop gets reset.

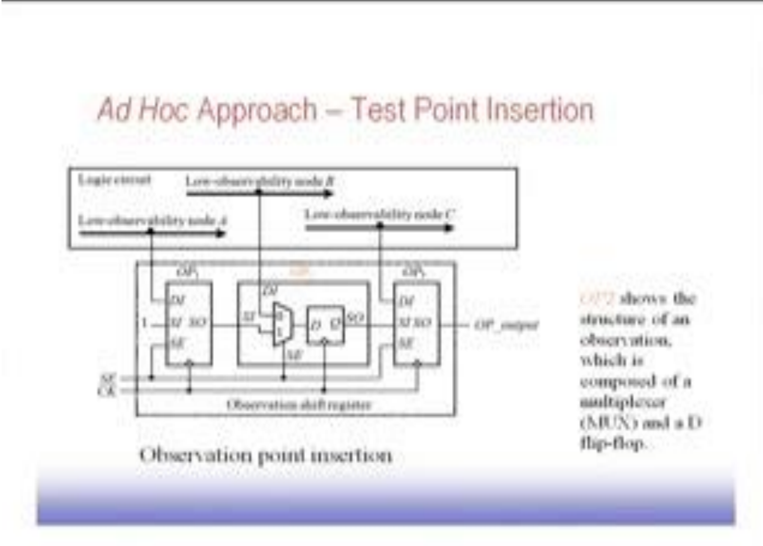
So, naturally that cannot be accounted for in the testing process. So, if my design says that if my design can be constant that I do not have asynchronous set reset then it is better. So, combinational feedback loop. So, combinational feedback loop if it is there than it actually makes the circuit circular, and there that makes it sequential circuits. So, sequential test pattern generation is difficult. So, it is better to break those loops. So, as a technique so we may have some extra component incorporated into the circuit so that this feedback loops are broken.

Redundant logic we like to avoid because you that will help us in generating test patterns we can detect more the faults will become more nonequivalent, as a result they can be detected; however, this redundant logic removal. So, this is not always advisable, because designers to avoid these hazard and glitches they may like to put some extra gates in the circuit which are actually redundant, but that helps avoiding them.

So, that way this redundant logic removal though this test engineers will ask for that design engineers may be telling the other way; then asynchronous logic. So, asynchronous logic if we can ignore then we know that the entire operation is synchronism with some clock signal. So, my testing is can be concentrated on this individual at individual clock transitions only. So, these scan cells shifting and all that. So, that can be affected easily, but asynchronous system does not have clock. So, as a result it a it becomes difficult to control the overall test session. And if I am allowed to partition a large circuit into small blocks, so that can be helpful because it can, it will help us to concentrate on smaller parts of the circuit, so as a result that testing process may be simpler.

So in fact, what happens is that this test generation of process algorithm so particularly. So, they are all NP hard in nature these problems in NP hard in nature, and for NP hard problems the difficulty is though we have to use heuristic methods, and heuristic methods they normally work well if the input size is small. So, if the input size is small the input circuit is small than this NP hard algorithm. So, these NP hard problem for them the heuristic algorithm they can give reasonably good result.

(Refer Slide Time: 22:29)



So, if we can partition into small block. So, it is easy, but again the same thing the it depends whether the design can be partitioned or not, it is very much design specific and it requires concurrence from the designer. So, the ad hoc test point insertion. So, suppose this is the situation. So, this is the logic circuit and there is a point which is these are some low observability points. So, A B and C, so they are 3 nodes, this 3 nodes they are not very much observable.

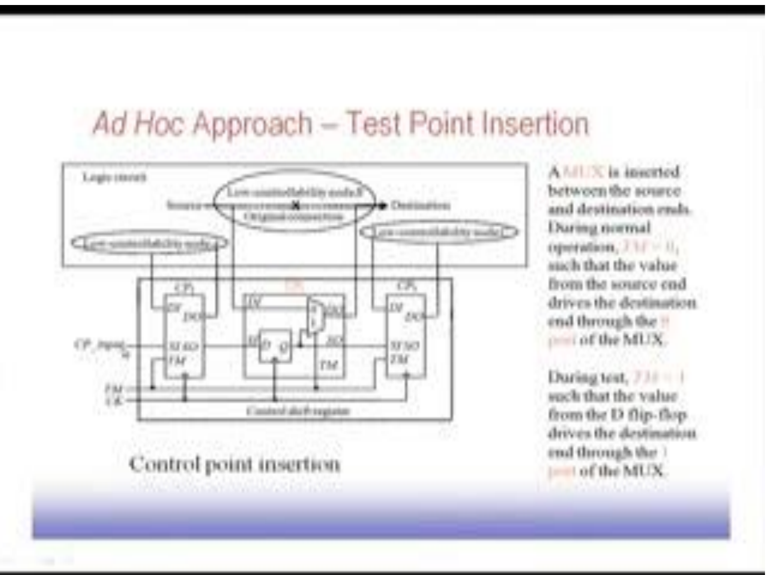
So, what is done we put them into we introduce a structure like this which is known as observation shift register. So, in this observation shift register what know? So, we have got this type of cells OP cells. So, OP 1, OP 2, OP 3 they actually form a shift register. So, what is happening is that. So, if you look into one particular such cell say the OP 2 cell. So, it has got one input from this low observability node b, another input is coming from this OP 1 cell and then it is going to the d flip flop through this multiplexer and this from this D flip flop it is going to this scan out to this scan out point or serial outpoint shift out point, and that goes to the shift input of this OP 3.

Now, if you are. So, if we apply some clock, so what will happen? This low observability points A B and C they will get captured on to this 3 flip flops corresponding to OP 1, OP 2 and OP 3. Now if you enable this serial shift enable or serial enable whatever you call it, and then apply some clock then. So, if you look into this multiplexer when this A C line is equal to 1 whatever is coming on the S I line. So,

that will go through it. So, this after one clock so this S O line this OP 3 that is this node three is available at the output, after one clock pulse this OP 2s s o b this cube it will be coming into these OP 3 is Q bit.

So, as a Q bit; so you will get it on the s o line. So, this B can be observed after one clock cycles, similarly A can be observed after two cycles. So, all the low observability points that we have in the system, so we can tap from those points and put them on to this observation shift register, and these observation shift register can be shifted by applying clock to it in a serial shift mode, and then we can get all these output. So, observation observability of the circuit is enhanced by introducing this observability points.

(Refer Slide Time: 25:07)



Another issue is the controllability. So, suppose I have got this type of situation. So, this node B is a low controllability node, similarly for this A and C also we have got similar situation. So, this is the point that you want to control, similarly here also we have a point that we want to control, the original connection was as in the dotted line. So, if this was the original connections.

Now, if you want to put some value here, externally we want to put some value here. So, we again take help of some shift register. So what we do? So, this line is this connection is broken from source to destination of load B. So, this connection is broken and this from these source the line is taken and it is connected to the data input of these C P 2 cell. Now when this for the normal operation of the system. So, this test mode this TM

input is 0, as a result this data input line will be coming from here, and it will be going through the D O line to the destination. So, this multiplexer will be transparent from D I line it will come to D O line and we will go to this measures. So, as if this connection get established.

Now in the test mode suppose we want to set some value to this point to this destination of node B So, what we do some how we set this flip flop Q to the desired value, and then this test mode signal is made equal to 1. So, what will happen? This Q will be available, this Q will be available at D O and then that will go to this destination. So, whatever pattern whatever b t we wanted to put at node these destination. So, that will be done.

Now you see that supposed depending upon our requirement. So, was the same structure is repeated in C P 1 and C P 3, now depending upon our requirement suppose we want to put this node A and value 1, node B at value 0 and node C at value 1. So, what we do first we apply these C P input; and in the C P input we give it clock. So, if you see this s i. So, this S i input is whatever if we give a clock to this flip flop then this flip flop will get the value. So, initially these TM value is made equal to 0, but. So, this C P input line with serially shift in the pattern 1 0 1 onto the three flip flops of this C P cells. After they have been shifted than we apply these TM line, we make this TM line equal to 1 as a result these connections get broken or this connection is not this connection is no more there which was taken we for normal operation.

So, it was going through this multiplexer I like this. Now once this TM line is made equal to 1, so this Q output will be coming here and it will go via these D O line to the destination. So, this pattern 1 0 1 that you wanted to apply at the 3 controvert nodes A B and C that can be done. So, this is the control test point insertion mechanism. So, using this type of shift register observation shift register, and this control shift register, so we can observe and control individual points of this circuit that you want to see.

So, we will continue in the next lecture.