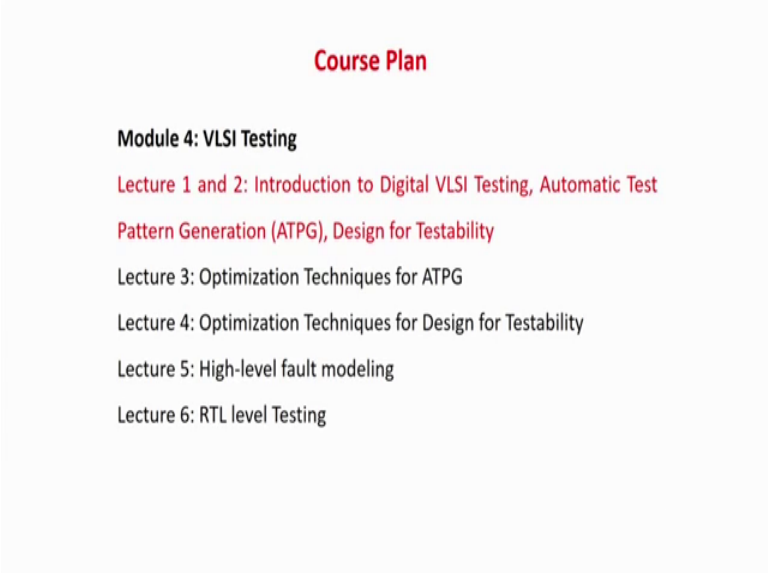
Optimization Techniques for Digital VLSI Design

Dr. Chandan Karfa
Dr. Santosh Biswas
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 14
Introduction to Digital VLSI Testing- II

So, welcome back on the course on optimization techniques for digital VLSI design. And at present we are in module 4 on testing.

(Refer Slide Time: 00:36)



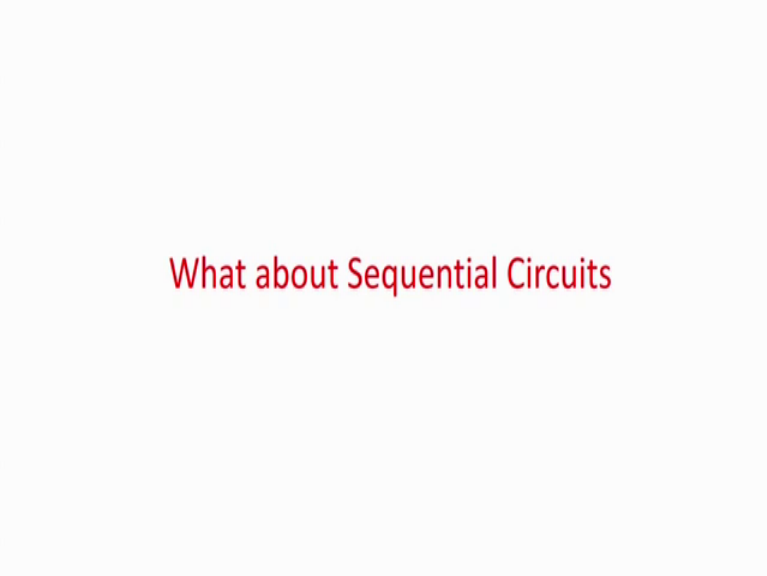
So, if you recall the last lecture, we were discussing on the basic preliminaries which are required to cover the advanced optimization techniques for this pattern generation and testing of digital VLSI circuits. So, the last lecture basically we in the late that is we are in the 2 to set up the set of preliminary lectures.

We have discussed about basically the basic idea of VLSI testing test pattern generation, and basically what are the fault models and what are the optimization techniques required so that we can test fairly complex circuits. And the motivation was to build a background of testing of circuits which are actually at the level of systems, which are said to know what complex theory left from the level of systems, and then when will be going through the lectures 3, 4, 5 and 6, which will be building all the basis we are trying

to develop in lecture 1 and 2, how they can be optimized so that we can handle larger circuits.

So, in the last lecture basically we covered the testing required or the test techniques required for handling combinational circuits. Now basically we will be discussing about sequential circuits.

(Refer Slide Time: 01:38)



So, what we discussed in the last day? That to handle complexity of even a fairly simple circuit and simple in the sense that it is lower complexity than NOC's, and associates to handle even a VLSI kind of a circuit like a multiplier or a simple mixed processor.

So, what does the level of testing we require? We actually have to handle the circuit not the functional level. We again handle all the circuits from a level of fault models. So, for model idea is that, we say that some of the lines can be stuck at 0 or stuck at 1. So, that is basically the fault model. And we have to generate test patterns so that we can verify that no such faults are present in a circuit.

So, basically this is an optimization step, what it compromises? It compromises that now the quality of assurance you can give to the people is not exactly same as the functional test, will be slightly lower, then that, but actually fault model the gain we have seen is that the number of test patterns required are exponentially less, then required for a

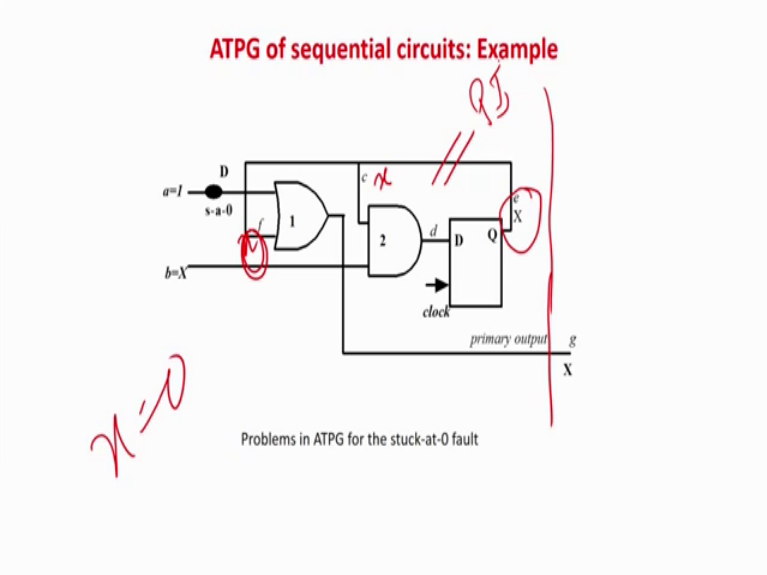
functional level of test. But the quality of assurance is varies is I mean near about the functional test which has been found out by statistical testing of large number of circuits.

Now, we will see; what are the challenges we face when you go for sequential circuits. Because most of the circuits basically now what is a sequential. Very clearly you will find circuits which are purely combinational in nature. So, what was the basic assumption we learned from the last lecture that we are going to go for structural tests, and that comes with fault models; that means, will be and also our assumption is that we are going to take stuck at fault model.

Stuck at fault model means, each line can be either stuck at 0 or stuck at one, and we are going to take only one fault at a time, and you have to verify that by a test pattern that that fault is not existing in this circuit. If we can do it for all such faults, then we will say that whole camera is 100 percent, and then the circuit can be shipped to the market. And statistically, it has been found out that if you do such a kind of test, then the correlation with functional test is more than 99 percent plus.

Now, we will see the challenges for our structures sort of testing for a sequential circuit. Actually, will now be able to appreciate, that wherever we go to sequential circuit the complexity will actually blow up. And it will become a more difficult problem to solve.

(Refer Slide Time: 03:47)



So, in the last class, we have taken different circuits, and then we have seen that there are 2 ways of test. One is actually called the random pattern test, and another is called the basically the synthesize propagate and justify kind of test, where the different difficult falls are tested. So, in when you are talking about sequential circuits, will be mainly focusing on sensitize propagate and justify approach, because mainly the difficulty of the complexity starts when you are going first as difficult to test faults.

So, for example, now I have taken a simple sequential circuit you will be able to appreciate that we have a flip flop. So, one thing that I like to think that in case of circuits, generally in the when you are going for large scale circuits we generally talk about the flip flops on the; anyway, flip flops are interchangeable that the theory will remain same, but most for most of the implementations we generally use a d flip flop. And if you have forgotten no detail fundamentals just a recap that a d free plus will take the value from here from the input to the output in the inner clock edge. So, that is the; what is the idea is a simple the input. And now let us see that this is your very simple a sequential circuit, and we want to go for listing a stuck at 0 fault here.

So, we will see what is the complexity involved, and how it in how it becomes more difficult in a sequential circuit like this. So now, one very important thing you have to remember that for any power of a circuit the value of the output of the flip flop may be 0 or one. It is totally un deterministic, not so non-deterministic that whether it will be 0 or it will be one, we cannot predict only power. It can be any arbitrary value.

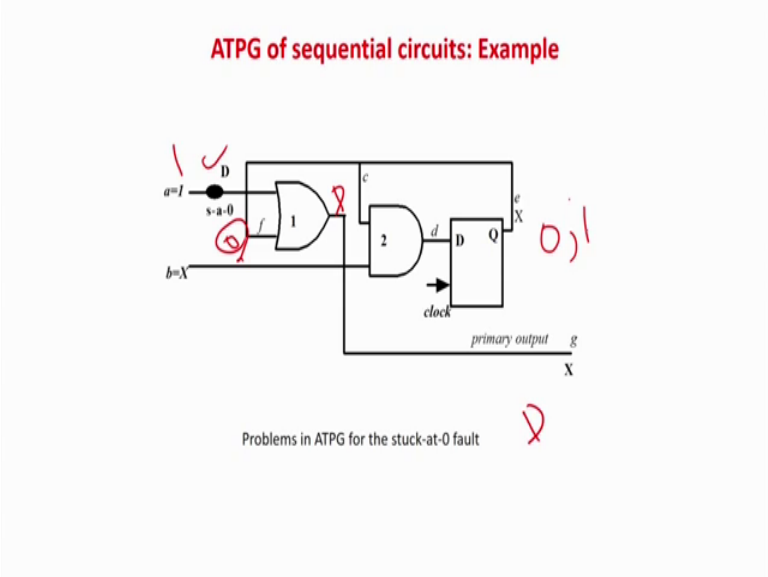
And as we do not know what is the value, and there is no prohibition of measuring because as I told you this is the input size and this is the output side. You can either apply some patterns here, and you can prove the values here. That is again another advantage of structural testing using fault models. That is, you do not require any probers inside the circuit; which is a very difficult problem because bringing so many peanuts or a circuit is a very difficult problem.

So, when we say X at the output of a flip flop, which is always in the start up of a sequential circuit, that we cannot predict whether the output is 0 or 1. So now, anyway so that fundament is gone we do not know whether it is a 0 or a 1. So, I have put an x, and then basically ah; so, this one is also b X, I do not know what is the value this one is also going to be X.

Now, we know that is a stuck at 0 form. So, what we have to do? We have to put up basically a 1, that is the sensitization step. So, what is the value? The value here is d that is normal case one failure case 0 there is very obvious. Now let us make it slightly clearer they figure. So, this is what is required. So, this is now the case. Now, obviously, the X because this is an or gate.

So, we have to make the X equal to 0. In fact, when we require 0 over here to move it. Now this is where the complexity increases compared to a combinational circuit. It is a combinational circuit basically this line will be cut and this will also be a sorry this line will be cut. And so, this thing will become and this also will be a primary. So, very easily by you are this will be sorry I mean let me go instead. So, basically let us try to not jump these steps basically.

(Refer Slide Time: 06:56)



So, the idealized a stuck at 0 I put a 1 over here. Now I require a 0 over here, and the value will be propagated so, if it is 0. So, the value will be a d over here, normal one faulty 0, and this d will be propagated over it. That is what is the requirement. This is sensitized, the value is propagated over here, and the value of 0 here is the justification, that is done.

But now the problem here is that all other things will be simple, d I am putting over at d, I am putting over there, but how do you put a 0 over here f? That is a very difficult problem, because when you start up the circuit the value is the value is 0 or 1, I do not

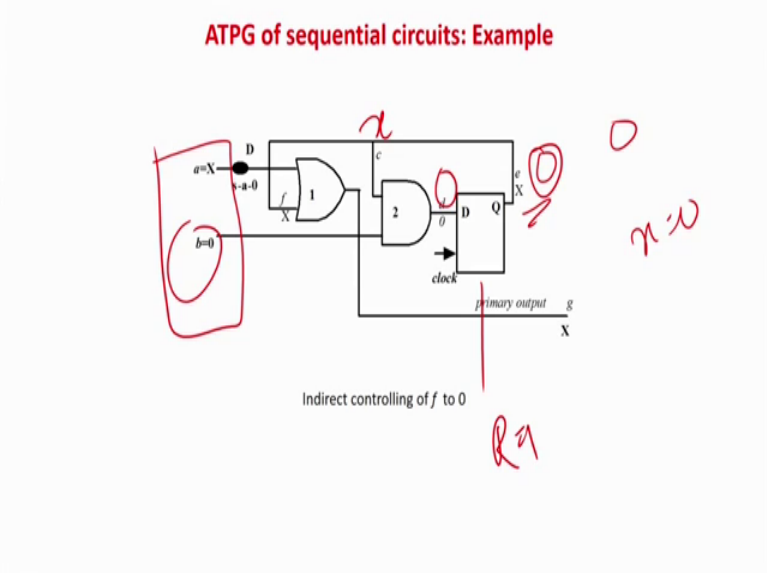
know that is why I have written an X. So, basically this line is also X, and now I require to control it. Now you can see there is no way of controlling directly line f or even line c because the flip flop output. And this toggle of the circuit will make it as a 0. Then what I have to do.

So, that is the difficulty of sequential circuit testing, because if it have been a combinational circuit, this would have been a primary output, and you could have directly put as 0 over here, and your job would have been done. Basic in a combinational circuit this one would not have been there, these are primary input and d will be a primary output.

So now the problem of sequential circuit, increases because of flip flop the controllability problem becomes more difficult; like in this, in combinational circuits the line f or c would have been a primary input, and it would have directly put a 0 over here. But now as is the sequential circuit X that is my f and c are output of a flip flop and it is a non-controllable line directly which actually makes the problem very, very difficult.

Now, you can understand there is only one flip flop at the level 2. 1 2 and this has a third level is a flip flop. But if you can give a very complicated circuit, then there will be flip flops at all levels and 100s and thousands of flip flops will be there. So, you can think how difficult is to control them. But in case of a combinational circuits, you can directly control them from the compare primary inputs, because there is controllability and there is no feedback from any flops. So, that is what makes the problem very challenge.

(Refer Slide Time: 08:58)



Now, how do you do it? So, of course, I require a 0 over here, that is my primary requirement I think where a 0 on the output of the few. So, how can I do? So, obviously, in sequential circuits, you cannot test by one test pattern. Like in combinational circuit, it was very obvious and also you have seen by example, that for a given stuck at fault you require maximum one test pattern to test it.

But in case of sequential circuit the philosophy not hold. Because first you have to make me free trust controllable to the respective required value. Like for example, at X i require X is equal to 0. So, first I have to bring all the flip flops with the required value, and then only I can apply the test pattern which in this case is, a equal to u, this was the case basically which is a equal to 1, and the final value has to be propagated to the output d is not required in that case this is X.

So, basically the test pattern a equal to 1 can only be applied after the flip flop has been settled to 0, that is X is equal to 0, then only when I apply the exact test pattern which is a equal to 1. And the value will be propagated to the output g which is d. So, all my in case of our sequential circuit all the efforts would be required to control the internal lines to the required 0 or one based on ATPG, which is the output of the flip flop. That is the main challenge of basically a sequential circuit compared to a combinational circuit.

So, how do we require? We require a 0 for X how do you get it. Basically, in this case it is simple. You can this one is already X we know we do not know what is the value of the output of the flip flop is start up. But fortunately, if I make b equal to 0 and this is an

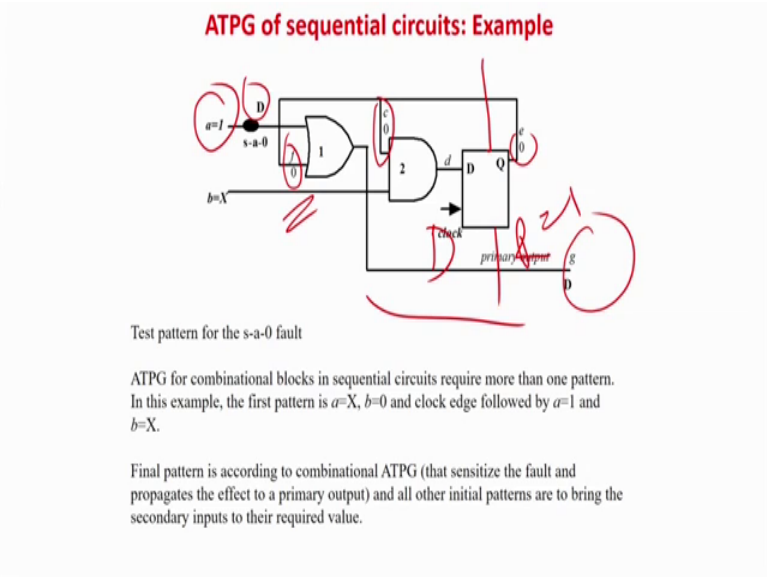
and gate. So, and of X and 0 is a 0. So, it will become a 0. So, this is a good value in this case which makes me to make it basically a 0 over here. So, if 2 would have been an or gate the problem would have been more difficult in this. So, anyway I put b equal to 0 .

And then d will be 0 then we will apply a clock pulse. So, the first pattern is X equal to 0 and b equal X a equal to X. I do not care about the value. And I put b equal to 0. So, if b equal to 0 output of gate 2 is 0, and then I apply a clock pulse, then automatically the X is going to become a 0.

So, this first step is actually controlling the value of the flip. So, this is an extra test pattern required if you want to go for our testing of a sequential circuit. So, these one flip flop so, I have required one extra pattern. If there are n number of flip flops there n different levels in the worst case you will require m different patterns to get them settled at the value.

So, for testing one stuck at fault in sequential circuits, you require a huge number of test patterns to actually settle the flip flops with a required values then only you can do the test. So, you can understand how the complexity starts rising. So now, what I have done in the first case I have applied a equal to X, b equal to 0 a X, means, it can be anything, then the value of d becomes 0 you apply a clock pulse then X is going to become 0.

(Refer Slide Time: 11:32)



So now your f is 0, c is 0, and basically your d is 0, q is 0; that q is 0 by the previous step. So now, I am getting the contouring value f equal to 0 and gate number 1.

Now, I can apply the real test pattern; which I am applying a equal to 1. Now the output will be going through the output of flop, and it will be going to output of g which is equal to d and your job is done. So, this is this case is very similar to a combinational test pattern. The idea is if all the flip flops are said to their required values which is required to do the testing.

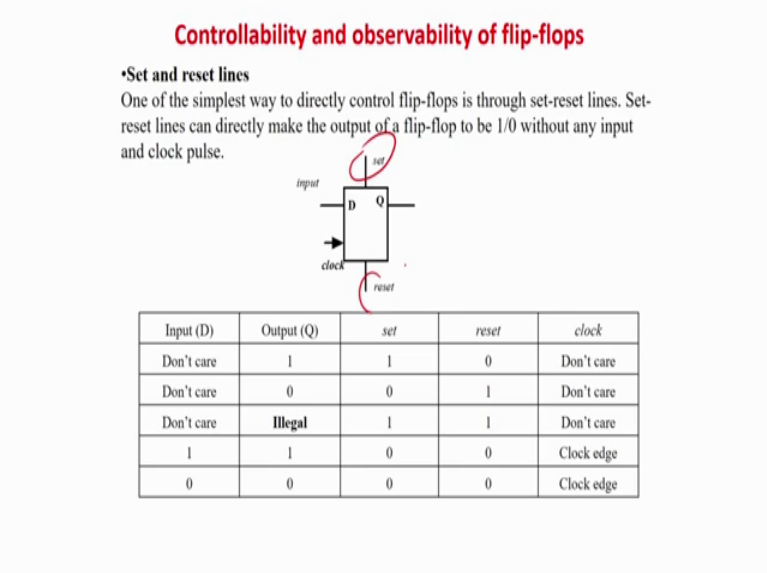
Once you have settled the flip flops with their values, after that the testing of the fault will thus become a simple automatic test pattern generation problem of the combinational circuit. Like, if when f is made 0 by the previous step where b was made 0, now it becomes a combinational circuit problem. That is you have to apply a 1 over here, it is sensitized d; d is propagated over here, and already you have justified it previously by making f equal to 0.

So, in sequential circuit this might be the other way around first you make everything justifiable. Then you sensitize our propagate. So, more complex here justification you have to plan beforehand. Because most of the (Refer Time: 12:37) will be uncontrollable in this case directly. So, you have to set all the flip flops. So, that all the lines are justifiable and propagate upon, and then only you can go to the previous step.

So, let us make the things more simple to understand that here all lines are not controllable directly because their flip flop outputs. So, whatever control you want to require, you have to settle the values in that manner using a previous sequence of patterns to get the flip flops the required values and then you can apply the test pattern. Now how to solve the problem. So, basically this is just the idea of a flip flop, just you can see I will come to this slide in a few minutes.

So, what is the problem now? The problem compared to communism circuit is that, you require more number of test patterns to test a single stuck at form. That is there optimization is again coming into picture. Even if it is a very small level circuit where you have 100 false to test the single fault.

(Refer Slide Time: 13:28)



Input (D)	Output (Q)	*set*	*reset*	*clock*
Don't care	1	1	0	Don't care
Don't care	0	0	1	Don't care
Don't care	**Illegal**	1	1	Don't care
1	1	0	0	Clock edge
0	0	0	0	Clock edge

If you require 10 or 20 patterns to test 100 false it will be in the level of thousand pattern; which is not at all possible even for a medium scale integration circuit forget about a NOC and SOC.

So, we require that our complexity should not grow more than the complexity of testing a combinational circuit. Combinational circuit testing what we have seen that even if there are 2 it stuck at faults, the number of these patterns will be much much less than that because one test pattern will cover multiple faults. And not only that basically for one pattern the one fall maxima one test pattern is required that is the worst case. But in this case as you can see the things have started growing up.

So, how to do it? So, one way is that you all might have recalled in our digital design lectures that all flip flops are basically something called a set and a reset line. So, if there is a set and reset line, then the advantage will be in that case we will not be equal to have to back propagate, and actually keep the settling values of the flip flop I in this case which is stuck at 0, sorry, in which case f is required to be made 0. If we have a set and I reset lying over here, then you can make the reset line equal to 1, means, annual the reset lines.

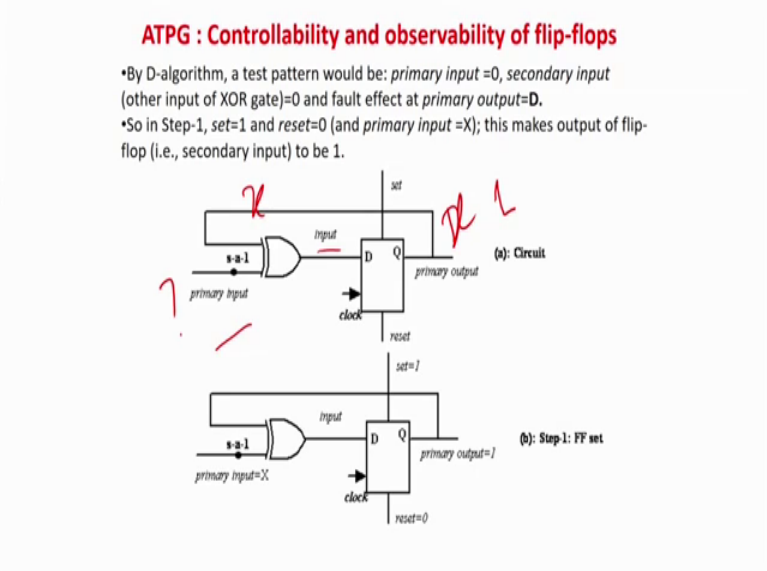
So, automatically your e will be equal to 0. And you do not require an extra test pattern like b equal to 0 in this story b equal to 0 in this case to admit the e equal to 0 or in turn f

equal to 0. Basically, you can directly make the reset line equal to 1. So, that directly it will become 0 and you can do the job.

And the advantage here will be if even there are 200 300 flip flops, all the set and reset lines will be controlled separately, and they can be brought to their required values. So, you can just one clock period, or you just require one test pattern to set and reset all the flip flops as required by the test pattern, like in this case, you require it to be 0. So, directly you can set the reset line. If there is some flip flop whose output has we made one, you can make it this headline equal to 1.

So, basically, we will go by this architecture. That is, we have set and reset line so, when you are directly control set and reset line basically, the inputs are do not care, and you directly get the value. And one thing we cannot make set and reset line equal to 1 together which is illegal, but for all our cases if you make set equal to 1, the output equal to is q equal to 1 if you make reset equal to 1, the output equal to 0 and others a normal d flip flops operation.

(Refer Slide Time: 15:33)



So, you can do that so, the same thing they have shown, I am just taking as a simple example to see if we have a set reset what will be the advantage. Like in this case, it is a stuck at one problem fault. So, you will require a 0 over here. So, if you have a 0 over here, basically, then maybe I assume that it is a xor gate. So, if I assume that I want I

require a value of 1 over here. So, one and a 1 normal case the answer will be a 0, but it is a sorry it is a normal case the answer is a 1, because 1 and 0 the normal answer is a 1.

But in the failure case the second line is stuck at one the other input is 1. So, one and I want same value the xor gate it will be 0 so, the output will be d. So, something I require like this so, this may be one test pattern. So, basically sensitized I propagate the value, through this because there is only one who do just mean values we propagated the input of the d flip flop, because you are the only one output.

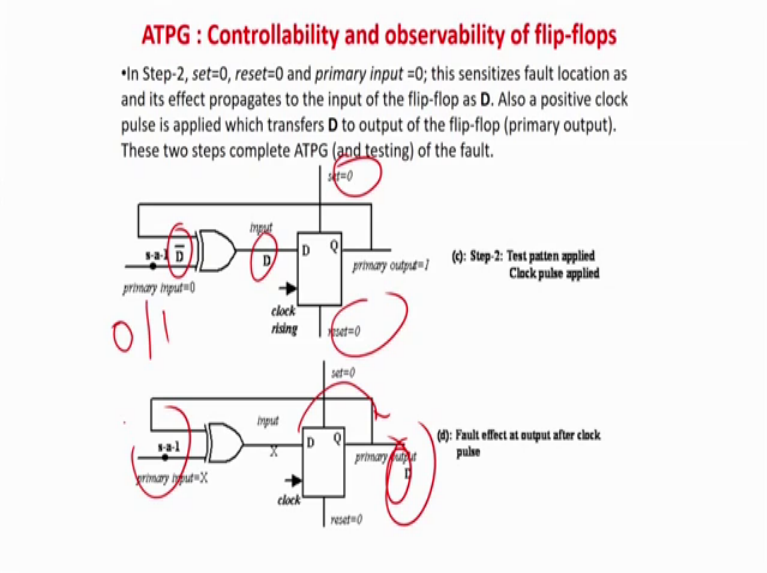
But actually, more important point has to note that I required a value of 1 over here. But again, as I told you directly you cannot control it because it is a flip flop pattern and now in this case if you see I will just give you homework, you can just say that I require to make this line as a 1.

But you will find out that in the start up this line is going to be X. So, unlike this previous problem; which was very nice that if you apply b over a of this line is becoming b and solving your problem, but in this case, they does not exist any impute which can directly control it to one, that you are not able to do you not be able to do it. You can take another homework and see that there does not exist any pattern over here which if this is X you can directly make it as a 1.

So, in fact, this fault will become un testable fault. So, that means, indirectly this line is not be able to be controllable, and this fall will become un testable which will be a very bad thing because your coverage will come down.

So, basically now I have to use set reset line. So, here I will make the set equal to 1, if you make set equal to 1 directly the output will be one, we will get a 1 over here.

(Refer Slide Time: 17:28)



And then basically as I told shown you basically I applied, sorry, I apply a 0 over here. So, normal 0 fault one. So, this is d prime output will be d and finally, you will get the value of the importantly is that, I actually require I made set equal to 1. So, that I get the one and I have control the flip flop.

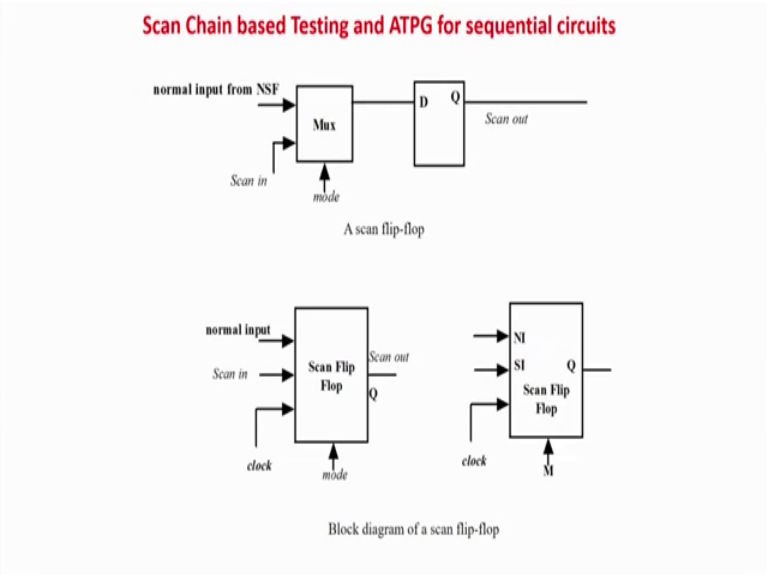
Now, once the flip flop have been set to the required values, I put set equal to 0 and reset equal to 0 because I want normal operation of the flip flop. So now, again b has been propagated, then I put the clock pulse your output of d will be here. So, if the value is one circuit is normal if the value is 0 this is a stuck at one fault over. So, what we have seen? That without set we said your problem is going to blow up. So, we require a optimization step.

So, what is the optimization? People have done they have put set and reset. Now this is again going back to the old ballgame to square one, that is of structural testing with extra pin outs. 2,000 flip flops means, 2,000 set reset lines has to be brought out as (Refer Time: 18:25) So, again we are boiling down to the same old story, that you test the sequential circuit we have to control the flip flops. If you do not control the flip flops directly a lot of test patterns have to be applied to make the flip flop settle to the required values.

And this is an example of an xor gate where you cannot even settle the flip flops to be made the line to be 1 to 2. So, the fault will become an un testable fault, which is a very catastrophic situation. So, in fact, cases flip flops are directly controlled. How you can

directly control of flip flop by putting a set or reset line. If you put a set on a reset line what is going to happen? As I told you, huge number of extra pin outs will be there and the story will be a again big flop and we again back to square one; which was actually testing off combinational circuit at structural mode without any fault a lot of pin out.

(Refer Slide Time: 19:17)



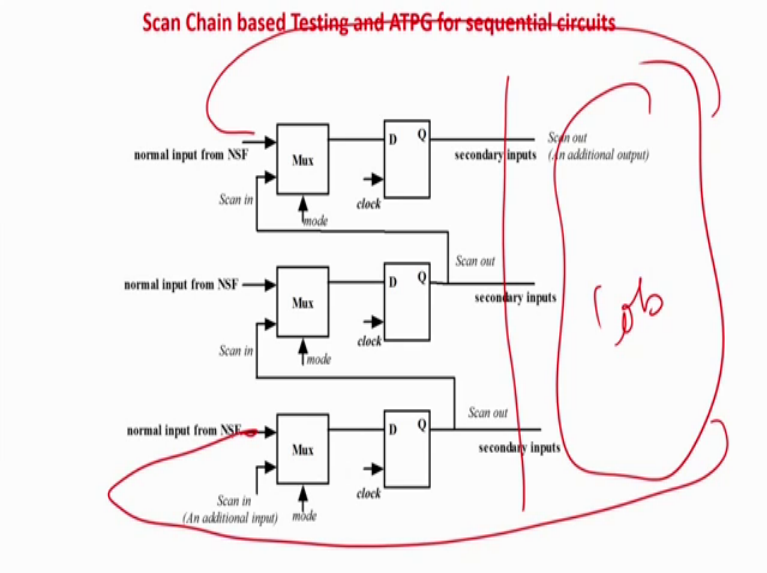
So, this story was also had come to a catastrophic end. Then there is something revolutionary that happened which have made a phenomenal just like testing of circuit fault model was a phenomenal jump, now I am going to tell you the story of a scan chain, actually we again made a phenomenal jump in testing of circuits, and it made the testing of sequential circuits a very low complexity problem just like stuck at fault made the story of number of test patterns and batman generation complexity, come down to a drastic level lower bound. So, the same thing actually happened for sequential circuits when people started using a scan chain. Of course, we total area over it and one beauty is in the in case of fault model, you might argue that actually I am not go going for functional test of the circuit.

So, they sight quality compromise in the level of test, but in case of scan chain base testing the complexity has been lowered, but there is quality compromising test; because we are not compromising in any kind of fault model, nor we are compromising in the number of test patterns. But what is the slight compromise we slightly increase the area over it.

So, that is very well accepted. So, scan chain you can think is a big thumbs up when you are going for testing of sequential circuit is actually revolutionized. And in fact, I am as is a cramped of tool I mean series 2 lectures of cramped I mean preliminaries of testing I am not telling the intermediate steps, because there are lot of intermediary words, mathematical formulations, I mean which you can find in the in any course or cad for VLSI that how people have come from the set reset problem to a scan chain, there is a lot of story in between before reaching to this final solution. But as we are covering up in 2 lectures we are cramping it up.

So now just I will tell you just I will I will come back to this slide later because the architecture was can flip flop, let us start to tell this story.

(Refer Slide Time: 20:57)



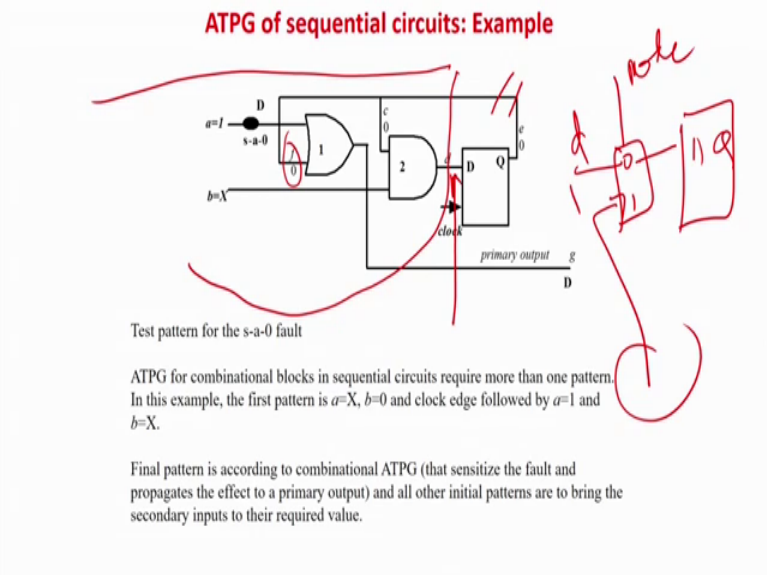
So now what is the problem? So, you have 10 flip flops. So, you have to control the 10 flip flops. One way is that you can use this patterns and backtrack and all these things you can do is a long story, or you can use set reset.

Another thing people thought is that if somehow, I can cut out the combinational part for some amount of time. And if I can connect the flip flops in series in a chain. Then I can actually directly put the values. Again, let me tell you this thing.

So, if you look at the this circuit, this will make the things more clear; say for example, if I look at this circuit it is the circuit. So, what I want to do is it, say, somehow for some

part say if I can somehow cut out this circuit. And then if I can directly put some value over here, then my job will be done, one way is up to set and reset. This is also you can do, but in that if I say that I will put a if the flip flop is there.

(Refer Slide Time: 21:46)

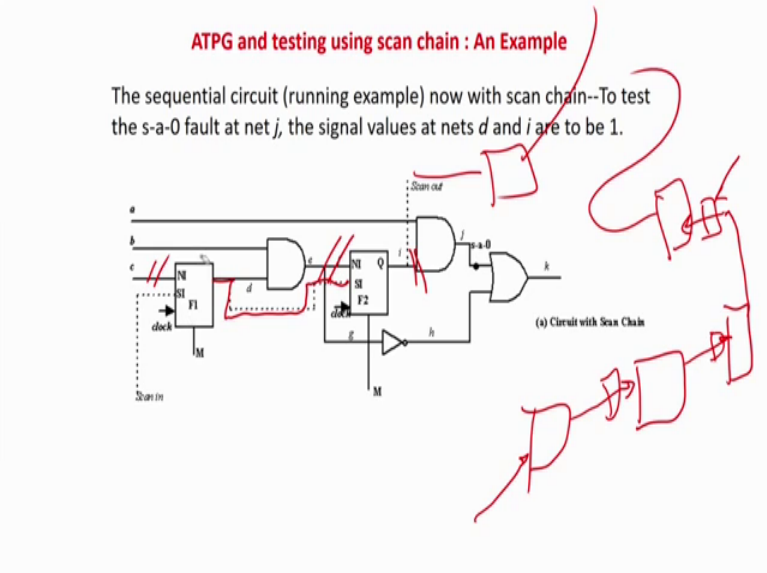


And it obviously whenever you have to go for indirect control, you have to put a multiplexer that you have already seen. So, this one in your normal d input and this you can say that I will bring some value from the output, this is 0 and one this is a take mode of this.

So, if you make the mode of test as 0. So, the d will go as the input to the flop. And if you make mode of test equal to 1 the output value we can put it and directly control the flip flop. So, anyway to this fine so, you will require dot of pin out. This will the story will remain same like a set reset problem. But then, but that is what is required basically also you can cut it out. Then also you can actually control the flip flop directly. Now, but problem is that I cannot have extra pin out.

So, what I can do. So, then people thought that some story like this, that for example, I will just come (Refer Time: 22:35) slightly complex example. And I will come say let is a slightly complex teacher. So now, what is the requirement? It is zoom it slightly. So, you can just see this is the figure slightly complex level to flip flop.

(Refer Slide Time: 22:37)



So, what do you have to do? You have to control these 2 flip flops to do some testing. So, somehow what if I can do is that if I can cut this, sorry, say somehow if I can cut it out ok, if I can somehow make a boundary something like this, then I will be very happy.

So, what is the boundary? Sorry, slightly I made the mistake the boundary I have to just eliminate out these 2 flops. So, what is the boundary story I am telling? So, if I can somehow cut these 2 flip flops out. So now, what is happening? Whatever is below this red line 2 flip flops will be out. So, that is one then I can directly give some value over here, and also, I can give some value over here, and I can do it.

But again, if you want to directly give some value you are require a multiplexer anyway, but you have to use 2 pins to do it. That will make the problem slightly difficult. Then what is the scan chain? These scan chain says that I will not put too many pins out. What I will do is that, I will actually cut this out, then again, I will cut this line as output and I will cut this not allow. So, basically these 2 flip flops are isolated.

Now, I will not put any direct control over this input, because if that 10 flip flops in sequence, then you have to put 10 control line whether what I will do? I will put output of this flip flop, like this, the output of this flip flop and I will connect it to the input of this flip flop ; that means, I will try to make a chain like for example, these are the 3 flip flops.

Which are I have now isolated from the normal circuit. So, this is one input of the flip flop. The output of this flip flop I will connect to this the output of this flip flop I will connect to this and it this another flip flop I will output will connect to this finally, I make it out of course, you remember that there are flops over there muxes over here, which is actually going to take 2 inputs one input is basically, normal input from the circuit the other input is through a chain.

So, therefore, we actually call this technique as a scan chain. So, what do you do? By multiplexing arrangement, you cut out the circuit into parts. So, one part the combination circuit will be out, and all the flip flops will be connected in a chain fashion. Like, output of this flip flop will be are connected to this is us output here, but it is another flip flop would have been here the output will be connected to this and will go. For and whenever the circuit will become normal or a normal operational mode.

All this scan part will be disconnected, and the circuit will be operating in the normal mode. Now what I will have basically, I have something like a single chain like these are the chain of flip flops, all these are connected in mode. This is one primary input and there will be one out. Now this becomes a shift register. So, once it becomes a shift register, maybe, if I equate the control of 1 0 1 1, you can easily send a sequence of running 1s and 0s of this and you can fill up the shift register chain.

So now the problem has been solved in a very simplistic design manner. In case of VLSI, always it remember that complex designs are complex, but the solutions are very simple. So, if you make a solution very complex hardware idea will be high higher power will be higher. And the problem will not be a very good solution here the problem is very complex that even all the flip flops integrated. In the very internal parts of the circuit how you can make the values 0 and 1, controllable with the minimum number of pin outs and minimum amount of time.

So, what they are doing over here? With multiplexing arrangement, they are cutting out of all the flip flops, but they are not bringing primary input and output of all the flip flops to the primary input output way, that there is only the extreme flop output will be the primary input. And the output of this one will be connected to the input of this flip flop the output of this flip flop will be connected to the another input, and it we made a chain. So, they will become a shift register so, wherever you want to make 1 0 and 1, you can

just make these running sequence, and you can just load the shift register using the sequencer across.

So, that is a your very simple secondary a digital fundamental, you can easily apply this and you can set the flip flops this one. And then again you can bring it to a normal mode and to the test. I give you an example to do on this, by do show you very explicitly. But I have just given you the idea. So, what problem we have solved? So, all the flip flops now will be set, there will be only 2 lines extra only the scanning and scanner.

But now you can say that and if you tell me how many test patterns are required, there will be mainly for to test pattern. One pattern will be required to set the flip flop to different values required, and one pattern will be required to do the test, that is the practical test. But now, test time will be slightly increased, slightly or rather it will be to a great level, why? Because one button will be required to do the test. But now all the flip flops you are connected in a sequence. So, you do not have direct access to all the flip flop to set or reset required depend on the requirement.

Now what is going to happen? If there are 100 flip flops in a chain you require 100 clock pulses to shifts values of 0 and 1 to the shift register to get the value. So, test time will be increasing. Pin outs will remain same, the number of test patterns also 2 one is the test pattern to do the test and the sequence to be entered. But the sequence to be entered basically require lot of time. Because if there are 100 flip flops in a chain you have to go over here.

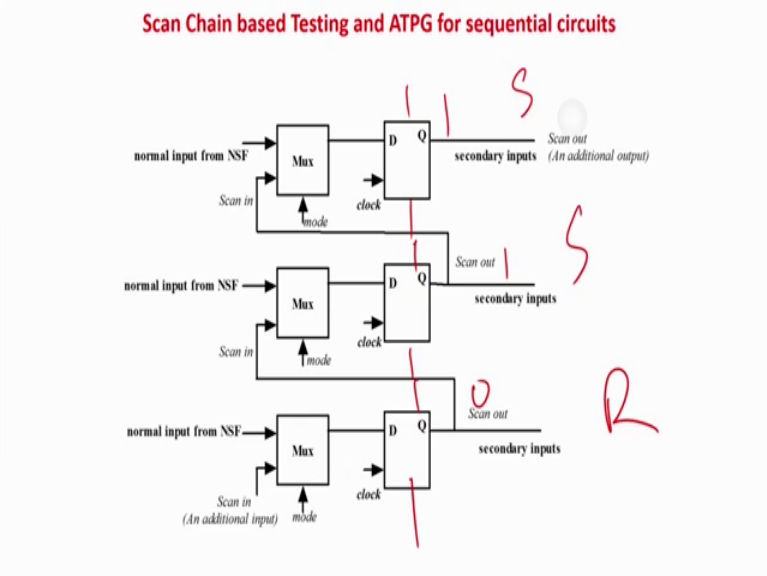
So, this story was fine where you are having medium scale or slightly higher scale of integration, but as we discussed in the end of these course, I mean end of this lecture today which motivate you automatically be motivated that if there are 20 thousand flip flops. To test one circuit, you required 20 thousand clock pulses to settle the different values of the flip flop.

So, you are going to be very problematic situation. So, when we are going for a NOC and SOC is scan chain has to be optimized that is what we are going to lag. So, any way so, I think I have given you the motivation, now I am going to give you a very abstract view. So, assume in this case there are 3 flip flops one to n 3 these are the secondary inputs. Basically, they are outputs of all the flip flops will be connected to a combinational cloud. And they are also the input from NSF. NSF means the net state

function block basic is a combinational circuit. Output of the flip flops are all going to be feedback basically this is your combinational circuit NSF block.

So, basically all these generally feed it, and the output of the flip flops anyway again go to the combinational cloud that is basically your NSF block, and then actually this is what is a typical structure of a sequential circuit. So, I am, I need not need not draw over this. But now basically say that somehow, I make I need make it 1 1 and 0 I need to do that.

(Refer Slide Time: 28:52)



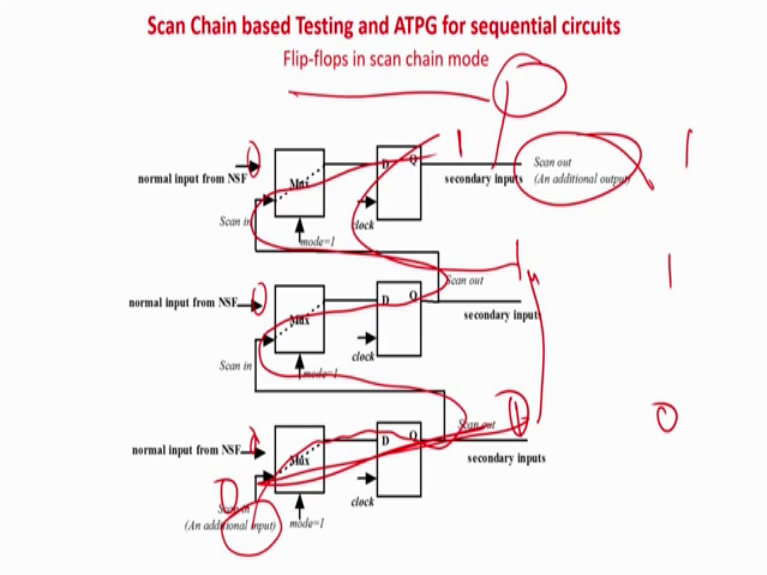
So, basically what I will have to do, either I have to make that set reset. So, in this case 6 lines will be output, but you can set I am resetting one mode. Because this is this is set, this is set, and this is the set. So, if they are 1 2 3 4 5 6 lines directly the output you can do it in one clock pulse. By making this set equal to 1, set equal to 1, and here reset equal to 1, your job will be done.

But in fact, that will actually blow up the number of pins because 6 additional pins will be required so, that is a very difficult problem. Now let us see how scan chain actually solves it. So, I assume that there was a there is a combinational circuit over here. We can actually feeding this d flip flop. Now somehow, I have to cut this.

So, how I can cut it? I have to put a multiplexer over here. So, they have put a multiplexer there is one pin from the combinational part, that is the NSF block. And the

other part actually basically follows from the output of the other flop to this other flops to this. So, you can see a chain has been formed, that is this one.

(Refer Slide Time: 29:48)



So, when these flip flops are in scan chain mode or a test mode basically.

So, you can see the out this is this is the primary input, and this is one primary output, this is scan out this one will be an additional primary output only 2 extra pins are there. So now, you see the all NSF block has been decoupled. So, all the flops have been separated out a separate chain. So, how it happens basically? This you can see I have made a chain. Say, as I told you what was the value we required 1 1 and 0, we require 1 1 and 0.

So, what you will do? You will put a first you will put a 0 over here, you will give a sorry first you will put a 1 over here, you will put a 1 over here, scan in and you will give a clock pulse. Of course, will be one will appear over here. Then again you will put a 1 over here and that you give a clock pulse.

So, once see the same clock pulse what is going to happen, this one will be propagated over here, and this one will come over here next what is going to do. Next you are going to give a 0, next we are going to give us 0 at this input and give a clock pulse. So, what is going to happen? This one will be propagated here, this one is going to be propagated over here, this one will be going over here by the chain and the shift register and this 0

will basically this 0 is happy anymore over here. So, you get it certain simple shift register fundamentals.

So, 3 clock pulses your pattern is applied. So, this is set take the flops. So, you assume that if there are 10 thousand flops in a chain, or in a circuit you have to make a chain, then 10 thousand clock pulses will be required to settle the flops. So, this is actually not increasing the number of test pattern.

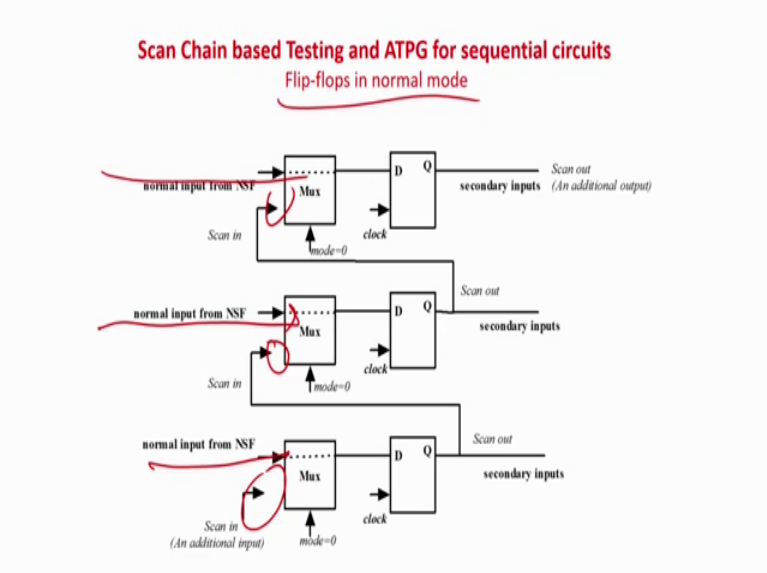
But it actually delays the amount of time to do what this thing. So, this if it the a flops. So, when it was a medium scale circuit or to typically slightly larger scale integration, people still try to handle this complexity by increasingly flip flops to scan chains to multiple number of partitions that yeah, I will not have one 10 thousand length chain, either I will have 10 chains of 10 1 thousand length each.

But in this case, what is the penalty paid you will have 10 extra pin outs for scanning and scanner. Because of course, you have to have a scan in and scan out for every change. But if you want to control the change parallely. So, more pinouts should be there. So, that is actually some kind of optimization to handle test time that started taking place. All these things will start discussing when will be going to the original part of the course from the next lecture series.

But this single scan chain actually solve problems still many years before moving to more complex circuits. Because as you can see as in the simple scan chain or the simple register. So, the delay is very, very less, because one shift will be connected to the other there will be connected to other and so forth.

So, from the 0 fundamentals, you can apply very high clock speed to transfer the data, because delay is almost very negligible, because one flop to other end so forth. So, there is no combinational circuit in between. So, this scan data can be fit very, very fast.

(Refer Slide Time: 32:32)



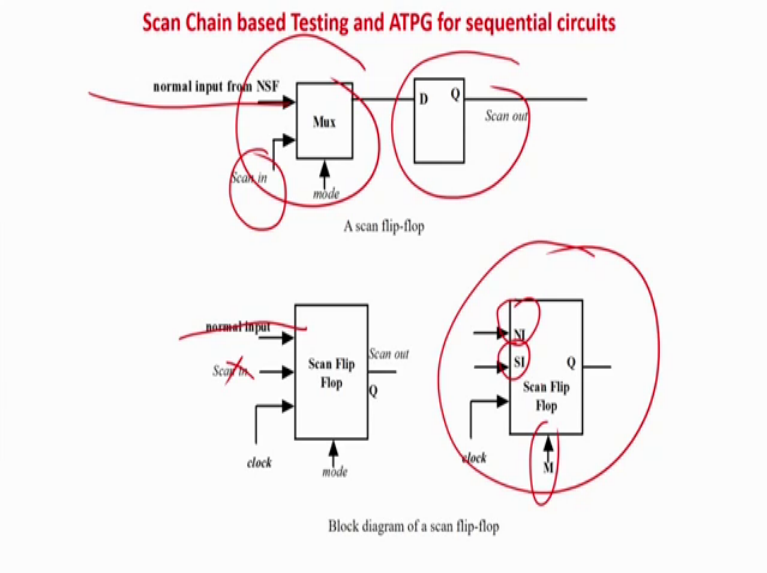
And now when the scan job is done, basically you can see the flip flops are in original mode.

So, you can see now, this scan in and the output of these all these lines are cut to lines from the combinational circuits are NSF block is all coming to these flip flop; that means, the circuit is not operating in the normal mode at this step you have to particularly apply basically your test pattern and you are going to do the job. And even if the circuit is operating in the normal condition, you will be in this mode only where I mean all the test the circuit is in the normal mode, you can see that all the inputs are going from the next edge block to the flip flop. And you are doing (Refer Time: 33:01) this scan chain is a cut mode.

So, advantage you can do the testing with only 2 extra pins, penalty paid is you have to put some multiplexers, anyway, I am not too much bothered about area these days, but only real challenge that remains is the test time. So, that has to be optimized people started it optimizing, when they have gone for advance level for testing for more complicated circuits.

So, all these things will discuss how to do this will be a major part of the course will be in the forthcoming lectures.

(Refer Slide Time: 33:29)



So now let us look at the typical architecture of this scan. So, what is happens basically we have a d flip flop we have a multiplexer. This is the normal input, and if this scanning, scan it can be a primary input or it can be the input from the other flip flop which is previous to that in the chain.

And so, we represent it something like this, we say that this is a scan flop, when the when the mode is a normal mode, basically this normal input will be there this scanning will be not there, and basically vice versa for the test mode.

So, we represent a flop in this way, and a scan flop. These are special kind of flip flop, where you basically have a multiplexer, and I if the normal input you also have a scan input and a mode to control this. So, in most of the VLSI circuits. Most of the flip flops if you see you will be surprised that all a scan flops. Because scan chain is a d factor standard that has been coming into picture for all circuits. Nobody, actually uses a normal flip flop.

So, if you have some experience in designing cad circuits. You will be surprised that all the flip flops basically has scan flops. Because any and nobody bothers much, but the only way to get access to circuits inside is a scan chain. So, even we will see that even if you are going for a NOC and SOC to get to the internals of the circuit you should have a scan chain. Of course, a lot of optimizations will do like breaking the single scan chain to parallel scan chain, and then some part of the scan will be dropping because you have

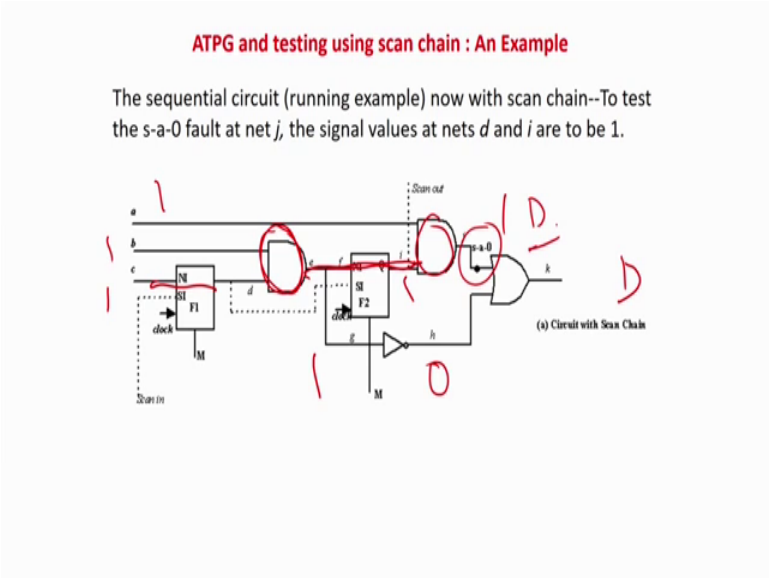
slightly compromised the area or test time to slightly know what the complexity to have a faster test time. So, all those compromise we will do to optimize the test time, or minimum amount of compromise to get a gain or benefit in terms of test and what do I say efficiency. But scan chain will still be the defect to standard to go to the internals of the circuit. For model will be still the de facto standard to go for testing. Of course, it may improve from stuck at to bridging etcetera, but still the story will remain same.

So, whenever talking about optimization, means one of the advances that this whatever we are discussing till now is something called the cad for test which applies to typically complex circuits. Like minimum, scales processors. But when we move processors to NOC and SOC, the time will be very high because the circuits will be very high.

So, this simple way of test will not be applicable over there. Even if you apply you will time you blow up. But still we are doing is a pre-requisite because the bottom philosophy remains the same. Fault borders and scan for sequential circuit you should have scanned to go into the circuit, and fault model this philosophy of fault model is mandatory to do test in a reasonable amount of time.

So, whatever you do these things has to be done.

(Refer Slide Time: 35:53)



So now what now basically we are going to see with an example how scan has to be done. So, we are taking a very simple circuit, and we are taking a stuck at 0 point over

here. So, same problem as you have seen that flip flop chain in sequence you have to do that. And we are not having any set reset lines because that will blow up the problem.

So, simple first you have to do an ATP. So, one of these steps, now I am going to think things more mathematically that what are these steps to do it. First basically you have to forget that there are flip flops, and we have to go for ATPG. So, I just say that there is no flip flops over here just assume that they are directly connected for timing you just as shown.

So, what will be that I am making a pure combinational circuit by removing all the flip flops connecting the input to the output and your job is done. Then go for ATPG combinational. Stuck at 0 means, sensitize would have one. Normal one fault 0 D output will be D, that is the propagation, because there is only one path here propagation is done.

Now, I have to justify so, how will justify? H output of another input of an or gate; obviously, has to be 0 to make it controllable inverter. So, I have to put a 1 over here. Next, I have to apply a 1 over here and gate. So, a is; obviously, equal to 1. So now, g is equal to 1 is equal to 2 my job is done, again I have to go back this is the and gate is is this and gate basically I am talking about this and gate.

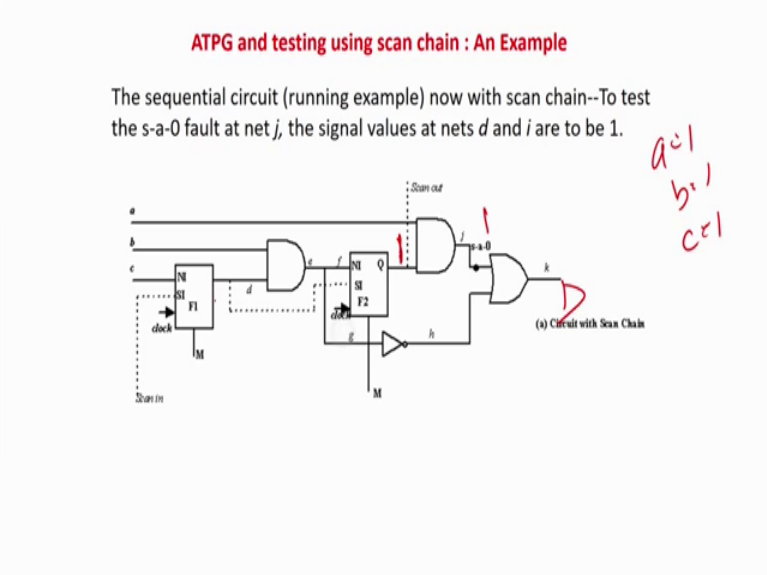
So, the output of the and gate is required to be 1. So, of course, b equal to 1 and c equal to 1. So, just see if a equal to 1 b equal to 1, and c equal to 1 your job is done basically. So, 1 1 and one means basically the output of this gate and is via this second and gate is going to be a 1, because a equal to 1, sorry, b equal to 1 and c equal to 1 implies the first and gate is a 1.

So, this gate is this line is basically one then [like] a is also equal to 1. So, the output of the second and gate is equal to 1. So, basically stuck at 0 for is sensitized. And then we are propagating it to the output d, basically h is also equal to 0. So, the d will be propagated over here.

So, d this h is equal to 0 implies that g is equal to 1 and g is of course, equal to 1, because b equal to 1 c equal to a no job is done. So, the sequence of combinational test pattern is 1 1 and one and the output is now I have to apply it.

So, here we seek problem of sequential circuits start coming into picture.

(Refer Slide Time: 38:07)



So, I know that a equal to 1, b equal to 1, and c equal to 1 is going to set my purpose, but the same time. So, I try to sorry I am slightly zooming this circuit. So, that you can look it nicely from the screen.

So now the problem r is let me just write it. So, I require a 1 over here, d over here, and also, I have seen that I require a 1 over here and also, I require 1 over here. So, that let me do a slightly a zoom version.

So, you can see that all the flip flops has to be made to 1 1. So now, what is the one way? If we are using a propagate I am not going to discuss the old story in which case without using any scan chain because if you want to make both the lines both the flops 1, you can do it, but you will require 2 or more patterns for that. Because to make first flop 1, you have to apply a c equal to 1 and apply a clock pulse you first. Required output will be directly equal to 1.

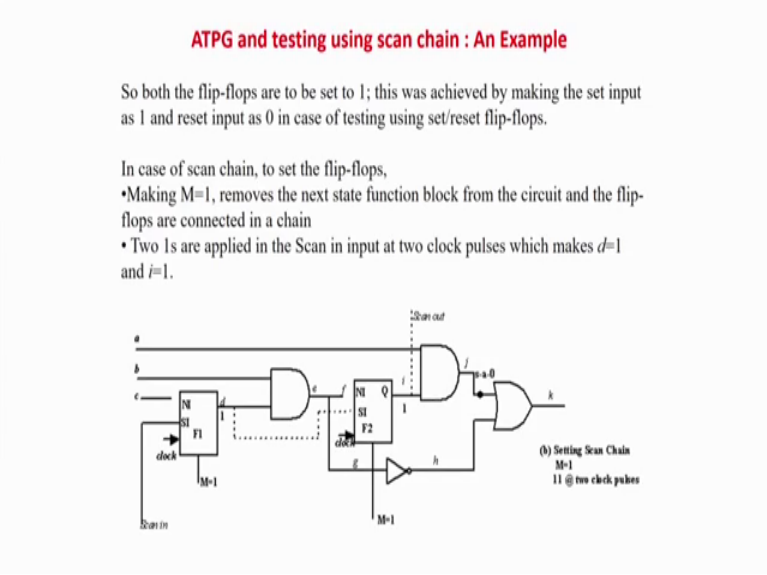
But then at the second step you have to make b equal to 1 and c equal b b first step c equal to 1 if you make the first flip flop value 1. Next, basically to make the second flip flop value equal to 1, you have to apply b equal to 1. So now, if they b equal to 1 this is what already d is equal to 1 apply a clock pulse the output will be one.

So, 2 patterns would be extra required to settle the business of the 2 flip flops in chain. If there are more number of circuits in the chain, this problem will actually blow up. That you have to find out an extra algorithms. In fact, people have already tried it you know the more complicated mathematical problem that if flip flops are inter twinned. Without using any scan or set reset how can you get to the real values required in a very short amount of time these are more mathematical problem.

But nowadays nobody uses it because, the story of scan chain has come in to be change. So, I think you are able to appreciate that, if you are not able to directly control the flip flops, you require more number of test patterns and also calculation. Like, here I am calculating c equal to 1 will make flip flop one equal to 1. But I develop some mathematical calculations to find out that b has to be also equal to 1, and in the next stage flip flop 2 will become one.

So, if the more inter twin circuit you can find out that it is going to become a more complicated binary equation solving kind of a problem. But anyway, now we have the story of scan chain. So, it is simple so now, basically this is this story.

(Refer Slide Time: 40:11)



So now I am first connecting it to a scan chain. So, let me just turn swimming the circuit for you.

So, you just concentrate on the circuit. So, it is a scan mode so, if you look at the scan mode you can see that this line is cut. And then this line is also cut. So, and the output of this flip flop is connected to this one else. So, so this flip flop one and flip flop 2 are in a shift register mode. So, what do you have to do? You have to have m equal to 1, that is test mode is equal to 1; that is, the scan chain is connected c and e are decoupled. So, this sequential part is one side the combinational part is totally decoupled. We put 1 1 here in the input and I applied 2 clock pulses.

So, once you do of course, first this one will come over in the next clock pulse this one will go over here and this one will come over it. Simple digital design shift register fundamentals. Now what the flip flops are settle, but I have to apply 2 clock pulse that is what you think time is being taken.

So, I required 2 clock pulses to do this, but mathematical computation is very, very simple in this case. So, when you see for testing optimizations are at many phases. One is the test time generation optimization that ATPG algorithm optimization. How much time you require to do the test plan or you have to optimize and how much time you require for that.

So, in scan chain it is very simple. You first go for in a combinational ATPG already which is not a very complicated problem, then you know the values of flip flop outputs, and then directly assume that I will be a shift register and I will do the problem .

But if the scan chain is not there. Then you have to use the combinational circuits to put the flip flops into that mode. Like in this case I have told you first you have to make c equal to 1, then you have to make b equal to 1 c equal to 1 which will make equal to 1 and the flip flop 2 will be 1. So, in this a simple circuit still you require some backtracking. So, is a very complicated circuit this backtrack will be very, very large.

So, the ATPG generation complex will become very high. Scan chain actually solves that problem almost. Because if you just when if you have assumed that you have a scan chain; that means, directly it will be able to apply the test pattern from the ATPG of a combinational circuit, which you are doing in this example.

So, ATPG complexity remains same it does not change, but what another important paradigm of complexity of our test is how much time you require to apply the test

pattern. That is not actually too much solve over here. Because in this in this case of scan chain of course, you can apply the vectors very fast. Just I am going to tell you this just you can look over here.

If I am going by scan chip. So, there is one flip flop another flip flop another flip flop. 1 1 I apply and I give a very fast clock pulse, when I am writing to clock pulses over here, the clock pulses will be very, very fast, but take to clock pulses are required. So, test time is still at remains a challenge for me, but if you see if I say that I go by the previous approach, in which case I say that b equal to 1 and c equal to 1. In fact, b equal to 1 and this time b equal to 1, and I am going to make flip flop 2 equal to 1.

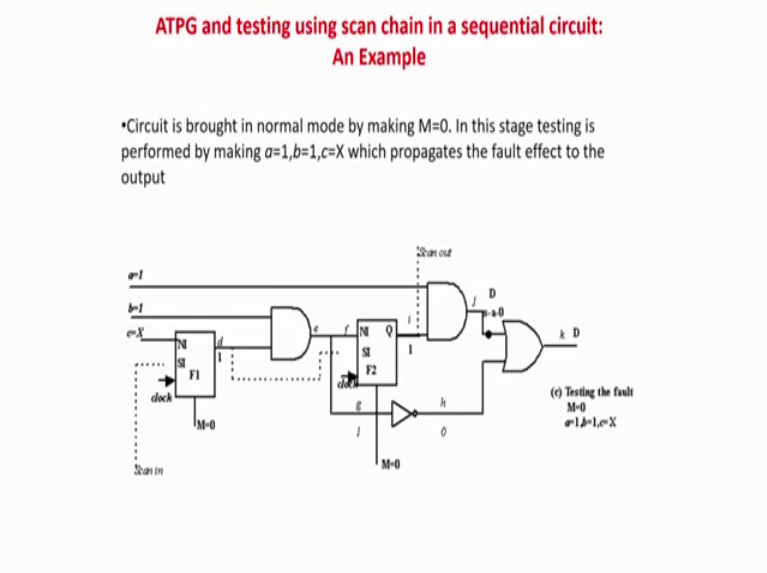
So, the clock speed cannot be that much high because there will be a delay of this and get e in picture. So, is the very big circuit instead of this and gate you have a large circuit. So, setting and resetting the flip flops using the combinational circuit will slow down the clock. So, in case of scan means you require say 2 clock pulses in this to settle the values. In case of without a scan chain also you will be requiring 2 clock periods totally.

But in case of without a scan chain, you will also have the delay of this and gate. So, clock frequency will be slightly less. So, in that clock is a just scan chain, number of test patterns may re remain same, but the frequency will be much higher compared to without a scan chain.

So, there is one optimization we can achieve without the scan chain with a scan chain. So, scan chain actually solves the complexity of automatic test pattern generation to a great level. Secondly, big another problem actually is sorts is that the clock frequency can be very, very high. But what challenge still remains is that if there are 100s of flops in a chain, you have to apply 100 to clocks you have to do. So, ATPG the application complexity that is test pattern application complexity still remains, which you are going to solve for in the next session of lectures.

So, anyway so now, if you look at this circuit. So, what they have done they have applied 1 and 1.

(Refer Slide Time: 44:10)



So, this flip flop is now 1, this flip flop is now 1, yeah, this one the values have been settled. Now we are going to simply apply your this patterns. So, what they are applying over here? So, this flip flop is 1, this flip flop is 1. So, you have to apply a equal to 1. So, a equal to 1 means it is d and of course, if the value of b equal to 1 this one already we have seen this you can just see that automatically you will going to get one and you are going to get the test.

So, what is the test pattern here? A equal to 1, b equal to 1 and c equal to x. But before that this can has as to be settled as 1 and 1. So, that 2 test patterns basically in 3 3 clock periods will be required in. In fact, 2 clock periods, and one extra will be required to test it. So, 2 clock periods will be required to settle the flops 2 1 1 and finally, you have to apply 1 and a 1 the real test pattern. And your circuit will be tested as d as the output, now what?

So, this brings us to a very concise representation of VLSI testing which we will use to do for medium to fairly complex circuits; say, around 20 years there are around 10 years back or so forth. Now what are the challenges we are going to face or what the challenge we have faced which we will be trying to address over here. First challenges even the circuit is very, very large, say we have tens of millions of gates.

So, at least 5 some 1000-100 faults will require test patterns because one test pattern can solve multiple faults we have seen. But still you require some 100s of thousands of test patterns to be applied to do the testing. So, if you have a NOC and SOC of brad

challenge. It is impossible to even at the same fault model you apply test patterns and you can do it still the number of test patterns will be so high that will not be able to do.

So, what is the optimization? Some people will actually try to find out which are the very good test patterns. That if the 100 test patterns, one first is patterned it takes 50 forms. The last 2 the test patterns did it only one or 2 very difficult to test faults. So, I have to compromise. So, I will select only such test patterns which had this huge number of patterns and slightly compromise on the coverage complexity.

Because there is no way I can go for a 100 percent testing and spends hours in the tester and make your chip some make one thousand dollar a simple chip for simple maybe a mobile processing unit. I have to have some quality control and will I have to go for the cost matrix. Because mostly when you are talking about VLSI systems major idea is consumer electronics.

So, where we do not require a very rigorous part, we require some quality of service of course but not as a rocket science. But you have to make it affordable. Because nobody can purchase some one lakh rupees mobile phone. And which we say that it will never had rather. I will prefer to purchase a 10 thousand rupees mobile phone, which may say has some issues may be very few maybe even one month some issue may come up.

That is basic philosophy that when you have to bigger trade off. So, one challenge is how can you bring down the ATPG or means number of test patterns. And still we have to have minimum compromising coverage. Secondly, about the it scan chain. So, if you can see the scan chain basically forming a very long scan chain. So, the time to insert the test pattern this chip rates will be very, very high.

So, what I can? Can I partition this scan chain? Can I put some of these for partial scan? Can I put some of the flops in partial more than some of the in non-non-partial means some of these flops will be in scan and somebody will some will leave it as it is. So, that at least some area is same as well as at the same time I mean I can have less time to push in the patterns because, now instead of one chain and have many parallel chains to do it

But many parallel chains mean at the same time many pinouts. So, I cannot have 100 parallel scan chain into picture. So, how will make a trade off? Secondly, fault model so, if there is a circuit with one million lines. So, there will be 2 n or 2 million faults in this

stuck at fault model. Or even if you are having a fault model in the a gate level that will also be very, very high.

So, what I can do is that I have to go at abstraction level. So, what is an abstraction level? Abstraction level means I will take block as a adder, I will take block is a multiplayer, I will take block as a multiplexer, and I will have some fault model at that abstraction level. I will not going to the internals of the like for example, here I am not going you the internals of the gates. Than I am taking whether the lines have no stuck at fault or something like that. I am I am not bothered about the internals.

But if the lines are also very I what I do? Then again, I actually make a side you turn that I take blocks, and I find out that whether blocks are operating properly or not. It is something like moving from structural test to some kind of a functional test, but at a very, very abstract level like adder a multiplexer and a simple do that.

So, they are actually called testing at high level faultmodels. Then again everything I am going to now not go out at gate level. I will go for and everything in the RTL this register transfer level that I would block level. So, can I go for some abstraction to solve it. So, basically what we are going to see what how what optimizations can be done to high level synthesis for high level circuits or very complex circuits like a NOC and SOC. Can I have a very abstract fault model? Can I have duty testing or a must higher level of abstraction so that I can ready block level? At the same time how can I reduce the number of test patterns? Because this RTL or high-level test pattern problem actually is very recent. But before that people try to optimize, how can I bring down the number of test patterns with slight compromise.

How can I optimize on the number of scan chains so that I have minimum pin outs extra, but still I can do this scan chains very fast? So, all these things actually will be required to do test that are circuits of higher complexity; which will be the main focus of the course from the lectures which will come up. So, these first 2 lectures just gave you a brief coverage of what is VLSI testing what is the philosophy, what people had done say about 15 year 10 to 15 years back, and why what needs to be improved?.

So, that idea have tried to given you, because we want to make this course is self-contained course. And we do not want you to go back and again listen to lectures or

books on cad for VLSI or VLSI testing. So, the basic philosophy we have built. Now we will try to improvise on that and see how more complexity can be handled.

Thank you.