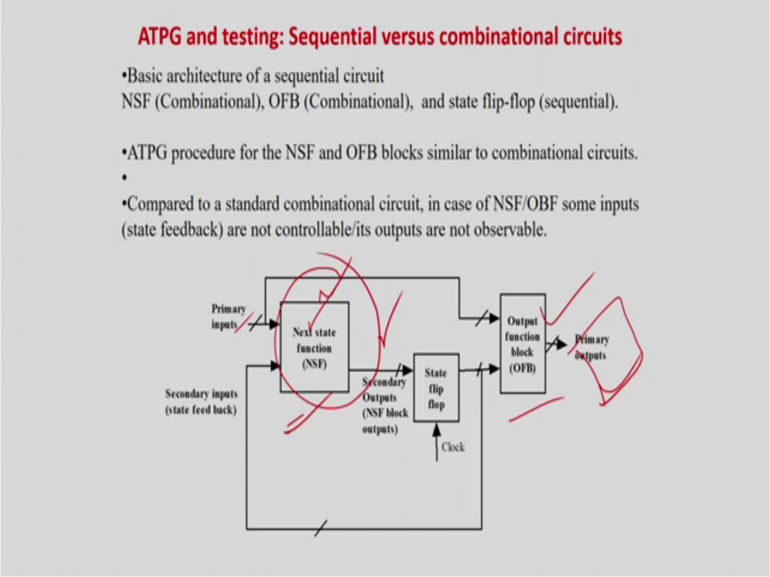
Embedded Systems – Design Verification and Test
Dr. Santosh Biswas
Prof. Jatindra Kumar Deka
Dr. Arnab Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Lecture – 28
Part-3: Embedded System Testing
Scan Chain based Sequential Circuit Testing

So, welcome to the 3rd lecture on the part of Testing of Embedded Systems. So, as you see today we will be talking on lecture number – 3, that is a Scan Chain Based Sequential Circuit Testing. Now, today we will see what is the complexity involved in the circuit becomes sequential and for as of you we all know that most of the practical circuits are sequential in nature. So, today we will see what are the complexities involved and how they can be solved.

(Refer Slide Time: 00:53)



ATPG and testing: Sequential versus combinational circuits

•Basic architecture of a sequential circuit
NSF (Combinational), OFB (Combinational), and state flip-flop (sequential).

•ATPG procedure for the NSF and OFB blocks similar to combinational circuits.
•
•Compared to a standard combinational circuit, in case of NSF/OBF some inputs (state feedback) are not controllable/its outputs are not observable.

Primary inputs
Next state function (NSF)
Secondary Outputs (NSF block outputs)
State flip flop
Clock
Output function block (OFB)
Primary outputs
Secondary inputs (state feed back)

So, basically if you look at this diagram actually shows you the basic structure of a sequential circuit. So, you have the next state function block, you have the straight register you have the flip flops and is the output function block.

So, you all know basically that this next state function block and the output function blocks a simple combinational circuits. And, so, then why we should talk very elaborately about testing of sequential circuit what is the problem. So, we all know that again our default assumption is that we will be going for structural testing based on single stuck at fault model which is the most widely accepted scenario.

So, now if you see, so, the primary outputs are observable and the primary inputs are controllable. So, let us assume that we will test on to the function next state function block. So, as simple as that this combinational circuit, so, you can ask me why I should have a separate topic on this. Now, you see the problems these ten registers are nothing, but flip flops. Now, there is a feedback from the feed from the flip flops to the next state function block, because the output on the next state function blocks will depend on the primary inputs as well as the present state and that will generate the next state.

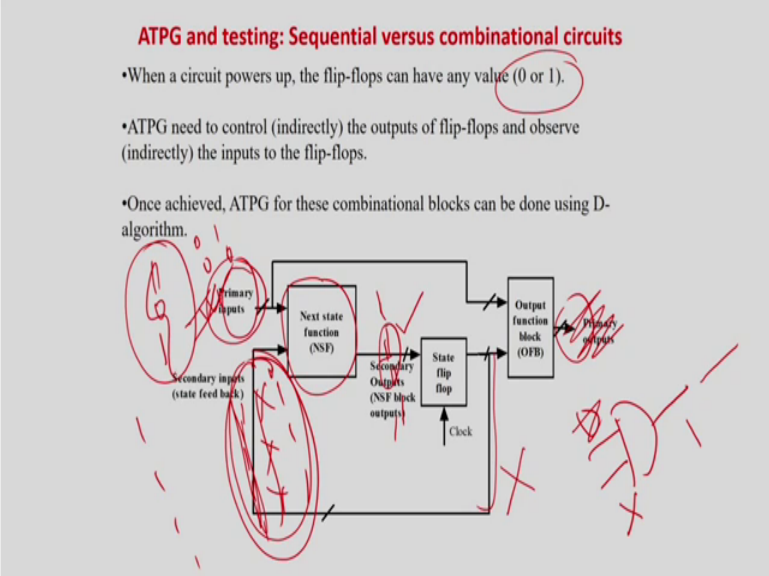
Now, this part is uncontrollable because, I should say directly not controllable directly uncontrollable, like the primary inputs you can directly gives some values so, they are directly controllable. But, this line which is a feedback to the next state function blocks which you also we can also call as secondary primary inputs are basically non-controllable as such. So, how to control them? Basically, you have the indirectly control them by adjusting the primary input values and then will be converted into next state values and indirectly you have to get the required values over here.

So, this is because of this problem that it we will see the test problem becomes much more difficult in case of sequential circuits. Also, we in case of combination circuits we can directly out see the output of the combinational function block and we can directly see match with the golden response. But, in this case also these lines are not directly observable. Why they are not directly observable because they are going into the flip flops as input or the state registers as inputs.

So, due to this problem of this secondary outputs basically which are next states as well as which are the present states, that is the inputs to the next state function block unobservable are controllable directly. So, the sequential circuit testing is becomes more complicated in nature we will elaborately we explained with examples. So, till now, we have just told that one of the issues involved in this one. Similarly, if you talk about testing of the output function block primarily outputs you can directly observe, but again these are not directly controllable. So, somehow we have to make these rights controllable.

So, in this class basically we will see how to mainly test the next state function block once you are able to do this testing of the output function block because more easier because at least the direct outputs you can directly observe inputs you can directly control only these things controllability is a problem. But, in case of testing the next state function block observer ability is also a problem. So, if we can test the output function block, the same algorithms can be directly applied for the output function block.

(Refer Slide Time: 03:57)



ATPG and testing: Sequential versus combinational circuits

- When a circuit powers up, the flip-flops can have any value (0 or 1).
- ATPG need to control (indirectly) the outputs of flip-flops and observe (indirectly) the inputs to the flip-flops.
- Once achieved, ATPG for these combinational blocks can be done using D-algorithm.

Primary inputs
Next state function (NSF)
Secondary inputs (state feed back)
Secondary Outputs (NSF block outputs)
State flip flop
Clock
Output function block (OFB)
Primary outputs

So, mainly concentrate on this part and other things can be easily extended to the output function block and very important these are the output of the flip flop. So, g are generally X, X means I do not know what are the values of 0 or 1, when a circuit powers up the flip flops have can have any values between 0 and 1, so, we make it X. So, all these inputs will be X and you can only control this and you can only observe this. In fact, these are non-directly observable lines.

So, this is lot of a concern as of right now because we are testing the next state function block. So, neither you can observe it directly observation levels come later, but in fact, to give certain inputs to your circuit to be tested these values are actually X X X in the beginning X means I do not know what they are because when the power of your circuit they will basically either 0 and 1 and we represent it by X. So, maybe I require this X X all X is to be 1 1 1 1.

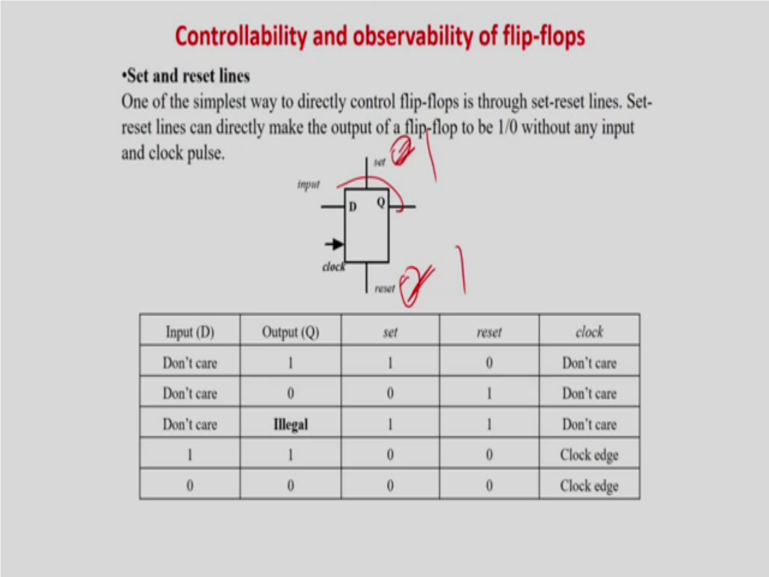
So, in fact, now we have to indirectly control this next state function block through only these primary inputs. So, that they be all become 1 1 1 1 and you can feel it how can I make it control line? I will just give you an example maybe this line is X, I do not know what it is either 0 or 1, but I can make this line as a one by fixing this. So, indirectly I can control this and get you without using this X value. So, in a similar manner we have to control this circuit in such a fashion by using only the primary inputs and try to make all the outputs as 1 1 1 1. So, all these X's will be converted to 1 1 1 1, but again and along with the primary inputs this will be only X's.

So, along with a suitable primary inputs without caring for the X's you have to bring all the ones, so that in the next stage I can get all the ones over here and the circuit can be tested. My assumption was maybe I required some 1 0 1 in the primary inputs and 1 1 1 1at the secondary inputs that is the feedback to test a circuit test a gate in this circuit.

So, initially you cannot do that initially everything will be X X because we do not know we at the power up what is the value of the flip-flop outputs. So, they are all X X by using these some patterns like some patterns maybe I can say 0 0 1 1 0 1 whatever using some patterns somehow you have to make them to all ones indirectly and then next state you can test to circuit. As I told you what do you mean by making a loud line output as 1 is in spite of X sorry, it cannot be made at X 0. So, I can have to make us using even in that with X I can get the value over here. So, if for the AND gate if I require a 1 with an X I cannot get it.

So, it is not always possible that you will be test able to test all the lines at the circuit in that manner sometimes it may not be possible also if it is possible you have to go by this approach. So, now I am just giving you the philosophy elaborately I will see lot of examples.

(Refer Slide Time: 06:35)



Controllability and observability of flip-flops

•**Set and reset lines**

One of the simplest way to directly control flip-flops is through set-reset lines. Set-reset lines can directly make the output of a flip-flop to be 1/0 without any input and clock pulse.

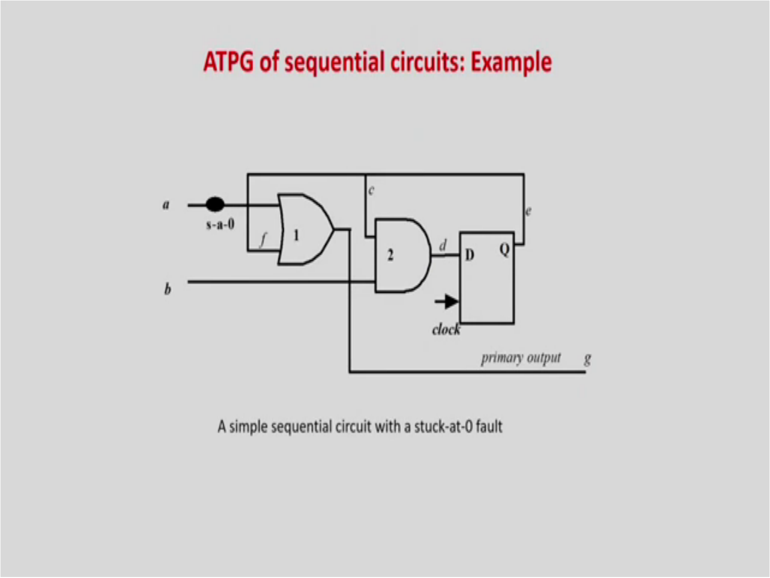
Input (D)	Output (Q)	*set*	*reset*	*clock*
Don't care	1	1	0	Don't care
Don't care	0	0	1	Don't care
Don't care	**Illegal**	1	1	Don't care
1	1	0	0	Clock edge
0	0	0	0	Clock edge

So, before that I think we have learned all about flip flops in our digital design course because we assume the prerequisite of this course in digital design the architecture. So, we know that there are a lot of flip flops J K S R D T, but in case of circuits be a circuit generally we go by the D flip-flop which is one of the most simplest flip-flop and we all know that all flip-flop can be interchangeable used. So, we are using a D flip-flop and which is more de facto standard in all cases of we are has a circuit designs.

So, the flip-flop is very simple whatever input you gives as the we will come to the Q was output in the next clock pulse, it has some set and reset lines. So, if the set is equal to 1 directly the output will be equal to 1 irrespective of the input and similarly if you make your reset line is equal to 1 this line will be equal to 0 irrespective of the input.

And, when you want to make your flip-flop functional you have to make this and this as 0 0 and then the input will come here at the level of a clock. And we cannot give set equal to 1 and set equal to 1 this is a non-feasible condition. I think if you look at this table this is enough to tell you the basic working of this circuit over again the flip flop, this is a recap of your original design course.

(Refer Slide Time: 07:45)



ATPG of sequential circuits: Example

A simple sequential circuit with a stuck-at-0 fault

Now, we will come with an example. So, they say this is a simple circuit example which I want to test. So, there is a flip flop so, you can think that this part this I can call as the primary output. So, this is your next state function block, right and we can say that as a sequential circuit so, some output will of course, go to the flip-flop which we can call the secondary output and also if you can see sorry, I should make it over here because input let me rub it up. So, you can see it has how many input is an input as a b they are the primary inputs and one more input is actually coming from the feedback which is the secondary input.

So, the secondary input is actually causing the main problem to me because, otherwise if you look at it if I could somehow control this line if I could have somehow cut this line then actually your circuit will have input as a b and I can call it as e. So, this is a primary input I cut it~~. So~~so, I can directly control it and maybe if you can directly observe d, so, it becomes a simple combinational circuit testing.

But, now life is not so easy here because in this case what happens neither d can be observed here fortunately there is another output line called g, but sometimes it may not be there also it could have been directly gone to the sequential circuit input like to d or someplace it would have gone, but here anyway we have extra line to observe or it may not be the case. But, in this case the main problem which you are going to highlight for us to observe is that so, this is the line which we cannot observe, right. So, this part if you can look at so, this is the light which cannot be taken into picture.

So, let us look at the testing of this part of the circuit so, this circuit, right so, this is basically a sequential circuit. Now, why is a sequential circuit because we will find out that there is a flip-flop which is given to this. So, this is the input which is going to the flip-flop with actually a secondary output in case of combinational circuits so, this one you cannot directly observe. Similarly, if you see this is the part which is coming into this one I can call this one rather combinational block of my circuit. Let me first you erased this as the output and again see 1 this half.

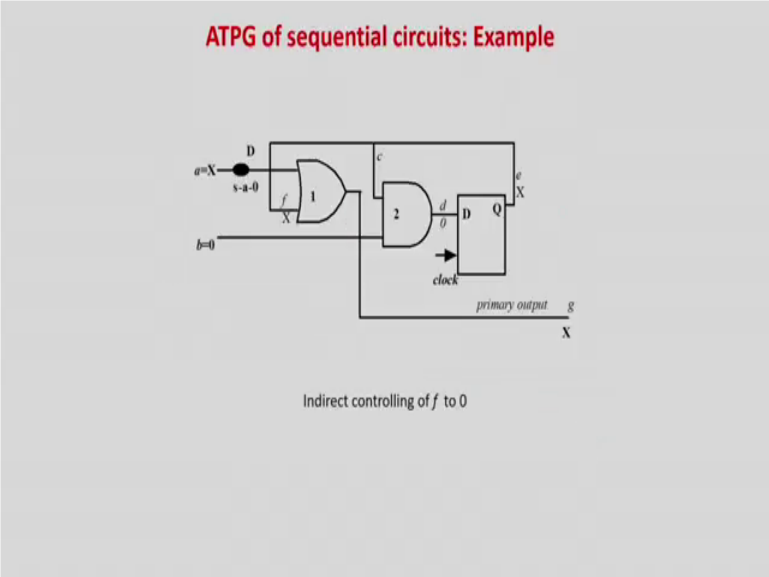
Now, you see, but this is a combinational circuit this part which I have made a box. So, it has input like a and b which you can directly control because there primary inputs. Problem is with the line e because, e is the line which is coming over here and you cannot directly control. If somehow I could have cut this line e then you would have primary inputs a b and e and I will show you could have easily disturb this circuit, but, in this case this line e is not directly controllable because it is coming as a feedback from our flip flop. So, we will see what is the problem that arises out of it, I want to test this circuit.

So, the you have to simply apply stuck at 0 so, you have to apply a 1 over here you are going to get the D over here. So, we I am going to propagate this as the output this is the only path, importantly of the observe that when the circuit starts up I have to put the X over here. For the time being just assume that I have not have the reset lines and if they are there I have made all 0, I am not using the set reset lines now.

Now, these are X; X means basically there either 0 and 1, I do not know. In fact, a circuit in a hardware can never have a value called X. X is not existing in hardware, but I do not know if I start it up whether it will be a 0 and 1. So, I have to be thing here at X is can be either 0 or either 1 my testing should hold even if it is 0 or 1 irrespective of this line e my test value should hold. That is I can easily sensitize this by making the line D equal to this as D I put I a equal to 1, done this is my propagation path.

Now, one main problem means is an OR gate so, the value will be passing over here. So, it will be a D over here and find it will be a D over here, right. Now, to get a D over here we are know that the input to the OR gate should be a 0. So, how do I get a 0? It is a very difficult direct to get a 0 equal to 0 because, e is now X. It can be either 0 or 1 if it is 0 your test can be done if it is that 1 your test cannot be done. If some of power up the value of Q was equal to a 1 so, it 1 one it will be become a value equal to 1, so, your testing cannot be done, If it is 0 it is guaranteed you can do the test because it is D. So, what I have to do I have to somehow indirectly make e equal to 1 and then only I will be able to test my circuit I will see how to.

(Refer Slide Time: 11:59)

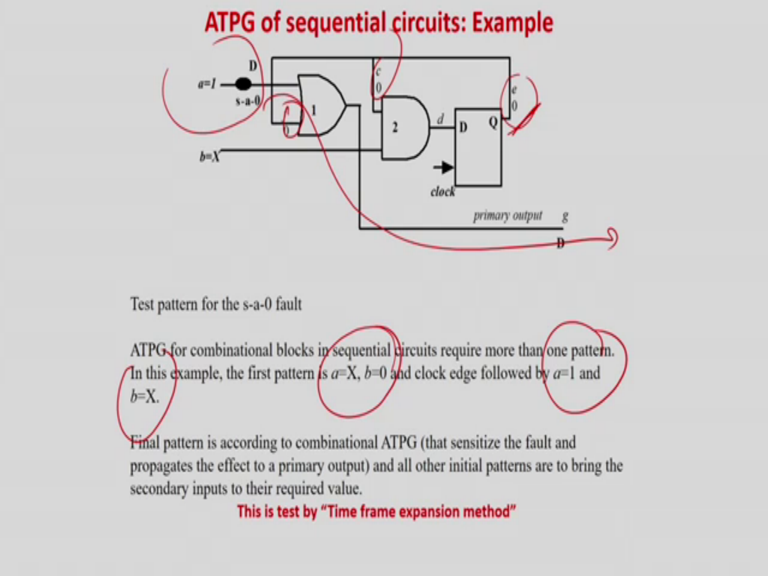


ATPG of sequential circuits: Example

Indirect controlling of *f* to 0

So, indirectly I have to make this line f e which is coming here as an f have to be made 0. How can I do that? So, one way is very simply I will make b equal to 0 because this part has a fault. So, this line I cannot use to make this gate as 0, this is a problem because this line a fault question is there. So, I cannot use gate number 1 to make the line e or f equal to 0. So, rather the game number two does not have a fault right now so, I am going to use it. So, I make b equal to 0 so, this c is equal to X so, irrespective of anything the value of d is equal to 0 now, I will apply a clock pulse.

(Refer Slide Time: 12:33)



ATPG of sequential circuits: Example

Test pattern for the s-a-0 fault

ATPG for combinational blocks in sequential circuits require more than one pattern. In this example, the first pattern is a=X, b=0 and clock edge followed by a=1 and b=X.

Final pattern is according to combinational ATPG (that sensitize the fault and propagates the effect to a primary output) and all other initial patterns are to bring the secondary inputs to their required value.

This is test by "Time frame expansion method"

So, once you do automatically your e is going to become 0. So, this is what I was saying as an indirect control. So, if this would have been just assume if this would have been cut, so, what I could have done? I could have made a X equal to a equal to 1, b equal to 0 or in fact, I could have made b equal to X there is no problem and I could have made e or c directly equal to 0, sorry.

So, I could have directly text like a combinational circuit I could make c equal to c e this e or c or basically f X equal to it means f equal to 0 even I could have direct control if this is a combinational circuit, then I would have put a equal to 1 d. So, these propagated over here and I go to add even in this distance.

But, now it is not possible why because this line e or the life is coming as f as input to gate 1 is connected to a flop. So, directly controlling is not possible. So, I now require instead of one, two test patterns to test it so, first I have to make b equal to 0. So, indirectly it will make this if this d equal to 0, I apply a clock pulse. So, what will happen e will become a 0 so, now, the feedback will come over it is a 0, now you can apply your required test pattern 1, it will be propagated over here that is tested.

So, now you require two clock pulses and two patters to do this. So, this is the first test pattern and second test pattern is equal to this. So, what do we see and this is called test by time frame expansion method so, what have we have seen. So, we have seen if there is a sequential circuit the number of test patterns required will be more than 1 to test it because, first you have to control all the flip flops to their required values so, that the fault can be sensitized or justification or whatever that is I have to bring certain nets of circuits to a some known values, like in this case I have to make it to 0. Had not be the combinational circuit you could have directly done this by using your in primary inputs, but here some of the primary input lines sorry input lines like the secondary input lines are not directly controllable.

So, first we have to control them the required values which I have shown you how can you indirectly control them and then only you can go for a combinational type of circuit testing. So, sequential circuit testing is simple like a combinational circuit testing only, but you there will be certain lines which you cannot directly control. So, we have to indirectly control them and bring them to the required values then only you can apply the primary inputs and get the circuit tested.

So, this whole backlog that all the required lines has to be brought to some control values before you can apply the real test pattern makes the circuit listing of sequential circuit extremely complex. So, and this method is called time frame expansion method.

What are the complexities involved? The complexities involved are first of all your test time will be higher because in combinational circuit you can apply one test pattern your job is done. Here you require n number of test patterns well I will tell you what do you mean by n over here? N is the sequential depth of a sequential circuit. So, that number of test patterns will be required in the worst case to control the required in directly controllable lines and then you have to apply the final pattern and circuit will be tested. So, tester will have more amount of time to test it.

One penalty as I told you having more number of test patterns to test a circuit actually is more expensive in terms of ATE cost or you have to higher the ATE. Secondly, this pattern generation algorithm will take more time to test generate the test patterns, like in give up sequence combinational circuits I have shown you sensitize, propagate and justify you do you get the test patterns for your difficult to test faults.

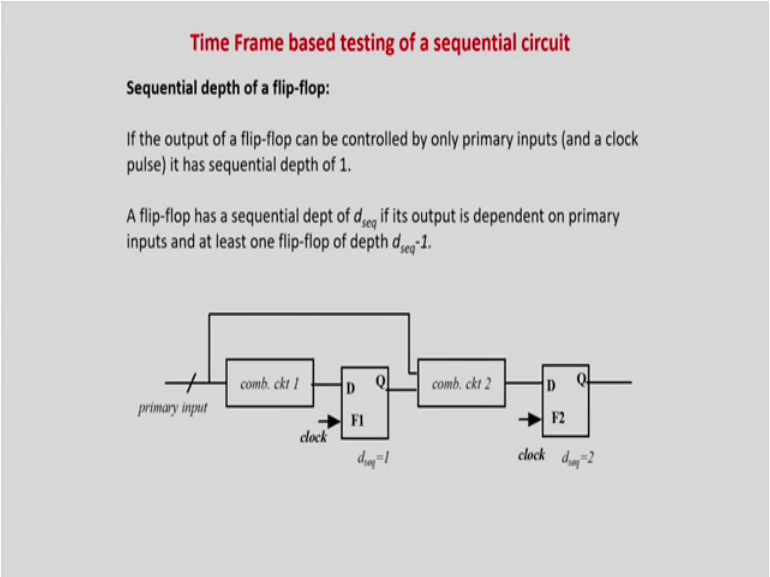
Random test pattern also you can easily appreciate that it will be more difficult to find out that which random bad. ~~because~~ Because, in this case also there will be no a single random pattern then be a set of back random patterns which will be required to test these faults so, it becomes a long process. Anyway, let does not go by this random test pattern business right now, let us assume that we are testing by sensitize, propagate and justify approach,

So, in case of some combination circuit three stage; sensitize, propagate and justify. If the have iterate you may have to go for another or some rounds of iteration if there is a conflict. But, here if you observe we are having multiple rounds of sensitize propagate and justify in a single circuit itself. Because not only you have to find out how to sensitize propagate this line also this line has to be controlled to 0. So, that means, I have to abrogate a 0 over here, to get a 0 over here means it can be somehow sensitizing it in fact, propagation will not be there, but at least justifying it.

So, sensitize propagate and justify will be for this test pattern to be tested, but I have to bring some of the code like internal lines like which are the secondary input lines to some desired values before the test pattern can be applied, for example, 0 in this case. So, at that means, what I have to sensitize this lines what I can say that to bring it to the desired lines and for that there could be lot of backtracks that is actually justifying of this lines of whose propagation will not be involved which is required to set this secondary lines.

But, still you can understand the complexity becomes much much higher. So, if there are hundred lines which has to be brought to the control values and there are hundreds of flip flops. So, the sensitize propagate and justify kind of approach has to be done multiple number of times.

(Refer Slide Time: 17:33)



Time Frame based testing of a sequential circuit

Sequential depth of a flip-flop:

If the output of a flip-flop can be controlled by only primary inputs (and a clock pulse) it has sequential depth of 1.

A flip-flop has a sequential dept of d_{seq} if its output is dependent on primary inputs and at least one flip-flop of depth d_{seq}-1.

So, sequential circuit testing not only requires more time on the tester, but also the test pattern generation complexity is much higher. This philosophy you could have easily observed over here, because not only the sensitization of the fault propagation and justification requires to be done, but for that lot of secondary input lines has to be brought to them required values and for that again I have to do a lot of processing.

Because, I have to bring these lines to a certain values and for that basically I have to do lot of back tracks that is justification how to get this values. Of course, there will be conflicts and then again you have to do recreation. So, it is a more complicated in nature, I think you have got the philosophy by now.

Now, I will give you some basic terminology so the I can you can get an idea how much time which require to go for automatic test pattern generation for a sequential circuit and how many test patterns will be required to test a circuit in the worst case. So, we call a sequential depth of a circus as d sequence. Now, what is that? Basically see this is one flip-flop. The input of the flip-flop depends on the primary input and nothing else we call it as depth of 1.

This is the flip-flop whose input depends on this primary input as well as on another flip-flop whose depth is 1 so; this sequential depth will be equal to 2. So, sequential depth of a flip-flop means if the inputs is dependent on lot of primary input as well as some flops, like in this case 1.

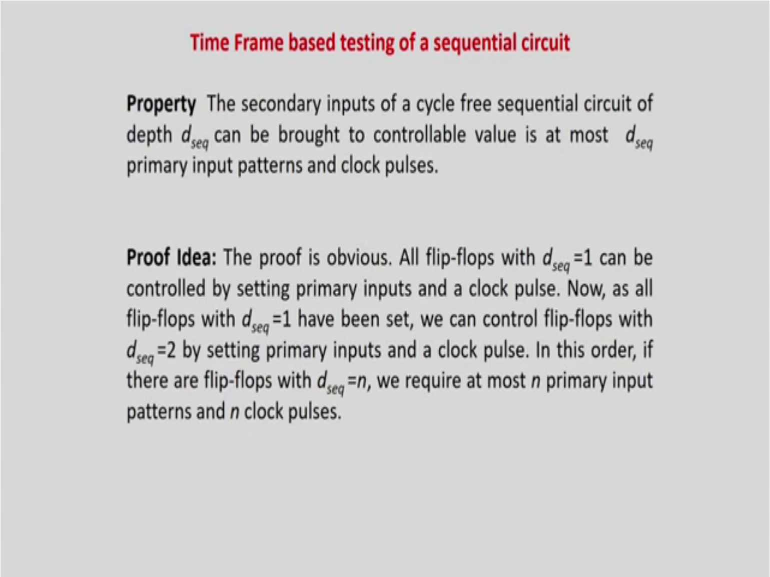
So, what is the maximum depth of this flip-flop is called the sequential depth of this next element plus a 1; that means, maybe say that this depth is 10 and then is deep this depth will be actually 11, maybe there are some other flip flops also involve whose depth is 1, 7, 9. So, we will take all of them and which one is the maximum depth on whose or whose output this input is dependent we have to add one just like levelization of the circuit so, that is actually called the sequential depth. So, the of flip-flop which with input is only dependent on primary inputs and combinational circuits my series of depth one like the very obvious definition.

Now, very importantly you can have a feel that how much time I require to control this line. So, in this case how much clock pulses will be required? First clock pulse will be required to control this and second clock pulse will be required to control this. So, as much the depth that may number of test patterns will be required to control the circuit. Of course, if there are no means of course, you will require time to generate those test pattern I am not saying that at the first iteration you will be able to generate the test patterns there may be conflicts a lot of iterations may be there.

But, mathematically you can easily prove that there exist as patterns or patterns in which in 1 time unit you can control this. So, once this is controlled this line is controlled so, and another state you can control this again I will reiterate. So, basically in the first step you can put appropriate values over here so, you will easily get the required values over here.

Next stage, these values are controlled and this you can already know. In the second stage you can bring your respective control values over here so that in this step 2 you can have the required values at the output. So, if a sequential circuit is with depth of n then with n test patterns maximum you can basically get all the these are the secondary input lines we can get it to a control level value.

(Refer Slide Time: 20:36)



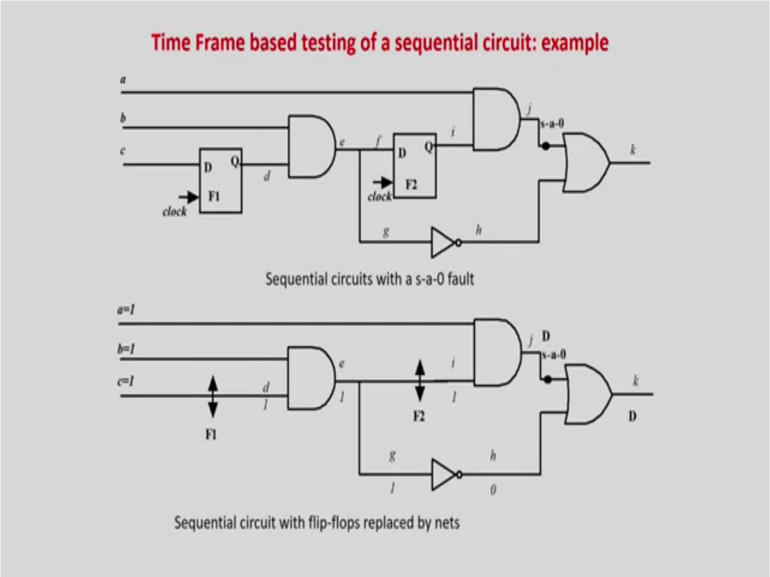
Time Frame based testing of a sequential circuit

Property The secondary inputs of a cycle free sequential circuit of depth d_{seq} can be brought to controllable value is at most d_{seq} primary input patterns and clock pulses.

Proof Idea: The proof is obvious. All flip-flops with $d_{seq}=1$ can be controlled by setting primary inputs and a clock pulse. Now, as all flip-flops with $d_{seq}=1$ have been set, we can control flip-flops with $d_{seq}=2$ by setting primary inputs and a clock pulse. In this order, if there are flip-flops with $d_{seq}=n$, we require at most n primary input patterns and n clock pulses.

So, that is the property says that the secondary inputs of a cycle free I will tell you with an example what do you mean by cycle free. The secondary inputs of a cycle free sequential circuits of depth this sequence can be bought to control level value is at most d sequence primary test patterns and clock pulses; that is even a circuit is a lot of sequential depth. So, we will require lot of test patterns basically to make the indirectly control labeled lines to control values then you can apply the final test patterns. So, the proof idea is given over here which I have told you is very simple and easily intuitive.

(Refer Slide Time: 21:07)



Time Frame based testing of a sequential circuit: example

Sequential circuits with a s-a-0 fault

Sequential circuit with flip-flops replaced by nets

Now, we will take a more elaborate example to tell you what do you mean by testing of a circuit using time frame expansion method? So, assume that the let us take that this is the first the top figure solar circuit there are some flip flops and I have to test this circuit. So, basically we know that stuck at 0. So, what I initially do what is the algorithm for this also I tell you the algorithm to test it algorithm to generate the test patterns is very simple.

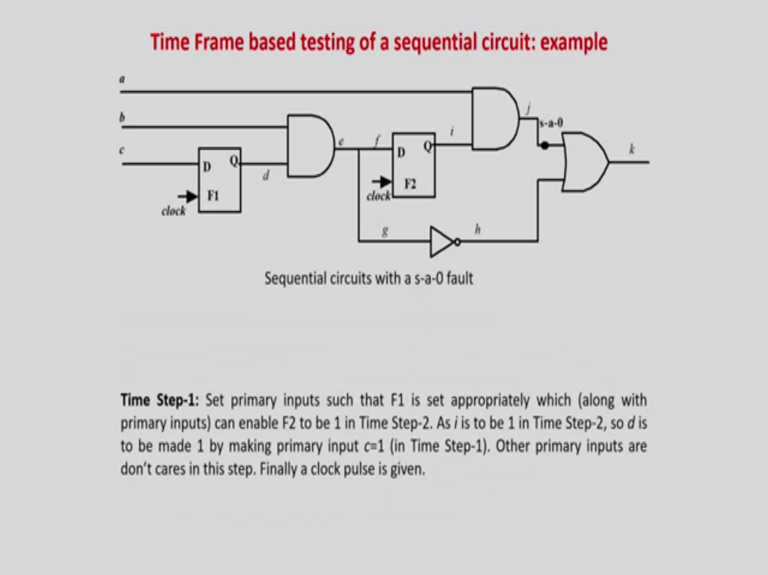
So, first you just forget that there is no flip flops over here you just put a marker that there the flip flops over there. Then just we have to go for a sequential combinational ATPG. So, stuck at 0, so, I have to put a 1 definite will be D this is the propagation path so, for propagation is done; now I have to get the values out.

So, this is an OR gate to propagate the value we require a D over here 0 over here, to get a 0 over here I required and over 1 over here. Similarly, this AND gate to propagate D over the I require 1 over here. So, fine there is no conflict, this one is 1 here and this one is also required at i so, e equal to 1 is a non conflicting value. So, if you require a 1 over here you require 1 1 over here and basically to get a 1 at the output of the end this is what is required.

So, very easily you can see that without conflict fortunately in one round we have done your combinational ATPG. So, any sequential circuit you get remove all the flops and go for a combinational ATPG so, that complexity is remaining it. Now, what extra? Now, you can see that I have to get d equal to 1 which I cannot do directly because this is a flop sitting over here.

Similarly, I require i equal to 1 so, that also I cannot do directly because it is a flop sitting over here. So, now, again I have to apply this sensitized and justify approach to line d 1 and line i 1, so that I can get the required values in the flip flops. So, more in the depth of the flip-flop that much computational complexity will be increased for automatic test pattern generation. Combination circuit text pattern generation anyway we have to generated, now special care has to be taken for these two lines now, we will see over there.

(Refer Slide Time: 22:58)



Time Frame based testing of a sequential circuit: example

Sequential circuits with a s-a-0 fault

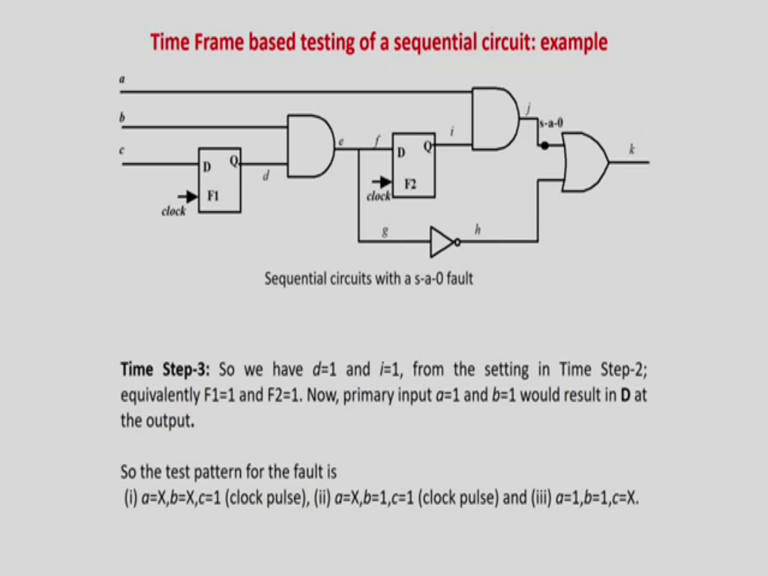
Time Step-1: Set primary inputs such that F1 is set appropriately which (along with primary inputs) can enable F2 to be 1 in Time Step-2. As *i* is to be 1 in Time Step-2, so *d* is to be made 1 by making primary input *c*=1 (in Time Step-1). Other primary inputs are don't cares in this step. Finally a clock pulse is given.

So, we require a 1 over here we require a 1 1 over here. So, what I require to do it over here first you have to make this line as a 1, I require a 1 over here so, two time steps ahead. So, I first require a 1 over here and I first require a 1 over here, I should get a clock pulse then the value will come over here. So, of course, to get a 1 over here I require a 1 over here. So, in the first time stage what I will do first time say I am targeting i equal to 1.

So, I will put a 1 over here and I can put a X X over here~~.~~, I will give you get a clock pulse so, this one will be coming over here as 1. Next what I do? Next basically I have to target that this line should become a 1 as well as this line should remain as a 1. So, what I will do I will put a 1 over here and instead of b equal to X, I will make a (Refer Time: 23:46) I will apply a clock pulse.

So, this 1 will now propagate over here and this 1 will come over here. So, you will go you are going to get a simple picture which is show shown over here. So, these are all shown in steps I have written down the steps over here.

(Refer Slide Time: 24:00)


Time Frame based testing of a sequential circuit: example

Sequential circuits with a s-a-0 fault

Time Step-3: So we have *d*=1 and *i*=1, from the setting in Time Step-2; equivalently F1=1 and F2=1. Now, primary input *a*=1 and *b*=1 would result in **D** at the output.

So the test pattern for the fault is
(i) *a*=X,*b*=X,*c*=1 (clock pulse), (ii) *a*=X,*b*=1,*c*=1 (clock pulse) and (iii) *a*=1,*b*=1,*c*=X.

So, finally, a time step 1 you are going to get this one equal to 1, this one equal to 1. Now, basically this is equal to 1 so, this is equal to 0 and now I will just require a equal to 1. So, first stage basically you have to have a situation like this that is c equal to 1 will propagate the value second clock stage basically you have to apply it a equal to not require b equal to 1 and c equal to 1.

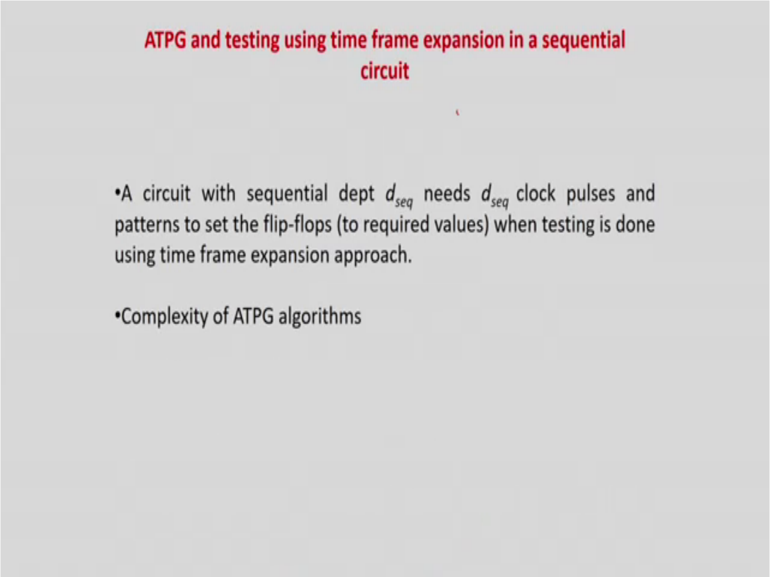
So, that you get the value of 1 equal to here and finally, we require to finally, when all this is set that is this one is sorry when this one is 0 this 1 is 1 basically this one is a 1 so, you require a 1 over here. And in fact, right in the in the of course, I require a 1 over here because this is 1, this is do not care your circuit will be tested with the value of D over here.

So, now you can see this is how a sequential circuit is test pattern is generated. So, now, you have to see you again I will tell you what is the complexity involved. So, first complexity involved was very simple, just like a combinational test pattern we have to do it. Now, these extra wherever there are flip flops you have to put extra load to find out how can I make a 1 over here and how can I make a 1 over here. In this case is a direct line~~. So~~so, it is very simple that I can tell that I put c equal to 1 are you are going to get the value of 1. Assume that this is a very big combinational cloud lot of primary inputs are there and I require a 1 over here.

So, you have to do lot of back tracks like we have done in case of a combinational test pattern generation to maybe I have a AND gate, there you will have OR gate, there even XOR gate. So, all fitting over here and I require a 1 over here, then again you have to be or back propagate and you have to justify that how do I get a 1 over here. So, you can understand it that for at least for all the flip flops involved to give the control level than if there is a combinational cloud over here you have to do a lot of justification approaches to get this value over here.

So, you can understand that if there n flops so, at least that many amount of justification of these values has to be done. So, it is so, the complexity remains like 1 time full or ATPG for the combinational circuits and then whenever you have such kind of flops whose values are required at least at that places you have to sensitize and justify. Sensitization is very simple you can directly say that I require a 0 over here or I require a 1 over here based on the requirement and then you have to justify how I will how I can get one of the primary inputs I have to send, so that I can get the required values over here.

(Refer Slide Time: 26:21)



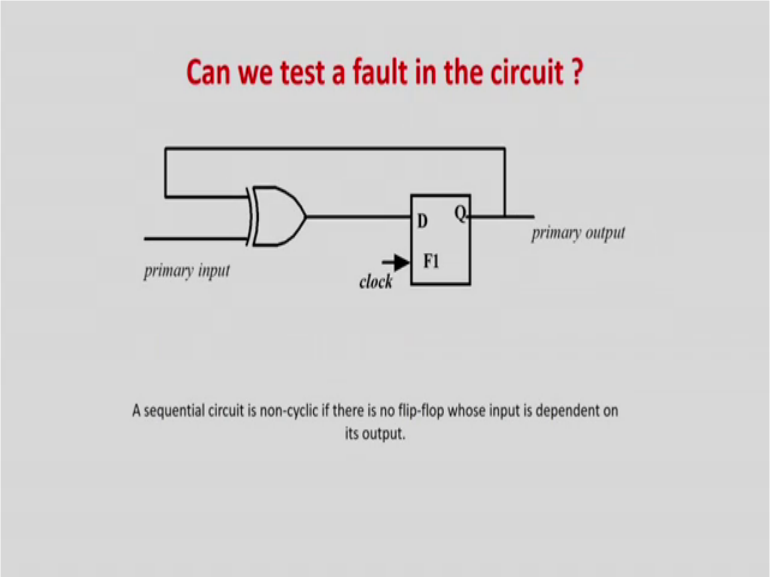
ATPG and testing using time frame expansion in a sequential circuit

- A circuit with sequential dept d_{seq} needs d_{seq} clock pulses and patterns to set the flip-flops (to required values) when testing is done using time frame expansion approach.
- Complexity of ATPG algorithms

So, basically the complexity becomes almost n into the combinational complexity. Combination complexity is only 1 sequence sensitize propagate and justify and in case of sequential circuit if the circuit depth is n, so, n times you have to go for sensitize and justify. Of course, propagation is not required to get the values of the flip flops the required ones. So, you can say in a overall complexity the if the combines combinational complexity is X so, it will be n X your order. What is the n? N is the sequential depth of this circuit.

So, that is actually complexity of the algorithms so, generally it is high. So, this algorithm I have told you can easily formalize and write it just like the combinational circuit ATPG algorithms. So, complexities at the high and secondly, the more important part is that one pattern can test a combinational circuit, but in sequential circuit testing we may require d q, d sequence number of test patterns and clock pulses to do it, this is the more killing factor.

(Refer Slide Time: 27:14)



Can we test a fault in the circuit ?

A sequential circuit is non-cyclic if there is no flip-flop whose input is dependent on its output.

Because, test patterns has to be generated only once that is the test plan for all circuits. The same test pattern will set will be required for all the circuits to testing, but for all physical samples you have to apply the patterns. So, here the number of patterns have also increased 1 plus sequential depth, tester will require more amount of time. So, therefore, sequential circuit testing is more complex compared to combinational circuits but, anyway we cannot create a magic that has to hold.

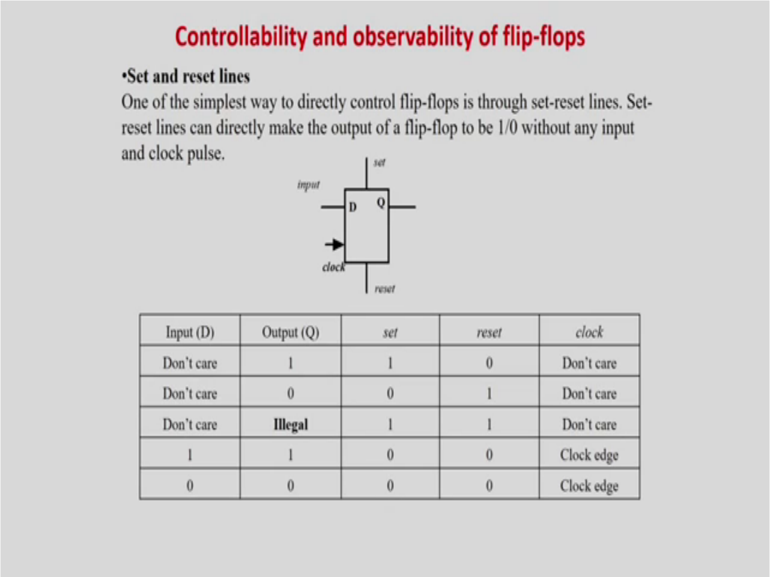
Now, the question is can I do it always? The answer is no. There are some circuits which are called cyclic circuit here why the cyclic circuit because, the input of this circuit and the output of the circuit depends on the input of the circuit. Like this one is the output of the combinational block this input actually depends on the again the out so, this is the cyclic circuit.

So, cyclic circuits may not always be testable as the example I am showing you. Let us assume that these are stuck at 0 fault over here I assume or make it let me make it more simpler I assuming that this is the stuck at 1 fault in this line. Now, this is X now to tests are stuck at 0 fault what we have to do we have to put the value of 0 over here. So, 0 over means is that D prime and I should get a D prime over here, to get a D prime over here I should get a 0 over here XOR gate, right.

Now, how do I get a 0 over? It is not impossible because indirectly I require a 0 over here to indirectly if I require a 0 over here you should have either 0 0 over here or 1 1 over here, but in this case to the cyclic nature of this fault you will always oscillate and you can never get the values to any control level value in this case. So, we cannot never can get a control value of 0 or 1 in this because I want this line to be controlled. So, you cannot get any time any controllable value of 0 or 1 in a indirect manner over here it will always remain X. This happens because this input and output are tied together.

So, if the cyclic circuit is high click there may not exist any test pattern to test it, then how do you do? And, lot many sequential circuit may have some cycles so, how you will do it? You have to remember that in case of combinational circuit untestable faults are very very rare they are basically our redundant force very there, but in case of sequential circuit these are not untestable faults. But, only because of a cyclic region of the circuit cyclic nature of the circuit you cannot control these lines and therefore, these faults basically are not able to be tested.

(Refer Slide Time: 29:35)



Controllability and observability of flip-flops

•**Set and reset lines**

One of the simplest way to directly control flip-flops is through set-reset lines. Set-reset lines can directly make the output of a flip-flop to be 1/0 without any input and clock pulse.

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

Now, we will come so, people started thinking about it they thought that this time frame based expansion method circuit testing will tell you anyway fail if the circles circuits are cycling. So, how they do? They started using the sorry the set and reset lines because, set and reset lines you can you can easily observe that if there was any kind of set and reset lines were there what I would have done it I would have made the reset line equal to 1, set line equal to 0.

If I make it the reset line equal to 1 so, directly this line will become a 1. So, reset line so, this line will become a 0. So, this 1 will be become a 0 stuck at 0 means you apply a 1 so, it is D. So, it is a XOR gate; XOR gate with 0 bits the non inverting circuit, so, it will be D and basically you can test it to be D as simple as that.

So, and of course, why you take this circuit you have to make this s is equal to 0 then I will come to that. So, the idea is that somehow if I could have got this set and reset lines controllable then I could have easily controlled all the flip flops so, that is what is the idea. The same circuit they have taken, so, what they have done. So, first they have made the set line equal to 1 so, if the set line is equal to 1 this actually becomes a 1.

Now, primary input s equal to 0 D prime. So, anyway you can you could have also made this line as 0 or this line as 1, anyway it will work for this case as an XOR gate because we know that XOR it is a control inversion in case of 1 because this is an XOR gate if you take the inputs as 0 0 the answer is 0 but, if one line is a 1 so, the answer is a 1.

So, this line is actually called the control line so, it is a 0 you will get whatever is the input. But, if I make this line as a 1 then if I put a 1 you are going to get the answers as 0 and if this line is a 0 you are going to get the 1 and line is 1. So, if this line is a 1 it is a controlled inversion becomes inverted and if the control line is a 0 whatever if the output will pass.

So, in this case what they have done, they have taken the set equal to 1; that means, they will be inverting the XOR gate. So, in this case you get a D; D prime and then basically first you basically make set equal to 1 reset equal to 0 so, directly this line is becoming 1. So, without this set reset line you cannot do anything you cannot indirectly control such cyclic lines so, set reset lines you have to use.

Now, you make both of them as 0 so, circuit comes is this working mode you put a 1 over here stuck at once or you put a 0 over here so, the stuck at 1. So, D prime is your requirement and then it will be inverted and D and of course, you can very easily test the fault effect to over here after a clock pulse the value will be latched over here. So, what is shows? So, it shows that if some circuits are complicated with cycles you have to have set and reset lines used.

Now, we are again going back to the old example in which case we have just tested our circuit in three clock edges. So, this was a real problematic circuit for testing could not be done without the set reset line, but if you look at this circuit we have already tested this circuit using three test patterns. Now, same I will try to do with set and reset lines. If you remember I require a 1 over here and I require a 1 over here and we have done lot of backtracking etcetera to do it.

Now, I will not do anything, I will simply apply set equal to 1, here also set equal to 1 thing is that the set reset lines are primary line primary input lines it is assumed. So, they are brought out of the circuit box that you are dual inline package is there from the package they are brought out an extra pin outs.

So, I make reset equal to 0, set equal to 0, so, obviously, this is becoming 1 and this is going to become a 1. So, now, so simple life is now so simple basically you need not go for any kind of a backtracking and also sensitize justify because so many things have to things that 1 propagates over here and then again 1 propagates here so much mathematical calculations I had to do. So, here if I have the direct hold up the flip flops so, I made set equal to 1, set equal to 1 reset equal to 0 so, I get these values directly.

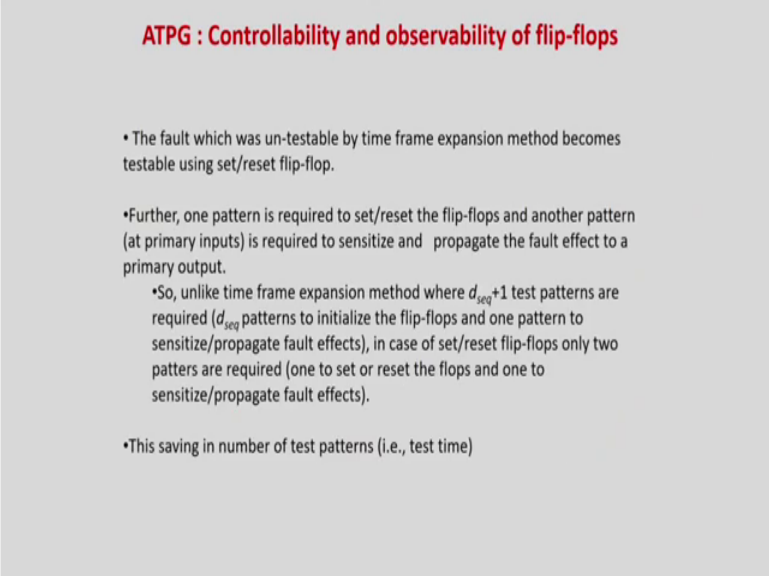
Now, next stage I make all of them as 0 because, I want the circuit to be in working mode and then what I require is the input this is the last pattern you have to apply and you basically job it tested. So, all of the advantages, they are lot of advantages the first advantage is ATPG is very simplified because now in the previous case I told you have to do an ATPG for the combinational circuit, then you have to do lot of sensitize and justify approach.

So, that how can I get d over here d equal to 1 over here, how can I get i equal to 1 one over here? Here I will not equal to do anything because, I know that if I require a 1 over here you have to make set equal to 1 and reset equal to 0, if I got a 1 over here I reset set 1 and reverse in the case of others.

So, whatever the flip-flop positions like in this case like as I told you have to first remove all the flops and you have to do a ATPG. So, I required this is equal to 1 and this is equal to 1. So, then in the time frame expansion method base testing you have to again try to see how I can get d equal to 1 and i equal to 1. Here I have no worry because I can control d equal to 1 and i equal to 1 and any other values in this flops by directly using the set and reset lines. So, I did not do any kind of any kind of ATPG for this. So, only one combinational ATPG will do the job and after that there is no more time step required only one test pattern will do the test.

So, what is the test pattern? The test pattern is basically your primary inputs and to test patterns basically first test pattern is you have to set and reset all the flip-flop lines as required and the second test pattern is the required ATPG for the combinational circuit you have to apply. So, in two patterns we are done so, the final what is the first test pattern? These two secondly, we are making both of them reset and this is test pattern your reply and your test is done. So, few the advantage test generation time is complexity is less, less test about generation time and less number of test vectors to test it.

(Refer Slide Time: 35:08)



ATPG : Controllability and observability of flip-flops

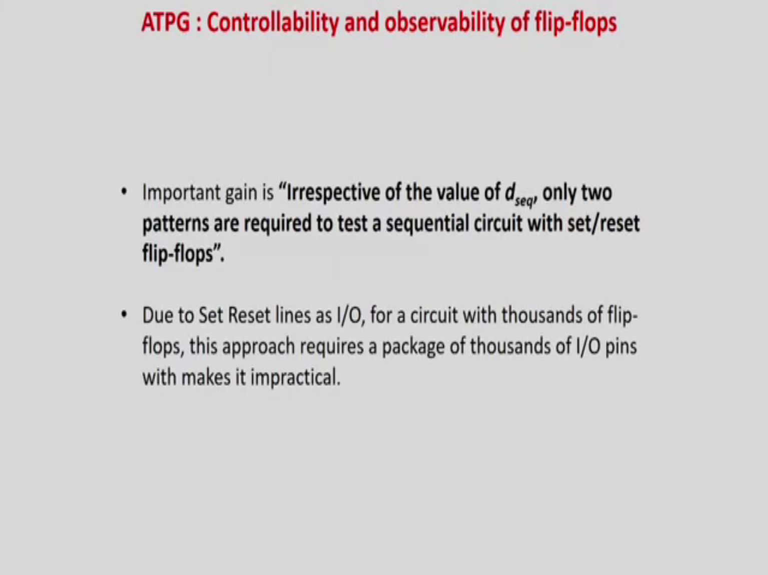
- The fault which was un-testable by time frame expansion method becomes testable using set/reset flip-flop.

- Further, one pattern is required to set/reset the flip-flops and another pattern (at primary inputs) is required to sensitize and propagate the fault effect to a primary output.
  - So, unlike time frame expansion method where d_{seq}+1 test patterns are required (d_{seq} patterns to initialize the flip-flops and one pattern to sensitize/propagate fault effects), in case of set/reset flip-flops only two patters are required (one to set or reset the flops and one to sensitize/propagate fault effects).

- This saving in number of test patterns (i.e., test time)

However, large your circuit is you just require two text tester two testing 1 is to set reset flip flops and secondly, you have to just apply the test patterns so, very lucrative. So, I think we should always go by testing of circuit with set reset lines, but what is the problem so, this is the good things about it. So, the fault we chose untestable by time frame expansion method becomes testable by set reset lines, only two test patterns are required to test it these saves in the number of test pattern and test time so, very good, I should use it.

(Refer Slide Time: 35:33)



ATPG : Controllability and observability of flip-flops

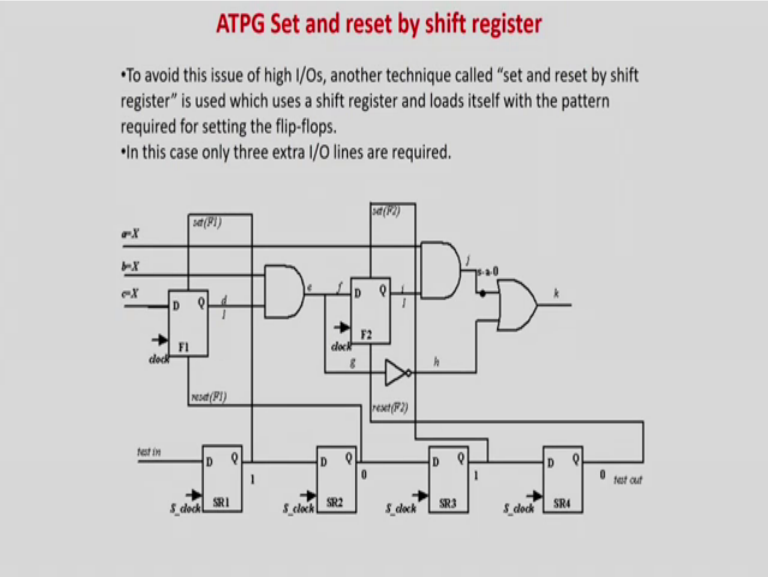
- Important gain is **"Irrespective of the value of d_{seq}, only two patterns are required to test a sequential circuit with set/reset flip-flops".**
- Due to Set Reset lines as I/O, for a circuit with thousands of flip-flops, this approach requires a package of thousands of I/O pins with makes it impractical.

But, and one important gain is irrespective of the value of d sequence only two test patterns and equal to test the circuit why? Because as I told you test pattern generation complexity is important, but not as important as the number of test patterns. ~~because~~Because, more number of test patterns more number of times you require in ATE and you have to apply the test patterns to all the physical samples. So, I am very much worried about the number of test patterns. Slightly more time I can give you to generate the test patterns offline that is a test time you can take some more time to do it.

So, therefore, you can see that it is very very good. So, you just require two parallels to test it and of course, test and generation time is also coming to lower because you can directly set or reset the flops. So, no need to worry about how to set and reset the flip flops by a back propagation approach, not required. What is the problem I think we should always go by this. The main killing factor is the number of set reset lines which are became the primary outputs 100 flip flops 200 extra pin outs will be there because the set and the reset lines, 10000 flip flops 20000 extra pin out.

So, again big problem just like we structural testing at the function of the gate levels. That is structural testing without a fault model was the killing factor over at the combinational circuit. Here all there what was the problem that if you try to test all the gate functionality that is structural testing without a fault model there was lot of pin outs, then fault model solve the purpose. But in sequential circuit even with the fault models that problem is not solved even if you are using set reset flops because all the set reset lines has to be brought at the output. So, that actually made the work very impractical.

(Refer Slide Time: 37:02)



ATPG Set and reset by shift register

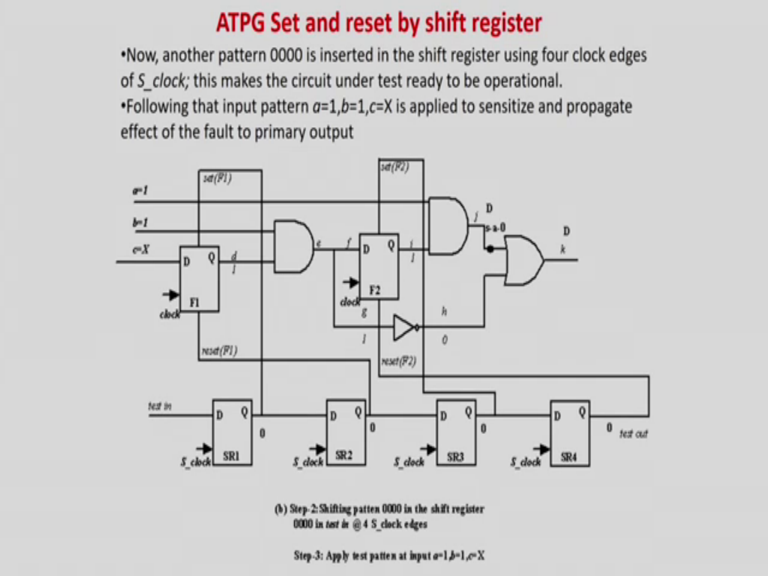
- To avoid this issue of high I/Os, another technique called "set and reset by shift register" is used which uses a shift register and loads itself with the pattern required for setting the flip-flops.
- In this case only three extra I/O lines are required.

Then, we will started thinking how we can solve it then, a revolution came which is called the scan chain. So, if you ask me what do you think is the most revolutionary discovery in testing, I would tell invention interesting I will say it scan chain such an interesting discovery a invention as we will see is solve all the problems or maximum of the problems of sequential circuit test. So, some people thought that the major problem is because of bringing all these pin outs circuit is a problem.

So, what I do, I put a internal memory that was the first that bringing this pin outs of the flip circuit is a problem. So, what I do I actually bring make a internal shift register of the internal memory and then I will put the values required for this. And let us see so, in this case there are four lines so, there is a four ship register fault bit shift register I will make. So, what was the requirement 1 0 1 0 so, I will not directly give it as a input, rather the output of all the sets reset lines are the outputs of all these are connected to the shift register output.

So, you see I require 1 0 1 0 so, I require our set equal to 1 reset equal to 0 once you. So, with the shift register so, what I will do I will have 1 0 1 0 and I will apply four clock pulses. So, this value will go in the shift register and it will be done so, no extra pin outs is required. So, for the shift register basically you require extra only two pin outs test in is a pin out and so, because you have to shift the values in this case. So, I require basically a extra only a single pin and a clock pin an extra clock pin for this because the clock for this one and the clock for this if registers will be different. So, first I put 1 0 1 0 in the shift register and apply four clock pulses your job is done.

(Refer Slide Time: 38:46)



ATPG Set and reset by shift register

- Now, another pattern 0000 is inserted in the shift register using four clock edges of *S_clock;* this makes the circuit under test ready to be operational.
- Following that input pattern *a*=1,*b*=1,*c*=X is applied to sensitize and propagate effect of the fault to primary output

(b) Step-2:Shifting patten 0000 in the shift register 0000 in *test in* @ 4 S_clock edges

Step-3: Apply test patten at input *a*=1,*b*=1,*c*=X

Then actually again I have to make all these lines at 0 0 00 so, again I have to put 0 0 0 0 and four clock pulses to do it your job is done and the circuit can be tested. Now, what is the problem? Problems are now twice. Test generation complexity is anyway not touched because I just use the same combinational logic and I just can directly control the flip flop. So, test generation algorithm time becomes reduced even with this approach, but what more I have to do? I require four extra flip flops.

So, for every 2 flip flops I will require extra 2 flip extra 2 that is n flip flops are there I required 2 n flip flops to make these lines controllable because, I am not controlling them output directly primary input primary output, only I am doing it on shift register. So, double the number of flip flops are required to control this. Secondly, I require 2 n number of clock pulses to test this circuit why? Because first I have to make the required set reset lines and then I have to make all the lines as 0’s so, that is not acceptable.

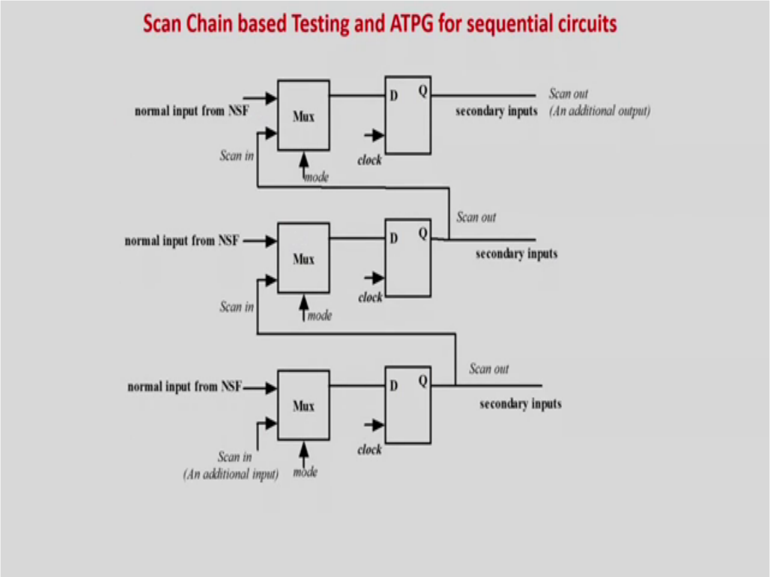
So, now the test time has increased so, the test time is now 2 into n actually it is 4n basically, where n is the number of flip flops. So, 2 n to set all set reset the flip flops and then again another 2 n clock pulses to make all the sled set reset lines to be 0 so that this circuit can be brought to a working known that is not actually acceptable. So, it is a huge time taking as well as a huge hardware over it is required for this. So, this happens nobody care, nobody took it, but then a revolution this is what is basically I have told you about the theory you can read this slide, but I will directly go to an example.

Then actually the revolution came then some people thought, just look at this circuit, some people thought that already these two flip flops are there. So, why do I require others four flip flops to make a chain like this? Can I somehow implement this as a chain so, this is the shift register I am putting to control this. So, can I do something like so that decide this existing flip flops where I can make a chain and I can directly control them?

So, there should be two modes of operation; one mode all the flip flops these features will not be there anyway this is not good actually because, we will have put it to test and to from and double the area you are using to test the circuit. Not only about area double the time you are consuming to bring in the test patterns that is what is the main drawback here.

So, what people say just forget about the extra flops. Can high somehow make a chain among these flip flops? If I can do this then I can then this basically shift register is implemented by this existing flops only and then without increasing any area over here I can get the values of 0's and ones whatever is required at the out. This is basically called scan chain and that revolutionized testing I will give you examples to clear it. So, just really look at this circuit I mean this look at this light whatever I have told you about the scan chain in the idea is written over here.

(Refer Slide Time: 41:36)



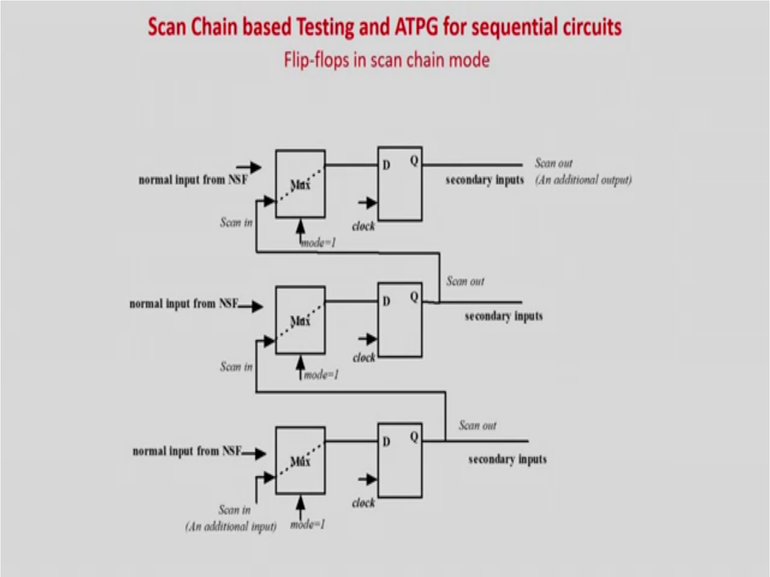
Scan Chain based Testing and ATPG for sequential circuits

So, there are they said that now there will be two modes of the circle. So, if there are two modes of the circuit you should have a marks to controlling so, that is your next state function block and is your flip flops. Now, what they have done they are put on extra marks if before every flip flop. So, one input to this is the directly input from the next input between normal mode another is basically coming from the output of the previous flip flop.

Secondly, for the seconds from flip-flop the same thing will be there one value is from the normal next step function block another will be the output from the previous catchy. And finally, it is the last flip flop so you will have an additional input which will be coming over here and the clock over here. Now, this is I have to tell you that these are all the existing flip flops only. So, only I have added some multiplexers in between that so that I can make a chain out of it when the circuit in the test mode.

So, you can see the circuit is a test mode you will have a chain like this and whenever the circuit is there a functional mode, sorry whenever the circuit will be in a functional mode then therefore, then what is going to happen these lines will all be cut. So, this will be the direct connection and you circuit will be the operating mode, this is the fundamental idea of a scan chain.

(Refer Slide Time: 42:44)



Scan Chain based Testing and ATPG for sequential circuits

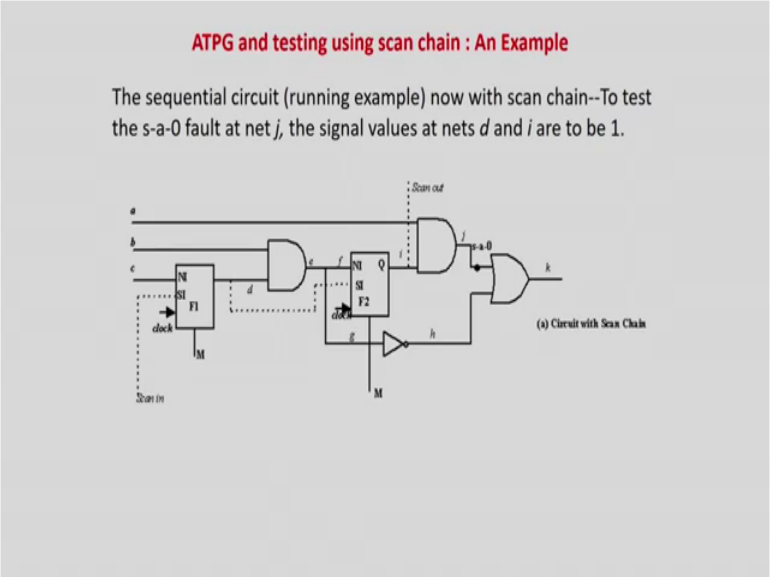
Flip-flops in scan chain mode

So, this is what is the figure? So, when the circuit is the scan mode. So, you can see they will be all connected in a chain and I can easily make it a shift resistor. If I require this as 1 0 1 so, I can put 1 0 1 and flip-flop process present and your flip flops are set. Of course, there is no set and reset lines in this flops. Having a set and reset lines is not at all helping in testing as I showed you either using an internal memory or bringing them as input output not at all solving the purpose because either you can have that many I O pins.

Now, that you can dedicatedly put such a huge number memory to test your circuit as well as having more number of test patterns so, nothing is being solved. So, this arrangement actually revolutionized the concept of testing, using the existing flops you make a chain and do it.

So, if you are using the existing flops extra hardware is only some of the multiplexer and multiplex 2 is to 1 multiplexer does not consume much area. So, without consuming much area I can easily set your flip flops to the required levels and even if it circuit is a normal mode you can see all these chains are decoupled and your circuit is operating in a normal mode. So, by using it just 2 is to 1 marks before all the flip flops I can easily convert it into a chain and when it in normal mode they will operate as a normal way.

(Refer Slide Time: 43:51)



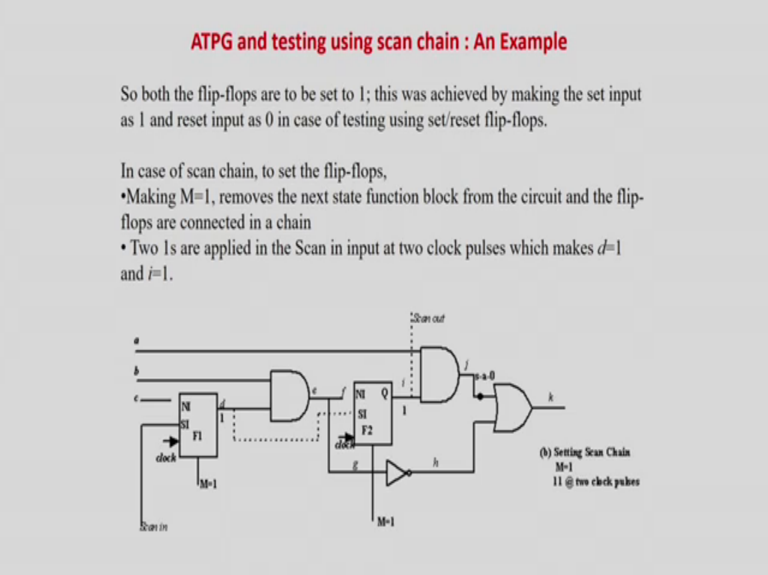
ATPG and testing using scan chain : An Example

The sequential circuit (running example) now with scan chain--To test the s-a-0 fault at net *j*, the signal values at nets *d* and *i* are to be 1.

I will give you an example to make it more clear. So, this was our circle and if you remember this is our stuck at 0 fault. So, I will put a 1 over here so, it will be D it will be a D it would be a 0. So, I require a 1 over here, I will require a 1 over here, I require a 1 over here~~.~~ ~~So~~so, of course, you require this as a 1 then require this as a1. So, one way of doing it is as I told you could have set reset lines then more pin outs other you can do a backtrack line. So, backtracking made mode number of test patterns as well as more number of test complexity.

In case of flip flops the scan chain based tests the complexity of testing remains same. So, I will just tell you that I require a 1 over here I require a 1 over. Now, I have to give the values, but neither there is any kind of a set reset line as the internal memory. So, I will have to use this, but there is a internal chain with this flops.

(Refer Slide Time: 44:41)



ATPG and testing using scan chain : An Example

So both the flip-flops are to be set to 1; this was achieved by making the set input as 1 and reset input as 0 in case of testing using set/reset flip-flops.

In case of scan chain, to set the flip-flops,
•Making M=1, removes the next state function block from the circuit and the flip-flops are connected in a chain
• Two 1s are applied in the Scan in input at two clock pulses which makes *d*=1 and *i*=1.

(b) Setting Scan Chain
M=1
11 @ two clock pulses

So, I have shown you in this figure so, I go to the test mode whenever you go to the test mode the primary inputs of this line and this line if you observe are cut. Primary input means the inputs from your or the main inputs which is the input for the next state function block or decoupled, but cut. So, this f 1 what I have shown is nothing, but your this one is actually f 1. So, what is this? This is the flip-flop with a multiplexer ahead.

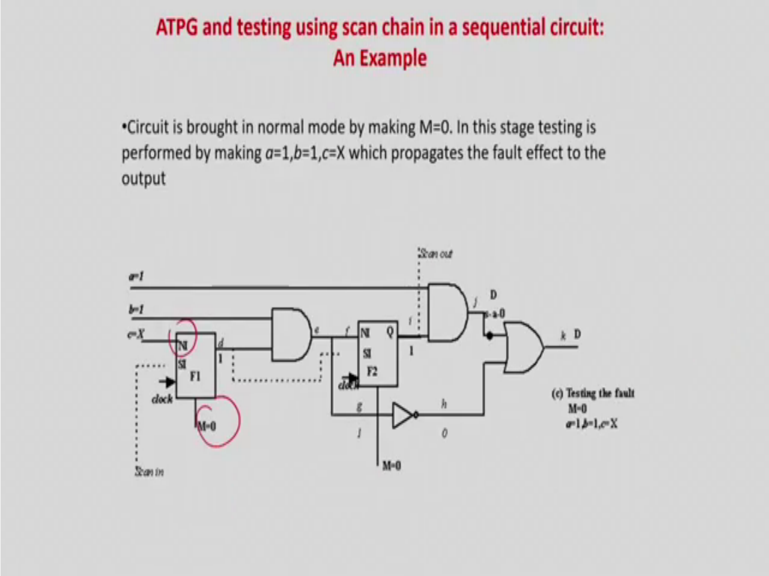
So, M is a mode called test mode so, if the test mode is equal to 1, so, in this case this line is decoupled and whatever is the other input is coming to the output of the flip-flop and if I make a equal to 0, so, basically a test mode equal to 0. So, basically the normal input is connected and that way so, anyway just cover it over here.

So, I will first make M equal to 1, if M equal to 1 the normal part is decoupled and the output of this flip-flop is connected to this and this is going at the output. So, this is the chain so, it has become a shift register by itself. Now, I have to apply 1 1 and 2 clock pulses so, it will become a 1 over here and it will become a 1 over here. So, directly I can set your flip flop so, here how I am setting your flip-flop~~.~~, I am connecting all these can chains, all the flip flops which are internal to the circuit in a chain and I have decoupled the normal combinational block out of it.

So, now your circuit is cut off and all these flops which are part of your circuit have become a chain that is if this idea was not good because I had to implement a chain; I had to implement a chain which was extra cost of hardware. Now, we are here what I am doing I am not putting any kind of extra hardware rather I am actually using the existing flip flops in the circuit to make a chain out of it and only I am paying the cost is some 2 is to 1 multiplexer.

This is very much acceptable because I have just putting a max which is equivalent to the number of flops if there n flip flops n marks are there and 2 is to 1. The set reset lines with the internal memory was not acceptable because n flip flops to an extra flip flops you are ready and flip flops are a much larger size correspond compared to 2 is to 2 so, this is still ok.

(Refer Slide Time: 46:45)



ATPG and testing using scan chain in a sequential circuit: An Example

•Circuit is brought in normal mode by making M=0. In this stage testing is performed by making *a*=1,*b*=1,*c*=X which propagates the fault effect to the output

Now, I have set reset the flip the my lines are set. So, now, what now I will actually test mode equal to 0, if your test mode is equal to 0 you will find out that this scan in lines are decoupled so and these are connected. So, you are circuit is now operating in the normal mode. So, what I have to do I know that I have to apply the pattern 1 1 and your circuit is getting tested.

So, now, what we have got so, as I told you why scan chain is so important. If you if some hardware design is very very complicated to explain very very complicated to implement nobody takes that. Same functionality if you can implement it as simple as possible then it is the greatness of a digital design or a any hardware based embedded system design we always call simplicity is the beauty of a design or simplicity is the ultimate complication in case of hardware, designs are simple good designs are always simple.

So, with a such a simple idea people have easily solved the complex problem of sequential circuit test. What directly you generate the test patterns as require in a combinational circuit then somebody we will tell you that I require say 1 and a 0 here I am just giving the example here. Now, I have to do it I need not bother how I will make this as 1 as this as ~~zero~~0, neither have to put extra memories to do it I will go for a scan mode where this will all be correct in a chain I will apply the pattern 1 0 and 2 clock pulses they will be set and reset as required.

And then I will again decouple sorry then again I will decouple this scan mode so, it will be always connected and I will go for the combinational circuit testing like I will apply these two patterns required and your circuit will be testing. So, this revolution as the test process of sequential circuits all the problems have been alleviated except one. So, what is the problem? If the scan chain is very very long like if this depth is d sequence so, that amount of test time will be required to push in the data, but still a revolution was still there because, these are just shift registers if you see that just shift registers corrected in this way. So, these clock pulses so, people use two different type of clock clocks to drive the scan chain.

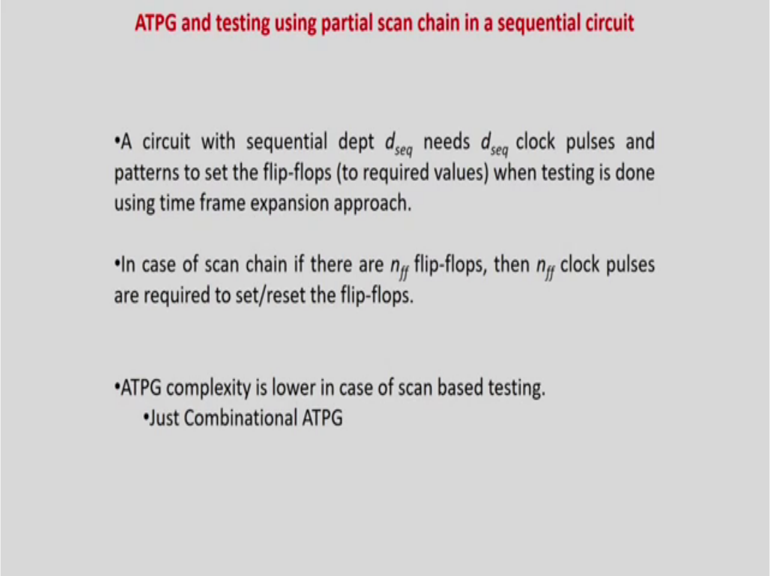
So, when you are driving is scan chain you can drive the clock at a much much higher speed because there is no combinational cloud in between that is just kind in a same. So, you can drive this can chain extremely fast compared to the normal operation. So, although the number of test patterns required to send this scan chain may be equal to this d sequence, but still the d sequence pattern can be entered in an extremely fast manner because there is nothing in between.

But, that cannot be done in a time frame expansion method. Why? Because if you look at it you cannot do that because the time frame expansion method you drive it because to get the value of 1 here using this whole combinational block, but in case of sequential circuit this one is directly connected here let me just erase it and show you. So, in case of time frame expansion method to get a 1 over here you are using the entire combination clock.

So, you cannot have a high speed clock over here, but in case of scan chain basically what is happening this is directly connected to this one end so, for there is no combination cloud in between. So, very easily you can set 1 and a 1 in two clock pulses and that clock can be much much higher speed compared to the normal operating speed of the clock so, you can do a much much faster level.

So, therefore, scan chain only issue I have is that if the d sequence is the length that much amount of patterns has to be shifted into the chain. So, that much pattern will be there, but without the problem you can do it much faster rate. So, still it could not solve the problem of large number of test patterns to be applied, but somehow it has solved by increasing the frequency of the in which the scan chain can be filled in. But test pattern generation complexity is extremely solved because this is just a combinational circuit complexity testing complexity ATPG complexity of a combination circuit you do not have to do anything extra as in the case of a time frame based expansion method.

(Refer Slide Time: 50:27)

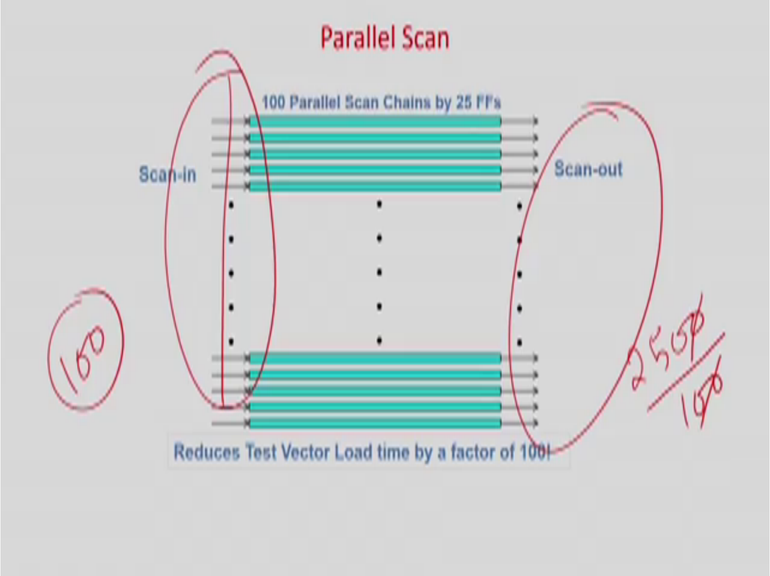


ATPG and testing using partial scan chain in a sequential circuit

- A circuit with sequential dept d_{seq} needs d_{seq} clock pulses and patterns to set the flip-flops (to required values) when testing is done using time frame expansion approach.
- In case of scan chain if there are n_{ff} flip-flops, then n_{ff} clock pulses are required to set/reset the flip-flops.
- ATPG complexity is lower in case of scan based testing.
    - Just Combinational ATPG

So, basically whatever I have told you is written over in this slide that the combinational complexity is lower in case of scan based this is just combinational ATPG, but only thing is that you require some n number of clock person that you d sequence to fill up this can. But, still it is solvable not that much of a concern because I can use a faster clock to put in the data.

(Refer Slide Time: 50:48)



Parallel Scan

100 Parallel Scan Chains by 25 FF's

Scan-in

Scan-out

Reduces Test Vector Load time by a factor of 100!

Then people said that I have large scan chain maybe of 100's and 1000's of line. So, what I can do? I can paralyze what is that? Maybe this is a scan chain quiet long scan chain 100's and 1000's of lines. So, what now you can do is that you can partially break up this scan chain, that is tenth flip-flop is one chain ten another ten in one chain and the ten in one chain some problem will happen of course, more number of scan chains you have more number of scanning pins will be there because, if you are making a chain of course, these is the scanning.

So, there two flip flops so, there is one scan in front it if they are ten, but still there will be only one scan in pin for chain, but if you breaking into two multiple chains more number of scanning pins are there. But, again say for example, I have around say 2500 flops then if I have a single chain your time required will be equal to 1 your time will be required is equal to 2500 clock period to do it, but say I break it up. So, if I break it up say that I have 100 scan chains I have another issue. So, if I have a 100 scan chain so, now, you can parallely feed it.

So, you are going to have a order of 100 speed up, more the number of scan chains and amount of speed up you will of course, because you can parallel feed in this scan chain. But, what is the penalty at least that amount of scan in and scan out pin will be coming out, but still in modern days 100 scan chains is not a problem because only you have 200 pin out says extra.

But, if the number of pin outs becomes for 100's and 1000's and 20000 that is not at all accept it because, in the safety separate case the problem was there for every flip-flop you are going to have one set line and 1 reset line, but here you have a control. So, if you have less number of pin outs you make 10 10's may make scan chain if you have some liberty having more pin outs you make 100 scan chain so, the full liberty is with you.

So, depending on this provision you can increase the number of pins and make the scan based business loading the scan chains are faster. So, the fill flexibility is given to the designer, but in case of IO set reset based scheme it is hard coded therefore, n flip flops you have to in output. So, people have not taken this approach. So, the idea as I have shown this parallel scan chain basically has solved most of the important problems of sequential circuit testing.

So, with this basically we come to the end of the first part of this in first part of the sequential. So, first part of the VLSI testing or embedded system testing lecture of this course this was basically prerequisite. Then I have given you what is this thing why is testing important how can you test combinational circuits, how can you take sequential circuits what are the algorithms involved etcetera. So, these are now brought you at you have just me you have just completed the basic understanding of what is testing and why it is important how can I do it. Now, with this basic concept from the next lecture onwards we will dive into details of test which is appropriate or required for embedded systems. So, those advanced concepts will start from the next lecture.

Thank you.