INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL
NPTEL ONLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification & Test

Dr. Santosh Biswas
Department of CSE
IIT Guwahati

Module X: Sequential Circuit Testing and Scan Chains

Lecture I: ATPG for Synchronous sequential Circuits

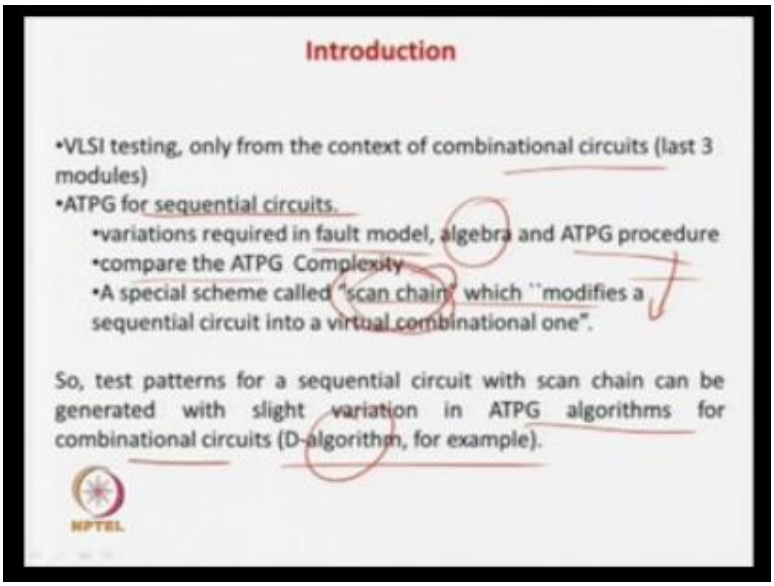
Design Verification and Test of Digital VLSI Circuits
NPTEL Video Course

Module-X
Lecture-I
ATPG for Synchronous Sequential Circuits

Okay, so welcome to a new module, module number 10 in this lecture series on VLSI testing. So what we have dealt till now in the test series the discussion on test with regarding combinational circuits. So we have started with fault models, then we have seen for random test pattern generation, then we have seen fault collapsing, course SCOP algorithms, sensitize, propagate and justify based algorithm as D-algorithm and so forth.

But most of them were limited to combinational circuits. So in the new module 10, we will mainly look about how these same things or all the technologies like stack at fault models, collapsing etc.. ATPG algorithms etc.. whatever we have seen for combinational circuits, how they map to sequential circuits. So this lecture, I mean today we will learn on automatic test pattern generation for synchronous sequential circuit.

(Refer Slide Time: 01:15)



That means you should also know that sequential circuits can be synchronous as well as asynchronous which we have learnt in our basic digital test. But actually here we will be mainly dealing with synchronous sequential circuits with the single clock, I mean to make big course a bit simple, because this course is mainly on VLSI design verification and test. So we are going to have a overview of almost all the three aspects of VLSI design.

So rather than going in between depth of asynchronous circuit testing and so forth we would rather thought that we will try to give a broad overview, so you need the depth of sequential circuits ATPG will limit or scope to single class synchronous circuit. And most of the digital designs we do this test are basically driven by single clocks, and it is mainly sequential in nature. So I mean this actually covers the main part of VLSI testing synchronous circuits and are very few limited circuits you can think of run asynchronous case.

So the more or less it is for not covering that, but that does not have a too much intact on our VLSI testing in study. So basically what we will study, so I mean we have seen combinational circuits in the last 3 days, so now we will, today we will see for the sequential circuit that is

having a clock. So what is the verification required, so we will see what is required in the fault model.

So we will just see that the fault model is not this absolutely similar in case of combinational sequential circuits as one or two small assumptions maybe there that we will point out. So we will see that there will be a requirement of a higher order algebra for sequential circuits and we will see the ATPG procedure and we will see how different in ATPG in sequential circuits compared to combinational circuit, that is what we will see.

In the next lecture maybe we will see something like a scan chain etc., so which can help you in ATPG. So we will show that the ATPG procedure complexity in case of sequential circuit is much higher then you have combinational circuit. So following that we will find out our technique detect technical scan chain which will help you produce the complexity and which will help in ATPG for sequential circuit that will be in the next two lectures maybe.

So I mean, so test pattern generation for when we will see the scan chain and all, so test pattern generation for a sequential circuit with scan chain can be generated with single ATPG algorithms D example and all. So this is about in details we will see slowly in today's class what we will discuss we mainly see what is ATPG about different, why is the ATPG different in sequential circuits.

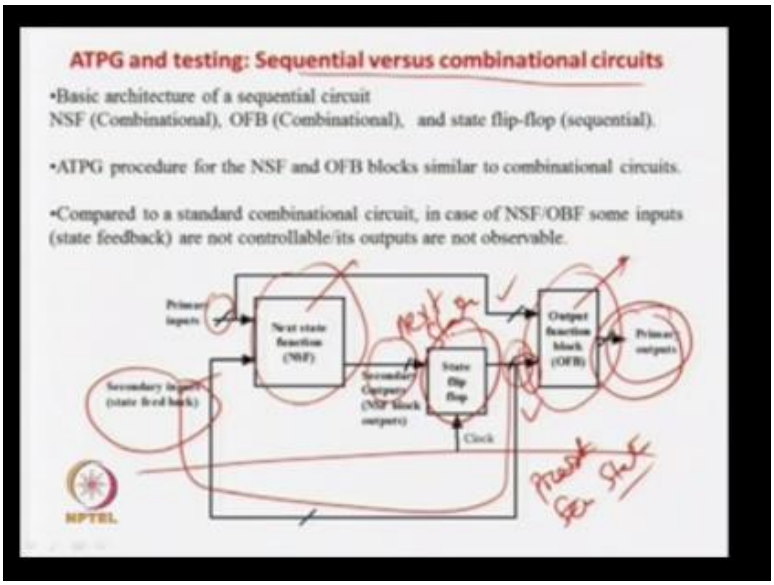
And then we will see that in the next lecture and also we will point out that ATPG for sequential circuit is more complex in combinational circuit. So followed by, so in next5 lectures we will see that how we can develop something called a scan chain which will convert your sequential circuit to a virtual combinational circuit. And these all algorithms which develop your combinational circuits can also be applied to sequential circuits with the scan chain without any much change.

So the intact will handle the complexity of ATPG for sequential circuits, but that will be in the later half of the lecture. So that is why we say that basic motivation of developing this three lectures or four lecture on sequential ATPG will make you convert, we will start for strategy

complexity of ATPG for sequential circuit compared to combinational circuits. And then convert the sequential circuits to virtual combinational ones using scan chain.

So that ATPG algorithm for combinational circuits the D algorithm etc., can be applied to sequential circuit with the variation. So that is the basic agenda of our three or four lectures on sequential circuit testing in this course. So as we said that first of all our main goal today or in today's lecture will be to see how combinational circuit ATPG is different or how sequential circuit ATPG is more complex compared to combinational circuit ATPG that first we will see.

(Refer Slide Time: 04:44)



That is what we will say that sequential for this combinational circuit ATPG and testing. What is the difference that is let us first study. So if you look at this we will look at the figure these are the basic architecture of a digital circuit. So here we have the next stage function block which takes input as the primary inputs as well as the feedback from the flip-flop. So feedback from the flip-flops we consider them as secondary inputs or virtual primary inputs.

Because they are not totally primary inputs they are virtual, because they are actually feedback from the state registers of flip-flops. The output of the next state function block actually tells you

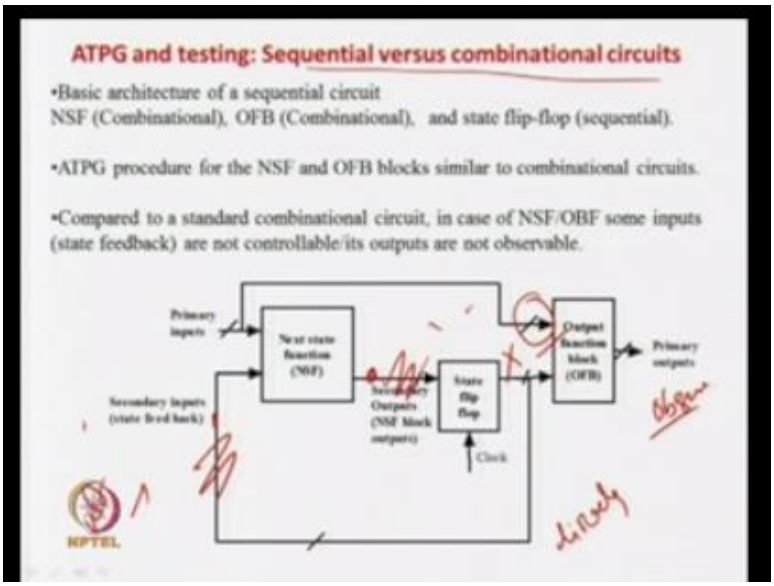
what will be the next state of the circuit which are also called secondary output, because the primary outputs will be output from the output function block which we will see later.

But the output of the next state function block N is we will tell you that this is your secondary output, because this is not directly output, but it is the secondary output which you determine the next state of the circuit, so this is also sometimes called the next state bits of the circuit. And similarly the output of this state represent state, so we all know that from the digital design. So this compression present state and also they are actually feedback to the next state function block.

So we call them as virtual primary inputs of secondary inputs okay. And then there is a last block which is actually called the output function block which takes the input, the present state as well as the input. So if it takes this input we will call it the mini machine and if it does not take this primary inputs only it depends on the next state, I am sorry the present state. So it is called the moved machine.

So already we have studied the digital design, so anyway to make these things here the output function block will take the primary inputs and whether the primary state and it will generate the primary output okay. So I mean if you look at this block carefully, so we have three main blocks the next state function block, state presents an output function block okay. Among them this is a combinational circuit, this is the combinational circuit block and state flip-flops are the sequential circuit blocks okay.

(Refer Slide Time: 06:33)



So now if you look at this from the sequential circuit basic block diagram from the combinational circuit point, so you can see that this is the combinational circuit, this is the combinational circuit. So you could have done ATPG using or D algorithm or whatever we have learned. But there is a small problem here, so in case of D algorithm or automatic test pattern generation by random test pattern generation what you assume that in a circuit the primary inputs and the outputs are controllable and observable respectively.

That is you can apply some patterns in the primary output directly and you can observe the patterns output directly. That is no need to for etc., circuit to control this inputs are observed in the primary output. Even if you may request some circuits to control immediate lines, but we have again discovered that approach, but if you look at this very carefully when your circuit will start up, so output of the flip-flops are not known.

Sometimes they can be 0, sometimes they can be 1, you do not know acts leave out in the value. So you generally call them as X only you can control the primary input. Similarly for the output function blocks these set of inputs that corresponds to the state flip-flop outputs are X that is they

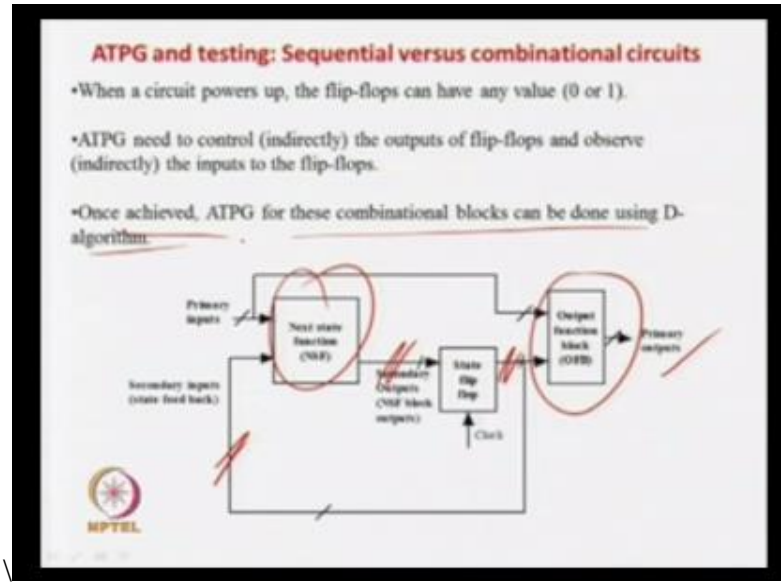
depend on what will be the value when your circuit starts. So if you look at the, so this is a primary output so very good.

So you can observe them directly, so there is not a problem, but you can control only half or one-third or a fixed fraction of your inputs of your circuit for this combination block and the output function block. Similarly for this block if you see you can control this portion of your primary inputs for this next state function block. For the secondary inputs of the virtual primary inputs you cannot control it directly.

Similarly, again another problem with this block is these outputs actually also you cannot directly observe, because they are going to be afraid to the next flip-flop next bits I mean as the inputs of the flip-flop. So this cannot be observed directly similarly this cannot be not controlled directly. So that is actually the difference, so we had it be the purely combinational circuit , testing algorithm, then you should have been able to, there should have been a decoupling over here.

So you should have been direct level to control these points, and you should have been able to directly observe these points, similarly for this.

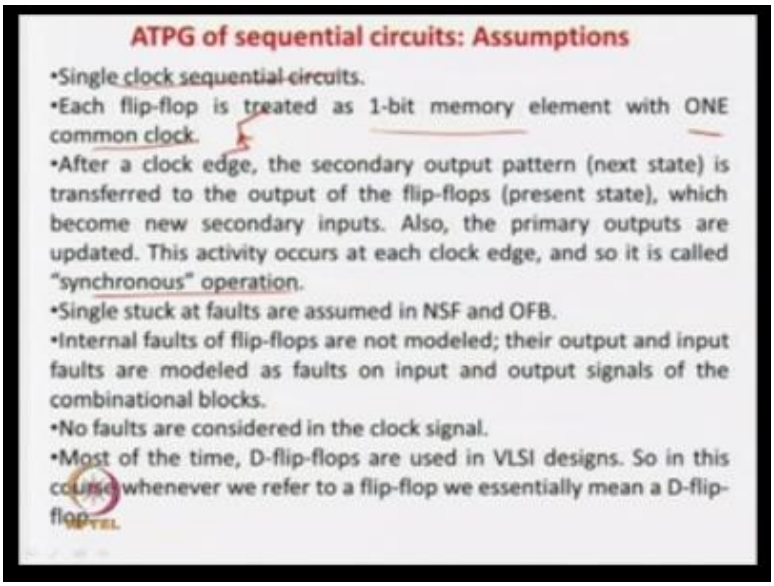
(Refer Slide Time: 08:41)



So as these things are not possible so there is what we require, so we require two etc., steps that is when you are going to test the sequential circuit they are two combinational block. So I think you can directly use the combinational algorithms like ATPG, D algorithm and all. But with the etc., constraint that somehow you should control them in directly and somehow you should observe them indirectly.

Similarly, for the output block you can observe directly for this part no problem, but somehow you should control this indirectly. So that is what it says, so once that is achieved, then ATPG for this combinational blocks can be done to D algorithm.

(Refer Slide Time: 09:16)



So that is what we are going to see in today's lecture how to do that, that is we will apply the algorithm for specially we have to control the virtual primary inputs and somehow we have to observe the virtual primary output this is what our job, then out job will be done. So ATPG for sequential circuits will be same as ATPG of the combinational circuits. So now we will see is there some assumptions which we will be taking.

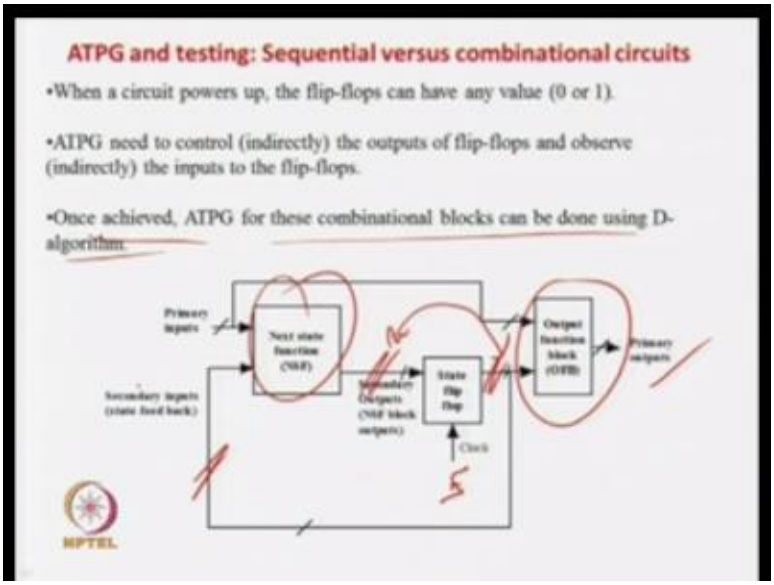
So for ATPG of sequential circuit that is some properties of these circuits is taken as assumption, to make of and this algorithm simple and on the same way, so these set of assumptions actually apply for a very, very high fraction of the digital circuit which are being designed. So we are I mean actually targeting the majority of the designs. So single clock sequential circuits we are taking there are clock as single circuit and it is sequential.

So each flip-flop has a one bit memory element that is okay obvious and one common clock that is one clock is being, I mean given to all the flip-flop. So all the flip-flops which in synchrony with the single clock. So what it says that and next assumption is after a clock edge, that you can say that we assume that everything is working at the positive edge that is your secondary,

primary output that is your next state is transferred to the output of the flip-flop in this present state.

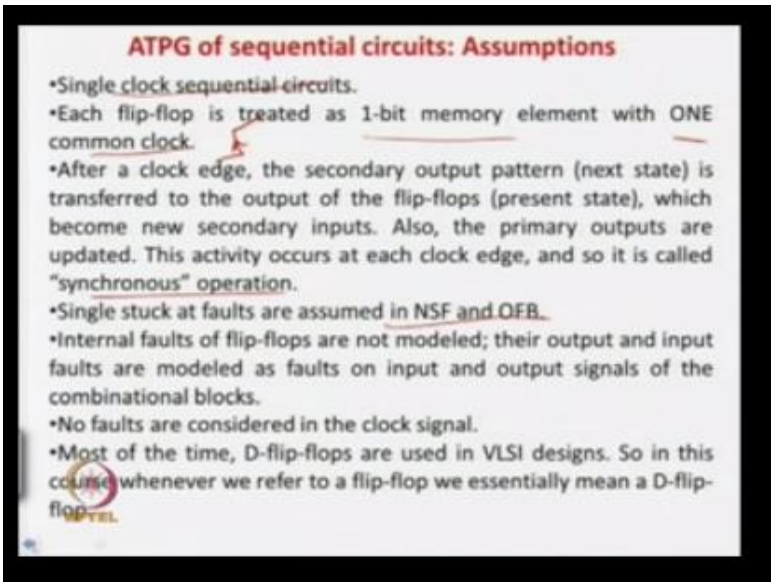
And it becomes secondary outputs also the primary outputs are updated, these activity occurs at the clock edge so it is called a synchronous operation.

(Refer Slide Time: 10:31)



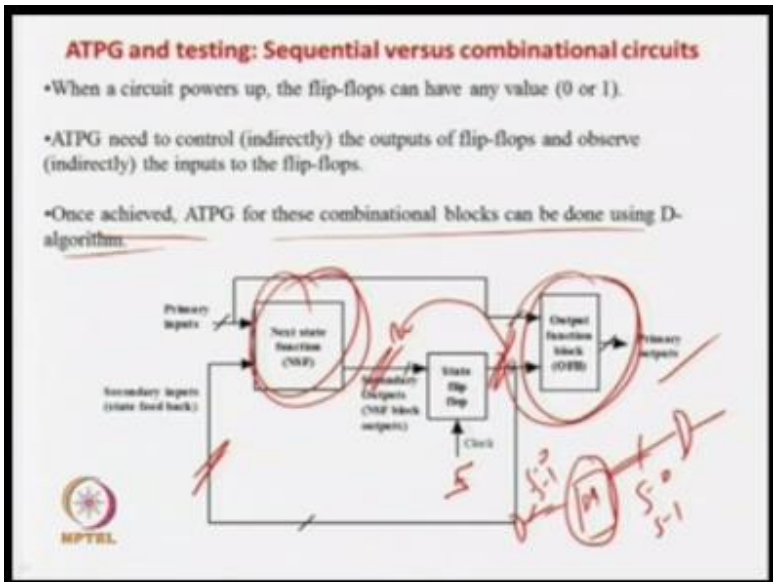
Basically what they have said is that so there can be some input here, so now you give up positive clock after the positive clock edge this output will come at this one. And it will happen for all these state flip-flop in one goal for this positive edge of the clock. So that is all changes that is all changes in this, all changes we are actually call secondary outputs.

(Refer Slide Time: 10:53)



Secondary output will become secondary inputs at the positive edge of a clock. And this will happen for all the flip-flops in the circuit. So it is called a synchronous operation so that is what is the idea right. Next important point is that single stuck at faults are assumed in the NSF block and the output block.

(Refer Slide Time: 11:07)

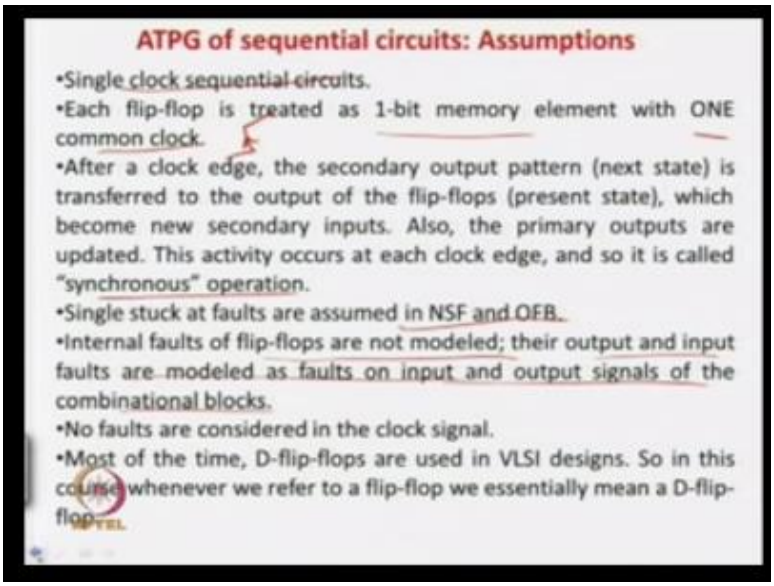


So we assume that this stuck at faults are at this circuits of the gates at this net and also at the gate. So I mean here you are actually having some you can say some deep flip-flops are there, so DQ, some S or JK or whatever some flip-flops are there, generally in case of circuits you sue the flip-flops. So we assume that all the deep flip-flops are available in this a state flip-flops. Now what happens we are saying that say this flip-flop will be driven by some gate of this next state function block.

So you can assume that there is a stuck at 0 fault here and the stuck at one fault over here. Similarly this is driving some other gate in the output function block okay, also the output function block or also some gate in the next state function block. So again you can have a stuck at 0 fault and stuck at one fault, but internal to the deep flip-flop there are some other case like feedback and all those things.

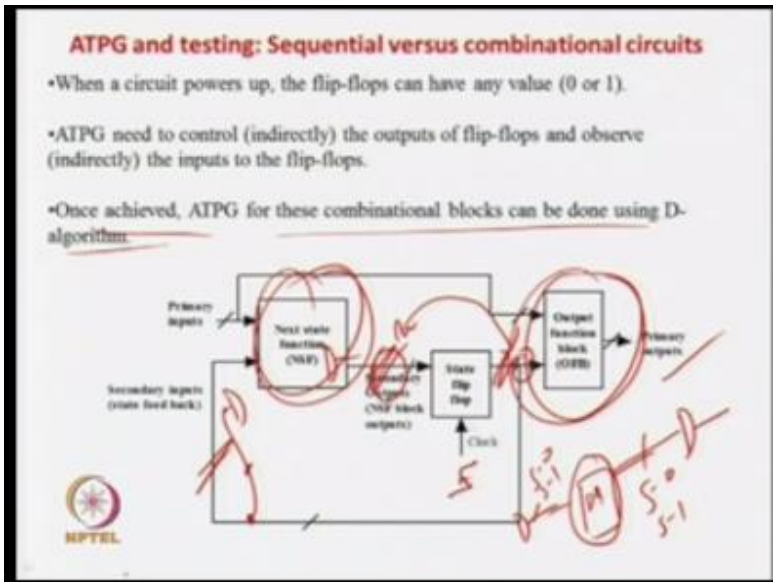
As we all know the internal structure of a deep flip-flop so it will assume that the faults are not present in though the internal gates.

(Refer Slide Time: 12:01)



So I mean if you the second assumption say that stuck at faults are the output of the NSF block and output block, internal faults of the flip-flops are not model, their output and input faults are model and faults of the input, output signal of the combinational blocks okay.

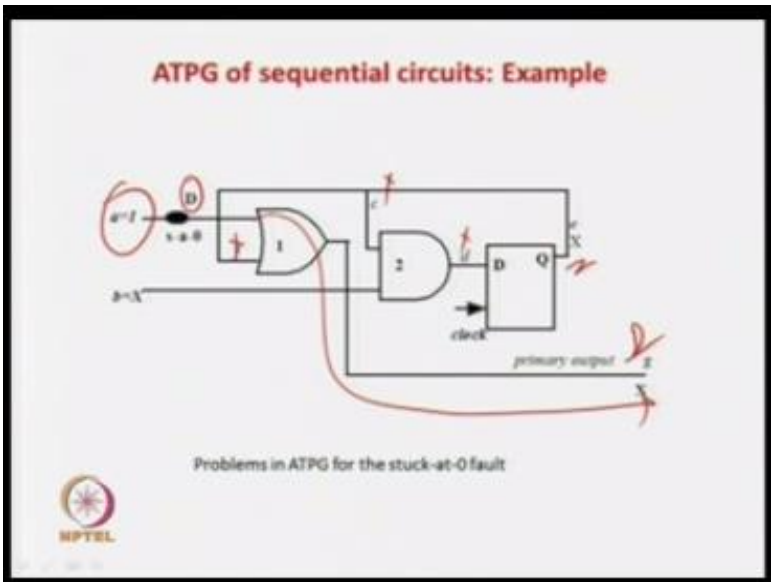
(Refer Slide Time: 12:11)



So that is what is being said that we model faults here, we also model faults here that is the input means of the flip-flop and the output means of the flip-flops will have faults model, because you are considering flip-flops you are considering faults at the outputs of the gates of the NSF blocks as well as also the inputs of the gates of the output function blocks, just like I have given an example.

So some gate here will be driving this which will have a fault which you can think of having a fault. Similarly there will be some gate whose input will be taken from here, so this is also you are considering as a fault. So faults at the input and output means of the flip-flop will be considered.

(Refer Slide Time: 12:48)



If you are not considered in the flip-flop faults of the internal of the flip-flop gates, so I mean this is to keep the things simple, because if you start considering faults of the flip-flop then you can, we can show that the circuit may not remain as sequential circuit at all. So also it may one flip-flop may start having multiple number of states and all those much more complex modeling will come into picture.

So the order of complexity will be very, very high. So in the very first few lecture we have said that what is the beauty of stuck at fault model we are considering stuck at fault model, because it is simple to model, you can do collapsing and all, number of test patterns is very, very less at the same time they give you an assurance that 99.9% cases if you are assuming stuck at fault you can be assured 99% then there is no defect if your circuit does not have a stuck at fault.

Same thing logic also actually holds for our sequential circuit if you consider faults only at the input and the output of the flip-flops and forget about the internal faults of the flip-flop, because if you consider the faults of the internal of the flip-flop maybe your accuracy improvement will be 99.9999+0.00001+0.000001 maybe there. But we improve that the amount of complexity that will come into the algorithm will not make any help.

Unnecessarily you will find that the circuit will have more number of states, the circuit may become a combinational circuit from a sequential circuit to handle all this complexities the ATPG algorithms will be so complex and the number of test patterns required will be so complex then we may loss what we are going to gain by adding this. So and people have followed experimentally just if you forget the internal faults of flip-flop still your coverage assurance or your assurance that you have to cover the stuck at faults there is no real defect.

That the assurance between this correlation between these stuck at faults and the defects then it is very, very high. So that is why we can assume that faults are only at the NSF block and output. And we can forget about internal faults of these flip-flops okay. Similarly, no faults are considered at the clock, because if you find for if you take faults of the clock things will be very, very complex like sequential circuit may not be a combinational circuit.

Some of this circuit may become asynchronous, because for some of the parts of the circuit clock may reach for some part of the circuit clock may not reach making it asynchronous and the algorithm will be terribly complex okay. And they will also not gain much in assurance level, so people have experimentally found out that you can forget about faults in the clock, you can also forget about faults in the internal case of the flip-flops still the correlation give the stuck at faults and whatever you call the real defects are very, very high. So we will consider only flops in the inputs and outputs of the NSF block as well the internal gates of the NSF block and also mainly our.

As I told where as the circuits are mainly considered what you called the D flip flops where the JK flip our J because once you can format do you all know but in the case of digital circuits we mainly consider about your D flip flop so in our assumption is that all the circuit will be taking hence forth made our D flip flop okay so now again let us go back go to an example tell what about the complex here is we are discussing so we may be lost so let u see the example is the complex circuit.

And is a very simple example circuit so you can see that if you look at this just a simple circuit now you can see that this gate actually depends on the one input of the flip flop and the primary input and rise the primary output so it will become your output function block you cam map it to the old diagram and find out so what in this gate map belongs to it depends on the primary input be and also it depends on the out of the flip this is actually the secondary input so you know that the block which depends on the secondary input as well as the primary input is called the next state function and the output of the next state function block goes with flip flop as input and we detrain the next thing.

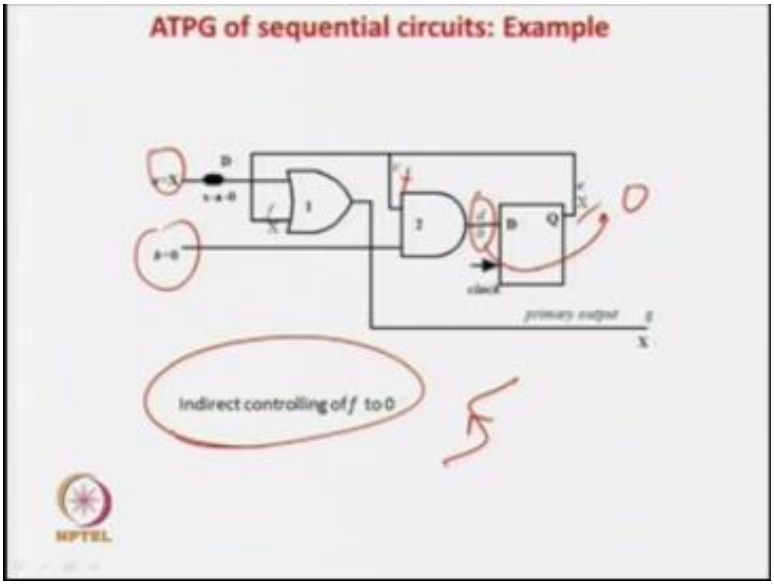
So one actually is the output function block and this is next state function block if you just map it to this one you will find out so let us consider a flop as a I mean sorry fault stack at 0 and one input of your output function blocks so now least us see how you can do the testing okay so this is the case so only have to remember that this is the output function block and 2 is the next state block.

Okay so now when something starts ups as you all know that initially every all the gates all the nest are having a value of x as per the algorithm so now these are stack at 0 so you have to apply a 1 so it will be D okay now one thing is that so those is actually this is the D so there is only one part through which fault of it can be propagated that is actually called a singular D path so you have to a fault of it should be there so out target is that so gate 1 is in the D frontier because one input is D another output is X.

So there is only path through which the fault can be propagated this is called the singular D path so this should be at the auto propagation of the fault effect they can again now image one may be now it is D output is D so this F is unknown x so J frontier algorithm the J frontier concept we know that F as to be a 0 but the fault affect to be propagated so we require that one but now you can see because the circuit because now you cannot directly F that is the difference between a sequential circuit testing as combination circuits head with the combination circuits and what they would have to don F could have be directly controlled and you could have applied 0.

But now is not directly construable because if the output of the flip flop and the output of the flip flop we totally depend on what is the value when it gets started so now it is x so if we require it to be 0 but now it is an x so we are in a problem because we do not know how to make at this one so we have to now do something so that we get F equal to 0 and once F equal to 0 then you can apply this one to get this okay get the fault tested so our main job is now to get a 0 at F so actually it is called indirectly control of F to 0.

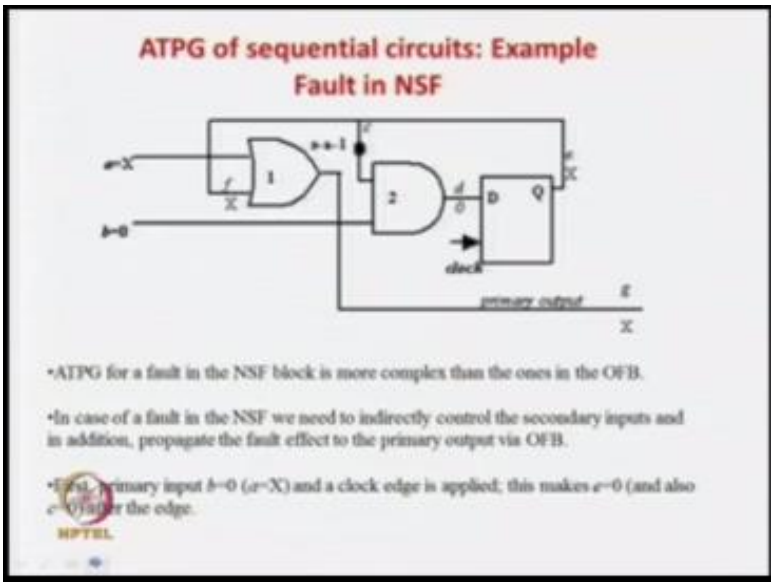
(Refer Slide Time: 18:16)



So how can you do that let us say that we put b = 0 so b = 0 this is x okay because output so if apply B= 0 then irrespective of what is the x can be 0 1 we do not whatever we already the x means either 0 or 1 we do not know because on the power of the circuit we can have any value at x so if even if it is 0 or what we have to case put if you B as o so D will be a o now you but B = 0 x do not bought at the time D and you apply a positive H of the curve so once you do this this0 will be coming over here.

Right so you apply a b=0 and apply you get a 0 and apply a clock pulse so once that so that so done.

(Refer Slide Time: 18:50)

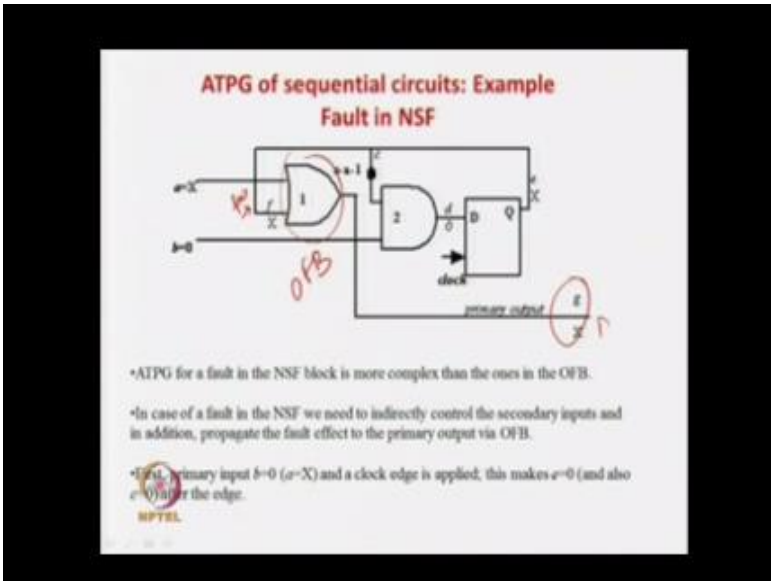


So once that is done so what you are going to get a 0 over here going to get a 0 over here now you indirectly you played F = 0 now you can put a= 1 b= 2 forget for the time being and then we will get this value of the output and circuit to test so now in case in case of sequential combination circuits what we have see there are only one pattern can test your fault correct so you find out by the propagate or justify or whatever by a random but not but one test pattern can deduct a directly a with that there is a fault in the circuit or lock but in this case we require to patterns as you can see first we have to apply axb0 a is we do know b = 0.

So which is determine we make that f= 0 in directly so once f = 0 now we apply a = 1 and b= x so that your circuit to test so this is a basic fundamental difference between sequential circuit testing a combination circuit testing so in sequential circuit testing what we have done is sequential circuit testing we require that first the primary output are secondary inputs whatever is sorry the partial primary inputs are secondary inputs whatever you call they have to be indirectly control because there the outputs of the flip flops that are indirectly controlled to some value that is required for ATPG.

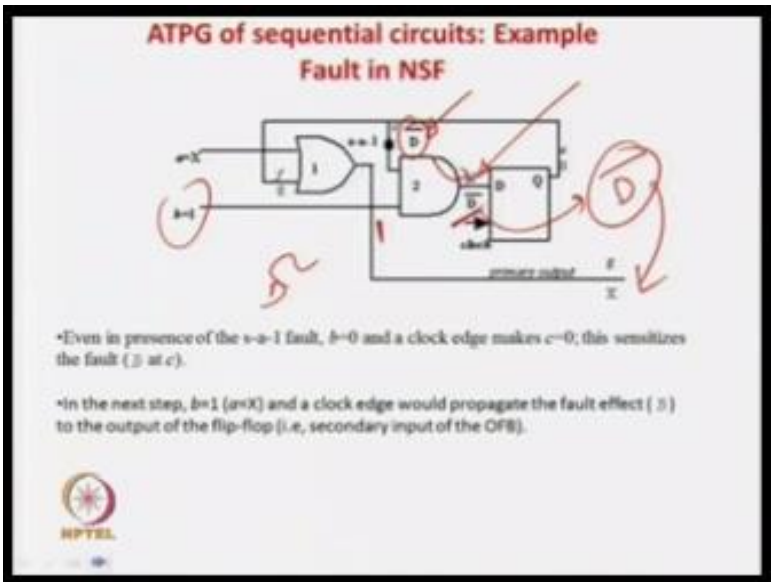
So you may in this case one flip flop is there so you require only one set slowly we will see how many steps are required to make this secondary input controllable so you require one pattern one clock one pattern some amount of patterns and a clock. In this case there is one pattern required and the clock has to be applied. So it will virtually or indirectly control your virtual inputs. Now once the virtual inputs are controlled then you can directly control your primary inputs and.

(Refer Slide Time: 20:30)



Then you can say what is ATPG is done. So in this case we need two patterns to do so now let us see the but in this case if you consider a fault in this next step function comes then in case of sorry this is the output function clock so this is a output function block so we consider a fault there so somehow we require to control this net to 0. So we have done with this indirectly but the output was directly observable so it could have been done in two patterns. But now we will take another fault in this block which is actually your next state function block.

(Refer Slide Time: 20:54)



So this is you like it is more difficult because you have to directly control this sorry indirectly control this secondary input as well as output of nested function is not directly observable so you have to some haw make it indirectly observable. So for set testing fault set the nested function block is more complex. Then you what you call this output function because in output function the primary output is directly is observable but in this case it is not stack at one fault so what do you require for one fault for that you require 0 over here incorrect. So we now this requires a 0 means it is D prime over here.

So but to get a D prime over here so you have to get a 0 over here but as you know that this is the output of flip flop is indirect so how you can do that indirectly so you do not forget XA You put as 0 so irrespective of all you get a 0 over here and then you8 get a clock pluses so once you do this you will get this as a 0 over here. So this is what has been done so if you make a B=0 apply a clock first then, then you get X=0. And you have FOR gate this is what we do in first step in indirect control correct.

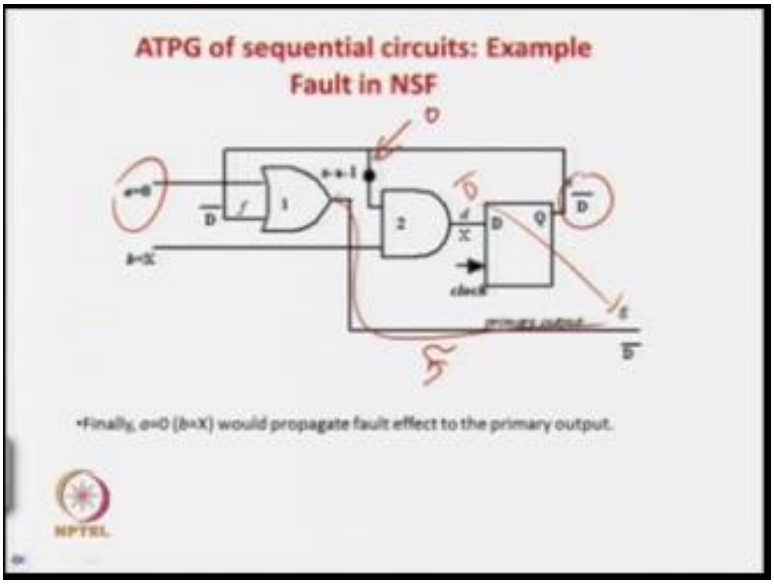
So you get this is 1, now let us take next thing you get from you get a D prime over here so now what you know so your fault is sensitized so previous case d=0 was the previous case so now

what do have to do so you know that if this is default this is the only AND gate so for this becomes the default here and you need default here or there is only a unique x path in this case so you have to somehow propagate the value of d prime over here, so for that you will require to have a 1 over here.

So in the next stage what you do you make forget about x make b=1 and then so it will be a 1 so this d prime, d bar will be floated over here and then you give a clock pulse. So once you give a clock pulse over there so d prime will appear at the output you have to observe that even if d prime has come to the output d that is the virtual primary output but still you cannot observe it directly unlike you nested function because that is going to be input of the d free flip flop so that is more that is why more complex faults should be tested in the what you call output function blocks sorry, faults which are in the nested function block is more difficult to be tested because not only you have indirectly control some lines.

But also to indirectly observe some lines which was not in the case of gate 1 which is the output function. Now some of you have brought DC at but now you cannot observe it directly so you have to somehow make this d prime appear over here so you have to go to the third step.

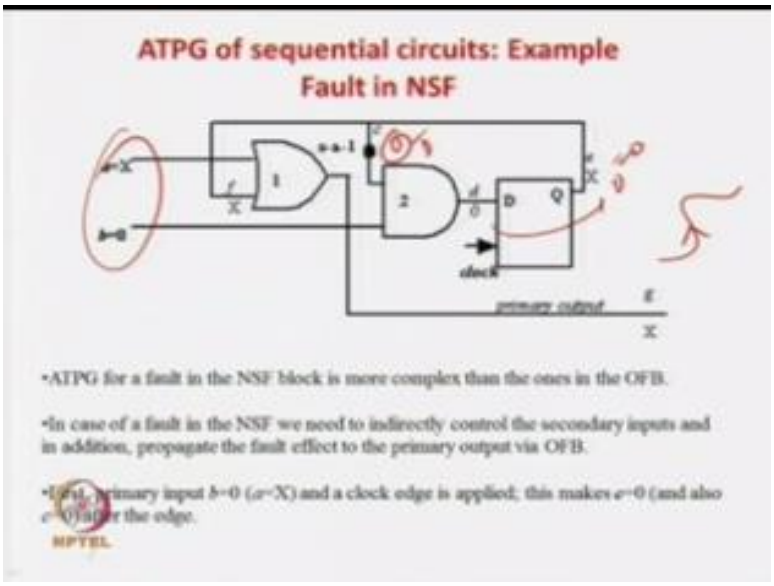
(Refer Slide Time: 23:15)



So last we could have done everything in two steps in its first we control get to some value and then we fault effect was propagated at here, but now in this case three steps are required first you sensitize this to 0 then you propagate at the fault value 2 here and then as this is not directly observable somehow you have to doing this value to here. so next state so in the second part what we have done so we got this one so second stage what we have done so it is 0 so sorry, it is 1 so this fault effect from here it has come over here.

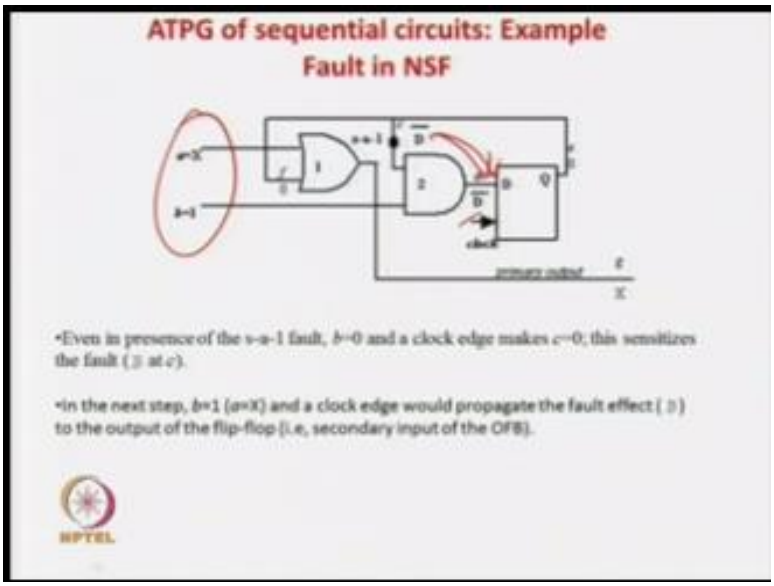
Now you apply a clock pulse so if you apply a clock pulse over here this is your third step you apply a clock pulse over here so your d prime will come here so now your d prime is here okay, so that is in the third phase you get a d prime over here but now again this has to be observed by this one so you have to make a=0, so we are indirectly applying three clock pulses if you see first a=x, b=0.

(Refer Slide Time: 24:01)



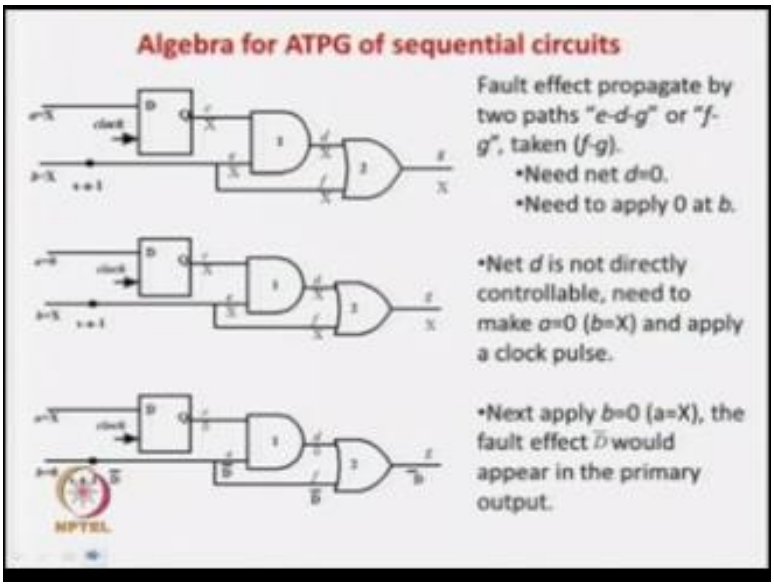
So you are in directly you are controlling this value this one.

(Refer Slide Time: 24:08)



Next you are applying Ax = 1 B = o so your fault effect is propagating to the output but the output is cannot observe so third step what we do again you could have clock gates so that it comes out in directly by your primary output.

(Refer Slide Time: 24:22)



So that is a more difficult if you are considering faults and the next step function block operating output function block but from all the cases unlike combination circuit you have to in directly control virtual and secondary inputs and in directly observed to secondary outputs that is what is the basic conceptual difference so in case of sequence circuit more than one test patterns may be required to test that so in directly so broadly speaking the complexity of sequential circuit testing is very, very it is more difficult.

Compared to our combinational circuit testing so in this case you want to observe that we are not considering the classes where they are can be in skin consistency and when they can be a black so you can understand that how the complexity will blow up this is not considering this so we may use that D algorithm to find out that this can be a good pass of propagation and or instead of one pattern we have genetic 10 or 20 patterns which are equal to in directly control these virtual inputs and in directly absorb the virtual outputs.

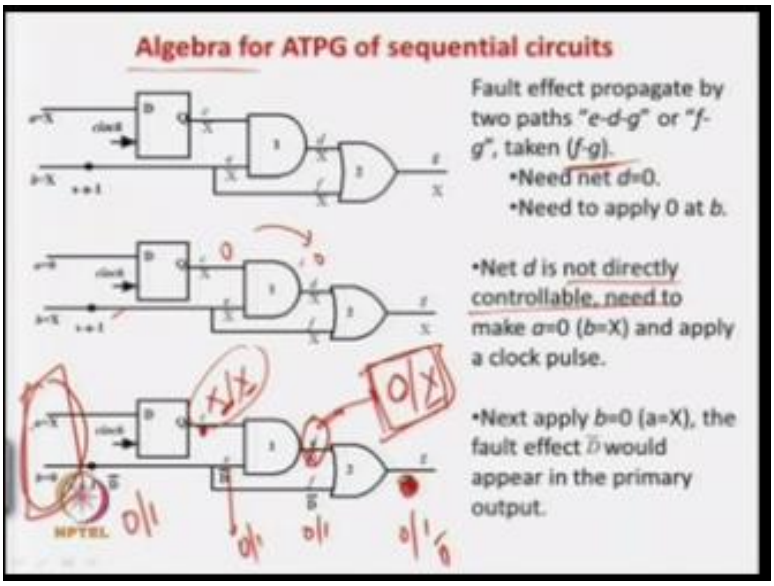
But now after that you could have found out that for some step there is in consistency and you have to do lot of black turkey so in combination circuit there was only one pattern for but now there are 10 patterns or 11 patterns or x number of patterns for so again if you remember these

are the back drag and the final phase so how much amount of completion work has been lost so that is why sequences are get testing is highly more complex in a combination of circuit so that is what we are going to focus today.

And in the next class we will see how we can solve this complexity okay so now we will see that you know what kind of a as a brought that means routes etc. We have seen for combination of circuits but now we are going to study here that do you do we require some extra kind of logic or extra kind of algebra we are asking the question that whether some extra kind of logic is required for ATPG for sequential circuits or whether find value logic goods are so if or not so let us just6 see that we will try to use the five value logic.

And see what the gain is and what is the laws and then we try to see if we can go for a higher order logic okay so let us see that this I get.

(Refer Slide Time: 26:18)



So these are stuck at 1 fault so obviously stuck at 1 fault over here there the stuck at 1 fault over here so was stuck at 1 fault here means you have to apply a 0 1 here so now these two paths so one path is fault can be propagated here and other path can be here so there are two x paths here

so for the time being let us see that if this path is smaller so by any rustic is apply over D algorithm so we can take that E and F may be a very good path through propagate the fault because the other one like.

e d and g the longer path correct, so stuck at 00 means if you get D' over here so you can propagate the value of D' over here so you can propagate the value of D' over here but immediately we know that D has to be a 0 over here so how to get a 0 over here so we know that directly we cannot get a 0 over here so better what we can do is so how do we get a 0 over here so one we can do so net D actually already told that net D is not directly control other okay so we have to make this net B = o.

Because actually this net depends on a output of a fixed up so this gate is not directly connected to the output of the faults but in directly dependent on net c which is the output of a flip flop so in directly we have to control this d2 0 so if you just look at it we can even is we do that you can make a = 0 so if you just make a = 0 now what you do you get and give a clock pulse so just if you would give a clock pulse a = 0 then this one will come here and this is the x so x an as 0 will be a 0 correct.

That is what is our idea so that this how we were doing by at D algorithm so just tell us quickly see once more what we have done so this is the faults stuck at 1 so we apply a 0 1 here we need to apply over 0 and this is the path propagation so we can get a D over D' prime over here so this the path and to get that these one we require a D over here so these4 have to get 0 over here to get fault of this one.

But now these not directly connected to the output of flip flop but it is connected to see which is the output of a D flip flop and these can be directly controllable c you mean that it is nothing but a primary secondary input sorry secondary output because it is the output of flip flop not directly controllable so somehow you have to make this source to be 0 so if want to make this one to be 0 because you are D to be 0 so somehow you have to make this to be 0 because you know that 0 8 is a singular cover.

So to make x = 0 what we can do we have to make a = o and give a clock pulse so immediately D will be equal to 0 and your D' will be propagated to the output so what are the two test but to test the for 1 test button is a is 0 bx and the next pattern is ax and b0 so this will test your fault so that two patterns is equal to test these now let us just do was height flick of Boolean algebra and see can we achieve something better okay so let us for totally just see this one and forget about all this so we can keep it as an x.

For the time be keep it is an x okay now let us see what we can do so let us see it is stuck at 1 and we apply a 0 so normal case 0 fault case one we can write like this correct so now in this case you can see that these also D' so we just write all this we require correct is what we required something over here and here we write also 0'1 now just have a very quick look at this circuit just sorry yeah just look at this, this point so this point is very important this net is this net e of you look very carefully.

So what is the idea so if just net d is 0 and now 1, now let us look sorry net e is directly you can net which is about net e you can directly get for this net because it is fan out from this step now let us look net d variants so net d is in fact x by D algorithm or robust etc. Now you can see because one value is x and 01 so if you do something like this so you can we will get 0 and x kind of a thing you will get at the output okay so but 0 and x is not a part of your algebra so let us see.

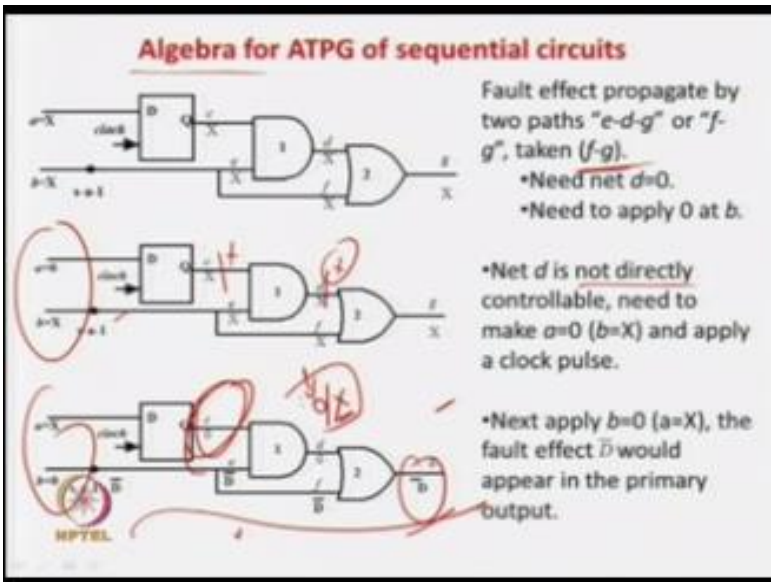
So we are in this case so we get a d = 1 and is that x but now look at that very carefully so now what happens say less forget about this been 0 let us assume that it is x we denote the output of this one so if fault is not there so what a fault is not there so what is the input over here it is a 0 okay and what is the value of this it is x because we are not considering we do not know in case form of our was not going output of this one so what is the output of the at fault so if the fault is not they are normal so the output will be a 0 because that the fault is dot the stuck at fault is not there so you apply a 0 over here so we will get a 0 over here so the fault your d is going to be a 0 okay and other things are fine now if these are fault then what will be the case if there is a fault then you stuck at 1.

So you are going to get a 1 over here and this is a x because we are not applying anything for the timing the schedule so it will be a x so it is 0 and x at the if you are not controlling this because the timing just assume correct so now the output of e is d, d prime that is 0/1 and the you can write as 0/x if the fault is not there you apply as 0 so the output will be 0 but if the faults stuck at 1 fault is there you do not what is the output so it can be our x now you just an f we know is 0 1 so if you do a r ring of this one.

So the output of G is what so it is 0 or 0 is 0 1 or x is 1 which is nothing but a d' so it is very surprising cases is happened here that you mean without controlling this guy tom 0 or this guy to 0 in directly which is required in the D algorithm or rods find was algebra so if we have a element in this which we are calling it has o/x because if you remember rows etc. There is no concept of 0/x we are x/x 01 10 00 and 01 sorry 00 normal 0 for 0 11 normally 1 fault 1 xx we do not know 01 that is d'.

10 that is d but we do not have any concept of 0/x but if we have a concept of 0/x then only pattern can be take this fault this is the very special case in which case I mean it is, is not required to control and all but many most of the times may have to require this but for some of this circuits you can find out that if you have the concept of 0 that is point b is 0/x then only pattern x a = x and b = 0 with test your circuit but if you are directly following the D algebra.

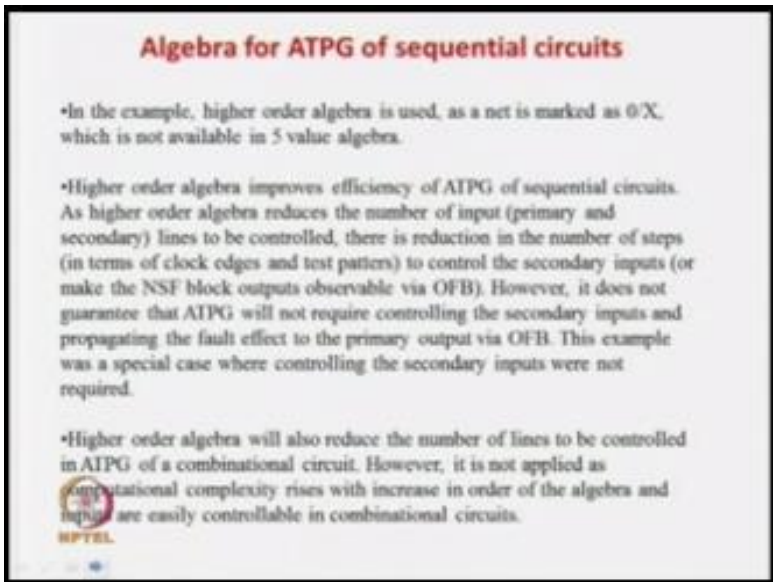
(Refer Slide Time: 32:54)



That is we do not have any concept that is there is only one concept that is rounds 5 value algebra there is only concept that it is x so x means series xx so it is x that means the xx it is also xx something like this you cannot write 0/x on x/0 so in that case you directly follow the d algebra so you require one pattern this one to make this d = 0 and then you require normal pattern to propagate the fault very but if you write something if you are either the concept of writing 0/x.

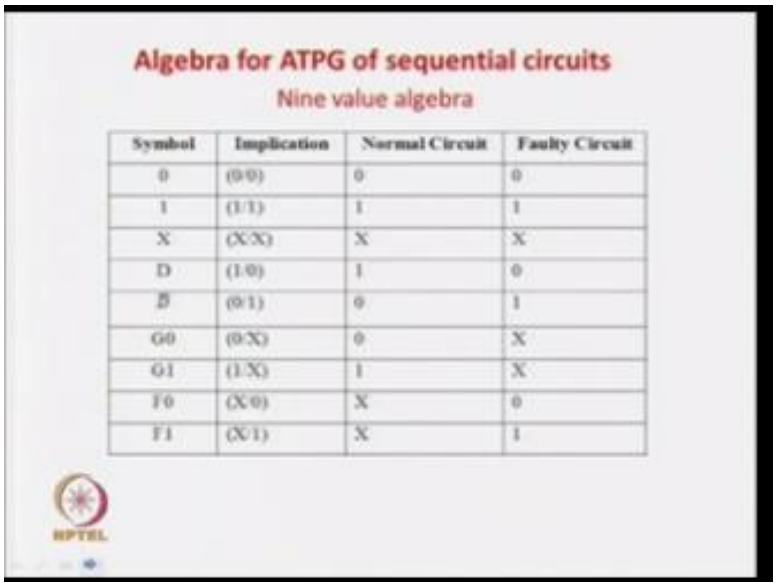
Which is the basically the case because if this circuit fault is there, there only the output is not more usually dependent on the partial primary output that is D but if the fault is there sorry then fault is not there then you apply a 0 so into that not does not dependent on the output of the slit from that at all dependent on the primary virtual primary output of the secondary output then you get the value 0 directly and by doing this we you can get a d' one here without controlling this there is a very special case of an example where you can test or circuit is sequential circuit without requirement of controlling the primary secondary primary output because of the fact but now for that you require some special symbols.

(Refer Slide Time: 34:02)



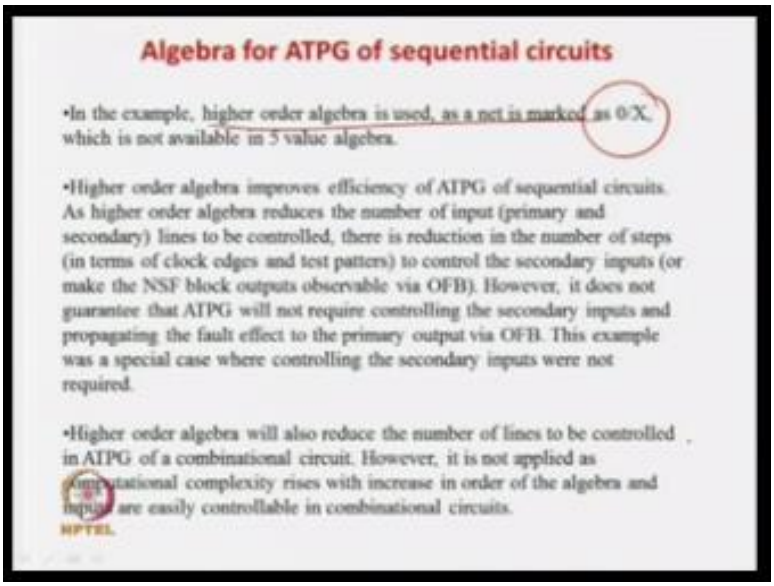
Which we are.

(Refer Slide Time: 34:03)



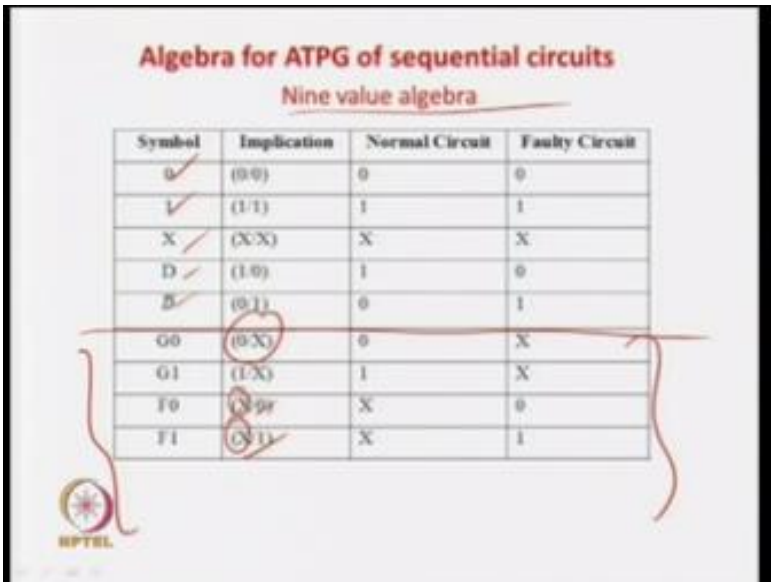
This was about the theory which we were discussing that higher or the algebra.

(Refer Slide Time: 34:04)



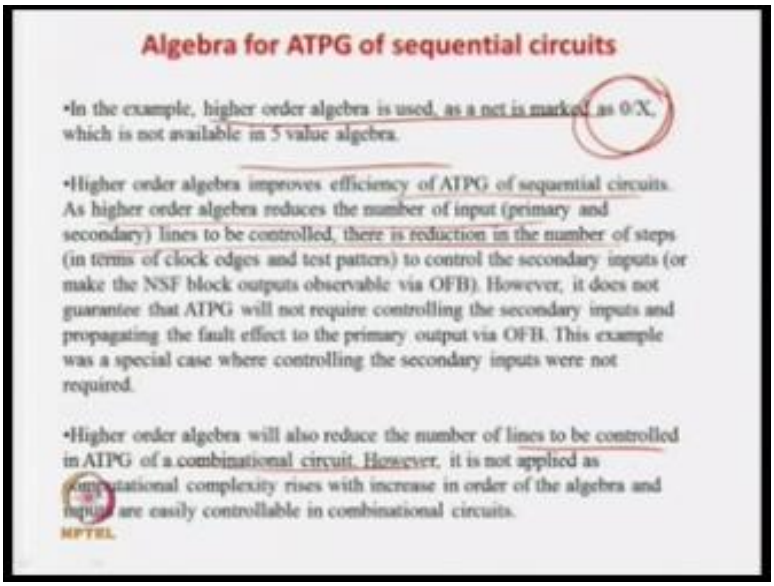
We require a symbol like 0x and all these things so higher or the algebra I mean virtual come back.

(Refer Slide Time: 34:13)



So by adding one extra symbol that is actually o/x we got that in case of a sequential ATPG only one pattern or at least one level of competition was less required so nine value algebra are use the sequential circuit that is 01 x d and d' these are all from rounds value of algebra but now we are added some more ox that is normal case 0 fault case unknown G1 normal case one fault case are known f0 normal case x fault case 0 normal f1 is normal case unknown fault case one so with this addition of these things you can see that for many times you will be requiring less amount of.

(Refer Slide Time: 34:50)



Computation to solve your problem now that is what is required in the last example higher or the algebra this one so which is required to reduce the number of competition or only test pattern can do it. And this available in 5 value algebra that is why we sometimes require to go for a higher order algebra that is 9 value algebra so higher order includes the efficiency of sequential circuit testing because you required less number of test for the competition but only one thing you have to observe that for always you will not be possible there even if you using a 9 values algebra that directly you do not required to indirectly control any virtual primary output or what your primary inputs kinds a thing.

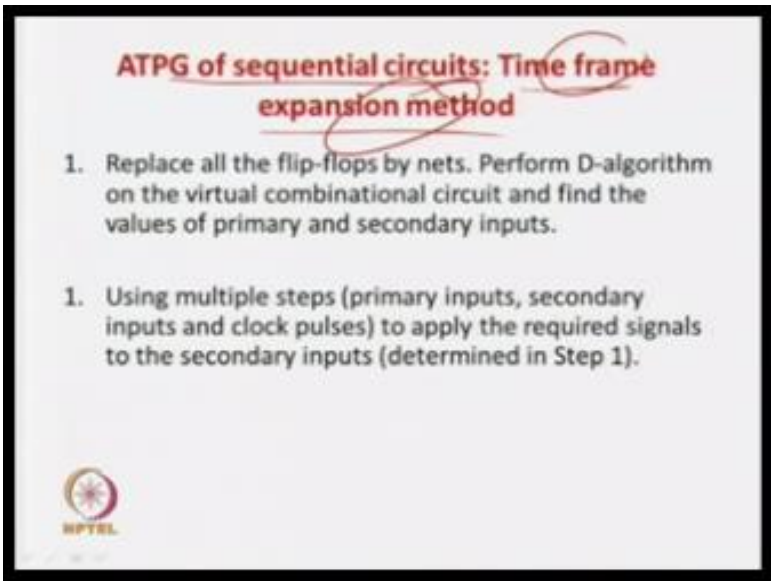
So the special case why did happen but only you can assume, you can assumed that or just you can found out experiment mentally you that if you using higher order algebra then the number of competition generally reduces because you have some more flexibility like x 0 1 ,x 0 x 1 x something like this so more the number of x in this circuit we have seen that more number of flexibilities there.

So that is why it will improve it improve your efficiency because incase of combinational circuit testing one pattern was used to detect this test and but here you required more than one

pattern so lot of competitions are required so you want a more flexibility in this circuit so you were using some other higher order algebra

So you can also question me that higher order algebra would have also helped in combinational circuit the answer is true because in the last lecture with an example we have shown that if you have more number of excess in your circuit then it is true that there will be more number of flexibility and ATPG may success better. So if you have 9 values for supplying combinational circuit you are correct you will get better solution, but having higher order algebra.

(Refer Slide Time: 36:23)



It can also computational difficult and it has been found out that the amount of gain you are going to get by using a 9 value algebra is not that much in a combinational circuit by the amount of inconsistency that you may reduce that is because in sequential circuit you have more number of test patterns the more number of patterns are equate to find out the test pattern, so we want more flexibility so use the extra four symbols like 0x, x0 and all this things.

But same thing if you bring out in our computational circuit and make it 9 value as algebra combinational circuits so some advantage we will get that is definitely because more number of

excess will be there in the circuit but the amount of complexity will income there will to help you to gain because in combinational circuit one pattern is enough to detect a circuit so for I mean inconsistency and bit more less computational sequential circuit.

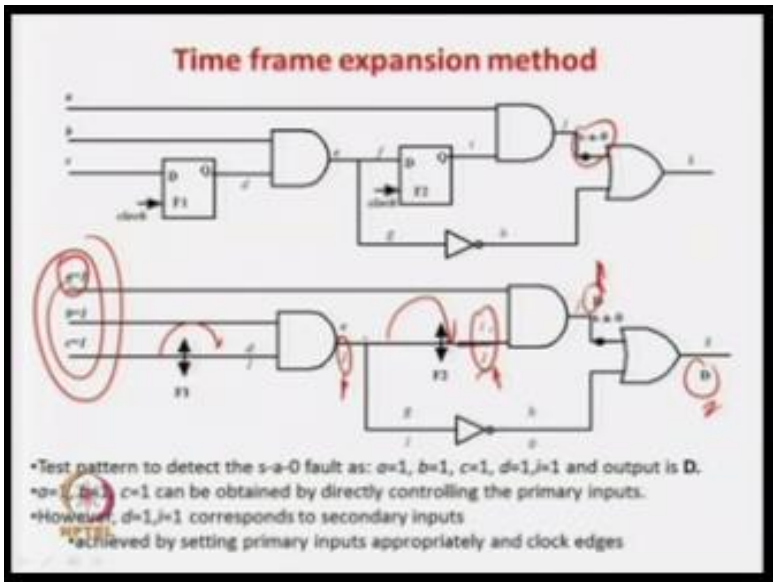
So that is what about the higher order algebra and we have seen the motivation so now what we are going to do so we are going to see ATPG for sequential circuit using time frame expansion like that, that is like D algorithm we have seen so in D algorithm what was the idea that is the Forman algorithm to find the test pattern for combinational circuit so in this case this time frame expansion method this is another formal method for what you can say is the very formal method like D algorithm which can find out test patterns for sequential circuit.

So let us see this in details with an example, so what is the first step replace all the flip flops by nets so first there will be sequential circuits will be there so there will be some flip flops over will be there so as just purely sequential circuit is a singular clock is the assumption so then you remove all the flip flops that is the first job and perform D algorithm on the virtual combinational circuits so whenever if you remove the flip flops so what will happen the output of the flip flops will become your virtual primary inputs and your that is what so we know and the outputs of your and the inputs of the flip flop will become your virtual primary output.

See what we have done we are removing all the flip flops so if you remove all the flips what is happen the outputs of the flip flops will become virtual primary inputs now you can also for see them as input and the outputs of your sorry, and the inputs of your flip flop will become the virtual primary output so some case which will see in the example that is what is going to happen. So virtually you are going to get a combinational circuit.

Now you use D algorithm on that and you will have to use so this step 2 you have to multiple steps will be requiring to control the secondary inputs and observe the secondary output.

(Refer Slide Time: 38:40)



So already we have seen that this will be the that for controlling the virtual primary inputs and observing the virtual primary output you may require multiple steps. So that maybe require but let us see with an example because that was the mathematical definition kind of a thing so this let this be your circuit to be tested and we have a fault over here, so there are two flip flops so you can consider this as your virtual secondary input because this output of a flip flop the secondary input you can say similarly you can also see this is the output of the flip flop this also a secondary input.

Because you cannot control this direct, secondary input yeah, you cannot observe this directly correct. Now this is one input to the flip flop and this is a special case so this is a primary input as well as the input to the flip flop so you can say that this is also both the primary input as well as a virtual input but in this case you can see that this is the output of this gate is going to the input of this gate.

So this is actually a virtual input, this is virtual primary input or secondary input because you cannot control this directly that is what is our idea. So this virtually primary input this is virtually primary output you can say or secondary output similarly this is output of the flip flop but you

cannot see that directly so you can consider that this virtually primary output kind of a thing, okay.

So that is what is our idea and also you can say that this again going to the input of another gate because this is also this output as well as it is had it been directly coming out then you could have called in a virtual primary output because or a primary output something like that but you can see that also it is going as a input to another gate so that can also be cannot be directly control or observe so you can say that it is a virtual primary input or virtual primary output that is secondary input output.

So it is not directly observable or directly controllable so somehow we have to do this indirectly, okay so you easily map this gates to which is the and say block and which is the output function now that can be easily done, but anyway that is not much of our concern right now so this is stuck at fault over here, so what is the first step, first step is remove the flip flop, so you have this flip flop here you have remove the flip flop here, here and here short at this so this is what is your idea.

What is that to be remember that you cannot control this directly and also you cannot control this directly that is even if this is connection over there if you want a 0 say or if you want 1 over here say you apply a 11 or 1 here you cannot get the 1 here directly because it is connect by a flip flop that assumption you have to know that similarly you cannot get at 0 here directly or 1 here directly not possible be very we have shorted this.

So that is has to be remember, so but it is now a, so but for the D algorithm for next step is that this now but with this small assumption in mind that is not very explicit assumption here so what do we have seen that is not a very explicit assumption but we just know in your mind that directly you cannot be controlled or observed. Now these are virtual computational circuit that was I was saying that is a virtual combinational circuit.

Now in the virtual computational circuit you apply combinational D algorithm that is it, so now you just see what is the computational D algorithm over here so we know that with a 0 so you

have to apply a 1 here so which is a D over here so if the propagating so it will at D now the other input should be a 0 OR gate this is a 1 here you require and next is that you have to apply a 1 over so apply a 1 over here means a will be has to be a 1 and this input also has to be a 1, correct.

Next what is the case so this has to be a 1, so the output has to be a 1 so 1 means the AND gate this has to be a 1, so if you have a D algorithm so you could have directly say that a1, b1 and c1 if you give and the output is the so your circuit would have been directly tested if is a computational circuit, but here is a big put over here so you require a 1 over here you require a 1 over here. But as I already told you that internally you have remember this big, big put that this directly not controllable this also not directly controllable that is the big put you have to remember.

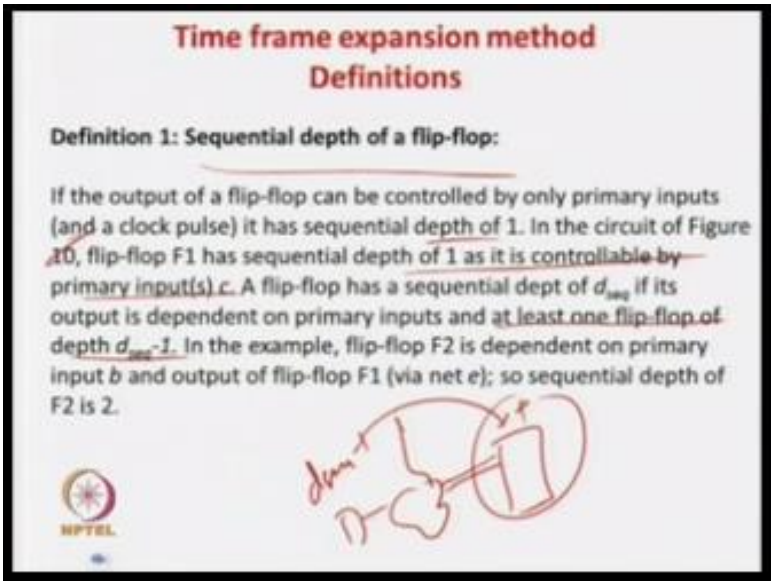
So how you can get a 1 over here you have to that indirectly, similarly how to get a 1 over here that also you have to indirectly because these two are output of the flip flops which are going to your input of some gates so there your virtual primary input they are not your direct primary input they are your virtual primary inputs and you have to control them virtually that is the idea.

Similarly if some continuous to observe this let that also not be possible because they are either internals or you can also even if they are feeding it to some outputs of the flip flops or even if they have the going as a input the output function because they cannot be done directly because of this virtual primary output and virtual primary input kind of the behavioral.

So now let bus forget about this so I mean let us see if it is your target so our target is that if we have to apply eventually to get to this part before that we require a 1 over here because sorry, we require a 1 over here for that we require a 1 over here similarly that is done so this is our requirement number 1 this is our requirement number 2 okay, and once this requirement is done so you get a 1 sorry this is not your immediate requirement so this is your requirement number 1 this is your requirement number 2.

So if you can get a 1 over here if you can get a 1 over this point then you circuit testing is done by applying this, so somehow you should get these two things indirectly so that is saying so this one is your input pattern a1 can be applied directly but d=1 and i=1, d=1 I have to we have to do by second by some other mechanism so that is what is very must required and we will see how to do that by time frame expansion method.

(Refer Slide Time: 43:59)



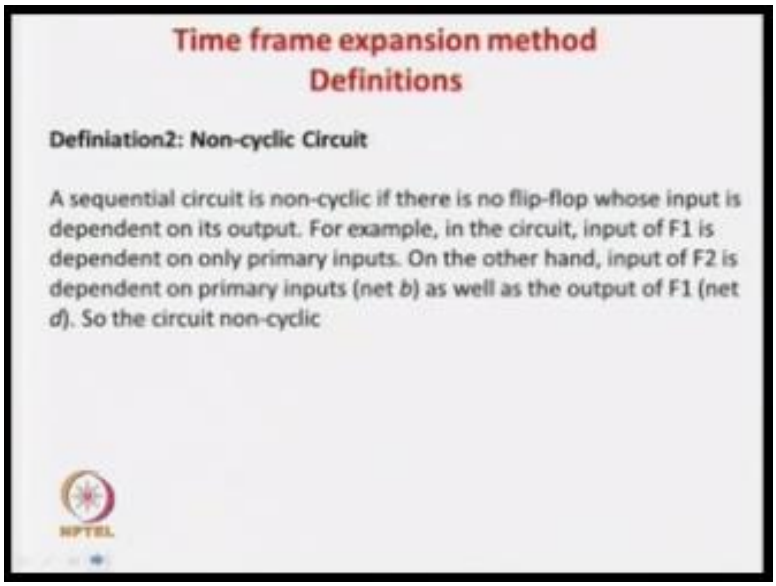
Okay, now so we just to very simple definition so one is called the sequential depth of a flip flop so the output of a flip flop can be controlled by only primary inputs and clock pulse then the sequential depth of 1 that is just look at this flip flop so it can be controlled directly by the primary inputs so the depth is 1. Now if you look at this flip flop so it depends on what it depends on this input, this input lying e which is again dependent on this flip flop which is having a so this flip flop is dependent on sorry, so this flip flop actually is dependent on this input correct and this input is dependent on the output and that flip flop.

So and this is having a sequential depth of 1 because this is directly controlled by the primary inputs, so depth of this one will be equal to 2. So what is the definition just if you look at it so if the output of a sequential flip flop is direct this one this is one then the circuit of last figure if you

take then this is sequential depth of1 that is directly controlled by C that is what, so directly controlled by c so depth is 1.

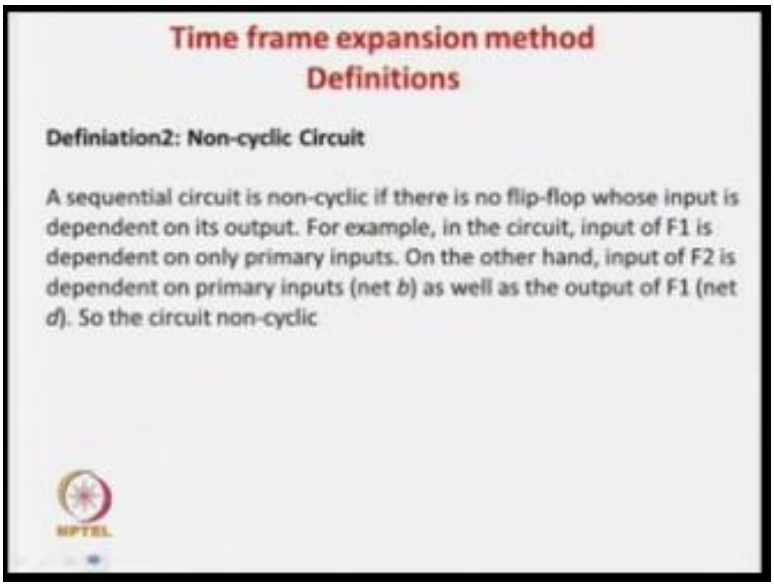
Now this flip flop is saying that a flip flop of sequential depth this sequence if the output is dependent on primary inputs and at least one flip flop of depth n-1, so that is you have a flip flop here so this input is dependent on a flip flop depth is d sequence -1 and some other primary inputs it can be there, okay. So that is in indirectly these are flip flops whose input is dependent on another flip flop whose sequential depth is this sequence -1 then you have to add a 1 for this flip flop and it will be d sequence.

(Refer Slide Time: 45:34)

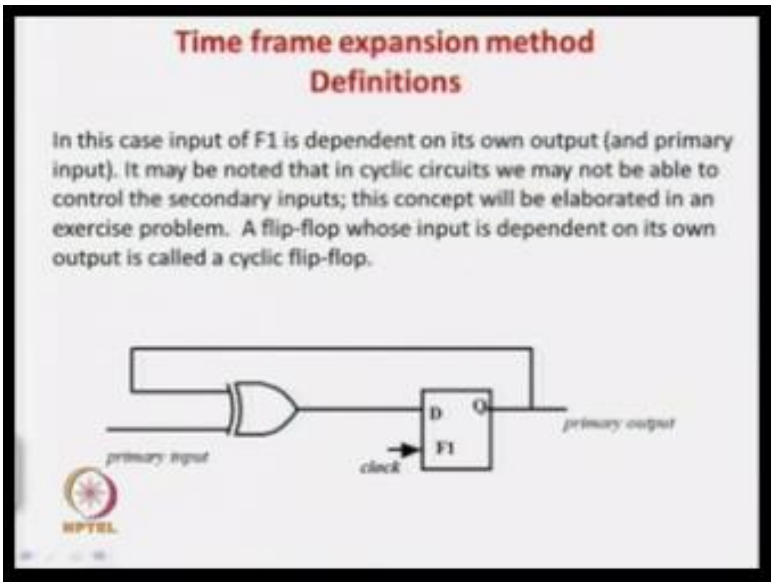


So that is what so first flip flop you consider which is dependent directly primary inputs that will be value 1, then you find out the next of flip flop which is dependent on primary inputs and all those flip flops which sequential depth is 1, so their value will be 2 and you can keep on doing it. So in our example, this is having a depth 1 and this is having a depth 2 so that is the next definition we have got, correct. So now there is a definition was cyclic circuit.

(Refer Slide Time: 45:55)



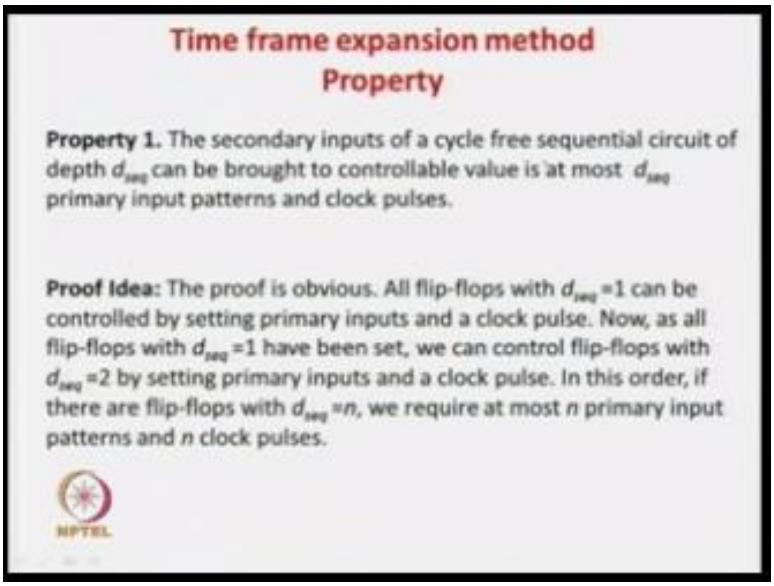
(Refer Slide Time: 45:57)



So what is the cyclic so this definition says that a sequential circuit is non cyclic if there is no flip flop whose primary input is dependent on output. For example, if you consider this circuit this is a sequential circuit because its output IS dependent on its this one you just look at the feedback so it is output is dependent on its own input so that is actually a cyclic definition, but if you look at this flip flop this is not a cyclic this one because this input is not dependent on this output.

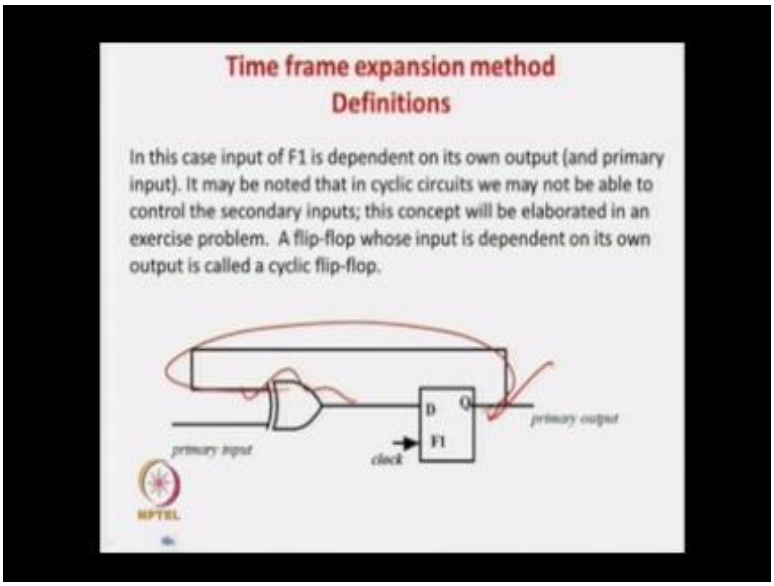
So that is the definition of a sequential what you call cyclic circuit, okay so non cyclic circuit means the output of the input of the flip flop should not depend on it is output, this one so this circuit is not definition so this is cyclic circuit and the other example is a n on cyclic circuit.

(Refer Slide Time: 46:44)



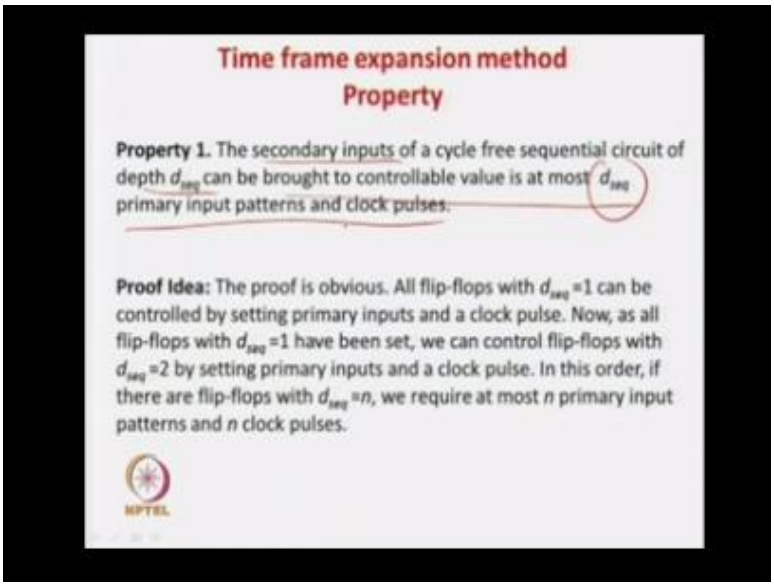
So now we are going to very quickly see a property so this time frame expands the property this property the secondary output if the circuit is cycle free okay, if the cycle is having or what do you call if the circuit is having a cycle then we will see in the end that no testing can be or it is very difficult to do testing for that kind of a circuit, because there will be continuous oscillation.

(Refer Slide Time: 47:08)



Like in this case the output will depend on inputs input is dependent on output and it will be oscillation and you cannot control any value. So circuit faults cannot be tested in the circuit it is very difficult to that but we will see. But for that case the circuit so there is no cyclic property then what we can do is that we can control all the virtual primary inputs and observe all the virtual primary output that is the properties saying. It is saying secondary outputs that your virtual primary inputs of sorry. The secondary inputs that is your virtual primary inputs of a cycle free sequential circuit they have this one can be brought under control in at most like this one.

(Refer Slide Time: 47:48)



In D sequence number of primary inputs and clock pulse. So what we are seeing in the last three examples is that we require some nets like, in this case if you look at that. So we require ton control this two one and also this two one. So we can now ask that how many clock pluses and patterns will be required to do this the property says that, if the maximum net is 2 so you can do it in 2 clock pluses and 2 inputs. Now how it can be done it is very simple the idea is very simple because this is sequential depth 1.

So sequential depth 1 means what it is directly connected to a primary or a set of primary. So give some whatever pattern is required at the output in this case it is 1 so you can apply a 1 in C and 1 in clock pluses directly set. That is the in other words all what you can call all these flip flops which are in sequential there are all in depth 1. Can be directly controlled by setting the primary inputs apply at clock 1, because they are directly dependent on the primary inputs. Now after 1 clock pluses and 1 pattern the entire sequential all the flip flops of sequential depth 1 has been set.

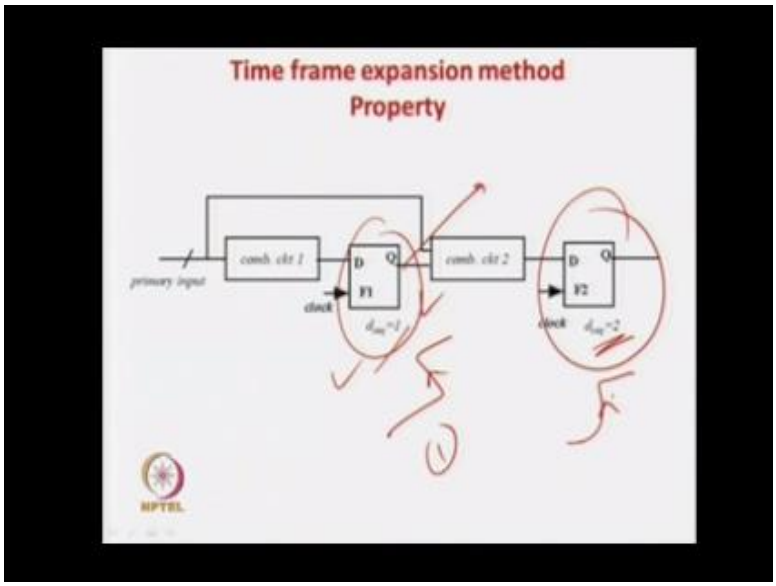
Now you can consider for the flip flops sequential depth 2 so now we know that after 1 clock pulse 1 pattern all the flip flops having a sequential depth 1 has been set. Now flip flops with

sequential depth to be dependent on what primary inputs and outputs of flip flops which are sequential at 1. So now indirectly you can say that 1 more pattern and 1 more clock pulse can be set in, because after one pattern this guy is set, because this sequential depth one and this guy is dependent on primary inputs and only on this output.

That is only on flip flops whose sequential depth is one, because the sequential depth is 2. So now you can apply another clock pulse and this value will be output. So by second clock pluses all the flip flops has been having a all the flip flops are having a what you call sequential depth 2 will be set. Similarly if you have a sequence depth flip flop free then you has to have a third clock pulse, because by second clock pulse all the primary inputs you can set6 it. And all the flip flops sequential depth has to be set by the second clock.

Now in the third clock pulse all the flip flops or the sequential elements with depth 3 be free. Because in the second pulse all it is inputs are red because the inputs over flip flop with a sequential depth 3 are, are flops with sequential depth 2 which is already been set in the clock pulse two. So another clock pluses you apply you are third level flops or sequential depth third flop will be set and so on. So the proof is very obvious that which we have discussed and we can see this pictorial this way.

(Refer Slide Time: 50:05)

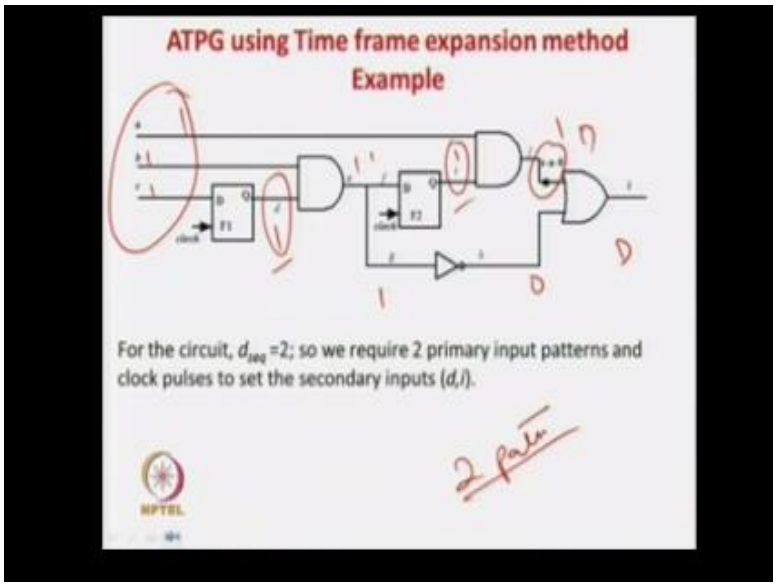


This is a sequential set of depth 1 so it is a primary circuit primary clock we apply, we apply a clock this is set. For the depth 2 now in depth 1 in first flop it had been set. Now in the depth two you require another clock pulse because now this because in first phase all this is set and in case of the second in case of the flops if there level of 2. All the level of flip flop 1 has already been set so just you apply the clock pulse and this output will be controlled.

Similarly dot, dot, dot, now if there is another flip flop whose depth required with third pulse because the second clock pulse all the flops of depth 2 has been set. And flop with level 3 are dependent on the output of the flops with gate 2. So third clock pulse is required so very obviously it is said that the secondary inputs of cycle free circuits of this can be brought under control at most D sequential primary inputs and a clock pulse.

So therefore if you have a circuit with 10 flips flops having sequential depth 10 so you require 10 clock pulse and 10 patterns to set the virtual primary inputs and then in eleventh period you are required to give the primary inputs and that will solve your problem okay.

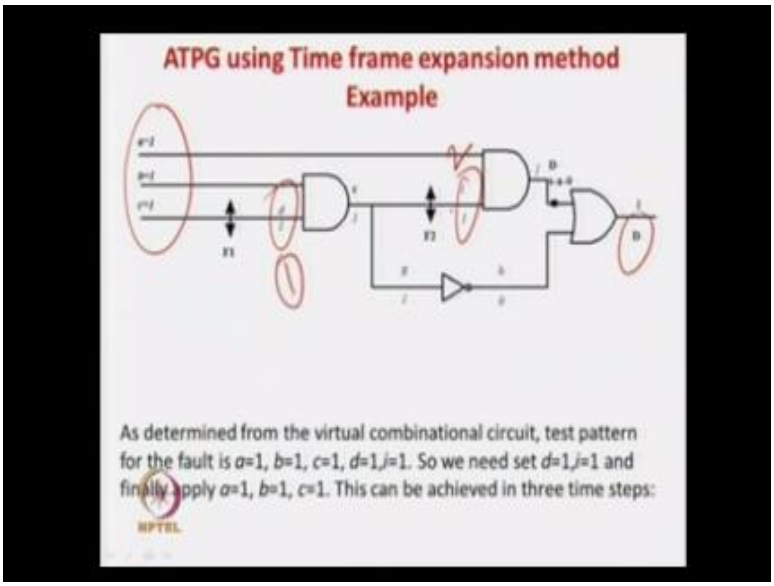
(Refer Slide Time: 51:17)



So again coming back with this property we will see how this time frame expansion method can be applied to the circuit. So you know that this is a stacked are there sequential depth was 2. So here we require 2 patterns two we have already seen that we require 1 over here we require a 1 over here correct to do this testing, because you are going to apply 1 this will be D this will be D we require a 0 over here, we require a 1 over here, we require a 1 over here, 1, 1 this is also required a 1 over here 1 over here 1 over here something like that. So this we need you control this one and we need to control to 1.

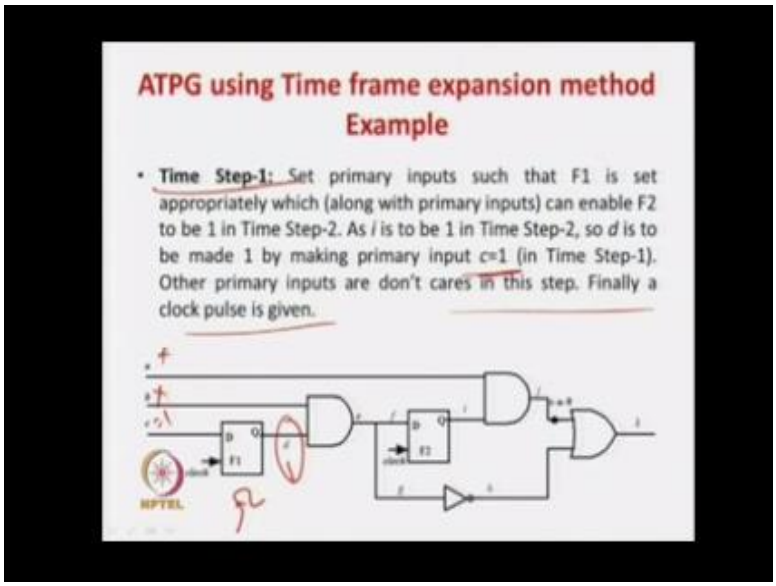
So and sequential depth is two so you are required to clock periods to make this one to this one and this one, and finally you have to apply A =1, is =1 to test this one. Let us see that the explicitly how it can be done so two patterns to set and third pattern to testing.

(Refer Slide Time: 52:01)



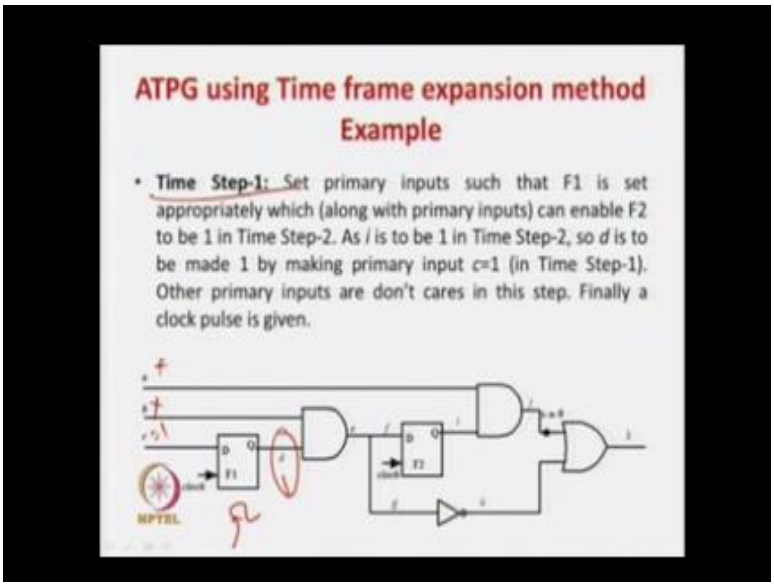
So what we do so first so this is what is the case so you require this is the test pattern as I told you require a D=1, I=1, to do this testing. So we are required to control this at level 1 clock, because it is at the level 1 flop and this is will be equal to do the second clock pulse, because this is the level 2 clock, so this is what is required.

(Refer Slide Time: 52:21)



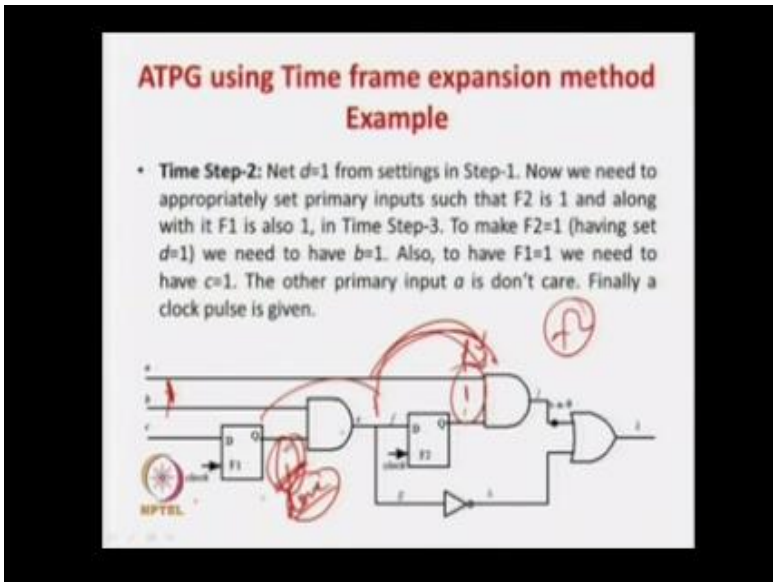
Now you see this is time step 1 so how to do that so time strep 1 what we do we require a 1 over here. So what we do we make C=1 others will do not require this at all and we apply a clock pulse. So what is going to happen we are going to get D=M, so this you can think that your clock pulse is set okay. Right so this is what we have done in this step1. Other primary inputs are do not care and your clock pulse is given C=1 and it is shown and we do this to get a 1 over here and it is done.

(Refer Slide Time: 52:52)



Now so what is happen so you can know that now D=1 already and this is x, this is x. we require a one over here this is very important like this the second phase is 1 okay. so this is requirement this is already done now what we have to do so now we require a 1 over here and if you have a clock pulse we can get it 1 over here, because what is already done in last phase. So now what we have to do to get a 1 over here why we do you need we need a1 over here and we need a 1 over here. So 1 over here is already done in last step, so if you apply a 1 over here and you apply the second clock here.

(Refer Slide Time: 54:01)



Then what happens this 1 will be reflected over here. But one more point you have to remember that this 1 and 1 will be reflected over here but the same clock is going here and same clock is going here. So it is nit that if you want to transfer from here to here you have to apply a clock over here and the clock is not coming here. The clock will also be coming over here so we should be very careful that.

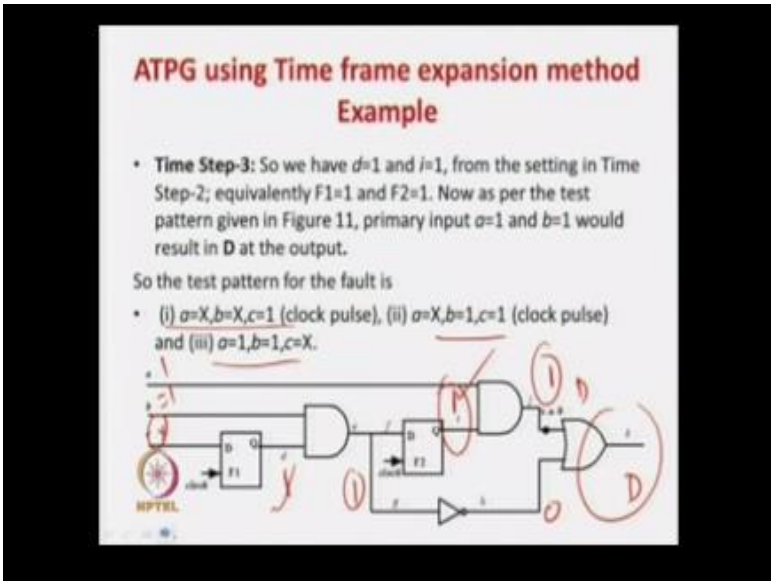
When this one or this one together is transferred over here so this should remain 1 because in the end of second phase we require a 1 over here and 1 over here. So you apply a 1 over here so what we have done so indirectly we have do not over here is the first phase we have a 1 over here and the second clock pluses you put a one over here.

So 1 and 1 you get a, a 1 so just apply the clock pulse you will get this one but you should be very careful that the second clock pulse also 1 is to be here so you also put a C=1. Now you apply a clock pulse same clock pulse will go both so 1 and 1. One will be coming over here and what this one will be maintained over here. So now you get after this the second clock was we get a 1 here a 1 here. So 1 we will get a 0 over here and 1 sorry a 0 over here that is what here

and after the second clock. We get a 1 over here we get a 1 over here okay. We get a 1 over here and a 0 over here.

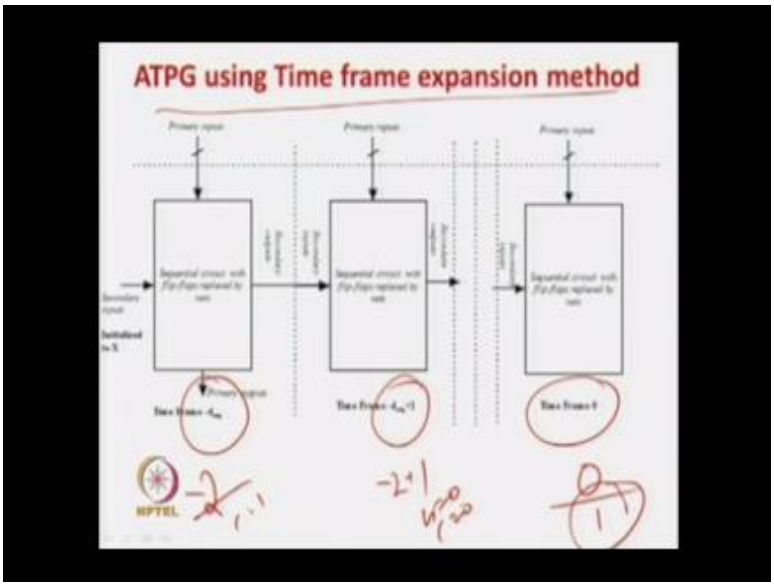
So this what we have got after your second clock and the times frame there finally you have to apply the clock pattern and you will be done/ so you are getting a1 over here you are getting a 1 over here it is already done in the first two steps. So now what we do we require a 1 over here that we will get a n then you have to propagate this it should be a 0 okay. this should be a 1 and you can do the testing so to do this what you require to get a 1 over here we require a 1 over here that is already set but this is a virtual primary input so you have already set by the clock pluses you just apply B=1.

(Refer Slide Time: 55:01)



So you will get a 1 over here and a 0 over here and this is done. so now you also require a 1 over here since it is a fault so for that this is 1 this is also set in 2 clock pluses this is virtual primary input and either primary input so you want a 1 here so you apply a 1 over here and this done. C=X, so three clock period so first we apply a C=1 set this as 1. Second clock pulse we apply D=1 C=1 D=1 and C=1, so you set this one. And third clock you apply D=1 C is do not care and you do the testing.

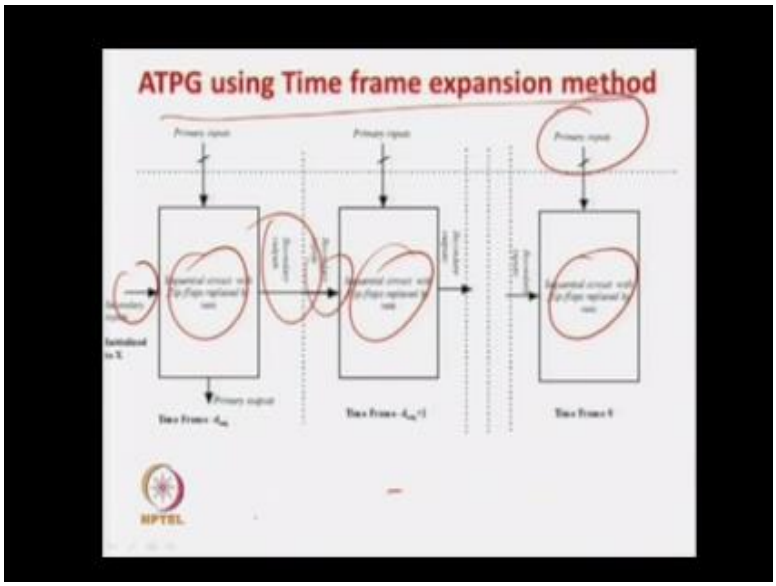
(Refer Slide Time: 55:48)



So that is what we have done it is actually called this is actually called a time frame expansion method because in there clock period we have tested this circuit in the first two clock period so what we have done we have set the non primary inputs or the virtual primary inputs secondary inputs and 1 and 1. And third clock pulse we apply the primary inputs and the testing has been done. so if I want to illustrate in a figure this is what has need done so if there are we the sequential depth is D then we require one more time which is –D then we actually –D+1 dot, dot, dot till D. So in this case we have a frame words with a depth of 2 so it is 2-1=1. Then this will be depth 0 and this is depth 1 kind of a thing okay.

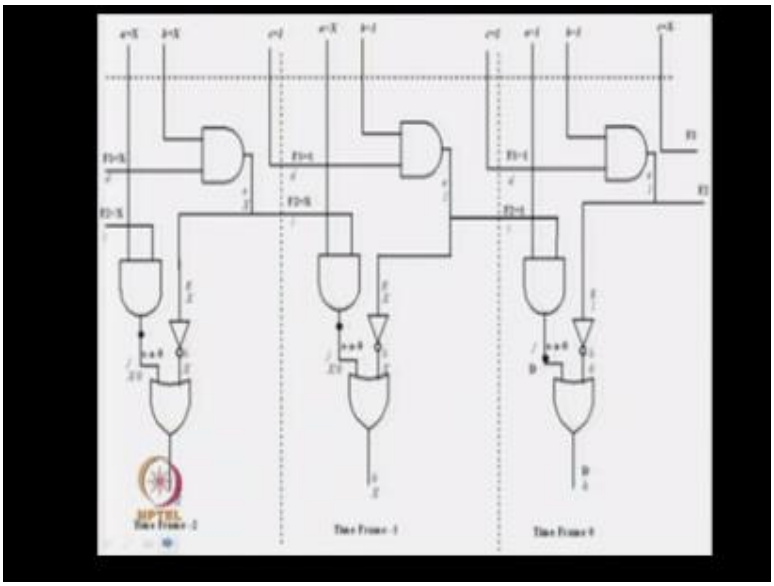
So you this is sorry so our sequential depth was 2 so it was – 2 then – D, D was 2 in this case then in this -2 and then -2 +1 that is 1 and then 0. And then finally so applied A=1 B=1 which will be the testing in this case we applies c=1 we set the value first output of the first flop up to 1. And in this case we apply D=0 and C=0. Then we set the values of the next flip flop of 2 1. And finally apply this pattern and test it. That is how we represent this circuit and this is the way of representing your time frame expansion method.

(Refer Slide Time: 57:10)



That is we replace the flip flops with the replace by net so they are eliminated so this is what we do and this is your secondary inputs not the primary input this is your secondary outputs. And this is how we represent this is your primary inputs and this is how you control so now just whatever we discussed.

(Refer Slide Time: 57:28)

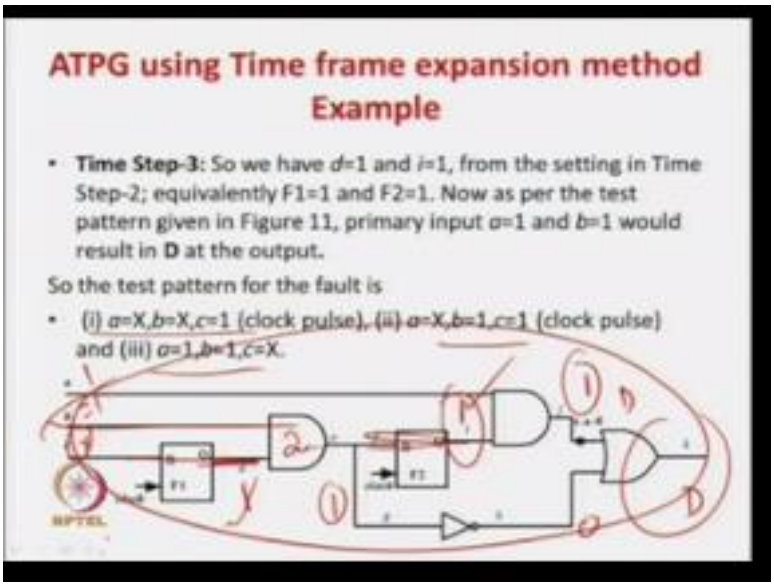


In that so let us just put in the time frame method and see what we have done so this was your circuit if you remember this part of the circuit we have drawn in a horizontal way this circuit we have drawn in a horizontal way so vertical way these things are the shorted and this thing is shorted this is what we have done. So just see this is your flip flop this you have entered so let us just mark this as say we can it gate number 1 and gate number 2.

So we have to just remember that input of gate one is D and D. input of gate number 2 is I and A. but this I is a virtual input and this D is for this case is a virtual input. So you can just see what we have done this is say we can think that see this gate is 1 and this gate is 2 kind of a thing see 1 gate it is directly connected from this is gate 2 and this is gate 1 this is gate 2 and this is gate 1 this is gate 2 and gate 1 gate 2 and gate1 okay.

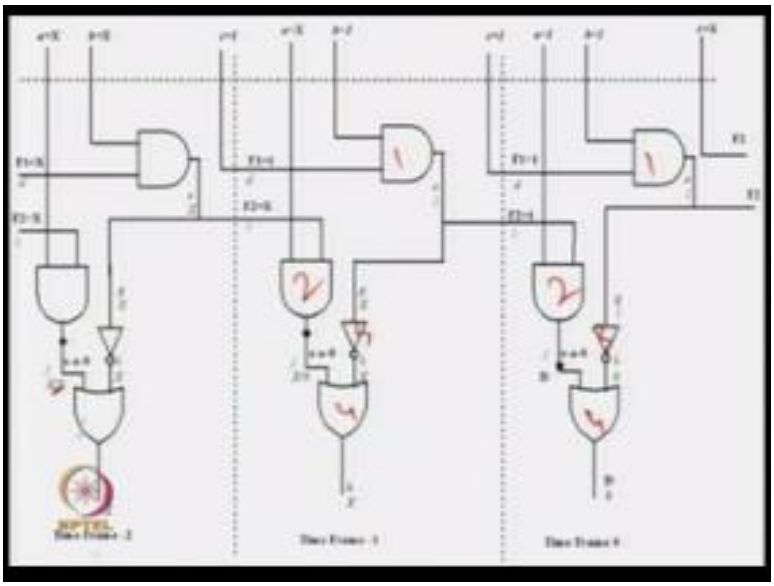
So see thus this gate 2 is dependent on A and the output of this second free flop this how we represent it so this gate 2 is dependent on at A that is primary and output of the second free flop. And gate 1 if you remember that was dependent on b as well as the output of first three flops. So this is your gate 2 this is our gate 2. So it is dependent on B as well as the output of the first flops.

(Refer Slide Time: 59.11)



So this is how it is we can call this 1 2 this can be called as 4 this can be called as 3 so this can be called as 4 this can be 3 or whatever you like this is that not much important so this only thing is important so gate 1 is dependent on B and the output of the first prefer gate 2 is dependent on A and this one. So we write it this one these are stuck at fault 0 so we write this is as extended algebra.

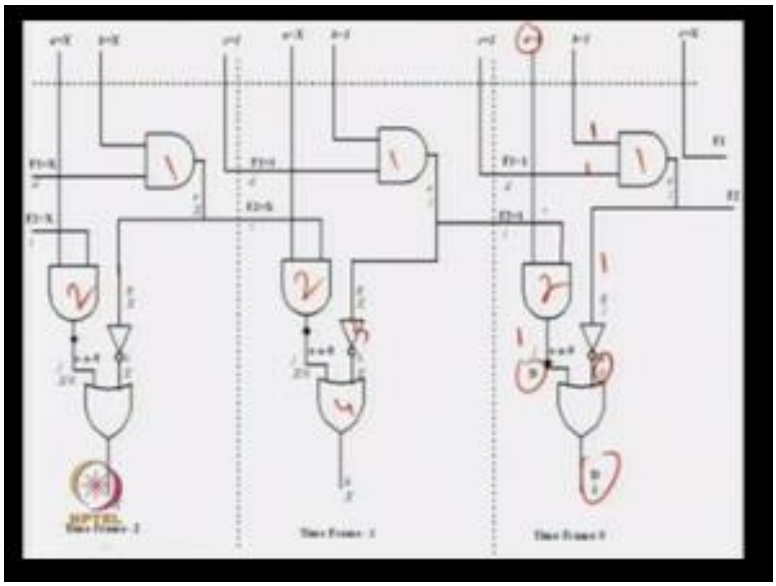
(Refer Slide Time: 59:45)



So it stuck at 0 so normal fault case it as 0 it is unknown so all other things we have marked as X so just this can very easily so all other things are X so only how you have written this circuit as instant of this gate number 2 and gate number 1 instant that the this is connected to the output of flip flop the eliminating flip flop then we say f2 as output.

Similarly for gate 1it is dependent on f1 output so we have say that flip flop is not a directly connected this output. So now you see what you require we require to test this circuit so what we require go to the last level of the circuit this is 2 and 1 so we require d over here. So d has to propagated over here so you require 0 over here just D algorithm okay 0 means you require 1 over here you require one over here.

(Refer Slide Time: 1.00:33)



Similarly so this is stuck at 0 so you reply 1 over here and you require 1 over here and 1 over here so this what we are requiring over this. So now this will actually test your circuit this is the last time to marked it. Now to get 1 over here so this is at the 0 times stuck so this case is connected to the output of second flip flop okay.

So this get to 1 the output of flip flop is to on primary input can be said as 0 x times step that is done. But the other input 1 has to come from the output of flip flop number so you have to write the output of 1 from this previous one. Similarly for this gate number 1 you require a 1 over here and require 1 over here so this sit her primary input can be controlled.

But the other input that is C equal to 1 that is and the gate that is actually also note that it is output of the flip flop number1. So you have to also know as that the flip flop number 1 has to be a 1 you make this one time step as 0 that is in other words we say that the flip flop is 1 the output has to be 1 in the previous step to get 1 in the next step.

So you write that flip flop is 1 is to be 1 over here flip flop 2 is to be 1 over here that is the times step 0 and the requirement when you can achieve it this flip flop 2 1 will be achieve as time step

minus one and this also achieved at the previous time step. So now in the previous time step if you are want to apply a x=1 so what you have to do you have to make is equal to 1.

Because input of the flip flop 1 is directly dependent on C so this kind can be done very easily by applying x=1 in the previous time step. Now but you require flip flop output number 2 is also to be 1 but how to achieve this that is some are not very simple because again the output of the flip flop 2 is driven by gate number1 which one of the input is primary input.

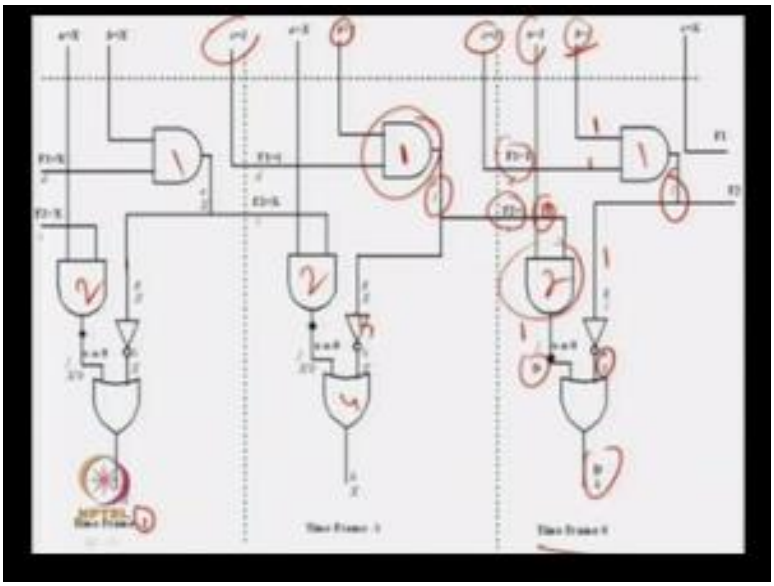
So you can directly said it as time step minus 1 so you can directly said it as a time step 1 but another input of this AND gate which is driving flip flop number 2 is dependent on the output of the flip flop number 1. So that has to be control in and the time step which is 1 and before so you to get 1 this which actually make flip flop in third time step so what you have to we can to get B has 1 in the second step.

But again another input that is actually the AND gate number 1 and flip flop number 1 so it has to be controlled at time step minus 2 so in minus 2 if you make set a=1 so what will happen that flip flop output 1 will be minus 1 that is what actually we are doing. So indirectly what we are requiring just very quickly look at this snap what we have been done we require 1 over here which is flip flop number1 so to get 1 over here we require this AND gate to be 1 by the primary input on fine.

But another input of the AND gate is actually output of the flip flop number 1 so you also have to be 1 over here so how we get that as this f1 is actually flip flop which is controlling gate 1 it has to be controlling gate 1 in the previous step. So in step minus 1 time step has one applies equal to 1 we get it.

In time step 2 again AND gate is here so we require 1 over here another 1 is by this primary input equal to 1. So it can very easily get in the time step 0 on the third step. But another input has to be 1 the output of flip flop number 2 so this has to be a achieved at previous time step minus 1.

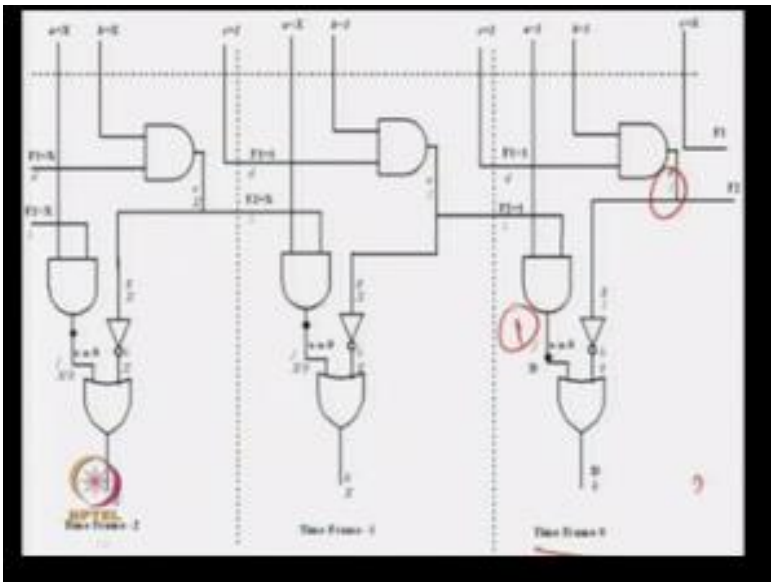
(Refer Slide Time: 1:04:19)



So to get 1 over here you require 1 over here so this can be done in the second step but again this AND gate is dependent on flip flop number 1 so it cannot be done in the correct instance. So you have to go one more stage before and you have to make is equal to 1. So if you make is equal to 1 in this time step minus 2 and then you make B equal to 1 at time step minus 1 then what you are going to get that output of 1 in the second stage which can be directly applied to the third stage.

And if you are making C equal to 1 time step minus 1 or step no 2 then what you are going to get in the third stage you are going to get a 1 over here indirectly what we have already seen in the long example before we are just writing it in the formal way that is just some requirements the requirements are 1 over here and 1 over here there are the two requirements at the time stage number or in step 3.

(Refer Slide Time: 01:05:09)

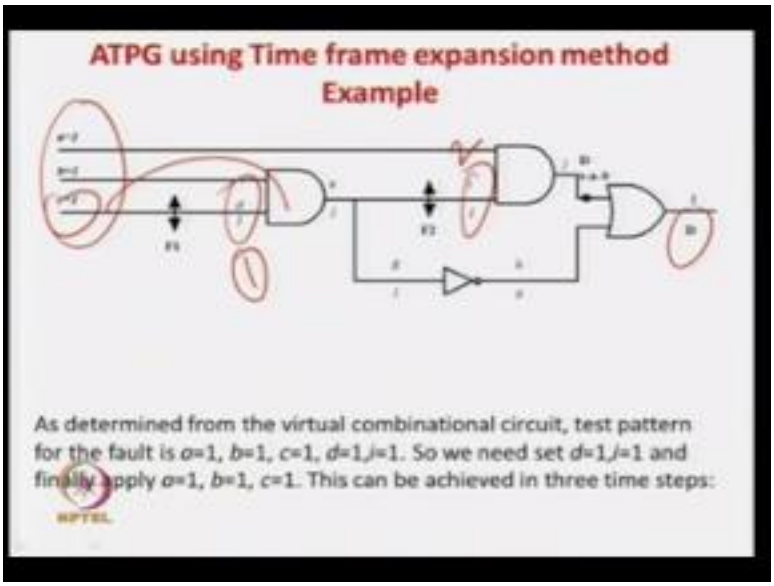


This step 3 you require this so this requirement this can be achieved by making b=1 another input has to be a 1 which we can make achieve by making c=1 in time step 2 you can say or time step minus 1. This requirement of 1 that is these are again virtual primary inputs which are direct controlling.

So now again these 1 is required to get this as 1 so this one is as 1 which we can easily get time step 3 you can say but another input is to get as 1 we require this step to make this as 1. To make this as 1 we require b=1 in time step minus 1 and also another input is there which is not dependent on the time frame minus 2.

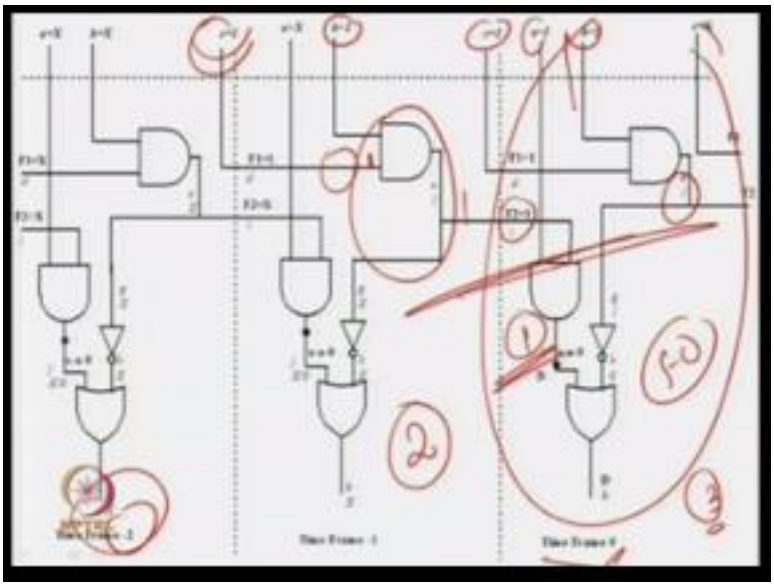
Because this gate is dependent on 1 flip flop level 2 and level 1 so again go back to time frame minus 2 and you have to said as 1 if you do that than this one will also become 1 and finally you are going to get the requirement. So these are nothing but whatever we are discussed nothing but these three stages.

(Refer Slide Time: 0106:23)



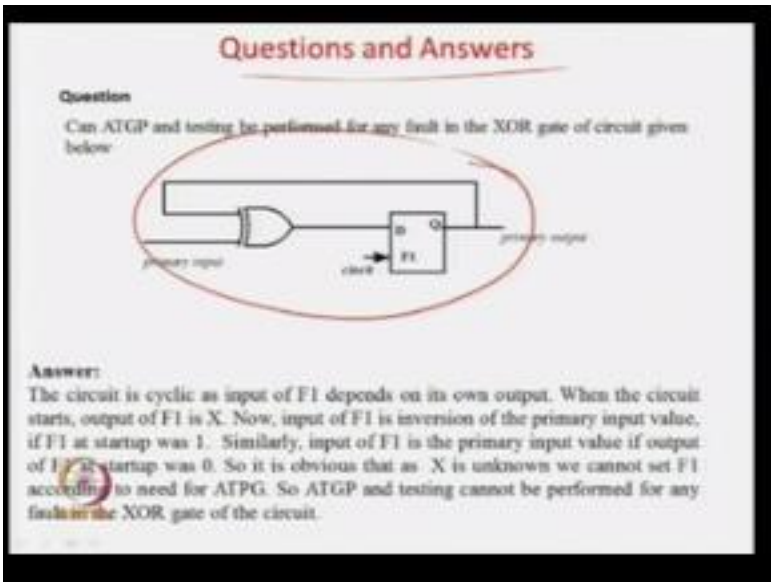
This stage 1 the very where we are making is equal to 1 and applying clock that is stage 1 then next is making 1 here and applying here and third stage is nothing but applying this one in the testing fault. Here are the three stages which we have done just representing pictorially in this stage so this are the very formal way of a time from expansion you can solve your this thing. So that about time from expansion method stage so in this case we have to observe that to test a single stuck at fault what they have to do only one step would have solved your problem.

(Refer Slide Time: 01:07:02)



But in case of sequential circuit we require much more number of stages and number of stages are equivalent to nothing but number of sequential data of flip flop that is why you can say that is why very, very complex for sequential and also remember for any stage you may have backtracks leading it to a big problem. So in the next lecture you are going to see how we can make this very complex problem has simple one. But before that very quickly you have see about the question so this is a cyclical circuit.

(Refer Slide Time: 01:07:32)



And we are not consider here any cyclical circuit so now if I ask you that can you test any fault in this circuit a question is can ethical and testing the perform any fault in this OR gate. The answer is no why just we will see say for an example the output is always have x so you will get x so now if you have the primary input as 1 so your answer will be the output is always x because we know that in case of x OR gate if the input is 1 as inventor.

Now primary input is 0 you are going to get a x so no point of time you can able to get a value of 0 over and 1 over here. If you get primary input as 1 if you get 0 it is an some other thing. So at no point of time you can get any control decision or control value for the decision over here. Because the input is dependent on own out as if we cannot said flip flop so if you have a circuit something like that is no point of view time control x to any other fault cannot be described.

So with this we come to the closing of this lecture and in next session how we can solve the problem of such a huge number of test patterns required for our sequential testing so now we reduce the number of test patterns required or number of sequential required for testing of fault less sequential circuit. Thank you.

Centre For Educational Technology

IIT Guwahati

Production

Head CET

Prof. Sunil Khijwania

CET Production Team

Bikash Jyoti Nath

CS Bhaskar Bora

Dibyajoti Lahkar

Kallal Barua

Kaushik Kr. Sarma

Queen Barman

Rekha Hazarika

CET Administrative Team

Susanta Sarma

Swapan Debnathi