INIDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL
NPTEL ONLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification & Test

Dr. Santhosh Biswas
Department of CSE
IIT Guwahati

Module X: Sequence Circuit Testing and Scan Chains

Lecture I: ATPG for Synchronous Sequential Circuits

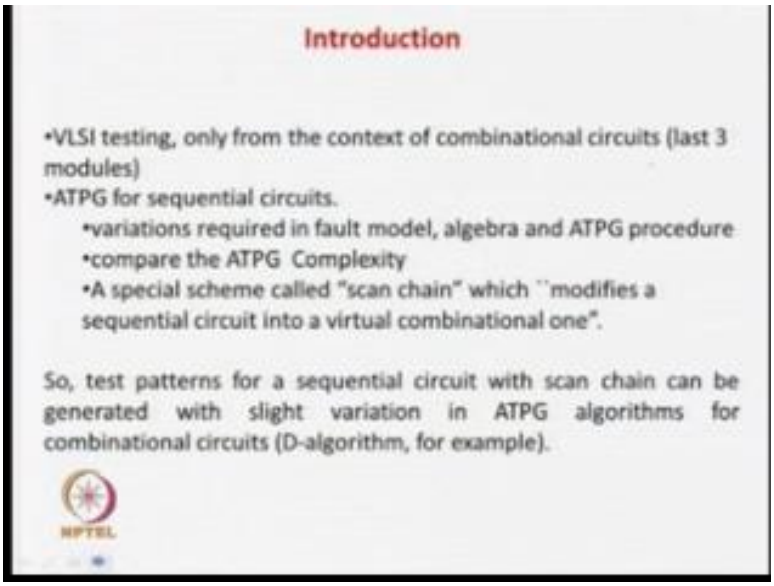
Design Verification and Test of Digital VLSI Circuits
NPTEL Video Course

Module – X
Lecture- I
ATPG for Synchronous Sequential Circuits

Okay so welcome to the, and to a new module, module number 10 in the lecture series of VLSI testing so what we have dealt till now in the test series in the discussion on test we are regarding combinational circuits so we have started with for fault models and we have seen for random test pattern generation then we have seen fault collapsing SCOPE algorithm sensory propagate and justify base algorithm as D algorithm and so forth but most of them where limited to combinational circuits so in new module 10 we mainly look about how the same things or all the technologies like stuck at fault modules collapsing etc.

ADPG algorithms etc whatever we have seen for combination circuits how they may up to sequential circuits so this lecture I mean today we learn on automatic test based on generation for synchronies sequential circuit that means you should also know that.

(Refer Slide Time: 01:15)



Sequential circuits can be synchronized as well synchronous which we have learned in our basic digital test but actually here we will mainly dealing with synchronies sequential circuits with a single clock I mean to make the course a bit simple because this course is mainly on VLSI design verification and test so we are going to have a overview all the 3 aspects of VLSI design so rather than going into in depth of synchronies circuit testing and so forth we rather though we will try to give you an board over view.

So to limit the depth of sequential circuit ATPG we limit our scope to single clock synchronies circuit and mot of this is design we do these days 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 for sequential circuits and a very few limited circuits you can think of run as synchronizers so more or less even if you are not covering but this does not have a two much impact on our VLSI testing study.

So basically what we will study so I mean we have seen combinational circuits in the last three days so now we will today we will see for a sequential circuits 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 seen

that the fault model is not absolutely similar in case of combinational sequential circuit but so one or two small assumptions may be there that we will point out so we will see that there will be requirement of higher order algebra for sequential circuits and we will see the ATPG producer and we will see how different is ATPL in sequential circuits compared to combinational circuits that is what we will see in the next lecture may be we will see something like a scan chain etc…

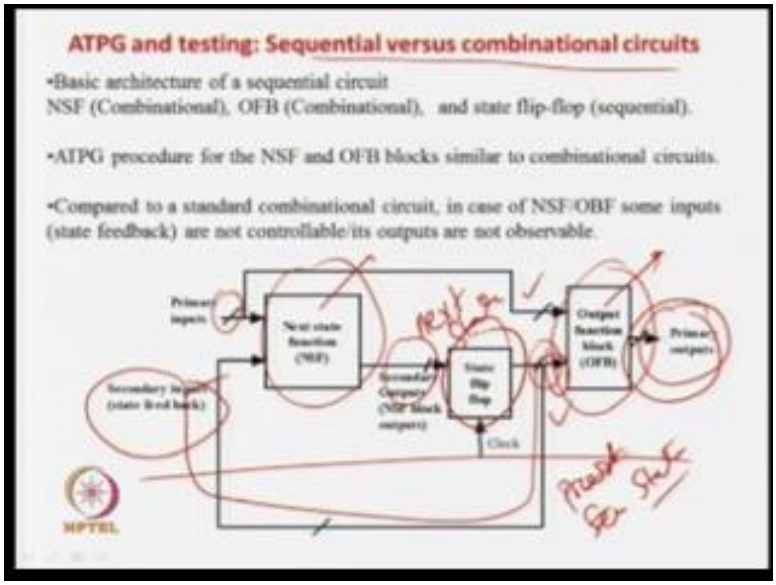
So which can help you in ATPG so we will show that the ATPG procedure complexity in case of sequential circuits is much higher then we have combinational circuits so following that we will find out the technique design technical scan chain which will help you to reduce the complexity and which will help me in ATPG for sequential circuits that will be the in the next two lectures may be.

So I mean so tested and generation for I mean we will see the scan chain and also test generation for sequential circuits with scan chain can be generated with single ATPG algorithms D example and all so this is about in details we will see slowly I mean in today's what we will discuss we will mainly see what is ATPG about different why is ATPL different sequential circuits and then we will see that in the next lecture and also we will pinot out that ATPL for sequential ATLP is more complex in combinational circuit so followed by so in next lectures we will see that how we can develop or something called scan chain which will consists your sequential circuit to a virtual combination circuit.

And the all algorithms which are which developed for combination circuits can also be applied to sequential circuits with a scan chain without any marked change so in that way we will handle the complicity of ATPG for sequential circuits but that will be in a later half of the lecture so that is why it is said that basic motivation of developing this 3 lectures are 4 lectures of sequential ATPG we will be to convert the first strategy complexity of ATPG for sequential circuit compared to combinational circuits and then convert the sequential circuits to virtual combinational once using scan chain.

So that ATPG algorithm for combinational circuits like D algorithm etc can be applied to sequential circuit with the variation so that is the basic agenda of our 3 or 4 lectures on sequential circuits testing in this course so as we said that 1st our main goal today our in today's lecture we will be to see how combinational circuit ATPG is different or how sequential circuit ATPG is more complex compared to combinational circuit ATPG that what today we will see that is why we say that.

(Refer Slide Time: 04:44)



Sequential versus combination circuit ATPG and testing what is the difference that is let us first study so if you look at this just look at the figure this is 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 straight registers or flip flops.

The output of the next state function block actually tell you which will be the next state of this circuit which are also secondary output because the primary outputs will be outputs on the output function block which we will see later but the output of the next state function block in is a we

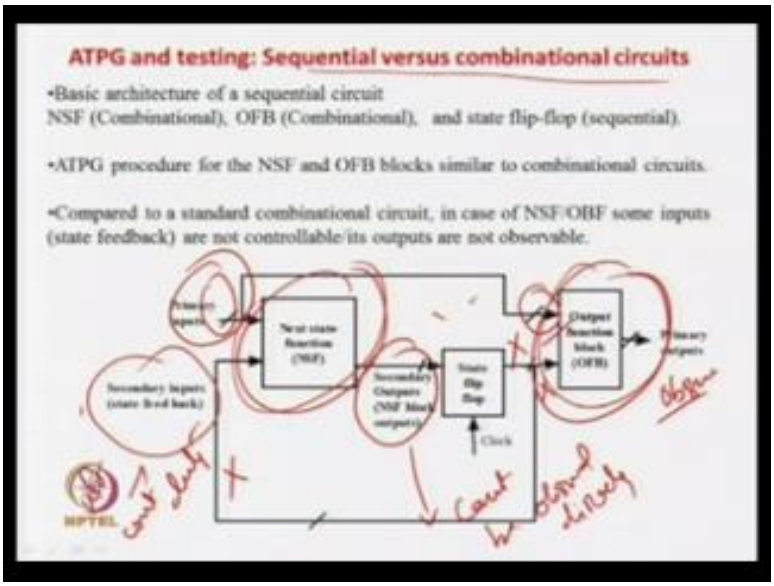
will tell you that this is your secondary output because this is not directly output but it is the secondary which will determine next state of the circuits so this is also come times called the next state bits of the circuit and similarity and may be the output of the state flip flop represents state.

So we all know that from the digital design so this combination represents state and also they are actually feedback to the next state function block so we call them as virtual primary inputs are secondary inputs okay and these are last block which is actually called the output function block which take as input the present state as well as the inputs so if we does if it take this input we call it the mini machine and if he does not take this primary inputs only it depends on the next state I am sorry the present state.

So it is called the more machine so already we studied them in the digital design so any way to make things general the output function block will take the primary inputs and well that the present state and it will generate the primary output okay so I mean if you look at this block we carefully so we have three main blocks next state function block state represent output function block okay among them this is a combinational circuits this is a combinational circuit block and state flip flop sub way sequential circuit blocks.

Okay so now if you look at this from this sequential circuit basic block diagram from the combinational circuit point so you can see that this is a combinational circuit so you could have done ATPG using or D algorithm or whatever we have learned there is a small problem here so in case of D algorithm or automatic test pan generation by random what you assume that.

(Refer Slide Time: 06:55)



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 extra circuit to control this inputs are observe the primary output you may require some circuits to control intermediate lines but we have again decreed that approach but if you look at this very carefully when you circuit will start up.

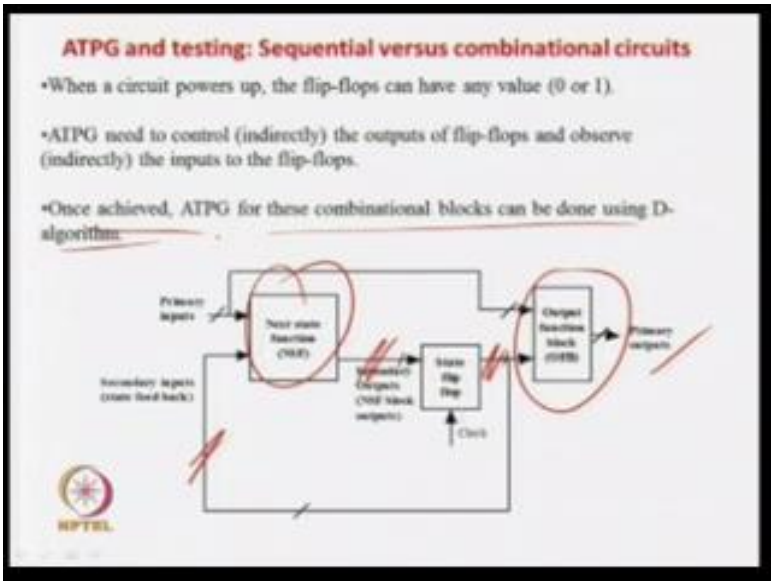
So output of the flip flops are not known sometimes they can be 0 sometimes they can be 1 you do not no acutely what will be the value so we generally call them as X only you can control the primary output similarly for the output function blocks this set of inputs that corresponds to the state prefer outputs are X that is they depend on what will be the value when your circuits starts.

So if you look at it so this is a primary output so very good so you can observe them directly so this is not a problem but you can control only half or one third or some fixed fraction of your inputs of your circuit for this combination blocks and the output function block similarly for this block if you see you can control this position of your primary inputs for this nested function block for the secondary inputs of the virtual primary inputs you cannot control it directly similarly again another problem with this block is this outputs actually also you cannot directly

observe because thy are going to be fare to the next flip flop next bits I mean as three inputs of the flip flop.

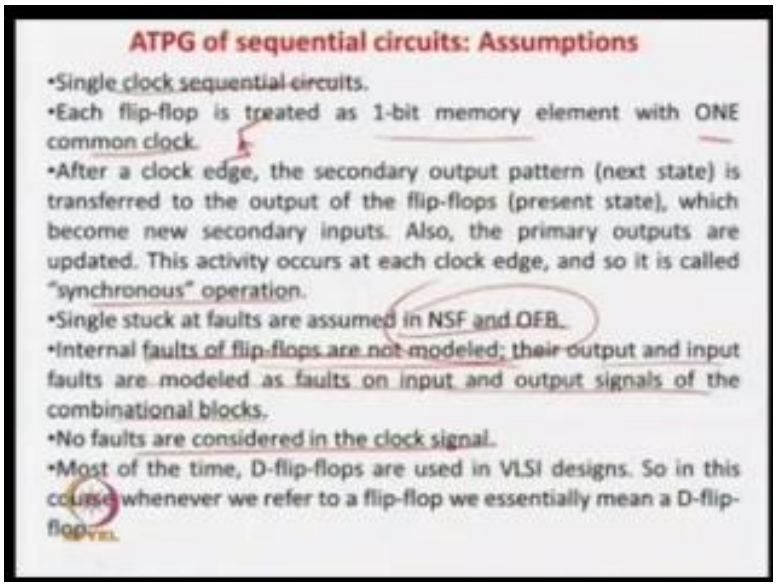
So this cannot be observed directly similarly this cannot be not controlled directly so that is actually the different so headed in a purely combinational circuit testing algorithm then you should have been able to there should have been decoupling over here so you should have been directly able to control these points and you should have been able to directly observe this part similarly for this so as these things are not possible so there is what we require so we require two extra steps that is when we are going to test a sequential circuit they are the two combinational block co I think you can directly use the combinational algorithms like ATPG D algorithm and all but with a extra constrain that somehow you should control them indirectly.

(Refer Slide Time: 09:00)



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 is it says so one state is achieved then ATPG for this combination blocks can be named D algorithm.

(Refer Slide Time: 09:14)



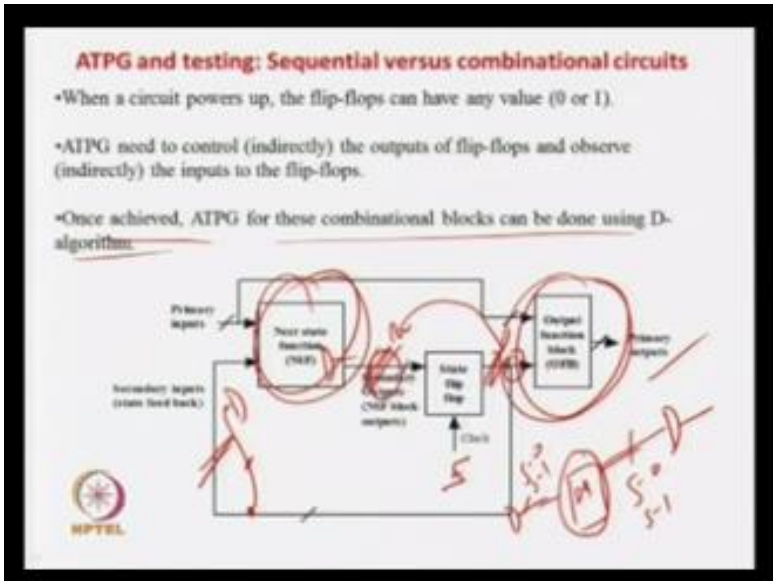
So that is what we are going to see in today's lecture how to do that that is we apply the algorithm what specially we have to control the virtual primary inputs and somehow we have to observe the virtual primary outputs this is what our job then our job will be done so ATPG for sequential circuits will be same as ATPG of the combinational circuits so now we will see is there some assumptions which we will be taking so for victual for ATPL of sequential circuits that is some properties of the circuits which up to taken as assumption to make out algorithms simple.

And the same way so this set of assumptions actually apply for 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 then our clock as single circuit and it is sequential so each flip flop is as a 1 bit memory element that is okay obvious and 1 common clock that is 1 clock is being given to all the flip flops.

So all the flips flops which in synchrony with the single clock so what it says that and next assumption is after the clock edge that you can say that assume that everything is working at the positive edge that is your secondary primary outputs that is you next state is transfer to the output

of the flip flop that is present state and it becomes in make new secondary outputs also the primary outputs are updated this activity occurs at the clock edge so it is called as synchronies operation.

(Refer Slide Time: 10:30)



Basically what they have said is that so there can be some input here so now you give of positive clock after the positive clock edge this output will come at this one and it will happen for all the state flip flop in one go for this for the positive edge of the clock so that is all change that is all changes in this all changes we are actually calling secondary output will become secondary inputs at the positive edge of a clock and this will happen for all the flips in the circuit so it is called synchronies 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 so we assume that the struck faults are at this nets and this circuits at the gates at this net and also at the gate end so I mean here we are actually have some you can say D flip flops are there so D, Q some D or JK or whatever some flip flops are there generally in case of circuits we used D flip flops so we assume that all the D flip flops are available in this S 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 these are stuck 0 fault here and this stuck at 1 fault over here similarly this is driving some other gate in the output function block okay also output functional block or also some gate in the next state function block so again you can have stuck at 0 fault and 1 fault but internal D flip flops there are some other case like feedback and all those things as we all know the internal structure of a D flip flop.

So it is assumed that the faults are not present in those internal gates so I mean if you combine this as second assumption say that stuck at faults are the output of the NSF block internal force of the flip flops are not model there output and input faults are model and faults of the inputs output signal of the combinational blocks.

Okay 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 flip flops will have some model because you are considering flip flops you are considering faults at the outputs of the gates of the NSF block as well as also the inputs of the gate at the output function block just like I have given example so some gate here will be driving this which will fault which we can think of having a fault similarly there will be some gate whose input will be taken from here so this is also considering as fault.

So fault at the input and output means of the flip flop will be considered what we have not consider in the flip flop faults of internal of the flip flop gates so mean this is to keep the simple because if you start considering faults of the flip then you can we can show that the circuit may not remain as sequential circuit at all so it 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 complex it will be very high.

So in the very first two lecture we have said that what is the beauty of stuck at fault model so we are considering fault model because it is simple to model you can do collapsing and all number of test patterns equal to very less as the same time they will give you an assurance that 99.9%

cases is what assuming stuck at fault you can be assumed 99% there is no defect if your circuits does not have a stuck fault same thing logic also actually holds for a 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.

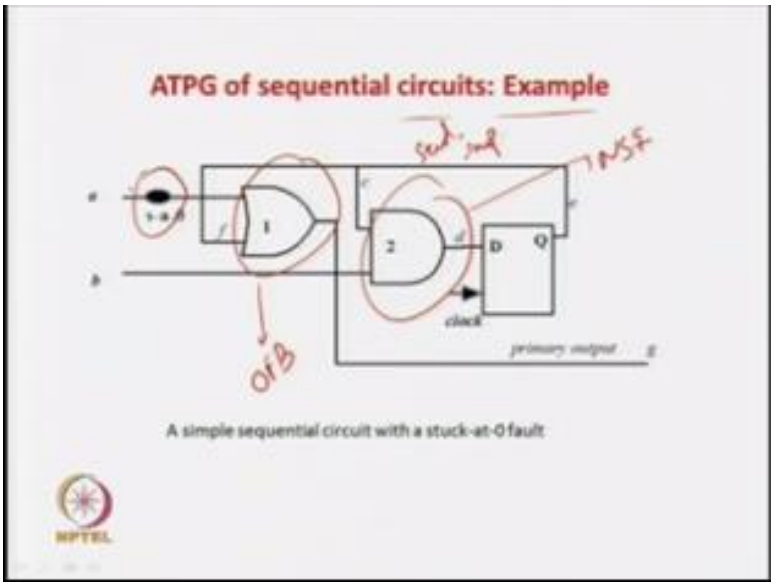
Because if you consider faults of the internal of the flip flop may be your accuracy improvement will be 99.9999 + 0.0001 + 0.0001 may be there but improve that the amount of complex is that will come into the algorithm will not make any help unnecessarily you will find that the circuit will have more number of states the circuits may become a combinational circuit from a sequential circuit to handle all this complexities the ATPG algorithms will so complex and the number of test patterns required may be complex that we may loss what we are going to again by adding this.

So and prepared the followed experimentally thus even if you forget the internal faults flip flop still your coverage assurance or your assurance that if you come out the stuck at faults there is no real defect with the assurance between this correlation between this stuck faults and the deep X then is very high so that is we can assume that faults are only at the NSF block and output and we can forget about internal faults of the flip flops okay similarly no flip flop faults are consider at the clock because if we find if you take faults at the clock things will be very complex.

Like sequential circuit may not be a combinational circuit some of the circuit may become as synchronies because for some of the parts of the circuit clock may reach or some part of the circuit clock may reach or some part if the circuit may not reach making the synchronies as the algorithm will be trouble complex okay and they will also will not gain much in assurance level so we been have experimentally found out.

That you can forget about faults in the clock you can also forget about flops in the internal case of flip flop still the correlation within the stuck at faults and whatever you call the VL defects are very high so we will consider only flops in the inputs and outputs of the NSF block as well as the internal gates of the NSF block and the output block and also mainly our as I told you.

(Refer Slide Time: 15:15)



Where as a circuits and mainly consider what you called the D flip flops they are the JK flip flop because once if the converted to the other do you all know but in the case of the digital circuits we mainly consider about your D flip flops so in our assumption is that all circuits will be taking hence forth where ever D flip flop okay so now again let us go back go to an example to tell what about the complex here is we are discussing so we may be lost so let us see the example is the complex circuit and is very simple example circuit so you can see that if you look at this is a simple sequential circuit.

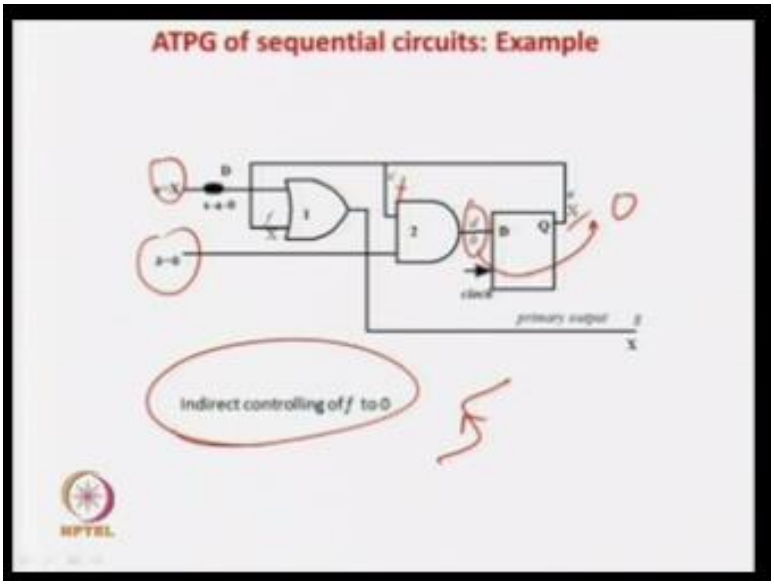
Now you can see that this gate actually depends on the one input of the flip flop and the primary input and it drives the primary output so it will become your output function block you can map it with the old diagram and find out so will this gate map belongs to it depends on the primary input B and also it depends on the output of the flip flop 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 through the flip flop as the input and we did in the next state sop one actually is the output function block and this is next state function block

if you just map it through this one we will find out so let us consider a flop as a I mean sorry fault stuck at 0 at 1 input of your output function block.

So let us see how you can do the testing okay so this is the case so only have to remember that this the output function block and 2 is the next state block okay so when the circuit starts up as you all know that initially every all the gates all the nets are having a value of X.

(Refer Slide Time: 16:46)



As per D algorithm so now these are stuck at 0 so you have to apply 1 so it will be D okay now one things is that so actually this is the D so that is only one part through which fault effect can be propagated that is actually called as singular D path so you have to F fault effect should be there so our target is that so gate 1 is in the D frontier because 1 inputs D and the other output is x and this is only 1 path through which the fault can be propagated this is called the singular D path.

So this should be at D after propagation of the fault effect again now it may one may be now which is D output is D so this F is unknown X so when the J fortalice algorithm of the J concept we know that F it as to be a 0 for the fault effect to be propagated so we require that one now you

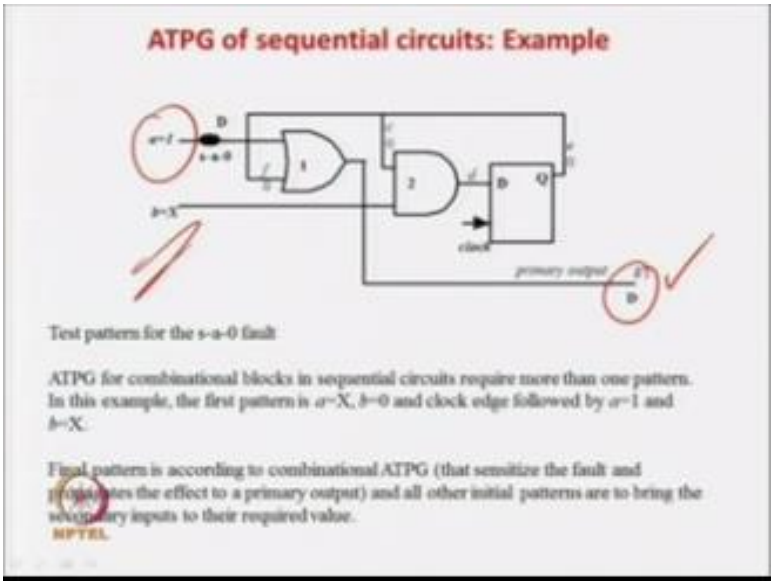
can see because the circuit because now you cannot directly control F that is the difference between a sequential circuit testing as a combinational circuit and in-between a combinational circuit then what would be have to done.

F could have been directly controlled and you could have applied 0 at there but now is not directly controllable 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 we require it to be a 0 but now it is that X so we are in a problem because we do not know how make a this one so we have to now do something so that we get s = 0 and once it is equal to 0 then you can apply this one to get this okay you get the fault tested.

So our main job is now to get a 0 at F so actually it has got indirectly control of F2 0 because you cannot directly do this one, so how can you do that, let us see that we put B = 0 so if B = 0 this is x okay because output, so if we apply B = 0 then irrespective of what is the x can be 0 or 1 do not know whatever it earlier told you the x means either 0 or 1 we do not know because the on the power of the circuit we can have any value at x so it even it is 0 or 1 whatever be the case but if we put z as 0 so d will be a 0.

Now you put B = 0, XY do not bother at the time B and you apply a positive edge of the clog so once you do this, this 0 will be coming over here right so you apply a B = 0 and apply you get as here I apply a clock pulse, so once that is done.

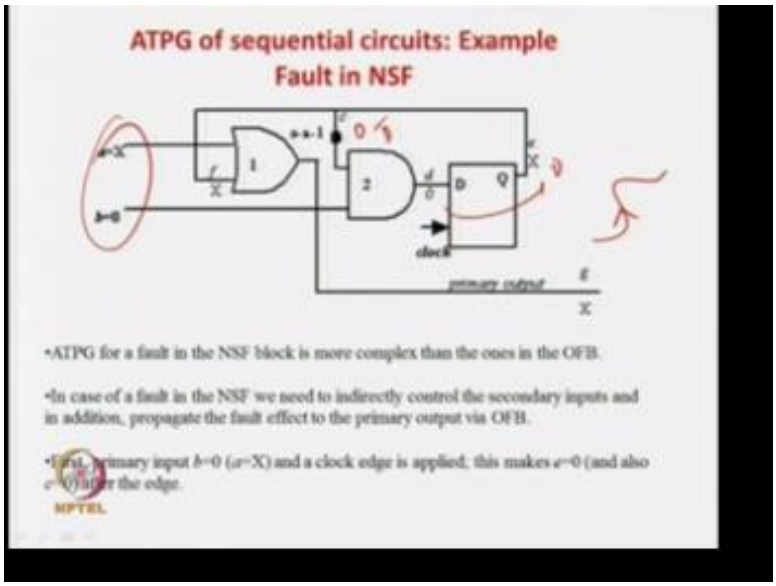
(Refer Slide Time: 18:49)



So what you are going to get you are going to get a 0 over here you are going to get a 0 over here, now you are indirectly if f = 0 now you can out A = 1 you can totally forget for the time B and then we will get this value of the output and circuit tested, so now in case of sequence how many such circuit what we have seen that only one pattern can test your fault correct so you find out by the fore say propagate or justify your whatever by an random by not but one test pattern can detect a directly a because there is a fault in the circuit or not.

But in this case we requite two patterns as you can see first you have to apply AX, B0 AX we do not know B = 0 so which will return my make that AF = 0 indirectly, so one F = 0 now you apply A = 1 and B = X so that your circuit be tested, so this is the basic fundamental difference between sequential circuit testing and combinational circuit testing so in sequential circuit testing what we have done.

(Refer Slide Time: 19:42)



See in sequential circuit testing we require that first the actual primary output or secondary inputs whatever sorry if the actual primary inputs or secondary inputs whatever you can they have to be directly controlled because the output of the flip-flop that for every directly indirectly controlled to some value that is required for ATPG. So you may in this case there is only Flip-Flop so you require only one step so slowly we will say how many steps are required to make this major secondary inputs control level.

So you require one pattern one clock then one pattern one clock some amount of where patterns are the clock so in this case this is only one pattern required and the clock has to be apply so it will virtually or indirectly control your virtual inputs, now once the virtual inputs are controlled then you can directly control your primary output inputs and then you can see what is the ATPG is done, so in this case we require two patterns to do it, okay.

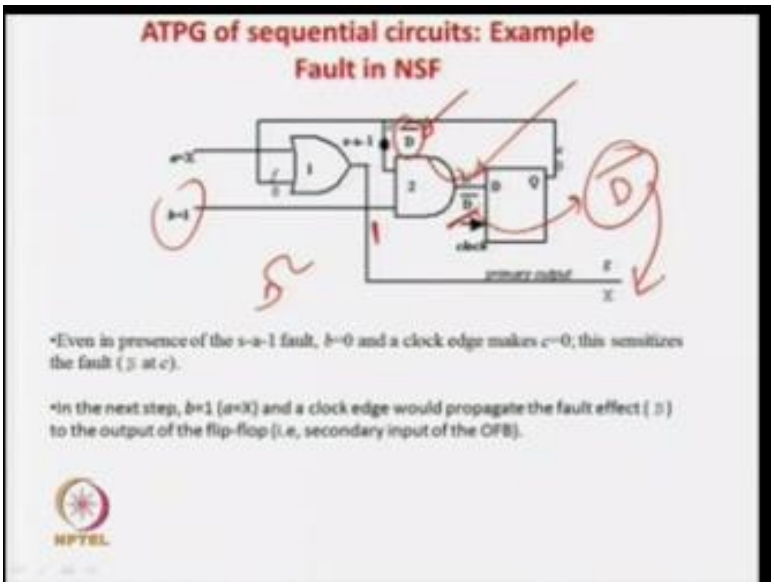
Now let us see that in this case if we consider a fault in this ouch next in function block this was NSF so in case of sorry this is the output function block sorry this was the output function block so we consider a fault here so somehow we require to control this net to 0 so we have done this indirectly but the output was directly observable so we could have seen done this in two pattern

but now we will take another fault in this block which is actually your next state function block, so here life has been more difficult.

Because you have to directly control this sorry indirectly control this secondary input as well as the output of the next function block is not directly observable so you have to somehow make it indirectly observable, so faults at testing fault set the nested function block is somewhat more complex then you are what you call this output function block because in case of output function block the primary output is directly observable but in this case it is not so start case one fault.

So what do you require for start gate one fault so obviously you require a 0 over here correct so now this requires a 0 means it is the prime over here so but to get the D prime over here so you have to get a 0 over here but as you know this is the output of the flip-flop so we have to do this indirectly so ho9w you can do that indirectly so you do not forget about XA B you put as 0 so irrespective everything you get a 0 over here and then you get a clock pulse. So once you do this you will get this as a 0 over.

(Refer Slide Time: 21:48)



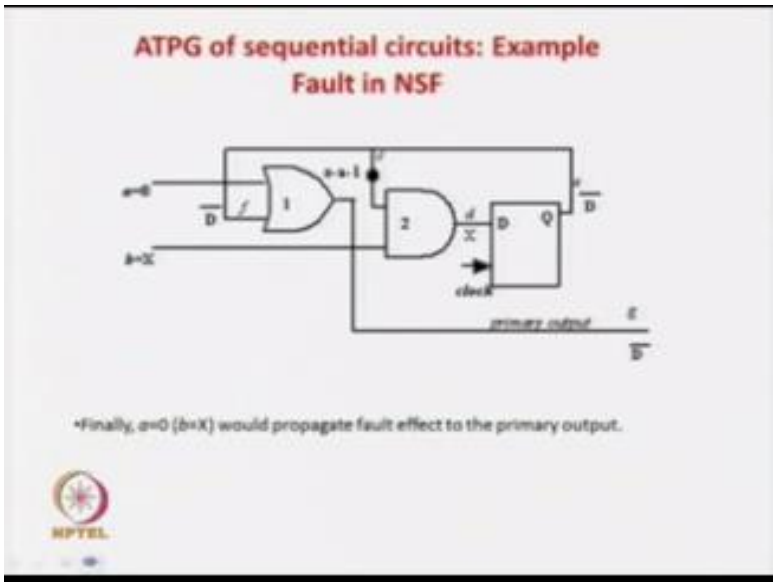
So this is what being done so if you make a B = 0 apply a clock pulse then you get X = 0 and your fault get incident this what you do in first step this indirect control correct so you get this one, now next thing what you have done so you get a D prime over here so now what you know so you fault this sensitize so previous case D = 0 was a previous case so now what you have to do so you know that that if this is the fault.

This is the only AND gate so therefore this becomes D front here and unique D frontier or this is only a unique X bar in this case so you have to somehow propagate the value of D prime over here so for that you require to have a one over here so in the next stage what you do, you make forget about X make B = 1 and then so it will be 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 here so D prime will appear at the out if now 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 your nested function block because there is going to be input or a D flip flop so that is more that is why more complex faults should be tested in the what you called the output function blocks, sorry faults which are in the nested function block is more difficult to be tested.

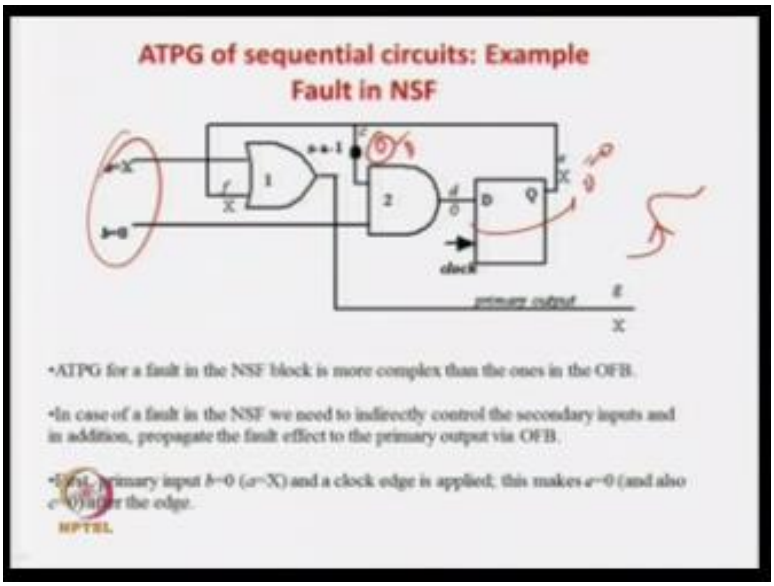
Because not only have to indirectly control some lines called also we indirectly observe some lines which is not in the case of gate 1 which is output function block so somehow we have brought this here but now you cannot observe it directly so you have to somehow make this D prime happy over here so you have to go to the third step.

(Refer Slide Time: 23:14)



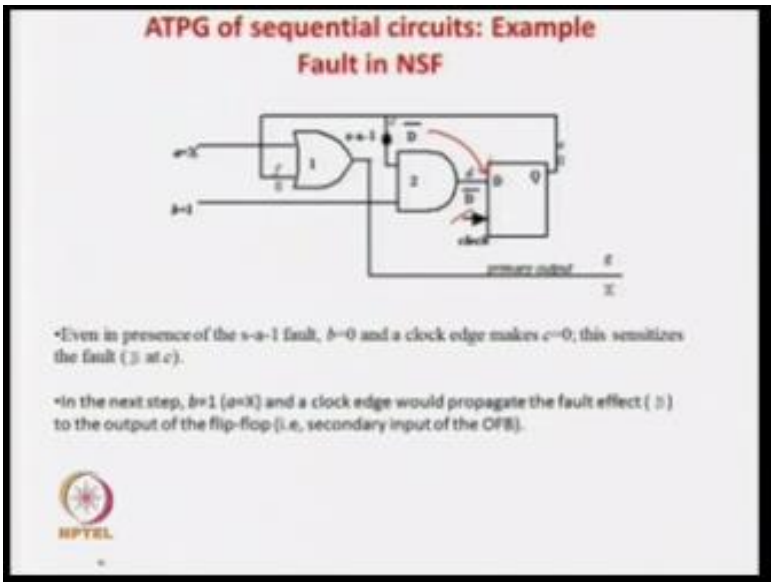
So last we are could have done everything in a single two steps in the first D control F to some value and then we fault effect was propagated here but now in this case three steps are required first you sensitize this to 0 then you propagated the default value two here and then yeah this is not directly observable somehow you have to bring this value two here so nest state so in this 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.

(Refer Slide Time: 23:44)



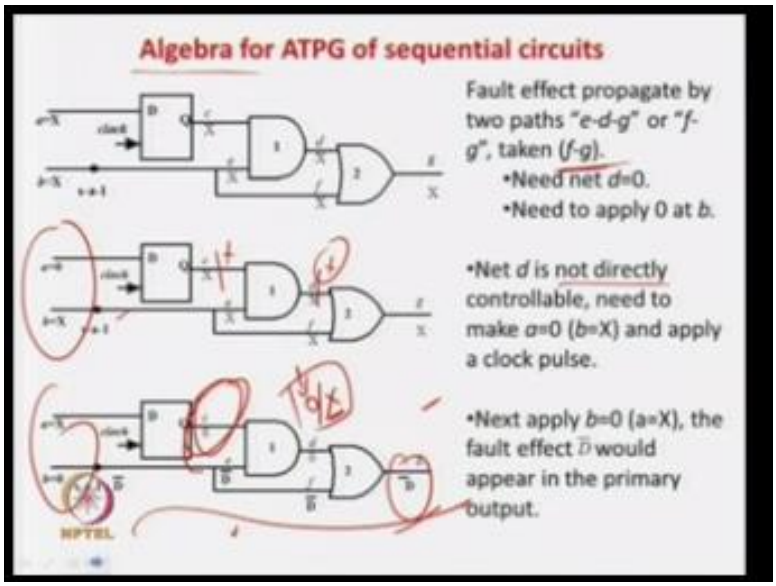
So if you apply a clock pulse over here the third step you apply a clock pulse over here so your D prime will come here so now your D prime is here okay so that is the third phase you get a D prime over here but now again this has to be observed by this one so you have to make A = 0 so we are indirectly applying three clock pulse did you see first A = X D = 0 SO you are indirectly you are controlling this value this one.

(Refer Slide Time: 24:08)



Next you are applying A X = 1, X = 1, B = 0 so your fault effect is propagating to output but the output is you cannot observe so third step what we do again we put a clock edge so that it comes out indirectly but a your primary output.

(Refer Slide Time: 24:22)



So that is the more difficult if you are considering faults at the next state function block competitive output function block but for all the cases I am like combinational circuit you have to indirectly control virtual and secondary inputs and indirectly observe the secondary output that is what is the basic conceptual difference so in case of sequential circuit more than one test patterns may be required to tested fault.

So indirectly what concept so the broadly speaking the complexity of sequential circuit testing is very is more difficult compared to a combinational circuit testing so in this case you have to observe that we are not considering the cases where there can be inconsistency and where there can be black prime so you can understand that how the complexity be blow up if you start considering this so we may use this D algorithm to find out that this can be a good pass or propagation and all.

So now instead of one pattern we have generated 10 or 20 patterns which are required to indirectly control this virtual input and indirectly observe the virtual outputs but now after that you could have found out that for some steps there is a inconsistency and we have to do a lot of

back trekking so in combinational circuit there was only one pattern for us, but now there are 10 patterns or relevant patterns for x number of patterns for our part.

So again if we remember that there is a back trek in the final phase so how much amount of competition work has been lost so that is why sequential circuit testing is virally more complex than a combination 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 what kind of as a breadth let me browse roads for five values algebra we have seen for combination circuits but now we are going to study here.

That 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 ADGP for sequential circuit what we or whether five valued logic could suffice or not so let us just see that we will try to use a five valued logic and see what is the gain and what is the loss and then we will try to see if we can go for a higher order logic okay.

So let us see that this the circuit so this is the socket one fault so obviously socket one fault over here this is the socket one fault over here so a socket one fault here means you have to apply a 0 over here so now this two parts so one part is it forth can be propagated here and other path can be here so there are two x parts here so for the time being now let us see that if this path is smaller so by any eristic if you apply over D algorithm.

So we can take that E and F may be a very good path for propagate the path because the other path like ED and G is the longer path, correct. So socket 0, 0 means it will be a D prime over here so you can propagate the value of D prime over here D prime over here but immediately we know that D has to be a 0 over here so how do you get a 0 over here.

So you know 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 what we can do so net D actually already told that net D is not directly controllable okay so we have to make this net D = 0 because actually this net depends on a

output of a flip flop so this gate is not directly connected to the output of the flip flop but it is indirectly dependent on net C which is the output of the flip flop.

So indirectly you have to control this B to 0 so if you just look at it we can even easily do that you make A = 0 so if we just make A = 0 now what you do you get a give a clock pulse so just if you give a clock pulse AA = 0 then this one will come here and this is X so X and a 0 will be a 0 correct, that is what is out idea so that is this is how we are doing by a D algorithm so just illustrate we quickly see once more what we have done.

So this is the faults socket 1 so we apply a 0 over here we need to apply over 0 and this is the path propagation so we may get a D over D prime over here so this is the path and to get this one we require a D over here so D has to be a 0 over here so to get the fault this one but now D is not directly connected to the output of flip flop but it is connected to C which is the output of a D flip flop and this can be directly controllable.

See you may note that it is nothing but a primary secondary input sorry secondary output because it is the output of the flip flop not directly controllable so somehow you have to make this has to be 0 so if you want to make this one to be 0 because you are not want S to be 0 so somehow you have to make this to be 0 because we know that 0X is a singular tower so to make X = 0 what we can do.

We have to make A = 0 and give a clock pulse so immediately D will be equal to 0 and your D prime will be propagated to the output so what are the two test that has to test d for one test pattern is A is 0 B x and the X pattern is Ax and B0 so this will test your fault so that two patterns equal to test the now let us just do a freak of Boolean algebra and see can you 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 being keep it as a X okay, now let us see what we can do. So let us see it is socket 1 and we apply a 0 so normal case 0 fault case 1 we can write like this correct so now in this case you can

see that these also D prime so we just write all this, this we require correct and what we require something over here.

And here we write this also 0 < 1 now just have a very quick look at this circuit so just sorry yeah just look this point so this point is very important this net E if you look very carefully so what is the ideas so you just net D is 0 and a 1 now let us look sorry net E is directly you can net decide about net E you can directly get from this net because it is the FAN out for this state, now let us look at net D very carefully.

So net D is infect X by DL algorithm or Rows 5 values 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 now let but 0 and X is not a part of your algebra so let us see in this case so we get D = 01 and ZX but now look at D very carefully so now what happens say let us forget about this being 0 let us assume that it is x we do not know the output of this one.

So if fault is not there what is fault is not there so what is the input over here it is the 0 okay and what is the value of this it is x because we are not considering we do not know the case normal or fault because we do not know the output of this one so what is the output of D at fault so if the fault is not their normal so the output will be a 0 because if the fault is not socket fault is not there so you apply a 0 over here.

So you will get a 0 over here so default your D will remain in 0 okay and other things are fine now if these are fault then what will be the case, if there is a fault then the socket 1 so you are going to get a 1 over here and this is the x, because we are not applying anything for the timing just assume. So it will be a x so it is 0 and x at d if you are not controlling this full timing just assume correct.

So now the output of E is D D' that is 0/1 and D you can write as 0.x if the fault is not there you apply a 0 so the output will be 0, but if the fault stuck at 1 fault is there you do not know what is the output so it can be x. Now you just and if we know it is 01 so if you do a or this one, so the output of g is what, so it is 0 or x is 0 1 or x is the 0 which is nothing but a D'. So it is very

surprising case has happened here that even without controlling this guy to 0 or this guy to 0 indirectly which is required in the D algorithm or Rose 5 value algebra.

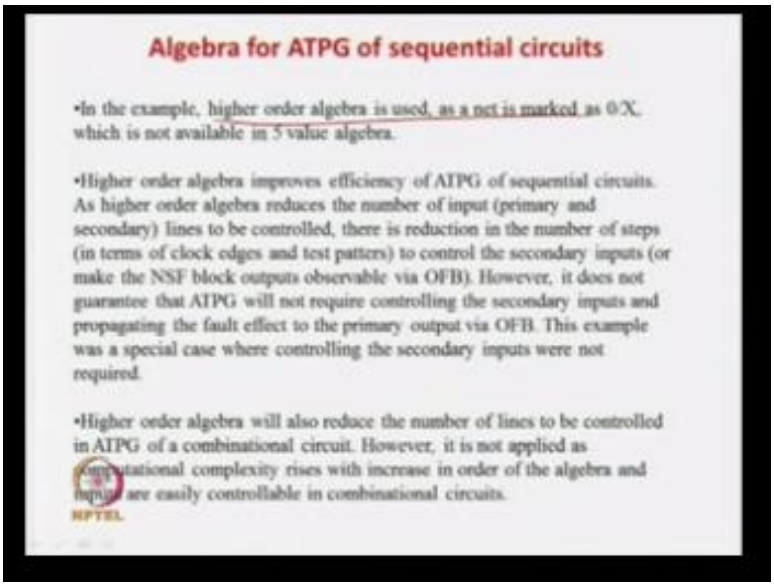
So if you have a element in this which we are calling it as 0/x because if you remember Rose algebra there is no concept of 0/x where x/x 01 10, 00, 01 so the 00 normal 0 fault 0, 11 normal 1 fault 1, xx we do not know, 01 that is D', 10 that is D. But you do not have any concept of 0/x, but if you have a concept of 0/x then only one pattern can detect this fault, this is a very special case in which case I mean is not required to control and all, but most of the times you may have to require this.

But for some of these circuits you can find out that if you have the concept of 0 that is point D is 0/x then only one pattern a=x and b=0 will test your circuit. But if you are directly following D algebra that is we do not have any concept that is there is only one concept that in the Rose 5 value algebra there is only one concept that it is X, so x it is xx. So it is x that means this is xx this is also xx something like this.

So we cannot write 0/x for 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 another pattern to propagate the fault value. But if you write some, if you have, there is a concept of writing 0/x which is basically the case because if this circuit fault is there then only the output is not known usually dependent on the virtual primary output that is D.

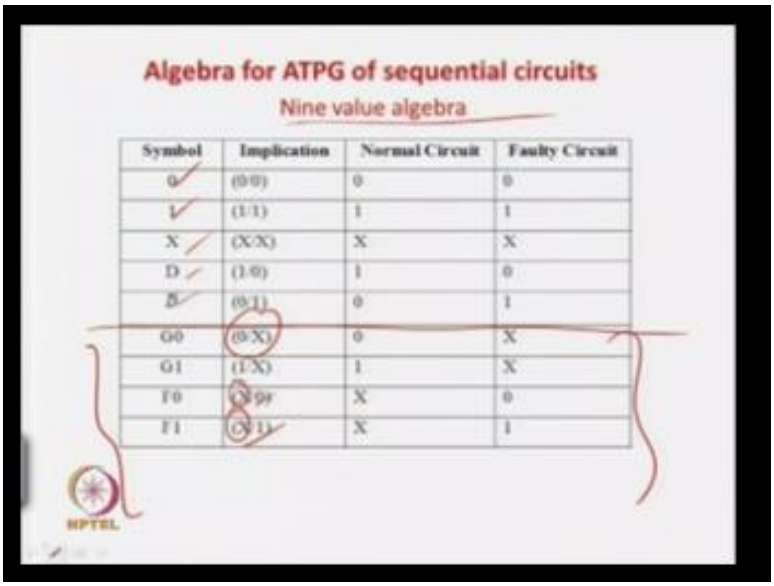
But if the fault is there sorry, if the fault is not there then you apply a 0 so at all does not dependent on the output of the flip-flop that is not at all dependent on the primary virtual primary output, secondary output, then you get the value 0 directly. And by doing this thing you can get a D' over here without controlling this, so this is a very special case of an example where you can test of circuit the sequential circuit without requirement of controlling the secondary, primary output.

(Refer Slide Time: 34:01)



Because of the fact but now for that you require some special symbols which we are for the theory which you are discussing that higher order algebra we require a symbols like 0x and all these things so higher order algebra and many advantage will come back.

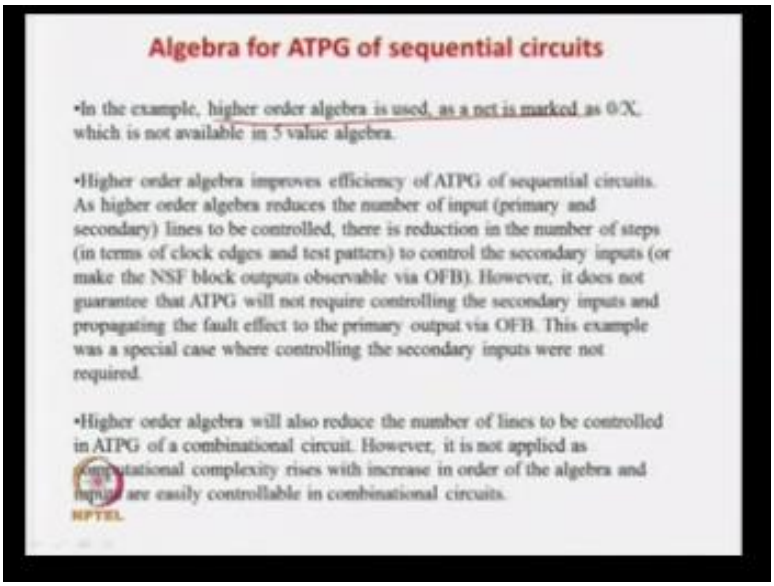
(Refer Slide Time: 34:14)



So by adding one extra symbol that is actually 0 < x we got that in case of symbols of ATPG only one pattern or at least one level of competition was less required, so nine valued algebra are equal use this sequential circuit that is 01,1 D and D prime these are all from rows five value algebra, but now we are added some more 0x that is normal case 0 fault case unknown G1 normal case 1 fault case are known F 0 normal case x fault case 0.

Normal F 1 is normal case f1 fault case 1 so with this addition of this things you can see that for many times you will be requiring less amount of competition to solve your problem.

(Refer Slide Time: 34:50)



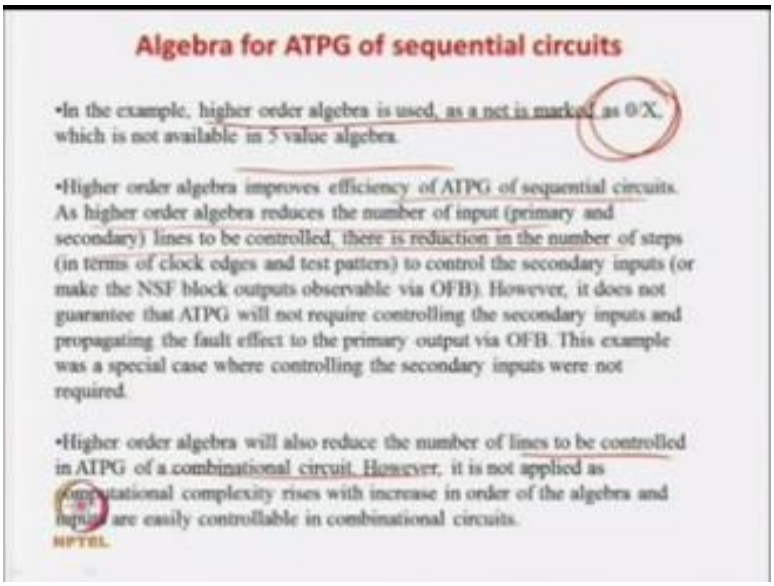
So that is what is the required in the last example higher order algebra this one so which is required to reduce the number of competition or only one test pattern can do it and this is not available in a five valued algebra, so that is why we sometimes we require to go for higher order algebra that is 9 value algebra so higher order algebra improves the efficiency of sequential circuit testing because you require less number of states for the competition.

For only one thing you have to observe that always it will not be possible to even if you are using a 9 value algebra that directly you will do not require to indirectly any virtual primary outputs or virtual primary inputs kind of a thing so these are a special case where it happen but only you can assume that or it has been found experimentally that if you are using higher order algebra then the number of competition generally be reduces because you have some more flexibility less x01, x0x, 1x something like this.

So more the number of excess in this circuit we have seen that more number of flexibility there, so that is why it will improve it if your efficiency because in case of combinational circuit testing one pattern was used to deduct this test and but here you will give more than one pattern so lot of

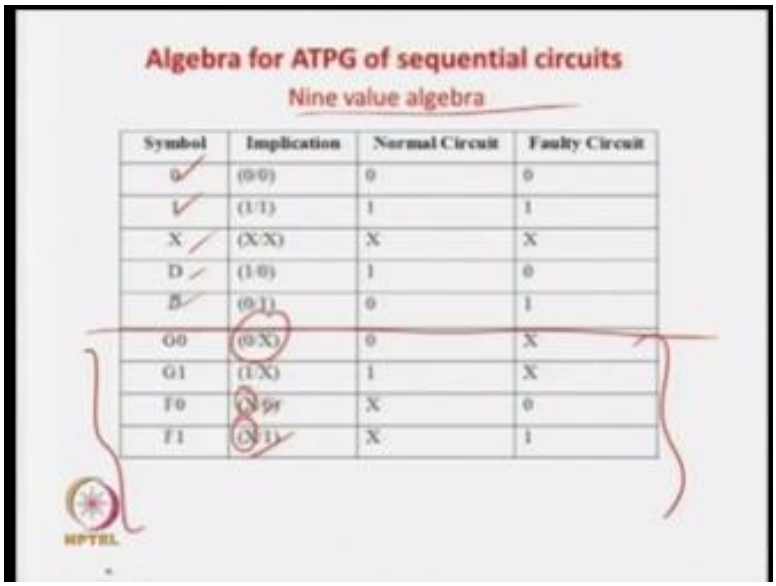
competitions are required sop you want to have more flexibilities in this circuit so you are using some other higher order algebra so you can also question me that higher order algebra.

(Refer Slide Time: 36:04)



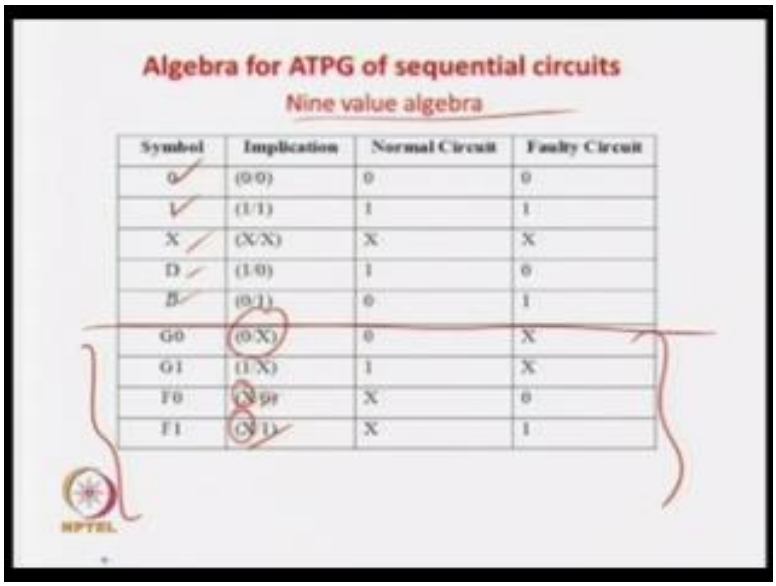
Would have also helped in circuit the answer is two because in the last lecture whether example you have shown that if you have more number of x is in your circuit then it is 2 that there will be more number of flexibility and the ATPG in a succeed better so if you have 9 values of combination circuit you are correct you will get better solution but in the higher order algebra.

(Refer Slide Time: 36:22)



Is also.

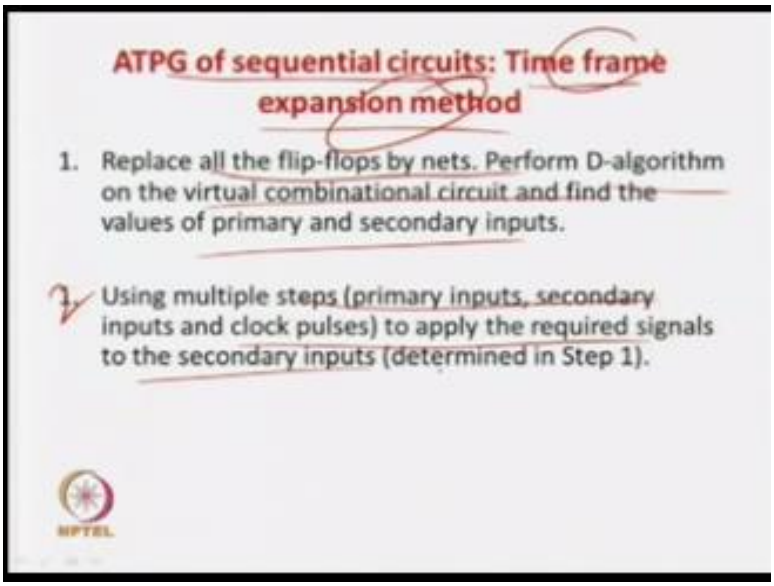
(Refer Slide Time: 36:24)



Combination in difficult and is have been found out that the amount of gain you are going to get where using a 9^{th} value algebra is not that much in a combinational circuit by the amount of in consistency that you may reduced that is because in sequences circuit and more number of days patterns in more number of patters are equal to find out a test pattern so we want more flexibility so you use this extra four symbols x0x x0 and all this thing but same thing if you bring out in a combinational circuit and make a 9 value as algebra for combinational circuits.

So some advantages you will get that is definitely because more number of x is will be there in the circuit but the amount of complexity will incurve there we will not help you to gain because in combinational circuit 1 pattern is an activate data circuit so because mean in consistency there are big more less computing sequential circuits so that is what 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 see ATPG for sequential circuits is a time frame expansion method.

(Refer Slide Time: 37:17)

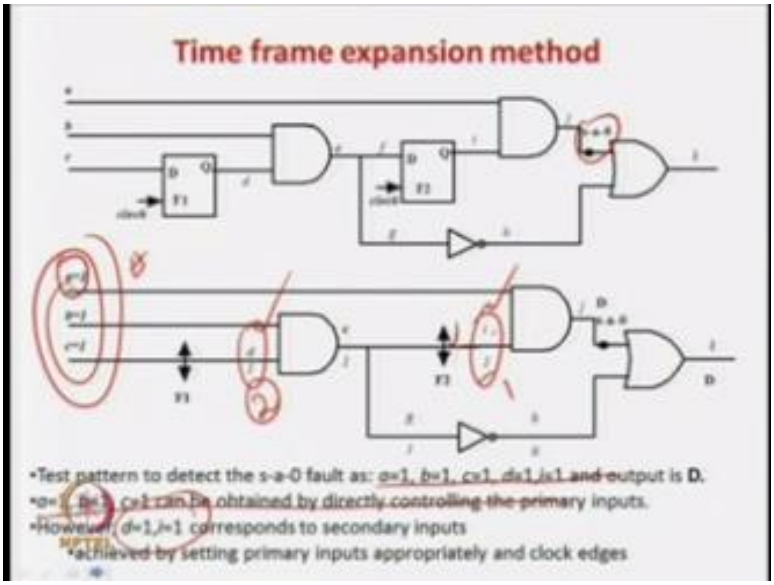


That is like the D-algorithm we have seen so in D-algorithm we have seen so in D-algorithm what was the idea that is our formal algorithm to find out the test pattern first is some of these circuit so in this case is our time frame expansion method is another formal method for what you can say I mean 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 their first step replace all the flip-flops by nets so first there will be sequential circuits will be there, there will be something for or so will be there.

So as purely sequential of circuit is a singular clock is assumption so then you remove all the flip-flops in 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 or the inputs of the flip-flop will become your virtual primary outputs see what we are done were removing all the flip-flops so if you remove all the flip is what is happen the outputs of the flip-flop.

Will become virtual primary inputs now you can also virtual time as inputs the outputs of your sorry and the inputs of your flip-flop will become the virtual primary output some changes there in which will see in the example that is what is going to happen so virtually we are going to get a combinational circuit now you use D-algorithm on that and you have to use so these step two you are going to multiple steps will be requiring to control the secondary inputs.

(Refer Slide Time: 38:40)



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

These also a secondary input because you cannot control this direct secondary input yeah, you cannot observe this directly correct now this is one input of the flip-flop and these are special

case so these are primary input as well as the input to the flip-flop so you can say that this also a both are primary input as we all as the virtual input and so but in this case you can see that this is the output of this gate is going to the input of this gate so these actually a virtual input these are virtual primary input or secondary input use you cannot control this directly that is what is our idea.

So these virtual primary inputs this is virtually primary output you can say or secondary output similarly this is the output of the flip-flop but you cannot see that directly so you can consider that this violently virtual primary output kind of a thong okay, so that is what is our idea an also you can say that these again going to the input of an that gate because that e also have these output as well as it is added mean correctly coming out then you put have all these virtual primary output because.

Or a primary output something like that but you can see there also it is going as a input another gate so that can also be cannot be directly controlled or observed so you can say that this is our virtual primary input or a virtual primary output that is secondary input output so these not directly observed be not directly come to level so somehow we have to do this indirectly okay.

So you can easily map this gates to which is the NSA block and was the output function though that can be easily done but any way there is not much of a concern right now so these are stacker fault over here so what is the first step, first step is they move the flip-flop so you have remove this flip-flop here we have removed the flip-flop here another short is so this is what is your idea what these 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 you want a 0 = 1 over here say we apply a 11or 1 here.

You cannot get the 1 here directly because it is cut by a flip-flop that assumption you have to know that similarly you cannot get at 0 here directly or 1 here may be not possible be very we have shorted this so that is has to be in but it is now a so but for the D-algorithm for next step is that this is now but with this small assumption in mind that is not a very expressed assumption

here so what we have seen that is not a very expressed assumption but we just go in our mind that, that directly we cannot be controlled or observe.

Now these the virtual combination circuit that all I was saying that is the virtual combinational circuit now in the virtual combination circuit you have right combinational D-algorithm that is it so now you just see what is the combinational D-algorithm of here so we know that were there 0 so you have to apply a 1 here so it is a D over here so they propagating so it will be at the now the other inputs should as 0 or get this is a 1 here you require and next is that you want to apply 1 over here so apply 1 over here means a is has to be a 1 and this input also has to be 1 correct, next.

What is the case so this has to be a 1 so the output has to be a 1 so the output has to be 1 so 1 means the AND gate this has to be so you their D-algorithm so you put have directly said that A1 B1 and C1 if you give and the output is the so your circuit would have been directly tested in the combinational circuit but there is a big but over here so you require a 1 over here you require a 1 over here but as I already told you that internally we have remember these dib but that this directly not controllable these also not directly control over that is a big body you have to remember.

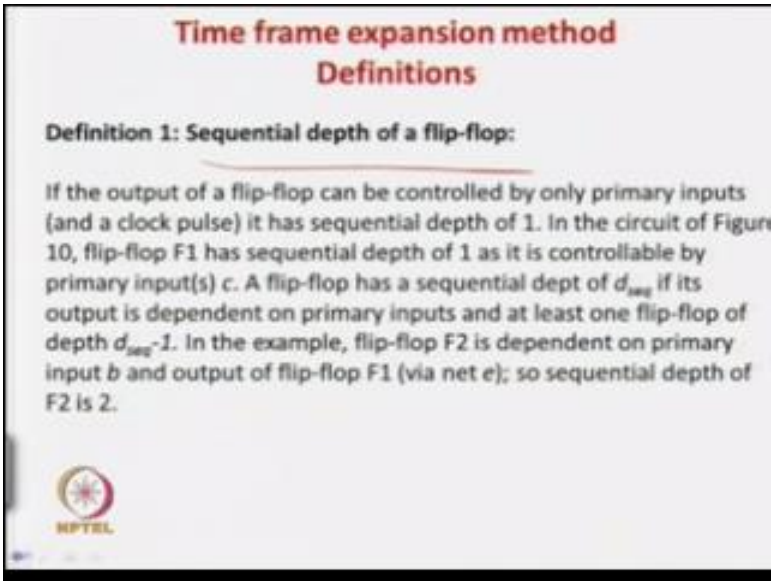
So how you will get a 1 over here you have to do that indirectly similarly how to get a 1 over here that also you have to do indirectly because these two are output of the reflex which are going to your input or some gates so there your virtual primary input their not your direct primary there your virtual primary inputs and you have control there virtual that is the idea similarly if some continuous to observe these then that also not be possible because there either internals or you can also even if.

Their fitting it to some outputs of the flip-flops or even if they are be going as a input the output function they cannot be done directly because of this virtual or primary output and virtual primary input kind of the be here so now let us forget about this so I mean let us see what is our target so our target is that you see have to apply eventually to take this part before that we

require our 1over 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.

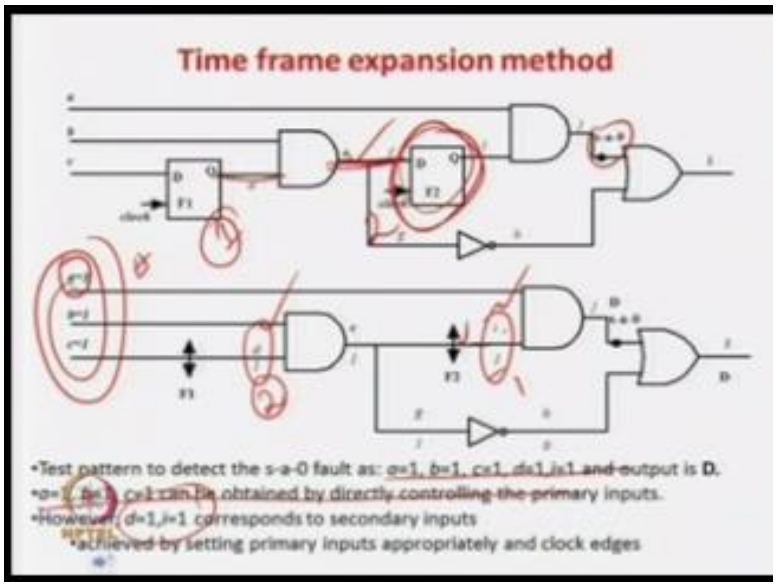
These our requirement number 2 okay and one this requirement is done so you get a 1 over sorry this is not your immediate requirement so this is your requirement number 1 so this is requirement number 2 so if you can get 1 over here if you can get a 1 over say 1 over this point then your circuit testing is done by apply this so some how you should these two things indirectly so that is it say so this one is your input pattern A1 can be apply directly but d=1 and i=1 d=1 I have to go by second by.

(Refer Slide Time: 43:54)



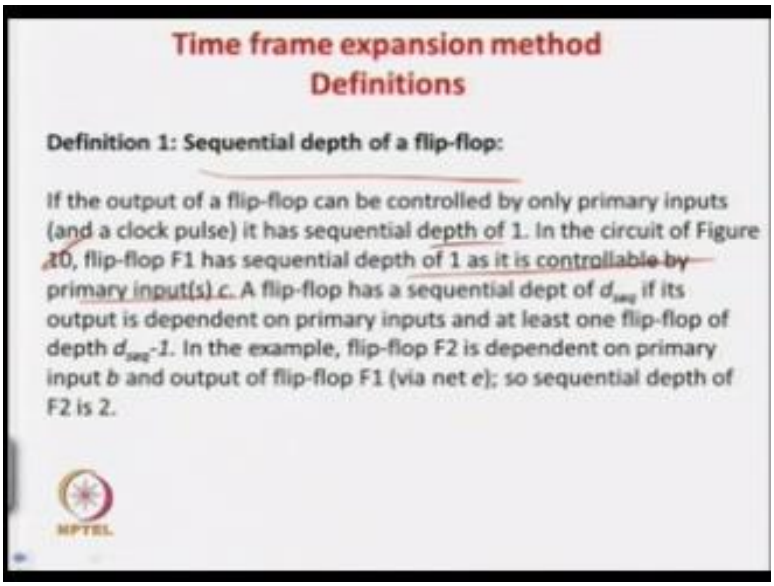
Some other mechanism so that is what is very much require and we will see how to do that but time frame expansion method okay now so just to very simply definition so 1 is call the sequential definition of flip-flop so the output of a flip-flop can be controlled by only primary inputs and clock pulse then the sequential depth of 1 that is.

(Refer Slide Time: 44:13)



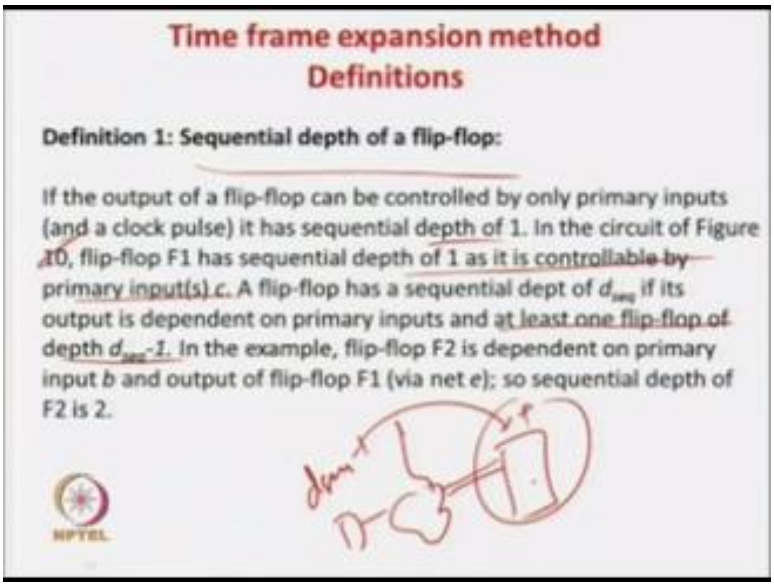
ust look at this reflex so it can be control directly by the primary input so if they depth is 1 now if you look at this flip-flop so it depends on 1 it depends on this input this input line e which is again depend on this flip-flop which is having a that is all if this flip-flop is depended on sorry so this flip-flop actually is depended on this input correct and this input is depended the output and the flip-flop so and this is having a sequential depth of 1 because there is directly control by the primary input so depth of this 1 will be equal to 2 so what is a definition just.

(Refer Slide Time: 44:48)



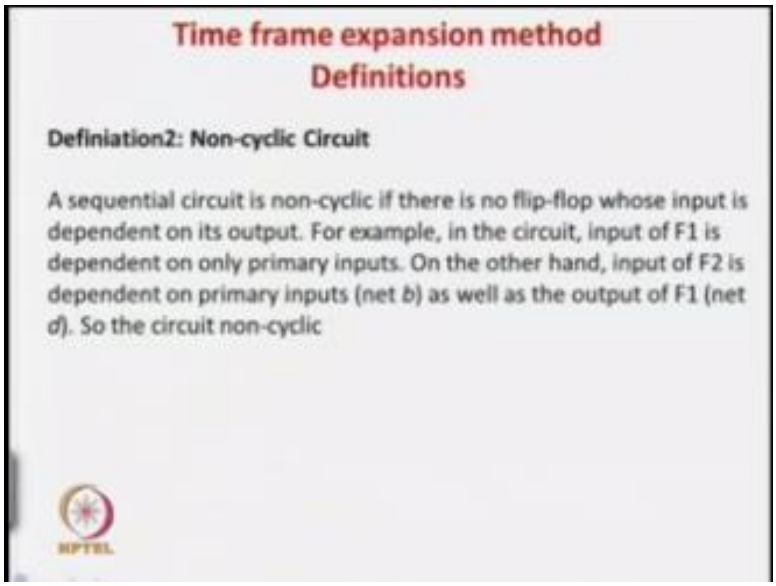
If you look at it so the if the output of a sequential flip-flop is direct this one is 1 then this circuit of figure last figure if you take then either sequential depth of 1 that is directly control by C that is 1.

(Refer Slide Time: 45:03)



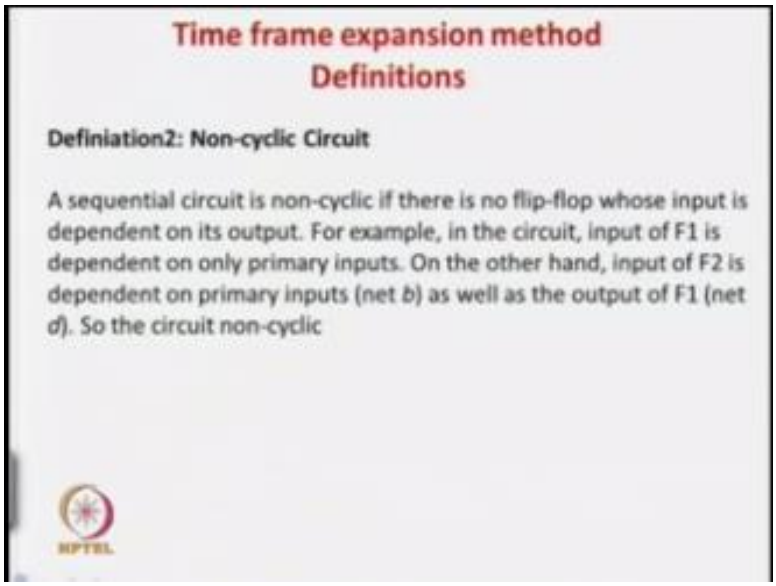
So directly control by C so depth is 1 now this reflect it is saying that a flip-flop sequential depth d sequence if the output is depended on primary input and at least 1 flip-flop of depth n – 1 so that is you have a flip-flop here so this input is depended on a flip-flop depth is our d sequence – 1 and some other primary inputs it can be there okay so that is in directly these are flip-flop whose input is depended on another flip-flop whose sequential depth is $d_{seq} - 1$ then you have to add a 1 for this flip-flop directly d sequence so that is what so.

(Refer Slide Time: 45:33)



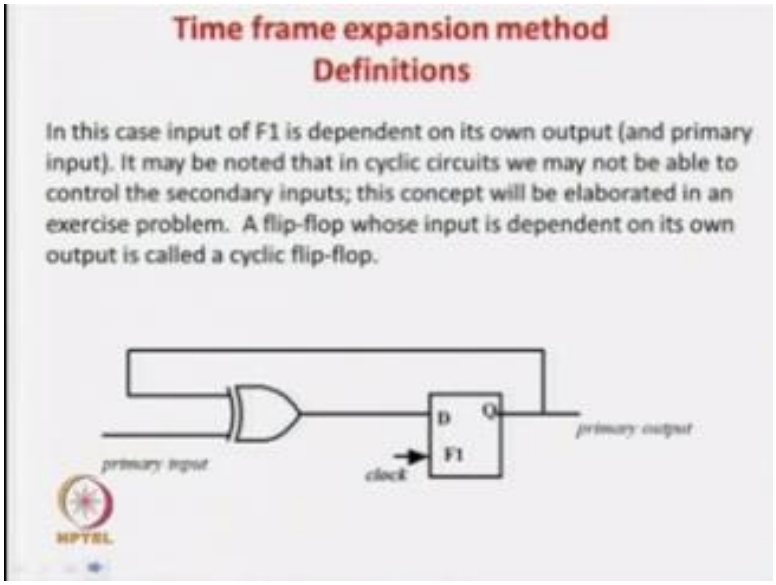
First flip-flop you consider so depended directly primary inputs that will be value 1 then we find out the next of our flip-flop which is depended on prior inputs and all those flip-flops whose sequential depth is 1 so that either value will be 2 and you can keep on doing it so you in our example this.

(Refer Slide Time: 45:52)



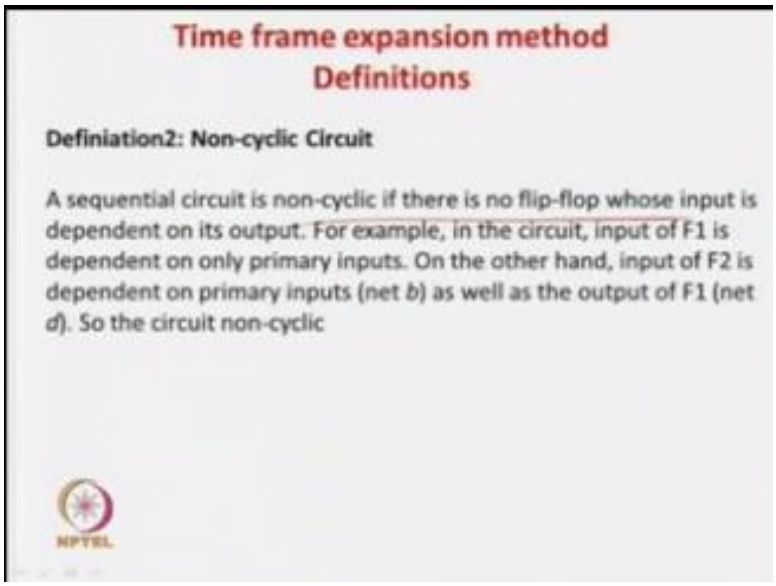
Is having a depth 1 and this is having depth 2 so that is a definition we have got correct so now these are definition was cyclic circuit.

(Refer Slide Time: 45:56)



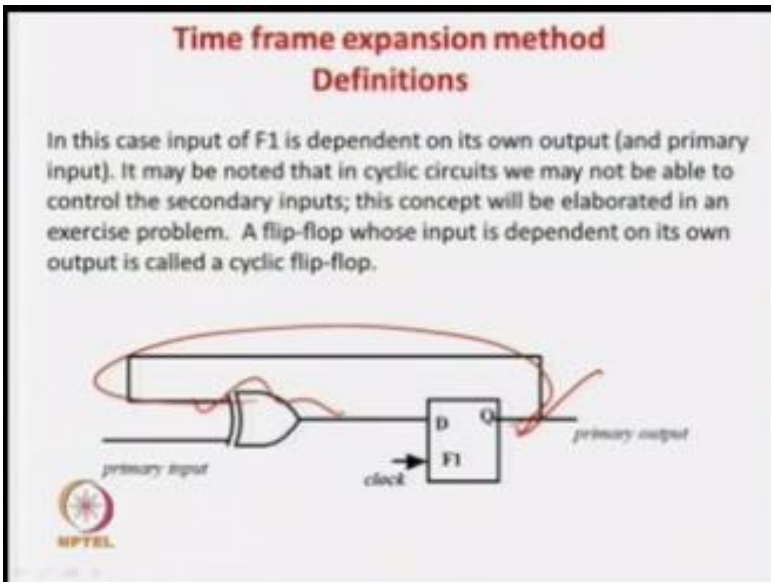
So what is a cyclic so this definition says that?

(Refer Slide Time: 46:00)



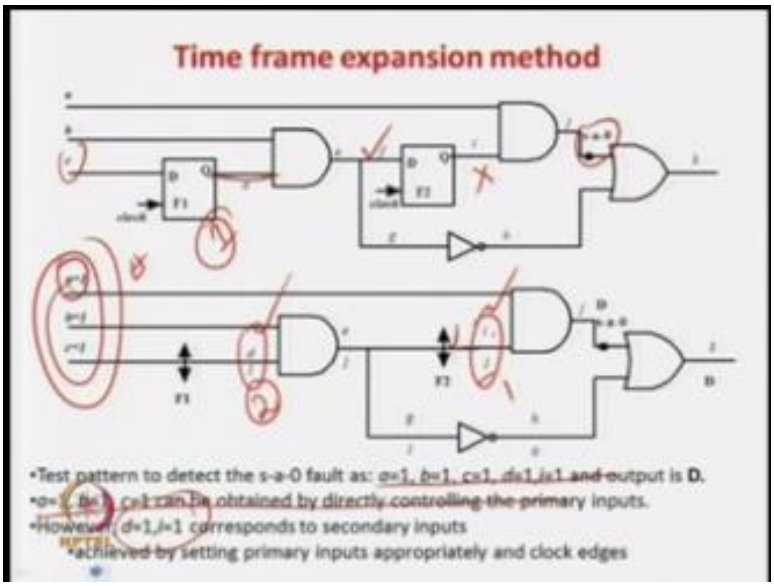
A sequential circuit is non cyclic if there is no flip-flop whose primary input is differ was input is depended in out for example if these.

(Refer Slide Time: 46:07)



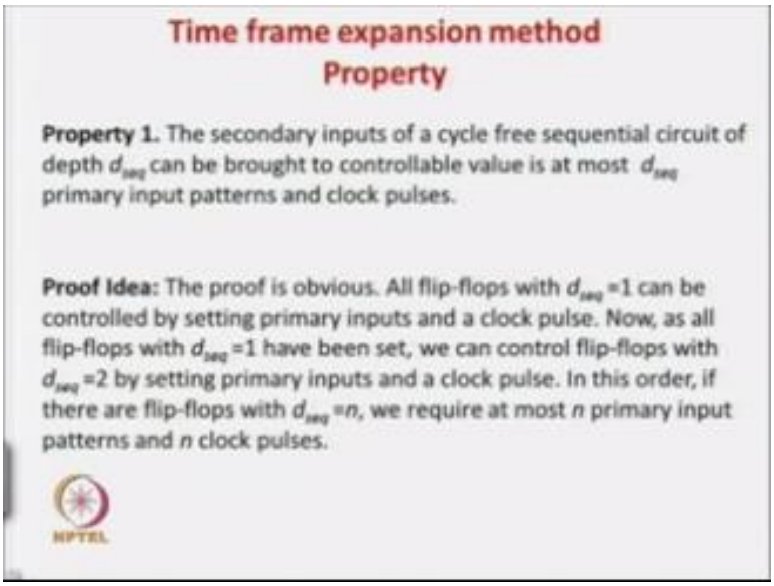
You consider this circuit this is a sequential circuit because his output is depended on its this one this is look at feedback so his output is depended on its owing so that is actually a cyclic definition but if you look at this flip-flop.

(Refer Slide Time: 46:22)



There is not a cyclic this is not a cyclic this one because this input is not depended on this so that is a definition of sequential of a cyclic circuit.

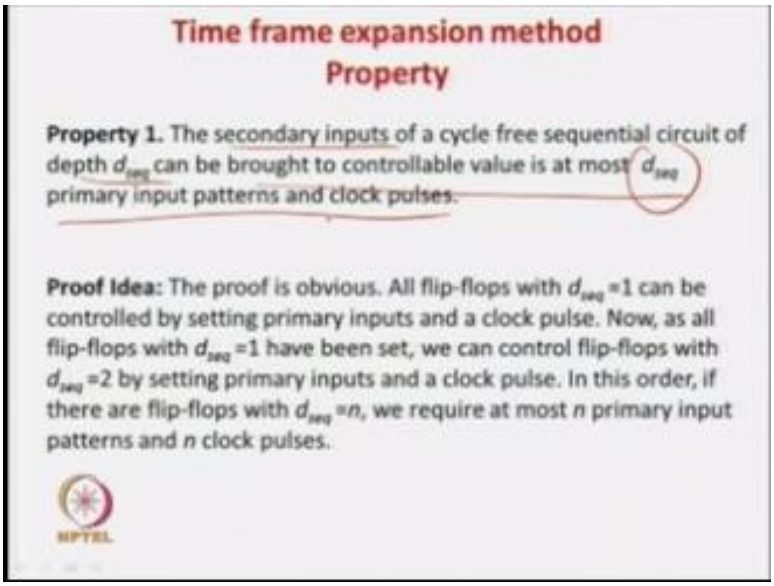
(Refer Slide Time: 46:43)



Okay so non cyclic circuit is the output of the flip-flop input of the flip-flop so not depended on is put this one so this circuit is not definition so this is a cyclic circuit and the other example is a non cyclic circuit so now we are going to very quickly see a property there is time frame expansions the property his property this secondary output is a in this circuit is cycle free okay if the cycle is having a or what you called if the circuit is having a cycle then we will see in the A and that no testing can be done or its very difficult to do testing for that kind of a circuit because it will be there will be continent oscillation.

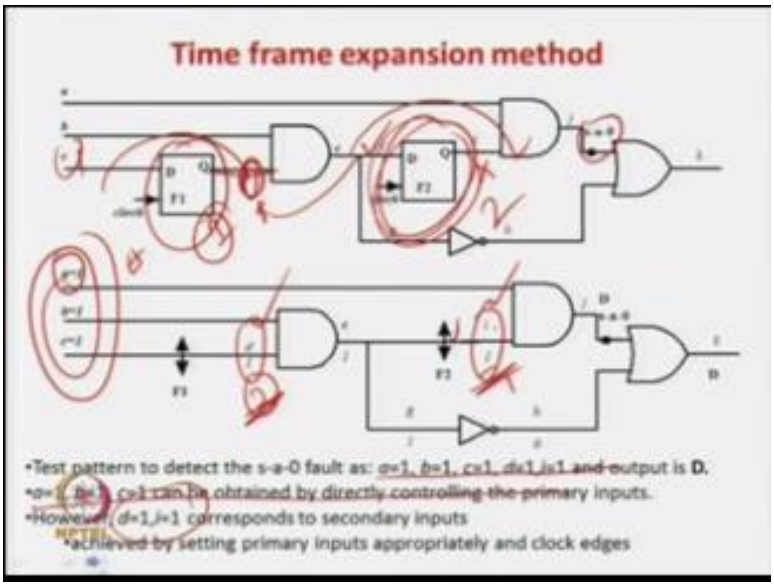
Like in this case the output is depended so in input the input is not depended on owns output and there will be oscillation and you cannot control any value so circuits falls cannot tested in the circuit is very difficult to do that but we will see that but for the circuits where there is no cyclic property then what we can do is then we can control all the virtual primary inputs and observe all the virtual primary that is the property saying is secondary outputs that your virtual primary inputs sorry the secondary inputs that is your virtual primary inputs of a cycle free.

(Refer Slide Time: 47:35)



Sequential circuit they have this one can be bottom of the control in utmost like this in these sequence number of primary input and clock pulse so what we have seen in the last of examples that we request some nets like in this case.

(Refer Slide Time: 47:48)



If you look at it so we require to control these two one and also these two one so now they ask that how many clock pulses on patterns you will be required to do this the property says that if the maximum depth is 2 so you can do it in two clock pulses and two inputs.

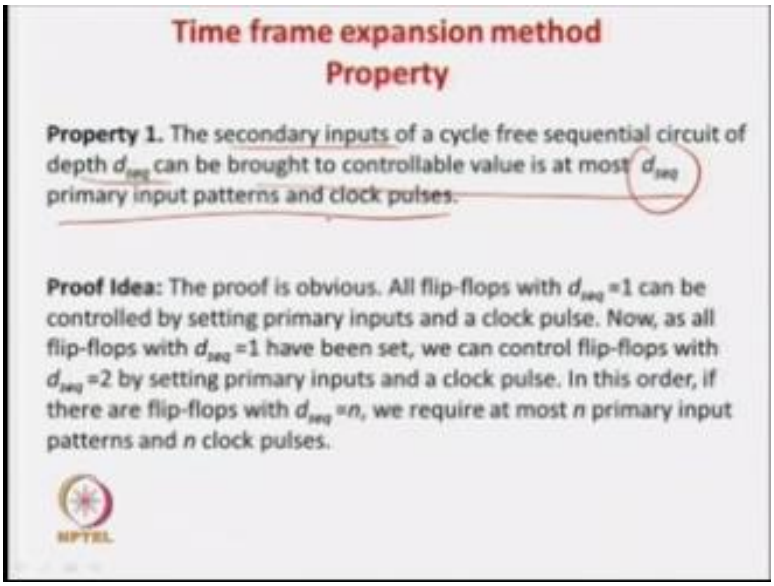
Now how it can be work it is very simple then that idea is very simple because this is sequential depth one so sequential depth once means what it is directly connected to a primary input or set a primary input so give some whatever pattern is required of the output this in this case it is a 1 so you can apply a 1 in C and 1 clock pulse will directly set it that is in another words all the what you can call all these as flip-flops which are in sequential depth 1 can be directly control by setting the primary inputs and apply a clock one.

Because they are directly depended on the primary inputs now after one clock pulse and one pattern all this sequential all the flip-flops of sequential depth one has been set now you can consider for the flip-flops in sequential depth 2 so now we know that after one clock pulse one pattern all the flops having a sequential depth 1 has been set now so the flip-flop will sequential depth two are depended on 1 primary inputs and outputs of flip-flops which are sequential depth 1 so now indirectly you can say that one more pattern and one more clock pulse will set it because after.

One pattern this gate is set because these sequential depth 1 and this get is depended on primary inputs and only on this output that is only on flip-flops whose sequential depth is 1 because this sequential depth is two so now you can apply another clock pulse and this value will be at the output so by second block pulse all the flip-flops having a all the flip-flops having a sequential depth 2 will be set similarly this if you have a sequential depth flop with three then you have to have a hard clock because by second clock pulse all the primary inputs you can set it and all the flip-flops.

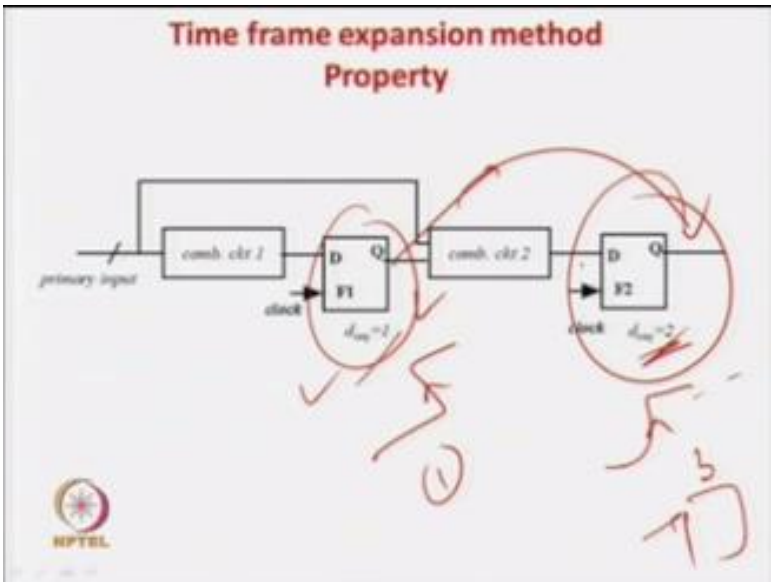
Have a sequential depth 2 has been set by the second clock pulse now in with hide clock pulse the all these flip-flops or sequential elements with depth 3 will be this because in the second pulse all it inputs are red because the inputs our flop with our sequential depth 3 are gate are flops is sequential depth 2 because already we will set in the clock pulse 2 so another clock pulse you apply you set 3rd level flops or sequential depth 3rd flops will be set and so on so the proof is very obvious which.

(Refer Slide Time: 50:02)



We have discuss and we can see this pictorially.

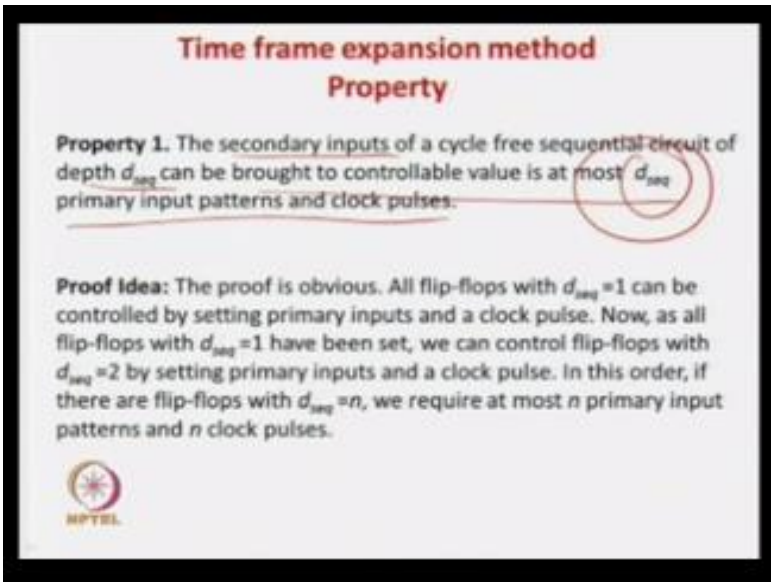
(Refer Slide Time: 50:04)



This way this is the sequential circuit of depth 1 so is the combinational circuit so primary of clock we have apply, apply at all this is set for the depth 2 now in depth 1 we will first flop it has been set now in the depth 2 you require another clock pulse because now this because in first place all this is set and in this case of so the second in case of the flops with depth level of two all the flip-flop with depth 1 has already been set so just you apply on the clock pulse and this output will be control.

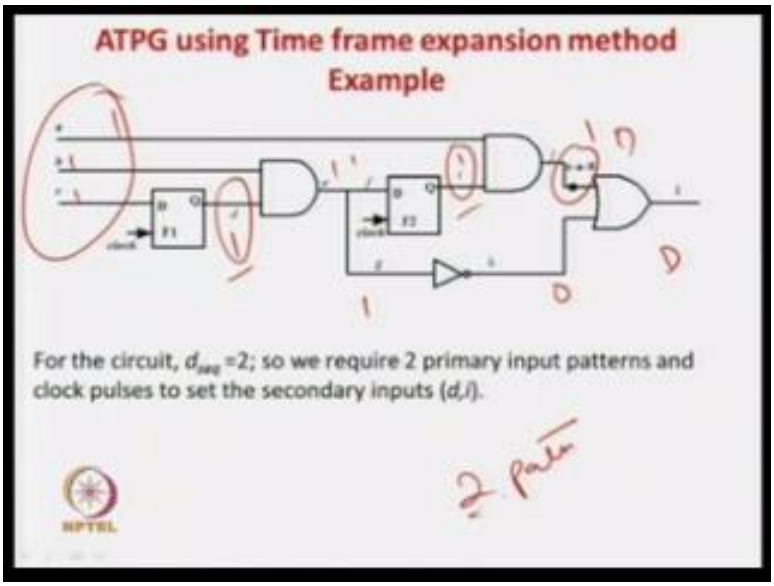
Similarly dot now if these another flip-flop whose depth is that you required the 3rd pulse because by the second clock pulse all the flops of depth 2 has been set and flop with level 3 are depended on the output of the flops with gate two so 3rd clock pulse is depended.

(Refer Slide Time: 50:51)



So very obviously it is said that the secondary inputs of cyclic sequential circuits of depth this can we brought in their control in utmost these sequential primary inputs and pattern so therefore if you have a circuit with 10 flip flop of having sequential depth 10 so you require 10 clock pluses and 10 patterns to set the virtual primary inputs and in 11^{th} period you require to give the primary inputs and that will solve your problem okay.

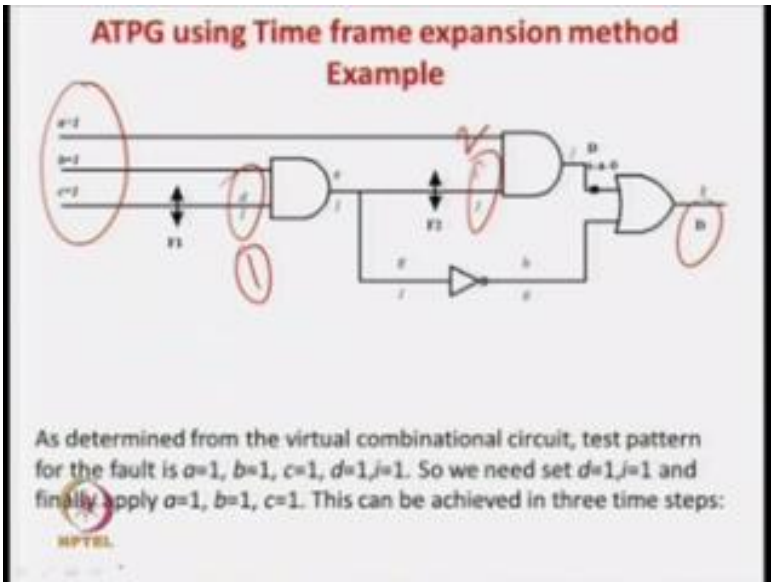
(Refer Slide Time: 51:17)



So again coming back with this property we will see how this time frame expansion method can be applied though a circuit so you know that this is a stuck at 0 fault over there sequential depth was 2 so here we require 2 patterns to bring we have already seen that we require a 1 over here we require 1 over here correct to do this testing because we are going apply a 1 and this will be D we require a 0 over here we require a 1 over here we require a 1 over here 11 this also require a 1 over here something like this we have already seen.

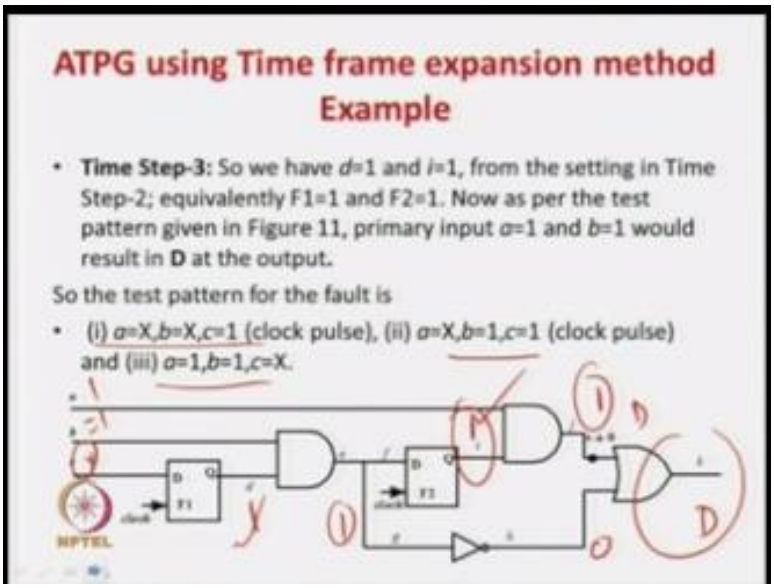
So this we need to control the 1and this we need to control the 1 so and sequential depth is 2 so we require 2 clock periods to make this one to one and this one to one and finally we have to apply a = 1 b = 1 and c= 1 test the fault then I seen that explicitly how it can be done so 2 patterns to set and the 3 pattern to testing.

(Refer Slide Time: 52:01)



So what we do sop first so these what is the case so you require a test pattern as I told you require a d = 1 I = 1 to do this testing so we require to control this at level 1 clock because this is level 1 clock and this equal to the second clock pulse because this is level 2 so this is what is required.

(Refer Slide Time: 52:21)



Now you see this time step 1 so how to do that so time step 1 what we do we require a 1 over here so what we do we make c = 1 other will do not require at this at all and we apply a clock pulse so what is going to happen we are going to get D = n so this you can think that your clock pulse is set okay right so this what we have done in this step 1 other primary inputs are do not acre finally a clock pulse is given c= 1 required that soon.

So we do this so we get a 1 over here and this done now so what is happening now so you can know that now d=1 already and this is x we require a 1 over here this is very important does this in phase 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 apply a clock pulse t hen we can get it over h ere, because what this already done in the last phase, so now w hat you have to get a over here what do we need , w e need a 1 over here and we need a 1 over here, so 1 over here is already done in the last step.

So if we apply a 1 an d 1 over here and we apply t he second clock h then what is happen 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 c lo ck is going here and same clock is going here so it is not that, that if you want to transfer this from here to apply you apply a clock over here and the clock is not coming here the clock will al so be coming over here, so to be very careful that when this one and this one together with this is transferred over here so this should remain a 1 because in the end of second phase we require a 1 over here and a 1 over here so you apply a 1 over here.

So what we have done s o in directly what we have done over here is say first case we have a 1 over here and the second clock pulse we put a 1 over here so a 1 and a 1 you get a 1 so if these as just apply the clock pulse we will get this one. But you have to very careful that after the second clock pulse also 1 is to be here so we also put c= 1 now we apply a clock pulse same clock pulse will go both s o 1 and a 1 , 1 will be coming over h ere and we 1 so this one will be maintained over here.

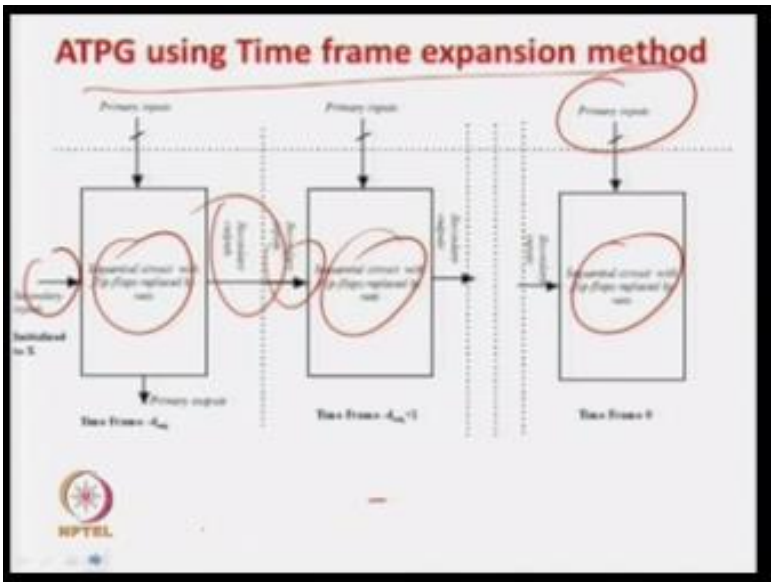
So now you get after the second clock pulse we get a 1 here a 1 here so this 1 was we get a 0 over here and a 1 over sorry 0 over here that is this one here getting after the second clock h, we get a 1 over here we get a 1 over here okay, we get a 1 over here and a 0 over here but this what we have got after your second clock pulse and the time step 3 finally after apply the clock pattern and you will be done so you are getting a 1 over here you are getting a 1 over here it is already done in the first two step.

So now what we require we require a 1 over here that it will get a D then you have to propagate this it should be a 0 okay, and this should be a 1 and you can do the testing so you can do this what you require to get a 1 over here we require a 1 over here that is already set by this a virtual primary input so you have already set by the clock pulses you just apply D=1 so you will get a 1 over here a 0 over here and this is done.

So now also you require a 1 over here to sensitize default so for that this is 1 which you already set in two clock pulses because this was the virtual primary input and either primary input so you want to 1 here, so you apply a A=1 over h ere you get and this is done C= x you do not require this, so three clock periods so first we apply C=1 set this as 1, second

clock pulse you apply b=1, c=1 b=1 and c=1 so set this one and third clock pulse you apply b=1, a=1 c is do not care and you do the testing.

(Refer Slide Time: 55:48)



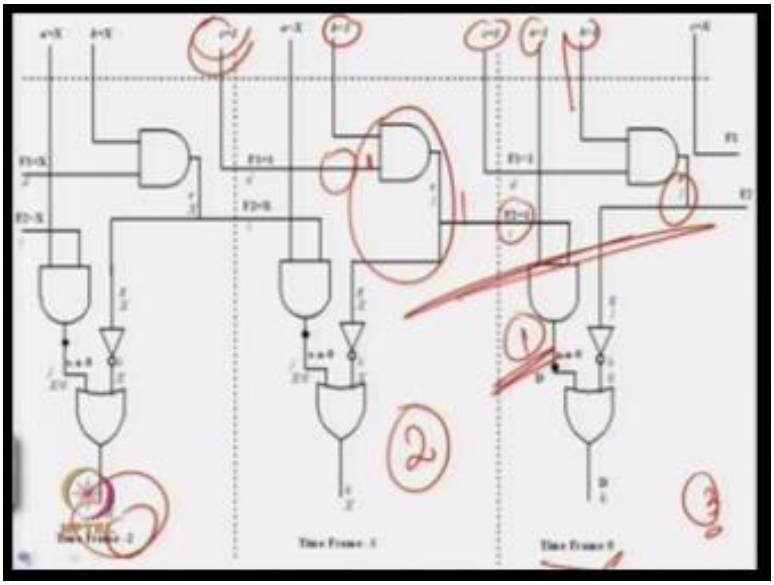
So that is what we have done there is actually call a time frame expansion method because in three clock periods we have tested the circuit in t he first two clock periods what we have done we have set the non primary inputs or the virtual primary inputs or secondary inputs to one and o ne and in the third clock pulse w e applied the primary inputs and the testing has been done.

So if I want to illustrate this as a figure so t his is what is being done so if there are if it is sequential depth is d t hen w e require one time frame which we call –b then we actually called -d+1…. Till d, so in this case we had a frame sequential depth of 2 so this was actually 2- 1= depth 1 then this will be depth 0 an d this is actually depth 1 kind of a thing.

Okay, it is sorry, so what sequential depth was 2 so it was -2 then –d, d was 2 in this case so it was -2 then -2+1 that is 1 and the finite was 0, s o finally applied a =1, b=1 which will be t es ting in this case we apply c=1 we set the value first output of the first flip flop to 1 and in

this case we apply b=0 and c=0 then it is actually set the value of the n ext flip flop to 1 a n d finally we applied this pattern and we did the testing t hat is how w e represent this the circuit I mean this is a way of representing your time frame expansion method that is we replace the f lip flops with replace by next so flip flops are eliminated so this what we do and this is your secondary inputs not the primary inputs this is your secondary outputs an d this we how we represent is your primary inputs and this how we control.

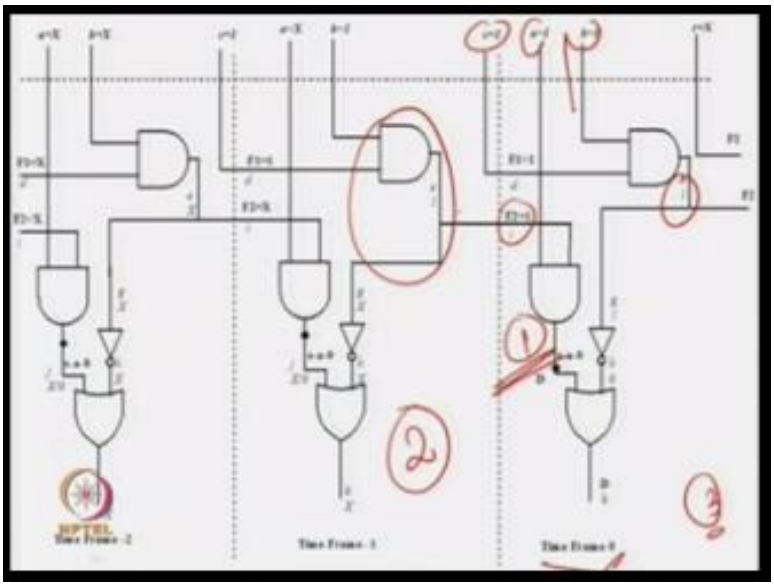
(Refer Slide Time: 57:25)



So now just whatever we discussed in t hat so let us just put it in the time frame method and see what we have done, so this was your circuit if you remember this part we have just done drawn here horizontal with this circuit we have drawn in a horizontal way sorry a vertical way and this things have been shorted and this thing has been shorted that is the only thing we have done. So just see t his is your flip flop t his are the way entered so let us just mark this a s say we can call it gate number 1 and gate number 2.

So just you have to remember that input of gate 1 is b and d and input of gate number 2 is i an d a but this i is actually a virtual input and t his d is fine and this for this case d is a virtual input, so you can just see what we have done.

(Refer Slide Time: 58:09)



This is say we can think that see so this gate is, this gate is 1 and this gate is 2 kind of a thing so you see one gate one it is directly connected from a, sorry t his gate 2 and this is gate 1 sorry this is gate 2 and this is gate 1 this gate 2 gate 1, gate 2 gate okay, so see thus this gate 2 is dependent on a and the output of this second flip flop so this how we represented.

So this gate 2 this gate 2 is dependent on at a that is primary input and output of the second flip flop that is height f2 and gate 1 if you remember was dependent on d as well as the output of the first flip flop so this is your gate this is your gate 2, so it is dependent on b as well as the output of the first flip flop, okay. so t his is how it is done so now you can call this one t o t his can be called a 4 this can be called as 3, so this can be called a 4 this can be 3 or whatever you like this is not that much important this only thing is important.

So gate 1 is dependent on b and the output of the first flip flop gate 2 is dependent on a and this one so you be write it this one and this a stack at fault 0 so we write this is c about a d algebra which you have written extended d algebra so it is stack at 0 so normal fault case it is 0 normal case is unknown. So all other things we have marked as x, so just this one you can very easily see, so all these things are x so only how we have written the circuit is that instead of

this gate number 2 and gate number 1 so instead that this is corrected at the output of a flip flop we eliminate the flip flop and we say that f2 output.

Similarly for gate 1 is a dependent on the f1 output so we have s aid that if flip flop is not there it is directly connected this way. So now you see what do you require so we require to test this circuit so what we require you can go to this last level of the circuit this is 2 and a 1, so we require a d over here so d has to b e propagated over here so you require a 0 over here just D algorithm okay, so 0 means you require a 1 over here, you require a 1 over here, you require a 1 over here, you require a 1 over here similarly so this is stack at 0 so you reply a 1 over here, so you require a 1 over here and you require a 1 over here so this what we are requiring over this.

Now this will actually test your circuit, s o this is the last time frame so mark it, now to get a 1 over here so this is the third of 0^{th} time step so this gate is connected to the output of second flip flop, okay. So to get this 1 the output of flip flop 2 so one primary input can be set at the 0^{th} time step that is done. But the other input 1 has to be come from the output of flip flop number 2 so you have to write a output of the 1 from this as the previous note.

Similarly for this gate number 1 you require a 1 over here and you require a 1 over here so t his is a primary input can be controlled but the other input that is c=1 that is AND gate that is actually also you can know that it is the output of the flip flop number 1, so you have to also know that if flip flop number 1 has to be a 1 you make this a 1 at time step 0 that is other words we see that when is c, that is the flip flop 1 then output has to be a 1 in the previous step to get a 1 in the next step.

So you w rite that flip flop 1 is to be a 1 over here flip flop 2 is to be a 1 over here but that is at the time step 0, and the requirement and when you can achieve it t his f lip flop to 1 can be achieved at time step -1 and this also has to be I mean achieved a t the previous time step. So now in the previous time step if I want to apply f=1 so what you have to do you have to make c=1 and apply a c lock pulse because the input of the flip flop 1 is directly dependent on c, so this can be done very easily by applying c=1 in the previous time step.

Now but you require flip flop output 2, output flip flop number 2 to also to be 1 but how do we achieve this t hat is somewhat no t very simple because again the output of flip flop 2 is driven b y gat e number 1 which is one of the input is primary input so you can directly set it at time step -1 but another input of this AND gate which is driving flip flop number 2 is dependent on the output of flip flop number 1 so that has to be controlled in another time step which is 1 head before.

So to get a 1 at this which will actually make t his flip f lop 1 a t the third time step so what you have to do you have to get a b=1 in the second step that is fine but again another input a n other input that is actually of this AND gate number 1 is connect with the flip flop number 1 so it has to be controlled at the time step number -2, so in -2 if you make set a=1 so w hat will happen that flip flop output 1 will be 1 in the 1- step t hat is what actually we are doing, so indirectly what we are requiring just very quickly look at the snap shots what is have been done we require a 1 over here which is the flip flop number 1 so to get a 1 over here we require this AND gate to be a 1 by the primary input on 5.

But another input of the AND is actually output of the flip flop number 1 , so we also have to be a 1 over here so ho w do we get that, s o this as this f1 is actually a flip flop which is controlling gate 1 so it has to be controlled in the previous step. So in step -1 time stamp -1 we apply c=1 and we get it, in time stamp 2 a gain the AN D gate is there so we require a 1 over here another one is by this primary input a=1.

So this you can very easily get in the time stamp 0 or the third step or third step, but another if this input has to be 1 it is the output of flip flop number 2 so this has to be achieved at the previous time stamp the time stamp -1, so to get a 1 over here you require a 1 over here so this can be done in the second step fourth time step -1 on the second stage, but again this AND gate is dependent on flip flop number 1 which also cannot be done at the current instant , so you have to go one m ore stage before and you have to make c=1 so if you make c=1 in the time stamp -2 and then you make b=1 a time stamp -1 then what you are going to

get is that, you are going to g e t a output of 1 in the second stage which can be directly applied to the third stage.

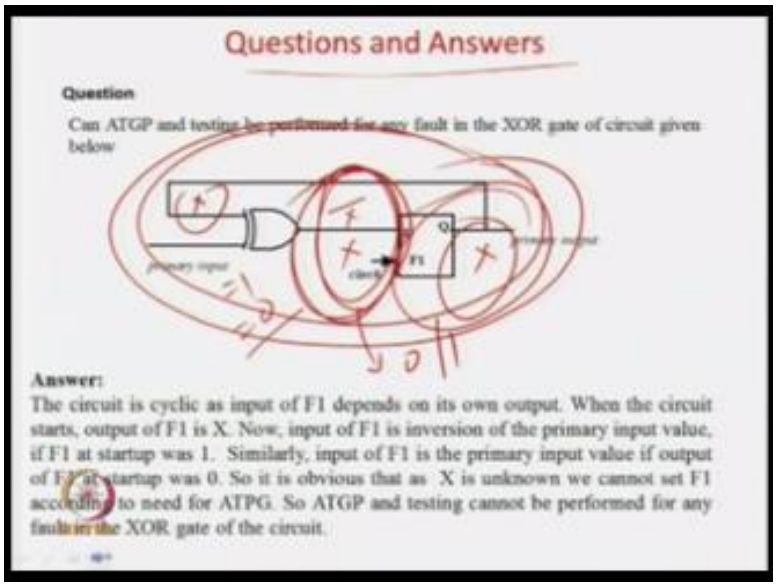
And if you are making c=1 in this time stamp -1 or the step number 2 then w hat you are going to get in the third stage you are going to get a 1 over here. so indirectly what we have already seen in all example before we are just writing it in a formal way that is just some requirements the requirements are 1 over here, a 1 over here these are the two requirements a t the time stage number, time step 0 or sorry or step 3 this step 3 you require this, so this requirement this can be achieved by m akin g b=1 and another input has to be a 1 so which we can make achieve b y making c =1 in time step 2 you can say or time frame -1.

This requirement of 1 that is again these are actually the virtual primary outputs sorry, virtual primary inputs which are direct controlling so now again this 1 is required, so get this as a 1 we require this one is a 1 we require this one is a 1 so we can which we can easily get it time frame 2 or step 3 you can say, but another input to get is the 1 we require at this stage we go to t h is stage to make this a 1 , to make this as a 1 we require b=1 in time e frame -1 and also another input is there which is now dependent on the time frame -2 that is actually because this gate is dependent on one flip flop level 2 and one flip flop level 1.

So at you have to again go back to time frame -2 an d you have to set c=1 if you do that than this one will also become a 1 in the second stage we will get a 1 and finally you are going to get your requirement. So this is nothing but whatever you have discussed is nothing but this three stages, the stage one there where you are making c=1 and applying a clock pulse that is the stage 1 this one, then next is making 1 here, 1 here applying here and the third stage is nothing but applying this one and testing default.

Here the three stages which we have done we are just representing pictorially in this way so these are very formal way of design.

(Refer Slide Time: 1:06:43)



Is having a time frame expansion method and you can solve your this thing, okay so that about time frame expansion method base testing so in this case you have observe that to test a single stack at fault what we have to do in case of ATPG for compression circuit only one stage would have solved your problem, but in case of sequential circuit we require much m ore number of stages and number of stages are if you equivalent to nothing but your number of sequential depth of the flip flop.

So that is why you can say that sequential ATPG is compression ATPG is star sequential depth, so that is why is a very, very complex procedure and you have to also remember that any stage you may have bad tracks leading into a big problem, so in the next lecture we are going to see how we can make this very complex problem a simple one. But before that there is a very quickly see a question so this is a cyclic circuit and we have not considered an y cyclic circuit.

So now if I ask you that can you test any fault in this circuit so the question is can ATPG and testing the perform for any fault in this OR gate can do the answer is no, why just we will see. Say for example we know that the output is always a text before the circuit is starting so you will

get a x over here so now if you have the primary input as 1 so your answer will be x prime because we know that in case of an OR gate if the input is 1 the other is inverter, and now if you primary input is 0 you are going to get a x.

Okay, and again this x or x prime will be feedback and this will be look so at no point of time you can be able to get a value of 0 and a 1 over here, if you get the primary input 1 it will be x prim e if you get a 0 it will be some other thing. So at no point of time you can get any control decision or control value or to get a decision over here because the input is dependent because this input is dependent is o now, s o therefore as you cannot set the flip flop so if you have a circuit something like this no point of time you can convert this x to any other stuff and.

(Refer Slide Time: 1:08:35)

Fault cannot be this step, so with this we come to the closing of this lecture and in next lecture we will see how we can solve the problem of such a huge number test pat terns required for a sequential circuit testing, s o we have to somehow reduces the number of test patterns required or the number of sequences required for testing a fault in a sequential circuit, thank you.

Center For Educational Technology

IIT Guwahati

Production

Head CET

Prof. Sunil Khijwania

CET Production Team

Bikash Jyoti Nath

CS Bhaskar Bora

Dibyajyoti Lahkar

Kallal Barua

Kaushik Kr. Sarma

Queen Barman

Rekha Hazarika

CET Administrative Team

Susanta Sarma

Swapan Debnath