INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

NPTEL
NPTEL INLINE CERTIFICATION COURSE
An Initiative of MHRD

VLSI Design, Verification &Test

Dr. Santosh Biswas
Department of CSE
IIT Guwahati

Module VIII: Fault Simulation and Testability Measures

Lecture II: Fault Simulation – 2

Design Verification and Test of Digital VLSI Circuits
NPTEL Video Course

Module-VIII
Lecture-I, II and III
Fault Simulation

Welcome to the second lecture on fault simulation as discussed in the last day that our topic on fault simulation will be a three hour lecture so in the last discussion we have seen that to generate test patterns for a given propagate mobilize secret model what we have to do there two approaches one approach was based on what you called that sensitize, propagate and justify approach so in that case given a fault the first at the sensitize the fall that is be one fault that second one fault.

we have to apply a one sorry ,you have to apply zero and if the second zero fault then you have to apply a one then if the balance to be propagated to some primary output then you have to justify the signals so that the balance can be propagated and sensitized so then we have seen that this algorithm has been complex and if you want to apply it to all the faults present it will take a long time then you have try to seen for already approach in which case we say it whether what we do we take a one test pattern.

And then we apply in the circuit and then try to verify what are the number of fault that is detectable by this one then if there is a forgiven random pattern we can find out that 70 to 60 or 20 faults have been detected then drop that and then we take another random pattern and try to do this procedure. So what is the advantage of this procedure about the sensitize, propagate and justify approach.

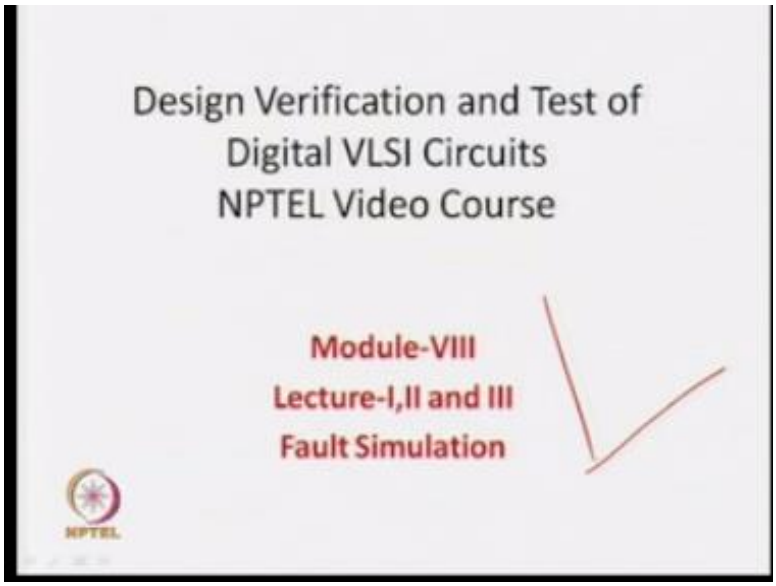
So see for example the primary importance of this one was that in test pattern generation or any test pattern they have single pattern for multiple faults in other words a single test pattern can be applicable to test more than one kind of a fault so in this case what is happen so if you are using sensitize, propagate and justify approach so fault can be tested for a pattern that is one pattern will be generated for this one, but if we are using random patterns then what happens multiple faults can be detected by a pattern that you had analyze and find out that is one random pattern is given 3 or 4 or 10 or 15 faults.

It determines can be brought down can be detected and can be follow that these are the this pattern can detect this faults so if the problem of multiple fault detection is also taking into picture, but several example like say one pattern say pattern I detects a 10 faults, fault 1, fault 2, fault 3 being sensitized, propagate and justify approach the same pattern maybe generated when you are trying to do this for all of this faults f1, f2, f3 and fn.

But for so you have to run the algorithm say 10 number of times and if it is the for a random pattern generation when that one is applied and tested for which faults are being detected then maybe random test pattern generation algorithm all the fault that is the f1, f2, f3, fn will be determine in one go or in multiple number of goes, but for the same random pattern you have able to find out that so when the faults are detected so we are save in two cases first the algorithm of random pattern generation is simple and secondly also.

If the same pattern detects number faults then you do not have to run the algorithm repeated. Then we have seen that so for some cases sensitization random pattern generation is the much better approach then sensitize, propagate and justify, but this is not the end of the story because we have seen that for if this one random pattern applying.

(Refer Slide Time: 03:36)



A random pattern generates some new faults are detectable then you have to apply a next random pattern then a few more number of faults get detected and so for but after some time there are 90% of the fault coverage so this saturates after that what happens if you apply a random pattern no new fault will be detected and the procedure will keep on going till some one or two new faults will be detected and so.

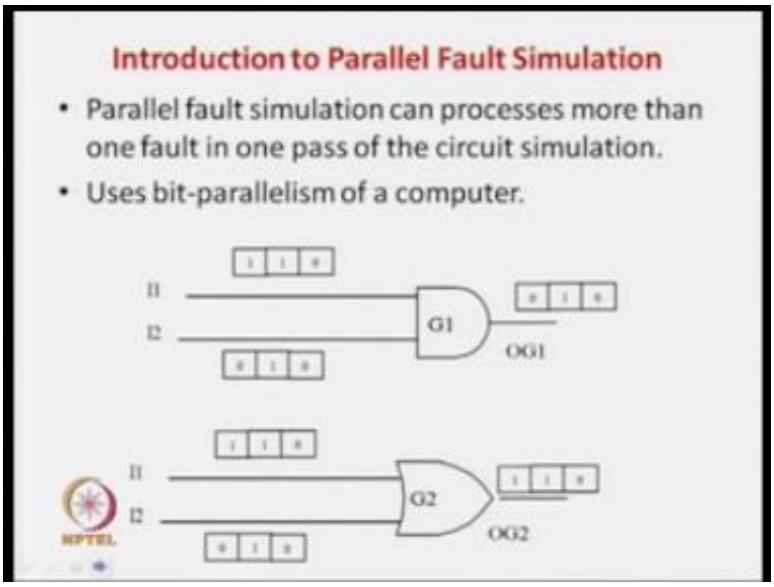
So in that case the number of random patterns and determining whether are fault is detectable by it or not because more complex of sensitize, propagate and justify approach so we turn them as difficult to test faults and for those we did have apply sensitize, propagate and justify approach, but for 90% of the cases our random test pattern generation is a very good approach that you have seen in the last lecture then we have gone for if you want to use random pattern generation or random pattern generation procedure.

In one very important part is that one that they should be procedure for fault simulation and circuit simulation that is given a pattern you have to find out what is the normal value of result that it the value of the circuit output without any fault and then you have to find out what is the value of the circuit output with a fault that is we have to simulate this results and then we have to

simulate this because you do not have the hard copy of the circuit and then if the output defers in the normal case these are the faulty circuit then we know that the fault is detected.

By the random pattern so fault simulation was a very important part of per random test pattern generation and we have seen that two types of fault simulation one is the even driven simulation and one is the simple one by good simulation and then we have seen that even driven simulation is very important because it is simulates only those part of the circuits where there are changes in the signals then we have taken a very simple.

(Refer Slide Time: 05:20)



What you called fault simulation approach this is a sequential or serial or sequential based kind of fault simulation which we called the serial fault simulation in which case one test pattern or for a random pattern you find out whether one fault is detectable or not then you take another fault and then find out with a reject able and so for so you take one by one for at a time for each pattern and then if the fault is detected whether pattern you dropping it and so it proceeds then you find out that this is not very official because for each fault.

In the worst case you have to simulate the in number of times if there is a end number of random patterns are tried so now then we saw that there should be some improvement on this one then we told that from the lecture onwards we will study some important other advance fault simulation algorithms which will paralyze the effort like which is done in a case of serial one fault at a time is considered to be tested whether the fault is detectable by the random pattern.

So instead here what we do we will take one random pattern and then try to verify if more than one fault can be detectable that is the verification that whether more than one than fault is detectable by that random pattern is possible or not that you are going to do. So there are many algorithms like pattern fault simulation then surely we will see the rest amount of algorithm so as the name suggests first we will try with the parallel fault simulation.

And then detect the fault simulation is there and many other so we will see three more algorithms which are namely towards the working on paralyzing the fault simulation approach so what is parallel fault simulation as the name suggests you can see parallel fault simulation actually can tell what is for a given random pattern it can tell you which are the faults that can be detectable in one run of the circuits.

So one run of the circuit means one parallel simulation one fault simulation of the circuit so what you can of simulation of a circuit means you generally take some inputs and then travel to the output this is actually carry you can say that and scanning my circuit once this called the scan of a circuit. So you are scanning the circuit one and in one scan you can determine for a given random pattern more than one number of faults that is parallel you can paralyze your effort you have remember that in serial fault simulation only decision about only one fault can be done so how are the base it actually uses or beat parallelizing of a computer.

So let me just elaborate of a simple example so if you see our computer of the hardware at multiplier, subs tractor which is the never processor. So whenever we execute some C program or any other programming language inside the processor they beat by the operation to finally do the operation because we know that our computer is maybe a 32 bit machine or 64 bit machine but in the processor level everything works at the bit level.

So if you add 3 bit number say 110 and 111 then what happens basically that is actually this have been this is 7 and this will know that 6 so if you do 6+7 in the C language basically what happens inside the processor this is what is happening binary it happens it is 1+1 is 1 then111 carry 1 and 1 in a something like this so basically what happens what I mean to say is that this hardware operation this bit wise and bit wise or whatever are happening by our hardware or our hardware multiplier or hardware distracter whatever is available in your processor so these are parallel and you know that our machine is 32 bit machine or 64 bit machine so they is 64 bit processor so 64 bit ALU.

So which may have a 62 bit wise and operator 62 bit wise or operator so all the hard ware's are available so parallel fault simulation actually takes what you can call takes advantage of this architecture of the machine say for example, I want to parallelize the output simulation I want to parallelize the simulation of the circuit the circuit is very simple as an gate but I what I want to do I want to find out I want to simulate for the circuit for 3 inputs one is 10, one is 11 and one is 00.

So even if you take a mean what you call that compile code simulation or even you could even driven simulation or whatever you do basically there if you symbols you got so what will happen first it will simulates for this one then it will simulates for this and it will simulates for this one, but if you have bit wise operation got that is you know that I was simulate for all then what you can do you can say that this is my one bit input number and for three bits are there bit 1, bit 2, bit 3 because for three instances you have to simulate your circuit.

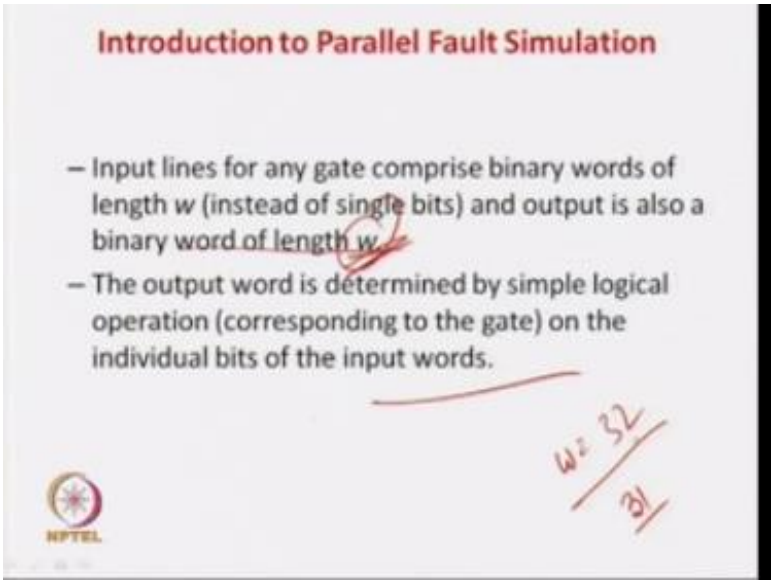
So what you can do instead of writing a for loop and then simulating for 10, 11 then 00. You better represent it this is a number called 6 and this one you represent as a number called 1 and you do bit wise and. So if you are doing bit wise and definitely we get the answers 010 which is nothing but 2 so what happens inside your circuit if whenever you are doing what you call bit wise this one bit wise and with 101 internally it can be done parallel because the internal hardware of your circuit or internal and operation of your circuit it can be a 32 bit processor it can support 32 bit parallel if it is 64 bit machine it can support 64 parallel bit wise operation.

So in this case in one go your output will be something like 101 and you can get this 3 bit parallel in words what I am saying so instead of giving 3 individual bits or 3 individual random patterns in 3 different goes what we can do is that we can think 3 random inputs or 'n' a number of random inputs you represent then as or vector you represent name as a vector of binary numbers and then for example if you are using 4 numbers for example you have to simulate to find out the value of the circuit for 4 different inputs.

Then you have to write say 1010 and this one maybe 1100 something like this then you represent this one as one vector and this one as one vector what you do then you do a bit wise AND operation then you it will be done parallel instead of first you simulate for 11 then 01 then 11 and 00 so it will be 4 but if you representing this 4 inputs as in the 2 input cases as 11 as a vector of this size 4 bit vector as binary numbers and then you do bit wise operation then it will be in parallel it can be done parallel and in scan of this circuit you can do it.

Similarly this for the gate and this is for the OR gate same thing you can do but instead of you do a bit wise or operation so this is the basic idea of parallel fault simulation.

(Refer Slide Time: 12:07)



So generally what happens so if that parallel fault simulation mainly depends on what is the parallel efficiency of your computer so if your computer 32 bit machine so you can do 32 wise operation parallel then 32-1 actually we will see how then 31 faults can be simulated in one go that we will see how so it is actually called we will see then input lines of any get compressed binary works instead of single bits and output is also binary.

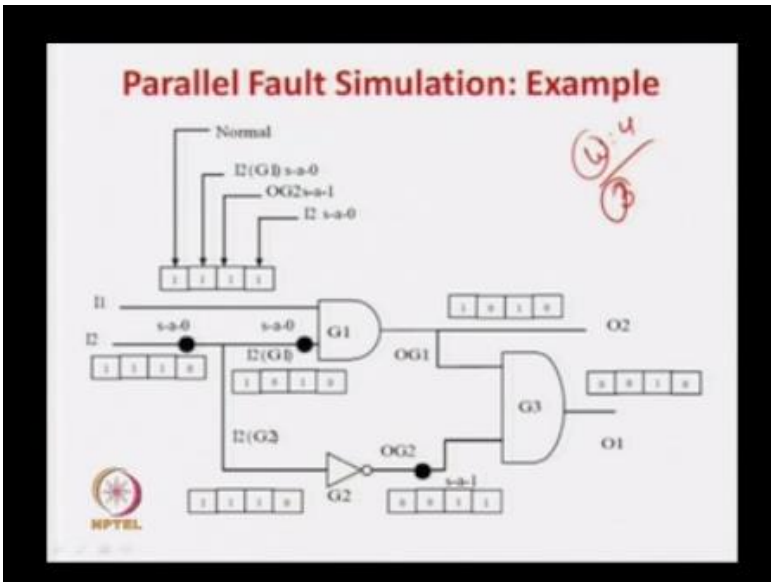
What of length W that is what I was discussing so if your computer is having so what can be the maximum W that means it is saying that now instead of each random pattern so what you are trying to do we are representing the whole thing is the binary vector so now what is the maximum length of the binary vector like for example as I told you so in the last example we have seen 4 bits like 11, 01, 10 and 11.

So there are 4 number we have tried this was one number, this was another number, this was one number, this was one number for simulating the circuit but then we represent then as two vectors so here the length of the each vector is 4 so now the question is what is the length you can go the answer is it is the length is maximum in a bit length of your machine so that you have actually called W.

That is the work length that is the W so output will also be a vector now instead of individual output it will be a Eigen of vector if you call this W. What is the maximum length of W. The maximum length of W is the bit level parallel or bit level paralleling supported by your computer architecture also can be roughly say that if your machine is a 32 bit architecture so it can go for 32 bit parallelism and the output is determined by simple logical operation are corresponding of the individual bits and actually this can be done parallel because we know that bit wise parallelism is normal operation.

Now we will see that if W of a computer is a 32 then for fault simulation we can go only for 31 fault and one bit has to be developed for normal operation that we will see.

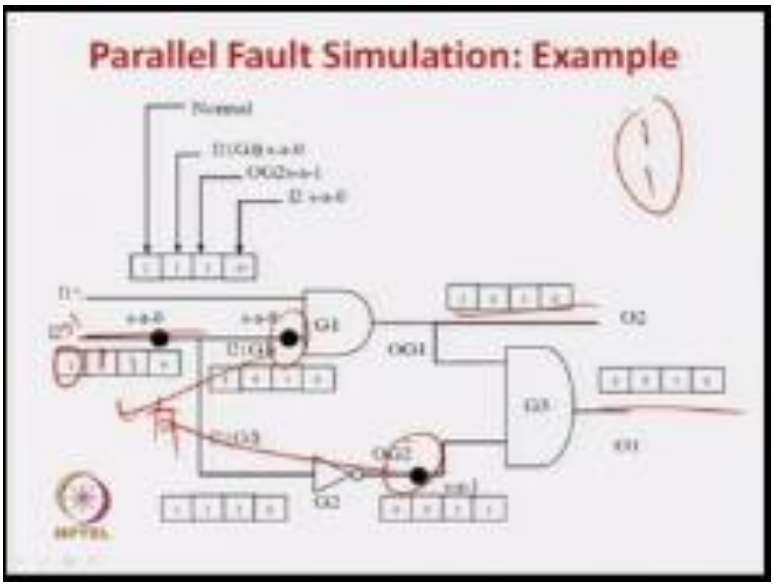
(Refer Slide Time: 14:08)



So that means if you know that computer has two bit machine then what we can assume that in one row we can reason about not having two but one less than that, we can reason only about 31 faults in one scan of the circuit. So we will see the concept by an example because the concept is otherwise bit listening directly from the theory as I was mentioning is not very helpful because it creates confusion.

So we will directly go in for an example and then we will see. So for this let us assume that w=4 that is you are making the four bit processor and only 4 bit numbers can be parallel EXOT and OR, Not etc can be done only four bit at a time. So this one as we have already discussed so in case if the machine w=4 then listening about only three faults can be done in one more not more than that.

(Refer Slide Time: 14:55)



So how it is represented so what happens let this be the circuit. So we know that from our earlier discussion that each individual line like this one, this one and this one the three fan outs they are individual lines like this one is one individual line, this one is one individual line and this one is one individual line. So why there are individual lines, because fan out lines will always be individual cases.

Next what we are doing, so now what happens and all the other lines obviously are individual lines. Now for corresponding to each individual line you should have an array. So what is the length of that array with the one-dimension array, so what is the length of the array, the length of the array is W. So here 4 so we have 4D arrays you can see at all the points of your NAND of the circuit.

Now each places or each box or each individual element of the array represent some special thing. So what is this, the first bit or the first block of your house correspond array corresponds to normal performance of the circuit. That is this is, it takes the response this array this array is corresponding to this net, so this place states that what is the value of this net, this place tells that what is the value of this net when there is no fault in the circuit.

Next there are three other boxes as I told you because w=4 here, so 3 can be used for fault simulation. So there are three boxes here, so each box corresponds to three different faults in the circuit. Say this one the first box corresponds to I2G1 stack at 0, so this one is I20, I2G1 stack at 0 so this box corresponds to this fault, next box is I2OG2 stack at 1 so where is OG2one minute sorry, I2G1 this is I2G1 stack at 0 okay, just a minute.

So this I2 sorry this one is OG2 sorry this one is Og2 so the next, next this one is for the OG2 stack at 1 fault, this one is stack at 1 fault, and I2 this one is I2 so this one is stack at 0 fault so this place corresponds to this one. Now this is for normal, so whenever this array is treated with line first box implies that what is the value of this net when the circuit is normal. Then the second place tells that is I2G1 it corresponds to this fault.

So it states that what is the value of this line whenever this fault is there, second this in third place tells O2 stack at 1 is for this fault. So this net, this box for this array say that what is the value of this line whenever this takes the fault over here this is the stack at 1 fault. And the last box corresponds to stack at 0 fault it also states that what is the value of this net whenever this is stack at 0 fault over here.

So and again obviously as you can see there is array at all the position so for example this array what does it represent, this place of this array represents what is the value of this net, when the circuit is normal. So this box again states what is the value of this net whenever there is stack as 0 fault here. This box states that what is the value of this net sorry this net not this one this find out net whenever they just stacked at one fault here similar last box state that what is the value of this net when these are fault at this net so as you can see.

So what you called these are one dimension array at each net in the circuits so the whole each this array at each point states what is the value of that net at normal condition and each of the fault condition is represented parallel. And you have to also point have I miss so we will start the fault simulation so with this representation we start they say for example we give 11 and the input that means this our random pattern here is 11 so what is the random pattern it is a 11 so

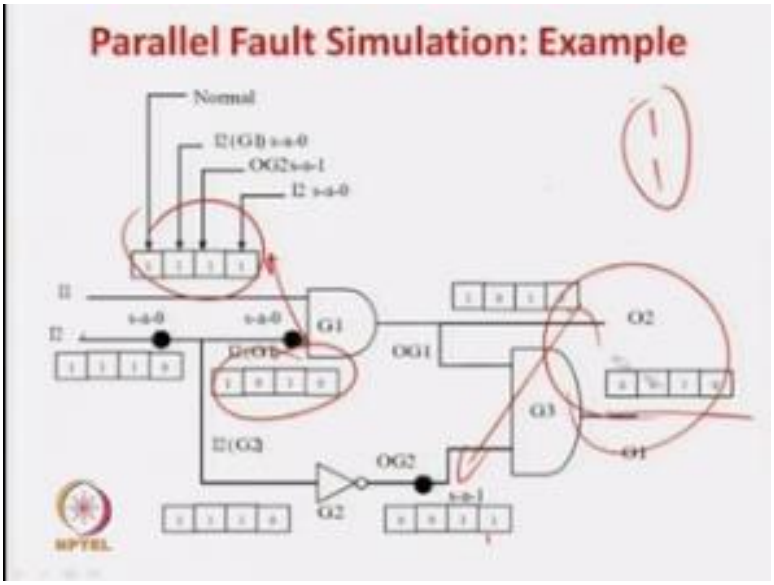
now for the random pattern 11 what is the value of this represents this box represents what is the value of this net when the circuit is
Normal as random pattern is 1.

So obviously the value of this net is 1 at this point so the fast box will have 1 second fast for similarly for this net so what is the value of this net when the input is 1 and the circuit is normal the answer is 1 so that is why this first two places in my array at I1 and I2 are 11 now let us see what happens now let us the first box the first box says that here in this case the first the box say the first box say that what is the value of this net when ever these are stacked at 0 fault here okay and input is obviously random pattern.

So the stacked 0 fault here as no affect on the slide so obviously the answer will be 1 here because if you apply a 1 over here over here then it is stacked 0 fault here this is the ministry of fault similarly the second box corresponds to this stacked one fault so the stacked one stacked fault here as no effect at this net so the value be one here because 1 is applied on the input similarly this stacked 0 fault as no effect here so the answer will be 1 in this case so this array will be all values will be 1111 here okay now let us see for this again works.

Now this one will be already 1 we know so the circuit under normal condition this net value is so it is 1 now you know that this stacked fault as no effect on this net okay no effect so the value will be 1 also this stacked fault as no effect here so the answer will be 1 but now the interesting thing is there let us see okay now just see.

(Refer Slide Time: 20:32)



Here actually the answer should be 1 because we are applying a 1 but this box corresponds to this stacked 0 for this stacked 0 for and this stacked 0 fault as average at this line so if this value will not be 1 but it will be a 0 which is shown here so this two box are unaffected because this fault and this fault do not have any effect at this net only this fault as an effect so what is we are going to observe that instead of 1 this value will be a 0 so this is how the input array are being filled.

Now what we have to do now we can easily see that these two values can be filled up because fan how do you just get propagated so this will be directly propagated here without any changes because this net, okay because this net with this is normal case if this answer is 1 this value of this net is 1 so it is done so this fault does not have any effect on this net so it is a 1 same value is propagated from here.

So again this for do not have any effect of this so value is propagated here and similarly also this fault that is 0 so this fault has an effect here so that means if the stack at 0 fault here so this 1 gets 0 here because the stack at 0 fault which is also effected here at this net sorry, this net is also affected because of the stack at fault so you get the answer a 0 over here, so basically what

happens this box shows that if the input is 1 this input is 1 then and the normal case this net is going to have 1 under this stack for the answer is 1 and at this stack for the answer is 1 and under this stack 0 for the answer is 0.

So only you can see that this stack at 0 fault has a effect on this strategy also obvious, so this array becomes now 1110. So now in this case also now when you are propagating this so if you having a 1 over here so in the normal case if this net is also going to be 1 correct, then this fault is having an effect on this one so this 0 this 1 gets converted into 0 over here because if you are applying 1 over here so this is 1 because this fault you are not having any effect here but this stack at 0 fault is having a effect at this net.

So now in this 1 will not be propagated here but it will propagated as a 0, but this for do not have any effect here so 1 will be remaining a 1, and this fault this stack 0 for also is an effect that this net okay, so this 0 remains a 0 over here. So now this array will be 1010 kind of a thing right. Now what you have to do now again let us see what happens so thus just you have to know this an OR gate sorry, a NOT gate so again you have to a logical operation so we know that we have a parallel logical operators are available over here.

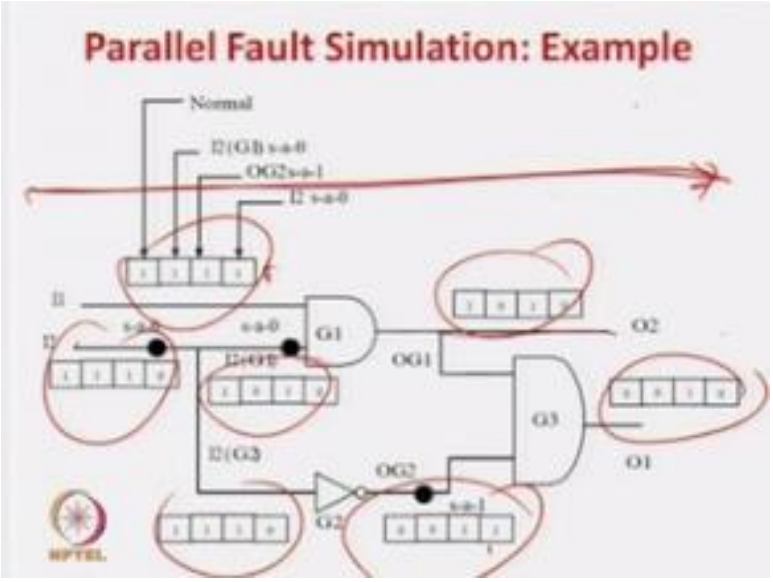
So what you can do so logic NAND logic NOT of this array so logic NAND not should be equal to 1, this is 000 and 1 so this should have be in the case here you are inverting this but you see that this is not a 0 over here so this is correct this is correct but you see this 0 has been converted into 1, so now why is that case the case is so because you should have been 0 because by this parallel NAND operation.

This should have been a 0 but this net this last how the third house corresponds to this thing stack at 1 and this net this fault has a effect on this net and which is converting this 0 to a stack at 1 fault and so the answer here will be a 1. So you are going to get as a 0011 which is the answer over here so this only this is a this thing have been converts sorry this is the mismatch or what you can say in one consist in this one you should have been a 0 by negation but again this scatted for it is taking its roll second fault is taking 0 and everything is getting changed over here right.

So now so this is your array at this point and now you can see again we can do a and operation between this two so in this case it is 1010 so this and it you will get 1010 because there is no fault over here so changes are there and it will get a consistory answer, so it is 1010 and this one is 0011 now again you can do a AND operation here again no fault is the same area so there is no affect will be here so you can directly keep on doing the AND operation.

So you are going to get 0010 so it is like this, only there will be only one over here so it is okay, so the final answer is 1001, now so what happens there is one interesting thing you have to observe that.

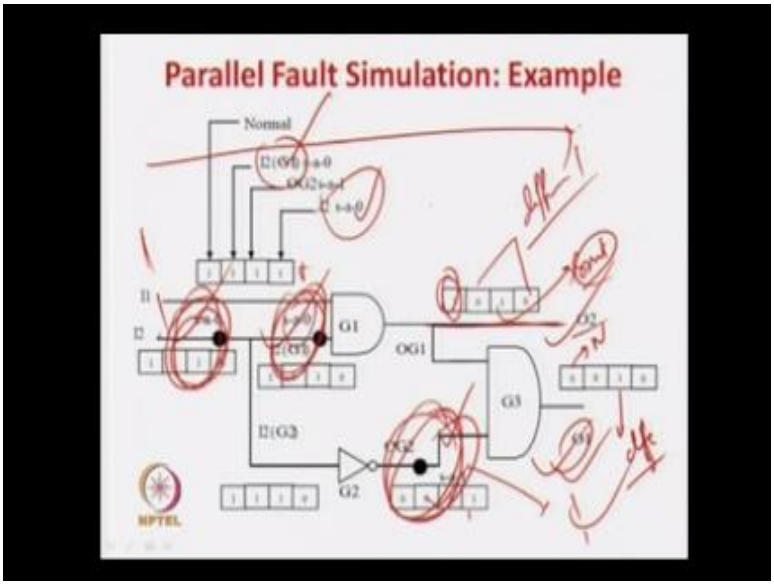
(Refer Slide Time: 25:11)



We have gone only one from this part of this to this part first we have to generate this array then we generate this array then you have to generate this array and then you have to generate this array then we have design negation obviously that reach each state you have to find out if there is any affect of the stacked fault and the plusses of the array otherwise you just simply do a AND operation OR operation NOT operation whatever is the case and then her also here also you get the answer this will be just a propagation.

That is not a different and then again you do AND operation now you see a V land and in one iteration of the circuit or once kind of the circuit we are able to complete our operation that is be very important, now you see what does this two array signify.

(Refer Slide Time: 25:49)

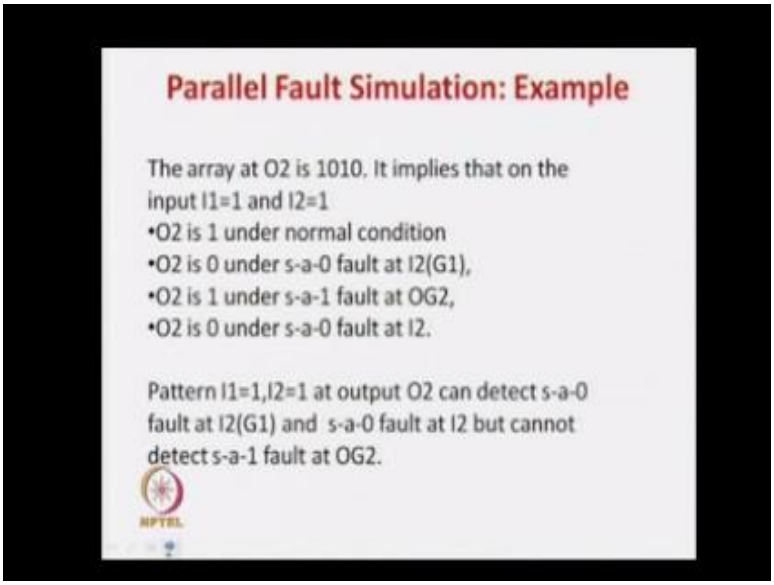


So in one scare of the circuit we have generated these two arrays say array one which for this output and array number two which is for this output, now you see what happens is very impotent se this corresponds to normal so it says that if your input is 11 then the output at this net okay is 0 under the normal condition okay and it says that if the answer if the input is 11 at this net the answer is 1 that is what is it say okay.

Then it says that the second place says that if the input is 11 and the i2 G1 stacker and this stacked 0 fall the output of this net is 0 so what is says that for random pattern 11 at this net okay and this fault okay the outputs are same, so this fault is not detected by the random pattern 11 through this one through this net because the input is similar similarly for the last fault that is your I2 stacked 0 so this brings a stacked 0.

So what is say that for random pattern 11 the output at this net under normal condition is 0 as well as for this fault is also 0 the fault cannot be detected. But on the other hand the third file like O2 at 1 is 1 for the random pattern 11 near the output is 1. Whereas but for the normal case the answer is 0 so this fault is deducted by the random pattern 11 deducted at this. So that is what is simulation, but there is another output over here we have to see. So in this case this is the normal point and there is one point which is the difference okay.

(Refer Slide Time: 29:18)

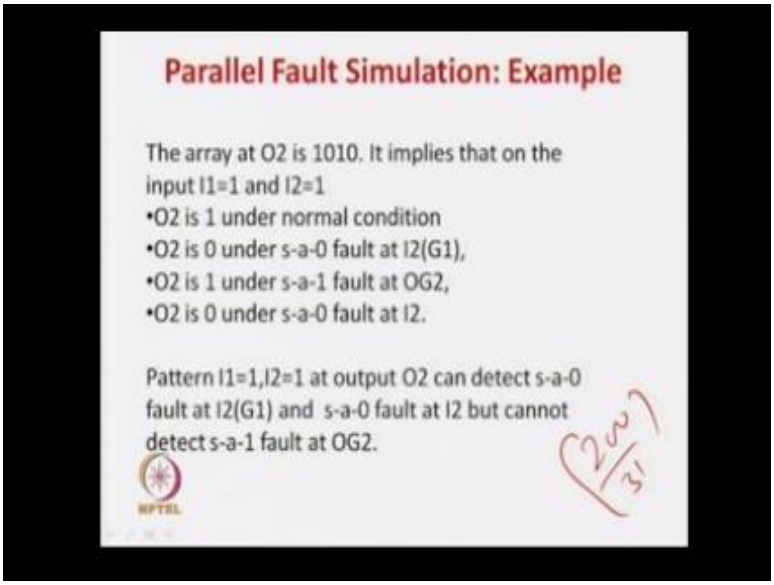


So this is different from the one, so this pattern this output of 1dedctes and this random pattern 11 we take this fault. So 1 fault is deducted now again this one more primary output you can see so in case you see this is the output of the circuit which is O2. When the random pattern is 11 it is 1. And there are two other faults in two faults this is t5he output is different. This is similar so if it is similar then at O2 this pattern cannot be deducted but this two fault I2G1 that is I2G1 pattern is for this one and this fault and last one is I2 second 0.

These two faults are giving answers 00 when the input is 11 and the circuit at the normal condition is 1 at O2. So this random pattern 11 deducts this fault, this fault at O2 and this for that O1 for the random pattern 1. So this experiment on for this algorithm parallel what it does, it

once can of the circuit it determines that random pattern 11. Takes this circuit, this at 1 for and this output and this one, this one at this, so what is the emphasize in monitor it is possible but if you know normal and random pattern procedure.

(Refer Slide Time: 28:46)



Then what you have to do then you have do this three times or three kinds of circuit is required. So on is for the normal then you have to apply 11 then you have to simulate this then you have to find this is different or this is different 1 will be different then you find and deduct and the fault is dropped. Then again you for simulating for simulate this circuit for this one and VPT. Then again you form single for this one and retain so three times this circuit scan is required. But here we did not require the three times of circuit scanning.

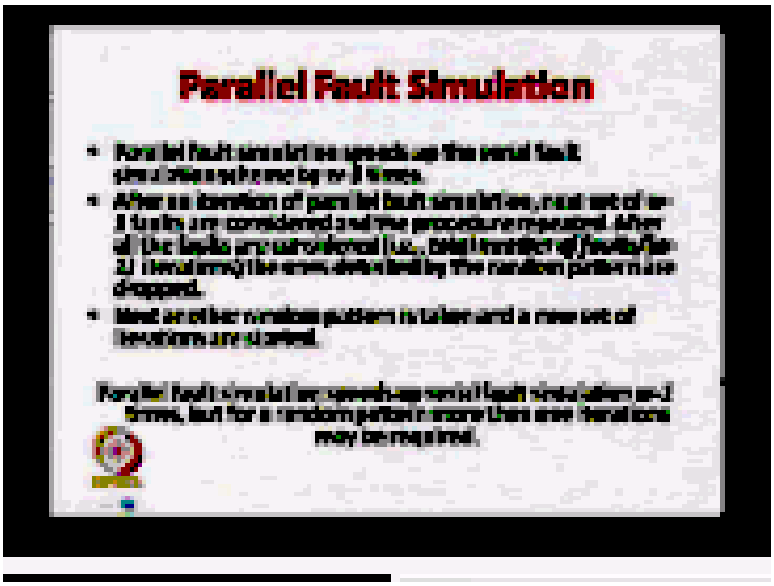
We have gone only by on parallel procedure and we have got the answer. That is one iteration we have got answer that three faults are deductible by random parallel one and when it is deductible then it also we have achieved. So this is what is parallelizing you serial fault simulation algorithm so that is this algorithm is very efficient but one thing you have to remember is that if a circuit is having a 200 faults still you have to repeat this procedure because

you can parallelize on up to that limit in at which you have the parallel beta ray or parallelism supported by your process.

So if your process to be processing parallel processing is possible then you can parallelize up to 32 bits. So 31 numbers of faults you can deducted in one move. So if you have till 200 faults then you have to rapt at least 200 by 31. So that mean number of times you have repeat the process. Here limited by the area size and area size is the maximum, which you can have, what you by called I mean the parallelism which is supported on the prospective. So again which you have discussed so this we presented in the area. The area O2 is the output then that means what it says that O2 is one under normal condition, O2 on the this fall, O2 is this fall and O2 is this one.

So you can easily find out this two falls are detected and this fault is not detected. What you have desire, we distal again writing showing in text. Similarly for oval this was the answer. This is the normal condition and so you can see these are the one point, which is the different. So this pattern, so pattern one and one at 01 deducts only the think first fast fall this fault. so this is the only fault, which is detected, which we have already discussed so just taken rewriting in the text form.

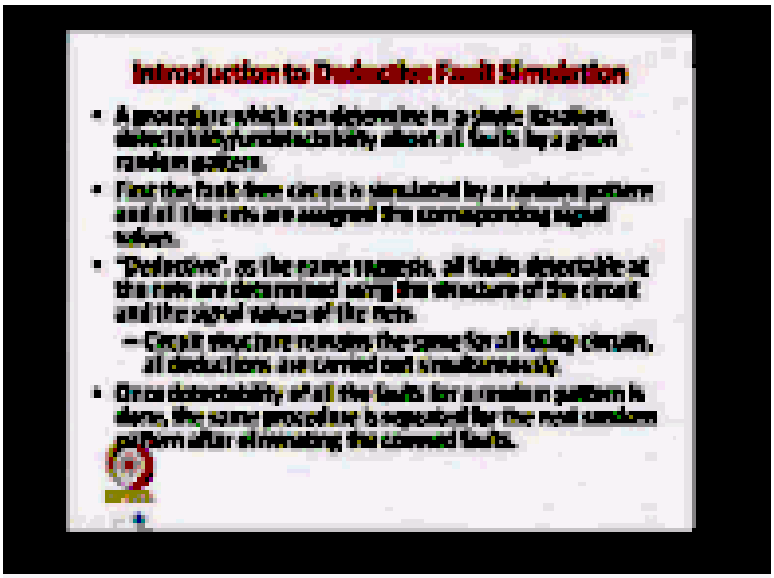
(Refer Slide Time: 30:57)



In summary what we have seen so we have seen that in fall simulation, which paralyze or speed up your fall stimulation procedure W-one time if you consider W is the bit wise parallelism, which is supported by our machine. So what is number of times you have to repeat is total number of falls by w-180 radiation again you have to take an upper sealing okay. So again if new pattern is apply so after that what you have to do so I mean first step apply one random pattern then you can try out your thinks for W-one number of falls and then again you have repeat it, repeat it how many number of times.

Total number of falls by W-1 upper sail and then when you have done to done this for fall number of falls that is for this value then what you can do, you can drop the falls, which you have been detected and then again you take a new random pattern then repeat the whole procedure. This is also very simple similar to serial falls stimulation, but only think is that you want to go you can listen about W- in number of falls so this way is parallelism you have obtained.

(Refer Slide Time: 31:59)



But now you see what happens so if your computer is very good, you have lot of parallelize possible then your answer you can get it very quickly, but on the other hand generally our modern computer have been very fast computer parallelize are generally not more than 32 or 64 with parallel because operations are possible.

So now we will see another algorithm, which is actually called detective fall stimulation in which in one go we can listen about all the falls in the circle whether computer has parallelize whether your computer lot has parallelize then does not matter that much. I mean beat voice parallelize does not matter will now handle it algorithm.

We will now handle it hard way level so the last this parallel false stimulation, we have taken, but you can say is that hardware advantage you have taken that is these process has beat twice parallelism, which can paralysis your effort so that advantage has been taken by the parallel false simulation algorithm, but now this is new algorithm which we are going to study the third false simulation algorithm in which case what we have seen we are not going to look into the we are not going to make a hardware dependent on what you called the hardware of the computer under architecture of the parallelism of the computer whether make a algorithm in such a way that in

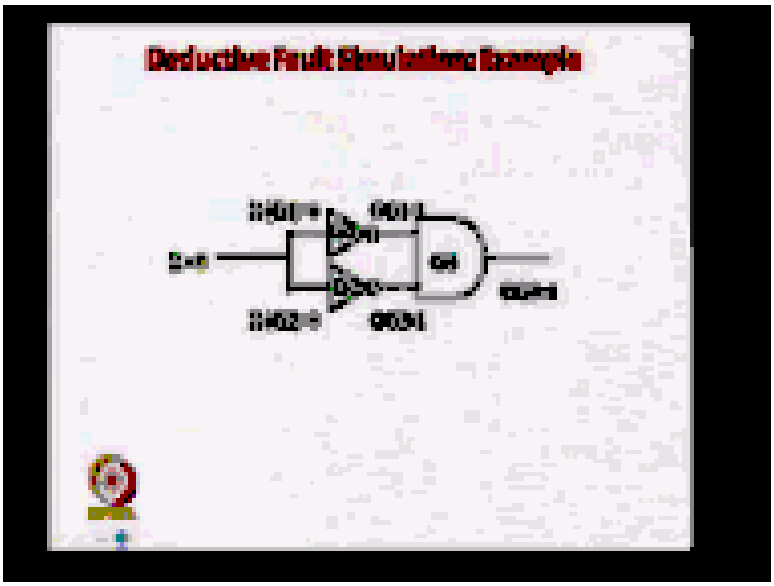
one gone listen in all the fault that is not mean 1st fault and then we are not going to use parallelism all beat level parallelism all those things into the picture.

So that is our main goal so now we are going to see detective false simulation okay so these are already told detectable name suggests at it can determining in a single iteration okay about the deductibility and un-deductibility about all falls okay in one random pattern that is there. We are going to apply one random pattern and then you are going to find out whether all falls detectable by that is what is very important.

So in this case what is happen basic idea here is so now what is the most equation in this case they say that the structure remains same for all the fault silicate that is generally it does not happen that for socket fault see more come from the year and they are going to be regenerated or because of some sockets some of the sockets gets vanished that is going to happen this say that the sockets structure remains same for all the fault sockets.

That is same in the items is number of gates or looking of the structural identify of the socket remains that is why they say that why we cannot find out in the algorithm, which can listen up about all the falls of the type in one go because if the structure of circuit changes then you have to go for the different creations of the circuit. So that is the basic philosophy that why this people have thought that this will idea algorithm so that should hold.

(Refer Slide Time: 34:56)



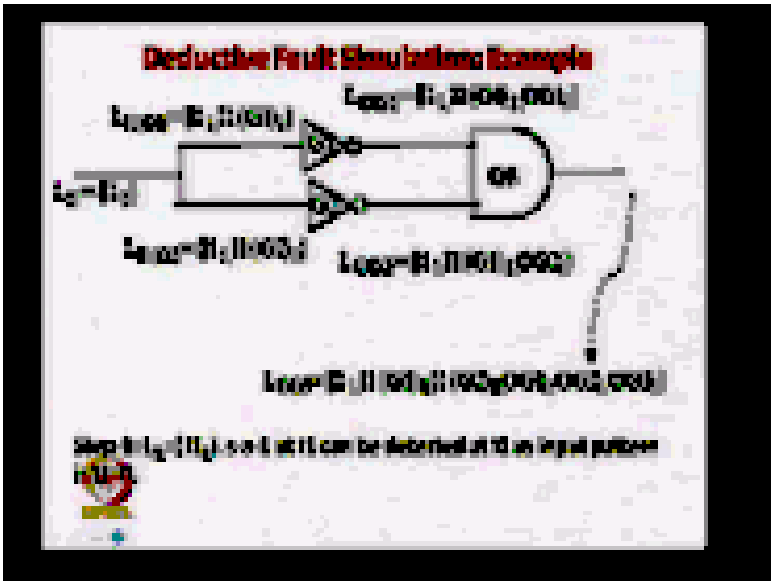
So let us see how it is going to happen so then what basic procedure will be will taken up pattern determine idea about all the falls in the circuit that which of the faults are detectable by the pattern and which are not and then again which will be detected will be dropped and then new renewal pattern will be taken.

So even if you are have 10000 faults in your surrogate and your computer has 32 beat paralyzed, but still your answer is going to be found only one of this surrogate. You need not go on 1000 or 10000 mode W-1 upper sally. Even if there is 10000 falls you can find this solution in one go it did not for this number of integration.

Now what is the philosophy says that the multiple integration should only be required if circuit structure is going to change for different falls, but here if the structure is same algorithm should hold so that is the idea of detective fault stimulation and detective as name suggest he says that did use about the falls which are detectable if one go for random pattern that is why name called detective fault stimulation.

So what you do here one thing we have understand we have taken very simple circuit this one and first you have given this random pattern 0 so if the random pattern I1z1 it will be 0 over here 0 11 and OG3 output is 1 that is very simple first are normal fault stimulation is done.

(Refer Slide Time: 36:16)



Now we are going to see for detective fault simulation of the circuit. So here one representation you should understand so if this lying I am saying as I one okay so then I will represent L I one that means fault at least of that net and when we say that I want underscore one that means what it is detecting I one socket zero that is for any net is named as I then LI is fault list that is forgiven random pattern, which are fault that has been detected at the line.

So that is LI that is falls detected at I net by our random pattern and then if you say that I underscore wants that mean I socket one is detected and if you say that I zero then I socket zero is detected that what is the notation we are going to follow okay. So initially just again you see Y one equal to zero has been given part to remember. So if you see that if zero has been given so obviously socket zero will be detected so we write L off I 1 equal to I 1 that is this list will have detect only one fault for the random pattern that is packet one so that has been return over here okay.

Now this one when are coming over here so in this net for the random pattern what are the falls will be detected so obviously socket zero will be detected here, you can also detect the socket zero fault over here okay as well as socket zero fault will also be able to be detected here. There is I1GI. Our socket zero one or socket one fault this net also be detected at this line. Why because your applying zero over here.

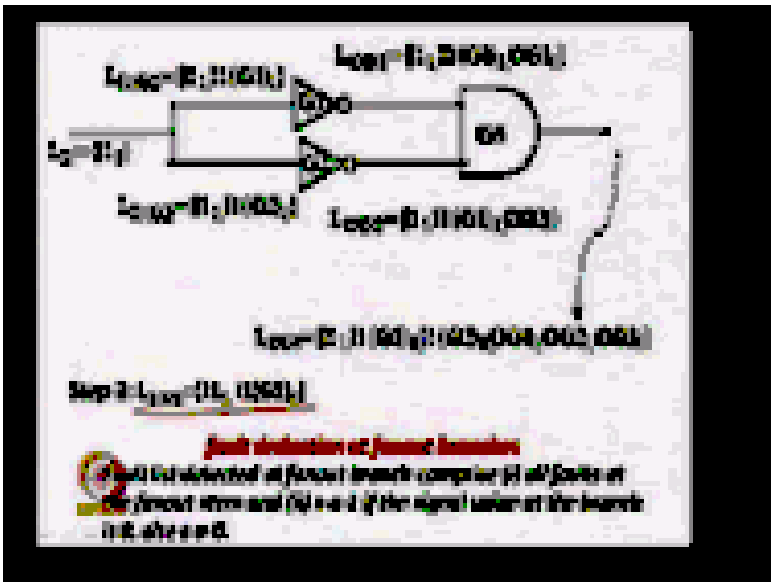
She is applying zero over here then if this may be started at zero so on socket one, the answer will be one in normal case also will be zero. Again your applying zero over here, if there is socket one over here the value of this net will be equally to zero in the normal case and in case of socket.

One it will be one over here so I1 and G1 that is this net will be equally to socket one fault here this thing will be here as well as any socket one fault will be detected over here so fan out, this value will be propagated here as well as a new fault is also detect that is this one, again for this fan out again obviously this one will be detected over here.

Because of socket one fault at this net is also detected here as well as socket one fault at this let that is by I1G21 one okay this is about second phone which is detected because of this random pattern zero. Okay so the lists and one more thing is obvious we have discussed many times that our socket one fault here is not detected at this name similar fault is not detected at this line so okay.

It is I1G11 and it is I1G21 so socket one fault here is detect at this net and socket one yet of this net is detected at this. So you see this list is to and this too okay. So this what we are getting for this fault list that is what your step one it is saying that this one so it is step 1 is saying that socket one can be detected at random pattern this one is sealing out this one.

(Refer Slide Time: 39:53)



So this is about step 2 so step two is this one, which already discussed about this similarly, you can get for this one. For fault detection one important rule has to be concluded so fault detection at fan out branches so what is that fan fault list is detected as fan out branch is all faults at the fan out stay so whatever fault list is here those things you can detected that is obvious as well as stacked one fault if the signal value of the branch is view else one and zero.

So if says that if there is fan out branch like this and so whatever fault list here can also be detected in this fan out as well as that along with that you have to make any of if the signal value is one then you can detect as socket at zero and if the signal value here is zero then you can detect a socket one over here so this is the detective falls simulation rule at the fan out so detected falls simulation says that all fall at the fan out same can be detected and socket one if the signal level are the branch is zero.

Okay else our socket zero fault will be detectable if signal value is obvious signal is one that is obvious. So this is the rule if you have understand about this detective fall stimulation is all about rules. So this is simple rule we have learned about the fan out branch. Okay now next is now we will go next step.

We will see what is the number of fall detect or what are falls that is detectable at this point. Now you see so what happen so this is a we have applied zero over here so now you see so if it is socket at 1 over here then the answer is 1 normal case zero, fault case 1, similarly for this pattern 1101, so that is we are applying zero fault over here.

For fault detection one important rule has to be concluded so fault detection at fan out branches so what is that fan fault list is detected as fan out branch is all faults at the fan out stay so whatever fault list is here those things you can detected that is obvious as well as stoked one fault if the signal value of the branch is view else one and zero.

So that is instead of doing some between operations or before this thing here we are having some detective fault so those ever using this fault rules we are populating slowly when our fault list, which can be detected here and finally we will go to the primary out then whatever will be fault list in this case are the faults which are detectable by this random pattern at this output so what will have any one go will be able find out all the faults, which are detectable by this random pattern so we will just see so still now we have seen the rules fault, this fan out branch so fan out branches investors now we will see for the hand gait, which is going to be very important.

So now what are the fault list we have this are the fault list for this net. This is fault visit for this net and then you have to find out the fault list for this now let us try to find out did use the rules of a inventor. So whatever fault will be listed just see the propagate here it is deductable as well as along with that that union it just propagate here it just propagate is union is signal value is one in case of normal and then you get a socket 0 you can be take and if is a 0 in case of normal then socket 1 can be deducted.

So let us find the rules for a this is for inventor for the AND gate so we just see for an AND gate so this is the list so you have to now find out for the AND gate what are the fault list for AND gate on detective rule for and now you see so this is now we are played as 0 over here correct and so now this is their socket one so if the socket one is here means normal case 0 fault one

normal case 0 fault one this is what you called 10 so in this the answer is going to be 1 or 0 so obvious this socket 1 fault is detectable by this AND gate correct.

So that is why in this list we are having this one. Now let us see for another fault one fault so we are applying zero over here so it is zero over here one over here in this case what happen we are applying a socket one over here correct. So now it was zero so now it has become one so it is zero one so it is one zero and in case the answer is one zero so okay in the normal case the answer is one in the fault case the answer is zero so this socket one fault is also in this list okay now let us see if the other faults of other net that is other part of this fan out branch so this is other plate zero over here.

So here we are having socket one so here it is zero answer is 1. Here is it socket one that means normal zero fault one 1 zero then it is one 0 so fault is again detected. Right so that is 12z1 is also at this list so now let us see for the fault list over here so now for the fault list over here is here this one and one let us see the socket zero fault over here so that is on zero normal one so it zero and one again fault.

Because of the socket zero fault is zero in the normal case let me know one here the answer is 1 by the normal case so it is this one in the output and again this socket zero fault is detected over here similarly you can easily verify that if have socket. We will now handle it algorithm. We will not handle it in the hard way level so the last this parallel stimulation, we have taken, but you can say is that hardware advantage you have taken that is these process has beat twice parallelism, which can paralysis your effort so that advantage has been taken by the parallel false simulation algorithm, but now this is new algorithm.

So whatever is happening in this can be reflected over here is very good i think about and get that this line is one and all the activities at this case get prepared here, similarly if this is out one and if this is one then also the other case so if this one then again all the activities at this point get propagated, but if it is zero then whatever happen here it does not get pass through similarly for other input of the that is in the case of iron gate.

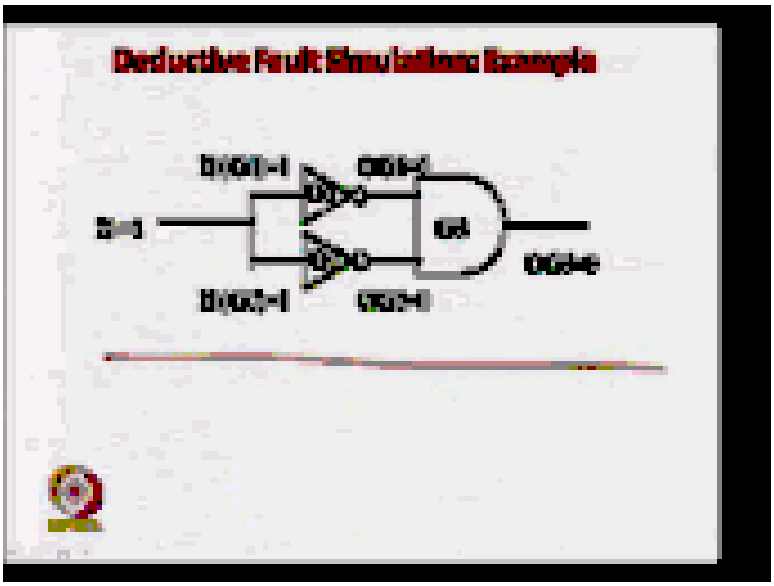
In this case so if this is one and this is two so if there is fault or some activity is here so this can be propagated here and you can observe it similarly as this also one so any type of activity fall detection probation etc at this point also will be reflected where here so is very helpful situation when if both the inputs of the gaits are one then what happening is that if this one so whatever is here will be propagated over here.

This is because this input is one so and what you mean by I want one here it says that the effect of this socket fault here socket fault is available over so this is one means the available fault effect of this fault socket one is available is here. This fault socket 0 for effect is available here and as this is one so the whole thing can be propagated over here.

So that is why whatever is available over here will come here. Similarly you also have a one over here okay so that means any effect so anything any fault effect here will be propagated here because this is one so this one here means that is socket one fault is effect can is available here so this is one is there means the socket one fault here affect is available here similarly this one here means this affect of socket zero here is available in this so.

As this one is one so the whole list will be available here. So this thing will come out along with that here the value is normal in this case it is one so the socket zero will also be detected. So if says that if there is fan out branch like this and so whatever fault list here can also be detected in this fan out as well as that along with that you have to make any of if the signal value is one then you can detect as socket at zero and if the signal value here is zero then you can detect a socket one over here so this is the detective falls simulation rule at the fan out so detected falls simulation says that all fall at the fan out same can be detected and socket one if the signal level are the branch is zero.

(Refer Slide Time: 53:51)



The list here will be propagated and for zero 1 the list here will be propagated and for 00 nothing will be propagated correct something like this, but it is not the case for let us see for 001 what gets propagated and then other case zero one and one zero are bit obvious as you have seen so we will see so it is not case that for zero 0 nothing will be propagated.

Something will be propagated we will see because by logic if both of them are 0 0 I mean this is blocking everything then this also blocking everything then nothing will be propagated over here accepting nothing will be propagated over here accepting only a socket one fault will be detected here the output, but this is not the case, but both input 00 also something gets propagated which is interesting to see and for 01 and 10 with straightforward so that we are going to see in the end.

Okay so now this input is 1 so here it is 1100 so the both input are zero and output is zero normal case so we will see what happens so if it is zero so obviously socket input is one here. So socket zero fault is detected so think is propagated here. This think is propagated here as well as it is zero it is 11 normal value so socket zero fault here and socket zero fault at the dates are detected so this is in this list.

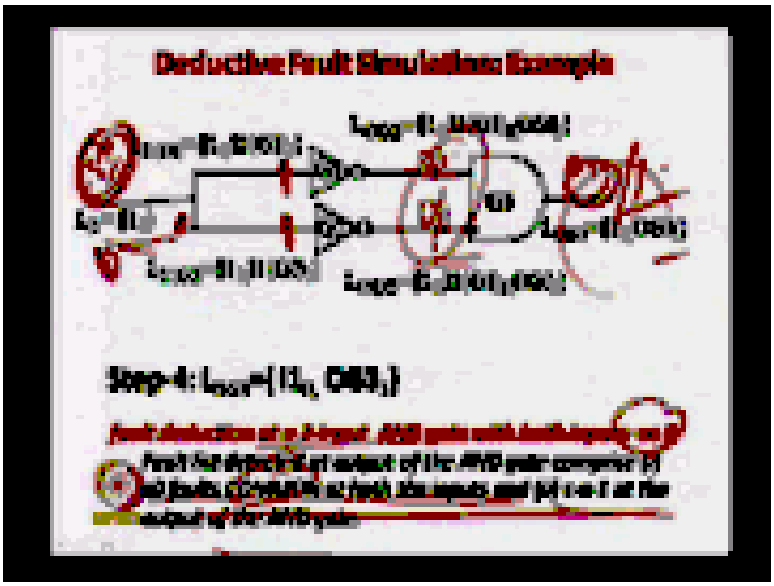
So this type 1 is far this one okay now step 2 says that these are two values this two fan out net, which is very simple because your are applying one over here so socket zero fault is detected here it is again 11 so if says that if there is fan out branch like this and so whatever fault list here can also be detected in this fan out as well as that along with that you have to make any of if the signal value is one then you can detect as socket at zero.

And if the signal value here is zero then you can detect a socket one over here so this is the detective falls simulation rule at the fan out so detected falls simulation says that all fall at the fan out same can be detected and socket one if the signal level are the branch is zero then we can format the rule for and get when both inputs are 00 so we are applying a one over here so this is socket zero fault is detected so 10 1 0 this is case this is zero 1 correct now you see what happens normal case 00 fault case this is 11.

So now you can see for this fault we are studying so the answer is 0 1 and this fault is detected so important interesting thinking is happening is that and the fault that is propagated so even if the value is 00 under the normal case, but still this socket zero fault here is detected now one important think you have observe that zero is blocking this zero at this input and zero is input at this one is totally blocking the floor here and obviously these thinks are not propagated and these thinks are not propagated that is true.

But only think has been propagated this is because this socket zero fault is effecting both this and it is effecting both input in one room that is why it is propagating the output and it is getting detected. So if says that if there is fan out branch like this and so whatever fault list here can also be detected in this fan out as well as that along with that you have to make any of if the signal value is one then you can detect as socket at zero and if the signal value here is zero then you can detect a socket one over here so this is the detective falls simulation rule at the fan out so detected falls simulation says that all fall at the fan out same can be detected and socket one if the signal level are the branch is zero.

(Refer Slide Time: 01:20)



Then obvious the whole fault list from this input and this input should be out as well as will be here as well as sockets zero fault will be taken into picture and for this both input 00 it is very interesting it is not that nothing will be propagated and only one socket one fault will be there, but if the some fault which is effecting both lines in common then that fault will be propagated here all other thinks will be and one socket one fault will be taken.

So there are two other combinations of 01 and 10 which will see tomorrow lecture and then will make a table kind of think that will give you the rules for detective falls simulation for AND gate OR gate NOR gate so whenever you have the circuit you can apply the rules and you can go for a false simulation by detective mechanism and in one go you can find out which are the false which is detective by a even in one shot okay following that will see another false simulation algorithm in the next class which will be finishing our discussion. 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 Debnath