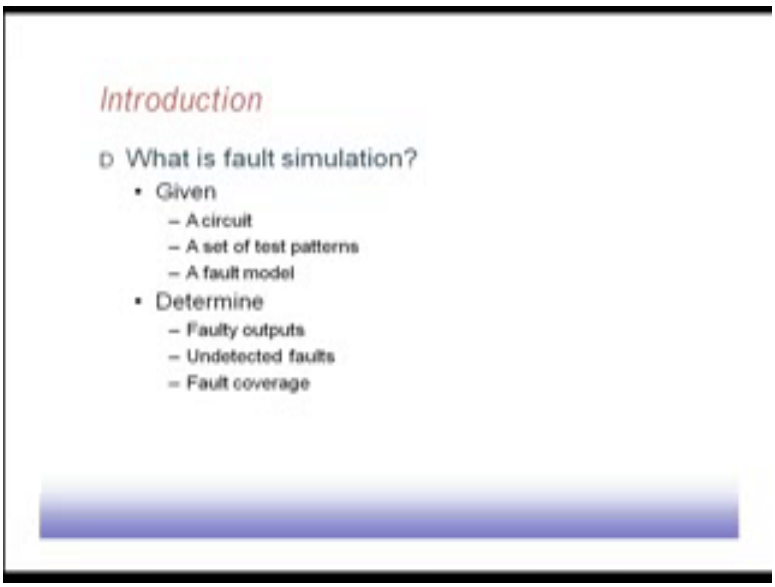
Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
Logic and Fault Simulation (Contd.)

We will see, what is fault simulation. So, it is ultimately a soft piece of software. So, may be nothing to do with the actual hardware part of the circuit. This is software which will be doing some simulation of the circuit in the presence of some faults.

(Refer Slide Time: 00:25)



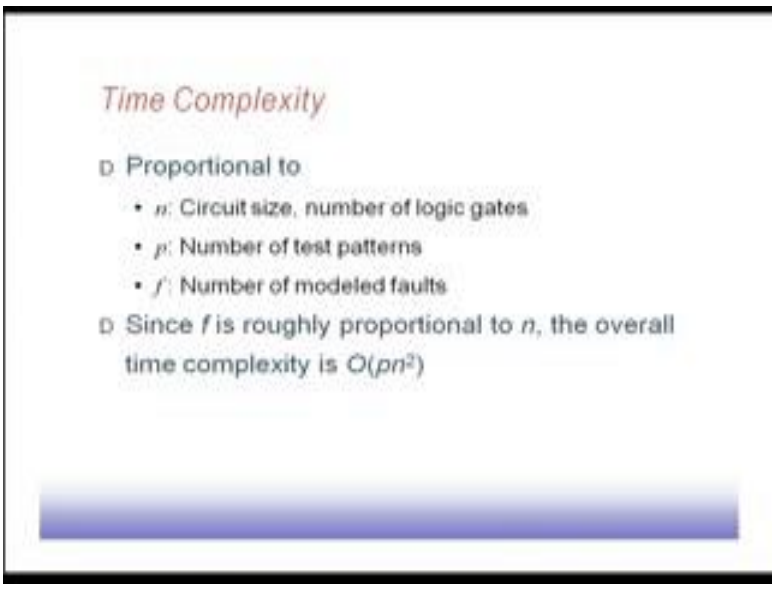
So, as an input, it has got a circuit description. So, you can say circuit description in some level may be gate level, may be model level, may be RTL level. So, it is some description of the circuit then a set of test patterns. So, this is given as input to the system and a fault model. So, like a set of test pattern may be I randomly generates a 100 test patterns called a circuit. So, if I have got n input circuit, any n bit random pattern is, it can be considered as a test vector for the circuit. So, if I have got a say 100 such pattern. So, that becomes a test pattern set then a fault model. So, we have said that every defect in the circuit. So, that is modeled as a fault.

What is the fault model? Like stuck at fault model, like single stuck at fault model or multiple stuck at fault model so, are you considering delay model; delay faults like bridging faults. So, like that you have fault model has to be specified and then what the

fault simulator will do it, will determine the faulty outputs like under the presence of fault of a particular fault where each primary outputs of the circuit differ from their correct responses. So, that gives the faulty outputs for every test pattern and for every fault and undetected fault there may be certain set of faults which are not detected by the given test pattern set. So, that is the set of undetected faults and once we have, once we know how many such faults are detected by the circuit by the fault test pattern set.

We can assign a score to the test pattern set which is known as fault coverage. So, naturally our objective will be that we will be using as small a test pattern set as possible at the same time we would like to have a very high fault coverage these 2 requirements are contradictory, but we will see how these are actually achieved when we go to the chapter on test pattern generation, but for the time being we are talking about determining the quality of a test set.

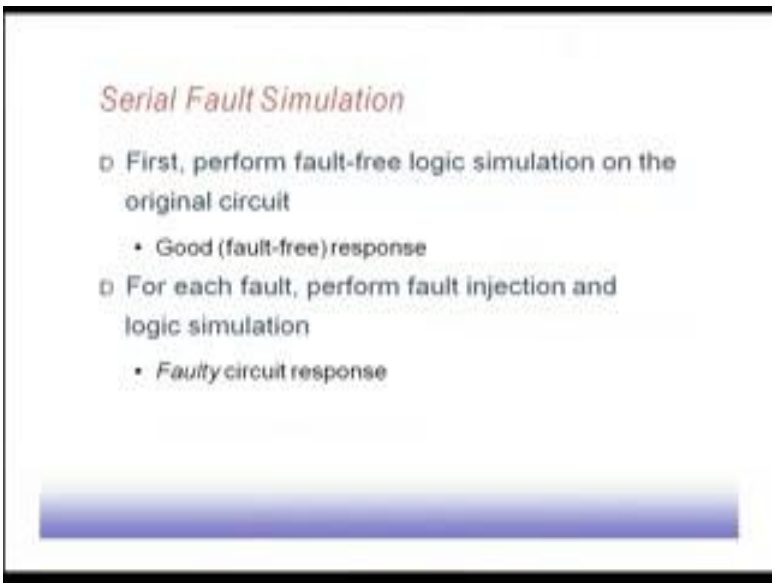
(Refer Slide Time: 02:46)



What is the complexity of this fault simulator, it is proportional to n; the circuit of the circuit size terms in terms of logic gates s, because every gate has to be simulated for every test pattern. So, they it will depend on that then p is the number of test patterns and f is the number of model faults. So, it can be shown that the size say f is number of faults being proportional to n. So, if we are considering single stuck at faults then if I have got n lines in a circuit then there are 2 n, 2 into n possible faults in the circuit.

That way the f becomes proportional to n. So, the overall complexity is n into f n into p into f and f if you replace by twice n. So, it becomes we go p n square of the order of p n square. So, as you are increasing the size of the circuit the complexity of fault simulator that will go up significantly as we are increasing the number of test patterns then the fault simulator type increases in a linear fashion.

(Refer Slide Time: 03:53)



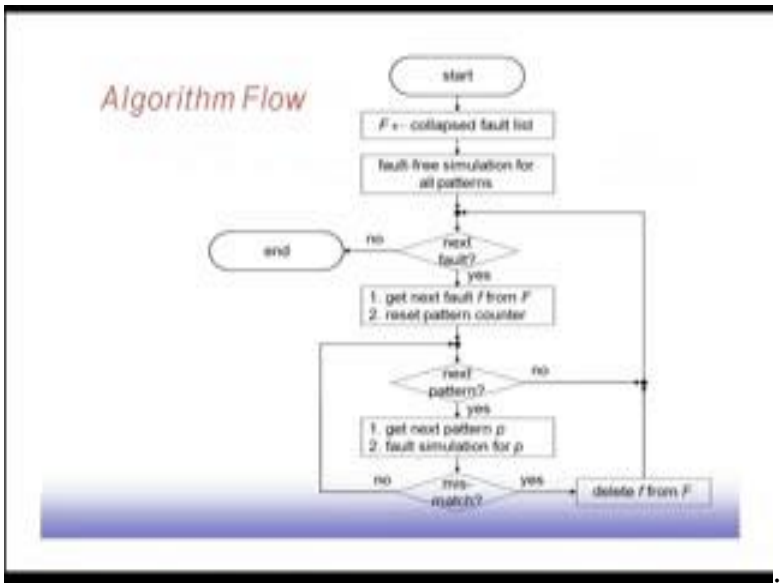
The most simple fault simulation technique is the serial fault simulation. So, what we do in this case we first perform a fault free logic simulation on the original circuits. So, we have got the test pattern set for the test pattern set we want to determine, what are the fault free responses from the circuit? So, these are called good or fault free response.

Then for every fault we perform fault injection and logic simulation. So, we perform fault injection means we modify the circuit description as if that particular fault has got in inserted into the circuit and then we again simulate the circuit. So, in the presence of that particular fault what is going to happen? So, that is the logic simulation after doing the fault injection. So, that will give us the faulty circuit response now there are 2 possibilities that for certain faults for certain patterns the faulty response may be same as good response; that means, those patterns are unable to detect those faults.

For if it happens that for all the primary outputs the patterns response is same as the faulty response then those faults that particular fault cannot be detected by their test pattern. Now if it is detected by some test pattern then naturally this faulty circuit

response has to be different from the good circuit response sometimes that they are called gold n response.

(Refer Slide Time: 05:22)



This is the way this serial fault simulation will work we start with f; f is assigned as the collapsed fault list. So, we have seen that when we do a fault collapsing by taking the only the non equivalent faults. So, for every equivalency class, we take only one particular fault. So, when we do this, we get the collapsed fault list and if the collapsed fault list find out like it is often leads to reduction in the number of faults. So, if number of faults reduces then complexity of fault simulation will reduce that is why start with collapsed fault.

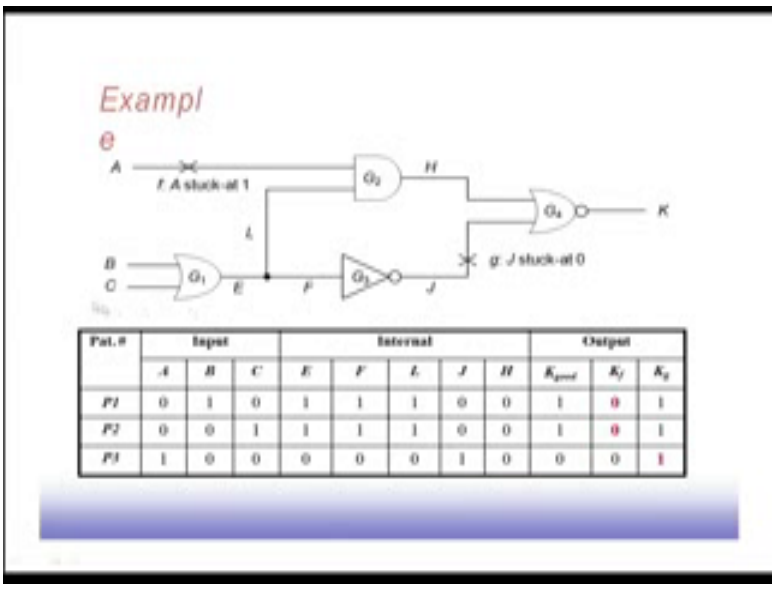
Then we will perform a fault free simulation for all patterns. So, test pattern is coming as an input. So, do a fault free simulation. So, then we pick up the faults one by one from the collapsed fault list. So, we will pick up the next fault. So, as long as there are faults in the list. So, we try to get; we get the next fault f from the set of collapsed fault list and then reset the pattern counter. So, we basically for this fault we again start from the very beginning of the pattern set if my pattern, if my pattern set has got hundred patterns then for all these hundred patterns we would like to simulate under the presence of this particular fault.

We will see then we see we get then next pattern and then we get the next pattern p and do fault simulation p after doing the fault simulation if we find that there is a miss match

miss match in the sense that the fault free response and the faulty response under the presence of this particular fault is different for pattern p if it is, if there is a; if there is no mismatch; that means, this pattern could not detect that fault. So, we try with the next pattern; however, if it can detect that fault, we delete f from the set of collapsed fault list because for this fault, we have identified we have got 1 pattern that can detect it. So, there is no point in keeping it into the set f because ultimately we are going to see whether the given test pattern set is covering the collapsed fault list or not. So, since this fault small f is already detected. So, there is no point simulating it again for the remaining test pattern that is why this is deleted from this set f deleted from the set f and then we try with the next fault. So, this way it continues.

This is a very simple algorithm, but time consumption is high. So, this may be an example like say we have got this say this particular cyclic.

(Refer Slide Time: 08:00)



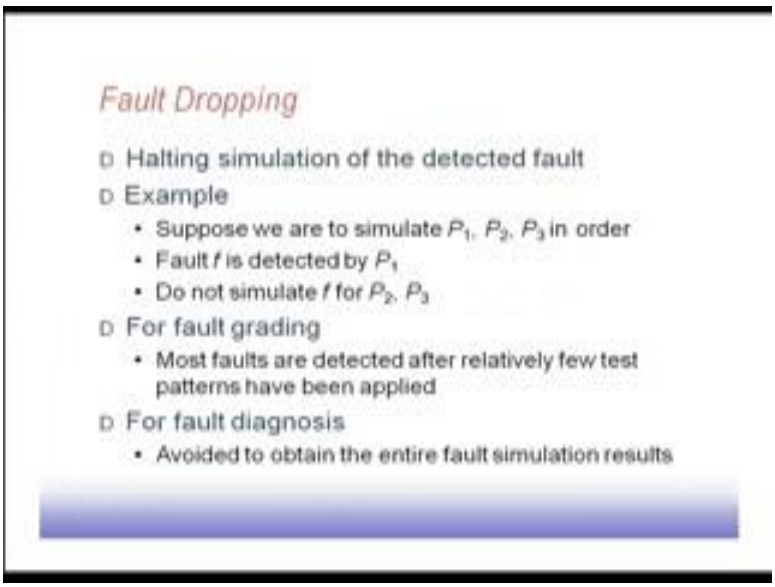
Pat.#	Input			Internal					Output		
	A	B	C	E	F	L	J	H	K_{good}	K_f	K_g
P1	0	1	0	1	1	1	0	0	1	0	1
P2	0	0	1	1	1	1	0	0	1	0	1
P3	1	0	0	0	0	0	1	0	0	0	1

We want to see that whether we are; we can get this thing that is f stuck at 1 and G stuck at 0. So, these 2 faults, primary input pattern, we have got 3 patterns, P 1, P 2 and P 3 and for then this A, B, C value this is P 1 is 0, 1, 0, P 2 is 0, 0, 1 and P 3 is 1, 0, 0. So, for first one is the; for the good circuit what is the response? So, we simulate this circuit for in the absence of all these faults F and G and we see that this K good is 1, 1 and 0 for the 3 pattern.

Now, if we assume that the fault F is present in the circuit then what will happen? So, these lio in the absence of fault line H was equal to 0, now this line is stuck at 1. So, this is this line is 1 and this B and A, C, B and C both are at least I think one of them is 1. So, C line is B line is 1 for the first pattern. So, this OR gate is 1, as a result this H line becomes 1. So, this k becomes 0. So, under the presence of fault F, the line K, we suggests the value 0. So, there is a difference from the good response. So, this is good response is one and this is 0. So, we say that this pattern P 1 can detect this particular fault f also when we try with pattern 2 we see that the response is different from the good circuit. So, good circuit response is one bad circuit response is 0. So, with that way there is a difference in this. So, that way we have got a fault faulty response for this fault as well.

For pattern P 2, either P 1 or P 2 can be used for detecting this particular fault, for P 3 also we see that the good response is 0; however, P 3 could not detect this particular fault because the faulty response for P 3 is also 0. So, it cannot detect this fault; however, these it can detect the fault K g. So, K g is 1, you see one unique thing about P 3 is that P 3 out of these 3 pattern, P 3 is the only one which could detect the fault G whereas, this P 1 and P 2 could not detect this fault G because they are faulty response is same as the good response. So, I can, if I have asked to do a choice from this pattern say P 1, P 2, P 3, for detecting the faults F and G. So, I definitely had to include P 3 in my set whereas, from between P 1 and P 2, I can have only one of this.

(Refer Slide Time: 10:49)



Next we will discuss about a concept called fault dropping. So, basically we want to fault simulation of the detected fault. So, that is the thing known as fault dropping like in this particular case we had to simulate suppose in this previous example that we are considering because having 3 patterns P 1, P 2 and P 3, suppose we are simulating in this particular order P 1 followed by P 2 followed by P 3 and fault F is detected by P 1. So, it is not necessary to simulate P 2 and P 3 for F again because we know that fault F is already detected. So, this is known as fault dropping. So, we drop F from the simulation part simulation further simulation of patterns.

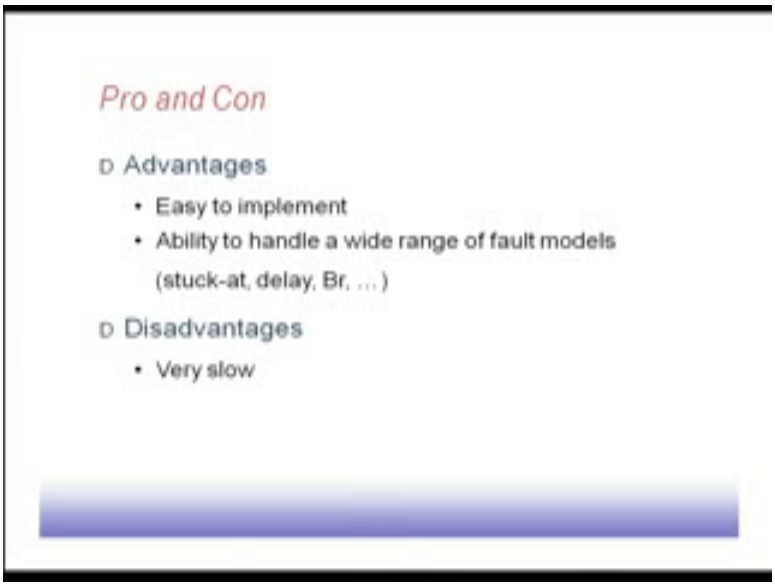
Now, most faults are detected are relatively they are detected after relatively few test patterns have been applied. So, this is a general observation like if you have a have a circuit and if you see that the test patterns are successive test patterns and their detection of fault initially as you as you increase the pattern size, pattern set size. So, your fault coverage increases rapidly because many of the faults that we have in the circuit. So, they are random pattern detectable faults because that they can detect. So, any random pattern can detect those faults. So, we will come to this issue whereas so that way this; your possibility of this fault coverage. So, that increases when we have got this thing now when we have another port another part where we have got this large number of patterns now for remaining patterns. So, it may not be very much effective in detecting many of the faults.

For fault, so this fault dropping will help us because those faults are already detected. So, we do not simulate them further; however, there is another application of this fault simulation which is known as diagnosis. So, diagnosis is actually a process by which after we have done the manufacturing and we have passed it through a test session suppose the chip fails for when the chip fails we collect all the faulty responses and try to see what may be the possible reason for this failure now you see if you have got only one pattern detecting a fault and we just note that for that one only then due to this equivalency of faults what can happen is that this particular fault may be detected by many other patterns and this particular pattern might have detected; could have detected many other faults.

Now, a better situation that can help us in pin pointing the exact location of fault is that if we have got a full history like if we have got for the entire test pattern set which are the patterns which have failed. So, if we know that pattern number P 1, P 10 and P 15 have

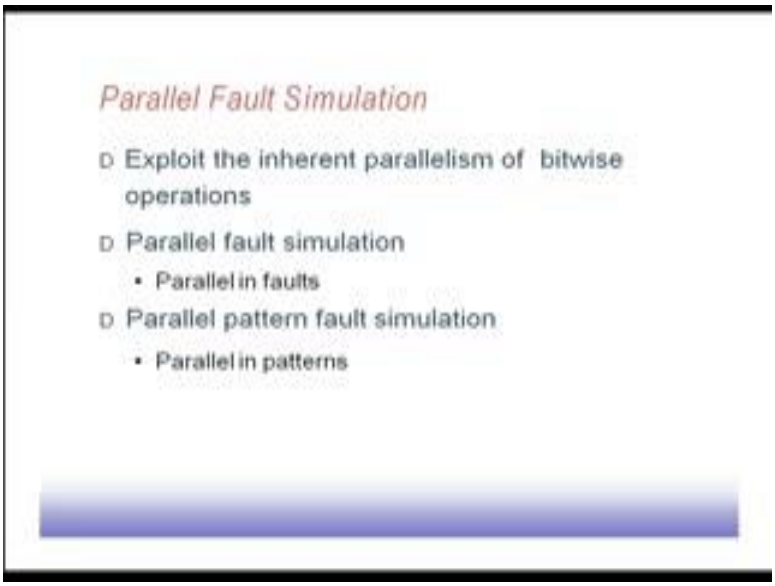
failed and if we know all the faults that could be detected by P 1 P 10 and P 15 then we may try to find out what is the common between all these faults as we depending upon that we can try to get some confidence about the location of the fault. So, that way this fault dropping is good for fault simulation.

(Refer Slide Time: 14:26)



But when we are looking for a diagnosis type of application this fault dropping is avoided. So, serial fault simulation it is easy to implement as we have seen the algorithm is very simple and it has got ability to handle wide range of fault models like stuck at fault, delay fault, Br fault. So, like that which has got ability to handle wide range of faults disadvantage of course, it is very slow because it is of the order of P into n square. So, that way if the pattern set size is large or number of lines in the circuit is large then it will take lot of time for doing the simulation.

(Refer Slide Time: 15:00)

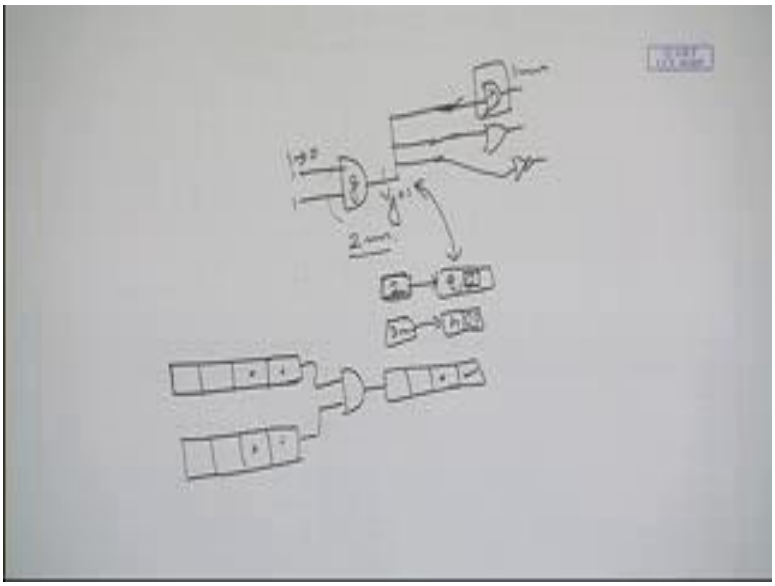


People started looking for faster version of this fault simulation process and accordingly many other simulators have come up.

One of them is known as parallel fault simulation. So, again the same thing as you are talking, this logic simulation part that we have got this bit wise we have got this word size is of this the processor or computer on which you are running the fault simulator. So, possibly you can do some parallel simulation. So, you can; if you have use this word of a computer to represent many possible status of the signal lines in the circuit and then may be if I have got say 32 bit word then I can do it like this that for 32 different status of a single line can be put into a word. So, that way it can help us in doing the simulation process.

Now, there are 2 variations that are possible one is known as parallel fault simulation another one is parallel pattern fault simulation. So, these are the situation like if I have got say a w bit word if I have got in my computer suppose it is a 4 bit word.

(Refer Slide Time: 16:16)

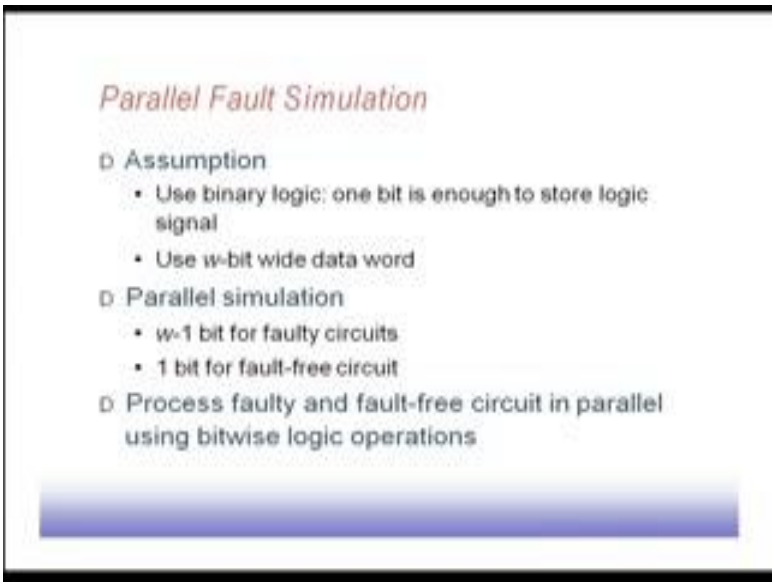


Now, for every single line that we have, we have got this particular word available, now suppose this is 1 AND gate. So, it has got 2 input and with both the inputs we put these 4 bits, similarly at the output also we have got this 4 bit space. So, the idea is that if the first bit will be computing and of this and this. So, if these 2 are ended, the results will be stored here.

Similarly, when these 2 are ended result will be stored here, by being a bit wise parallel operation. So, I can do this and operation of 4 bits simultaneously, now what do we represent in this 4 bits. So, that is one important thing. So, one possibility is that in parallel fault simulation. So, we represent this individual bits different faults for the same bah same line if in the presence of say 4 different faults what are the values of these 4 what are the values of these line?

Accordingly we get these 4 different values for the for the same signal line. So, that is the C O D of this parallel fault simulation, another one is parallel pattern fault simulation. So, here we are simulating for a single fault, but I take 4 such patterns and combine them into one junk and do the simulation together. So, both the versions are available. So, we will see them one by one.

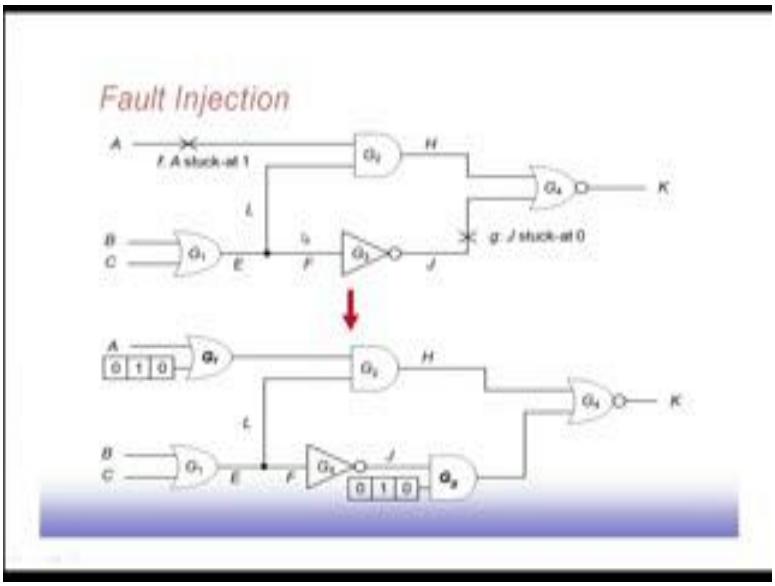
(Refer Slide Time: 17:47)



First one is the parallel fault simulation, it is assumed that we use binary logic that is one bit is enough to store logic signal. So, we are not looking into ternary or this higher order logics, but only binary logic.

If we have got another w bit word data here then for parallel simulation w minus 1 bits are dedicated for faulty circuit and one bit is dedicated for fault free circuits. So, what will happen? So, I will have w minus 1 faults simulated simultaneously and the first bit is taking the fault free circuit. So, that way we can process faulty and fault free circuits in parallel using bit wise logic operation.

(Refer Slide Time: 18:34)



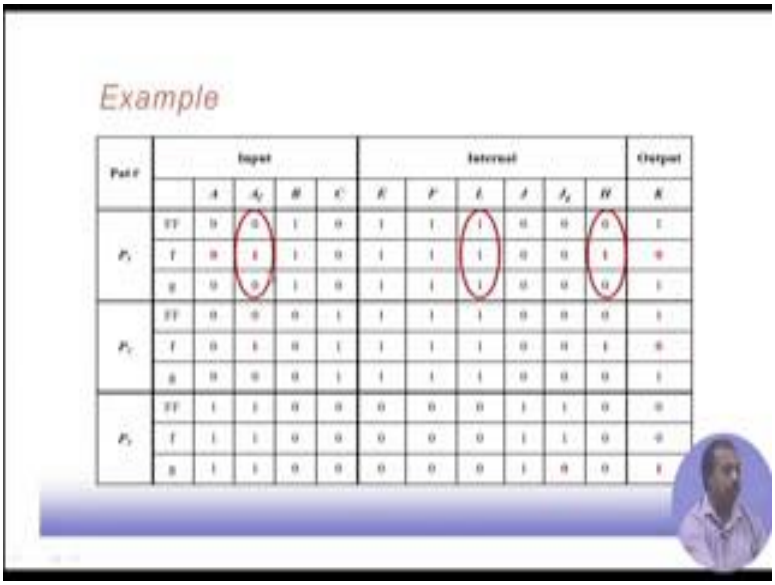
This is an example. So, you see, this is the original circuit. So, which has got 3 inputs A, B and C and what we do? So, we have got 3 A B line, we associate the 3 bits one is that we have got this for line A, we have got the good signal good circuit value which is 0 then we have got if the slide is stuck at 1, if A stuck at one then what is the value. So, that is there and the other fault is G which is J stuck at 0. So, if J is stuck at 0, what is the value of this line A? So, what we do? this particular fault, we replace as by an OR gate where 1 input is A and the other input is 0 for good circuit one for the fault F stuck at 1 and 0 for the G stuck at 0 fault.

Now, you see that when we are simulating for the when we are evaluating this or gates since a gate in good condition a line in good condition the value is 0. So, this point you get 0. So, that is actually circuit remains functionally same as the previous one under fault free conditions whereas, in the fault free condition. So, this one, F stuck at. So, this will become this OR gate will become 1. So, that way we are getting the value that F is stuck at 1. So, that way it is actually model. So, this new gates are introduced G F and G G into the circuit; into the logic circuit, we are introducing this new gate G F and G G and we have got these 3 bit words holding the good circuits and the F fault and G fault and the consequence.

Now, you see for B and C. So, you have, now, in the circuit, if we apply different patterns so you can simulate the circuit in a logic simulation fashion and get the

responses computed. So, this 3 fault, one good circuit and 2 faults, these 3 circuits are getting simulated simultaneously.

(Refer Slide Time: 20:42)

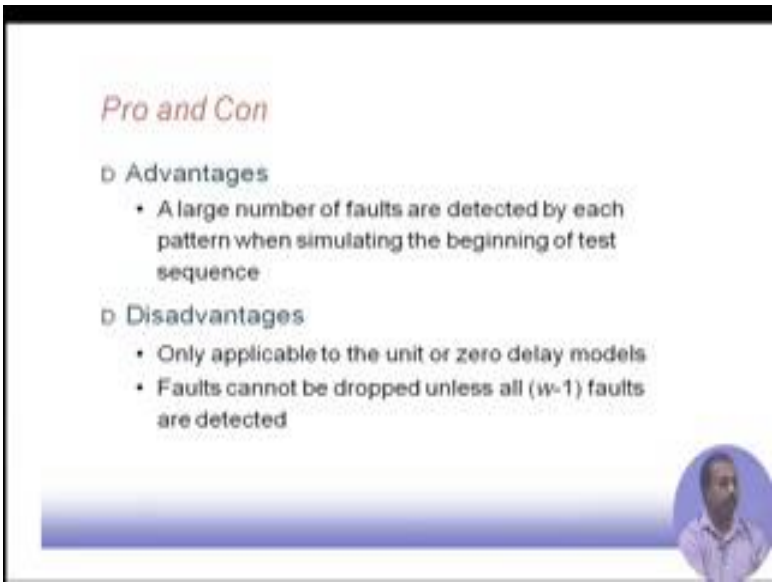


So, this is the situation that is suppose if I have got a pattern P 1. So, pattern P 1 under fault free condition A is 0, b is sorry, A is 0, B is 1 and C is 0 that is the fault free condition.

Now when we simulate, so, these under this faulty condition A will become equal to 1. So, under fault under if under fault F, fault F A will become equal to 1, this will become equal to 0. Now when you are computing say E, say these value say H. So, when we are doing this thing. So, L we are directly doing, we are directly doing bit wise ending 1. So, that way it is getting the internal value. So this is, you can see that by doing a bit wise operation, we can directly simulate the fault, we can directly simulating the circuit for a particular pattern, but for 3 cases, fault P, fault F and fault G, similarly for pattern P 2 again we can do a simultaneous simulation of fault free fault F and fault G.

Ultimately the amount of effort will be reduced much because many of the operations are being done parallelly.

(Refer Slide Time: 22:10)



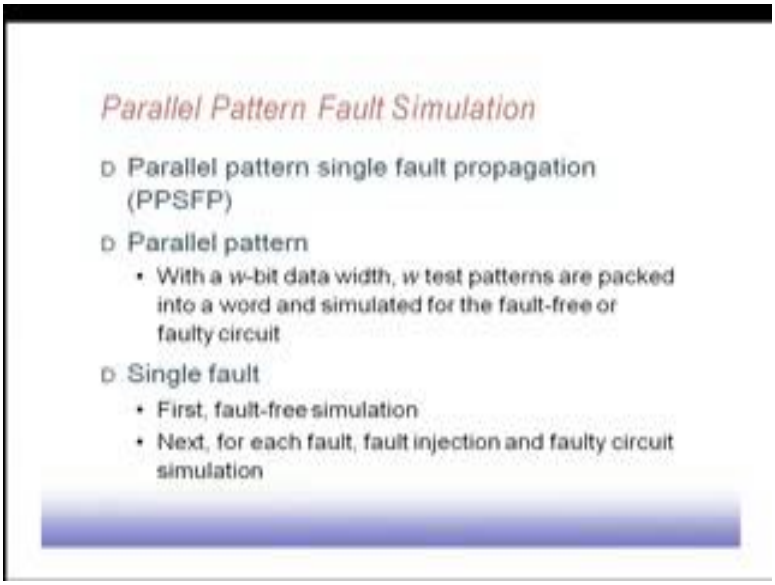
Several faults are getting simulated parallelly, advantages we have got a large number of faults that are detected by each patterns when simulating the beginning of the test sequence. So, as a result what will happen is that since we every simulations since it is handling a number of faults instead of single fault in case of serial simulation or series fault simulation.

Now fault dropping may be very effective like very easily many of the faults will get dropped. So, look at this. So, because of this parallel fault simulation, each pattern is simulated for directly all the faults in 1 sort. So, that way the fault dropping will be higher disadvantage it is applicable for unit or 0 delay model. So, unit delay model means all get delays are assumed to be equal to some unit value. So, that way it is under these 2 models only say we can do this thing otherwise we cannot we cannot take care of these different delays because then the structure will become more complex.

Second important problem that we have is that the faults cannot be dropped unless all w minus 1 faults are detected. So, because the structure of the computer of the processor word that we are using. So, there itself we are putting w minus 1 faults. So, if there is no point dropping a fault here because anyway you will be doing w bit logic operations over this bit patterns. So, that way if you cannot you cannot really help in dropping the fault. So, if all w minus one faults have been detected then you can drop it then you can say I

will construct a new word size or new word consisting of a different set of faults and that way it is greater, but ideally otherwise there is fault dropping is difficult.

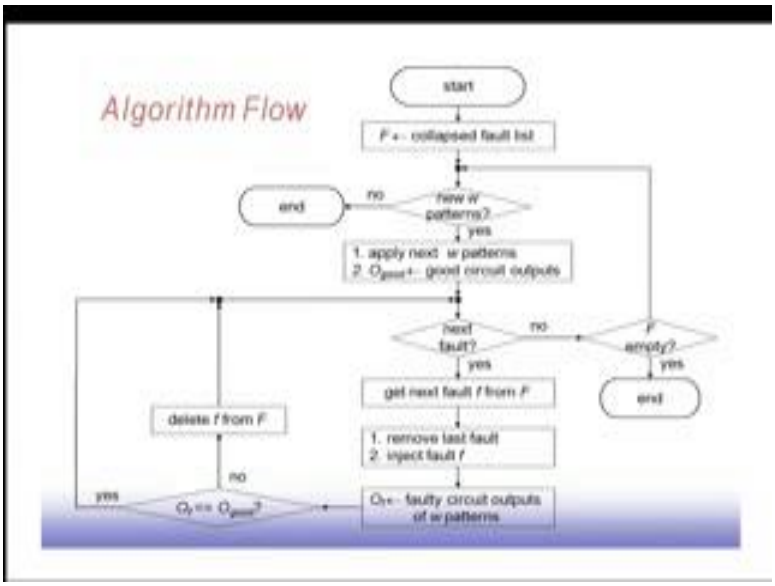
(Refer Slide Time: 24:03)



Next we will look into parallel pattern fault simulation. So, where in the previous case, we are trying to simulate a number of faults parallelly, now we will be doing single fault simulation, but for multiple patterns. So, again the basic idea is similar. So, we have got say w bit data bit then w test patterns are packed into a word and simulated for the fault free and faulty fault free or faulty circuit. So, now, at one time only one particular fault or the good circuit will be simulated, but we are packing w test patterns into the word that way the operation becomes parallel. Now it is the single fault. So, naturally single fault will be handled at a time, first a fault free simulation will be done and then for every fault will be a fault injection and faulty circuit simulation.

Now here we have got the advantage because in one short we are getting for a particular fault we are getting a large number of patterns that can detect that particular fault. So, this way we can see we can do fault dropping and also these faulty many of the patterns are being simulated simultaneously. So, it will be saving the effort.

(Refer Slide Time: 25:27)



This is the flow of the algorithm. So, start with the then we start with the collapsed fault list. So, new w patterns, as soon as you get a new w patterns from the test set. So, we apply the next w patterns and output good is the good circuit output.

This is basically the fault free simulation. So, we take the next w patterns then we have fault free simulation now for this w patterns. So, we will try to simulate for each of the faults. So, we take the next fault if the faults if the if the next fault is there some more fault is there. So, we get the next fault f from the collapsed fault list then we remove for last fault and inject fault F. So, in this circuit previously some fault may be existing or if it is a good circuit then there is no fault.

We will inject the new fault F into F then we simulate the circuit and see, what are the faulty circuit responses for w pattern? So, O f is the faulty circuit for w patterns, now if O f and O good, they are same then the faulty circuit are none of the faults could be detected from this otherwise. So, this particular fault could not be detected from if it is if they are not same if O the faulty response and good response are not same; that means, this fault was detected by some pattern in the w ok.

In that case, we delete f from F; that means, we have detected this fault. So, this fault need not be considered anymore. So, we take the next fault this process continues. If O f and O good are same; faulty response and good responses are same then this set w patterns they could not detect this particular fault. So, it will try with the next fault and

try to do that and when this f becomes empty when we have when there is no more fault that we can handle then we go back and try to get the next new w pattern. So, for one set of w patterns we consider say then for that we simulate all the faults then we take the next set of w patterns and simulate the next set of faults. We will continue in the next class.