Embeded Systems - Design Verification and Test

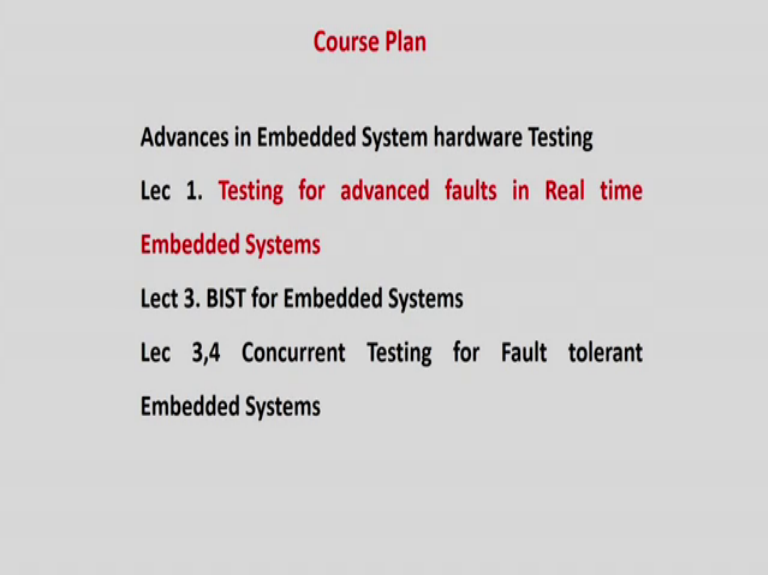
Dr. Santosh Biswas
Prof. Jatindra Kumar Deka
Dr. Arnab Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Lecture – 32

Testing for advanced faults in Real time Embedded Systems

Hello everybody, welcome to the 3rd part of the Embedded System course which is on Testing.

(Refer Slide Time: 00:32)



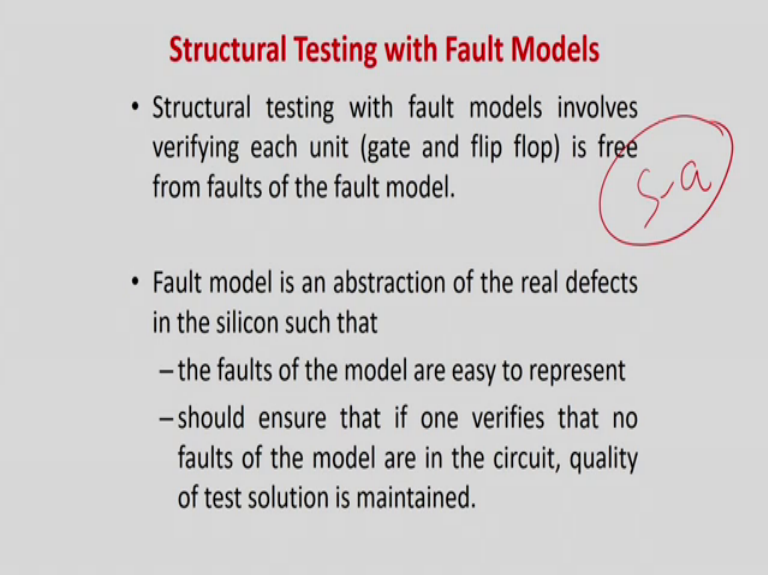
Today we will start the, the 3rd unit of this part on testing; that is actually testing for embedded advances in testing for embedded system hardware. In the last 2 basically parts of the testing unit; what we have seen? We have seen first 1 of the basics of testing and then you have seen some of the specific testing concept which are mainly applicable for embedded systems.

Now, in this part what will be mainly looking as you can see in the course plan (Refer Time: 00:57) advances in embedded system technology. That is which is more becoming sophistication of the digital hardware and more typical kind of constrains which is coming into embedded system design like meeting the timing constrains meeting the

power constrains etcetera. So, what are the different advance test techniques that are require for such cases.

So, as we see we will first see what are the different fault models which require to verify the real time operation correctness. Second we will be looking at build in self-test and then after that we will give two dedicate lectures for fault tolerance of embedded systems. So, these things are mainly required in the modern technology circuit which are build using the modern technology. And which are applicable for a large class of embedded systems where we require fault tolerance, where we require real time guarantee, etcetera etcetera.

(Refer Slide Time: 01:48)



So, the first lecture on this is testing for advanced faults models in Real Time Embedded Systems. So, whenever you talk about embedded systems; now a day's the term called real time is becoming very important. What you real mean by real time, as already discussed in details by Arnab Sarkar Sir and Professor Deka. That basically whenever we say a embedded system, we have some kind of deadline boundaries that is some kind all the operations not only have the functionally carried, but also have they have to be temporarily carried. That means, we have to generate the correct solutions and at the current time; that means, you should meet the deadline.

Also you have talked about delay hard deadlines and soft deadlines. So, hard deadlines are mainly for mission critical application like a car, avionics. Then you can tell about

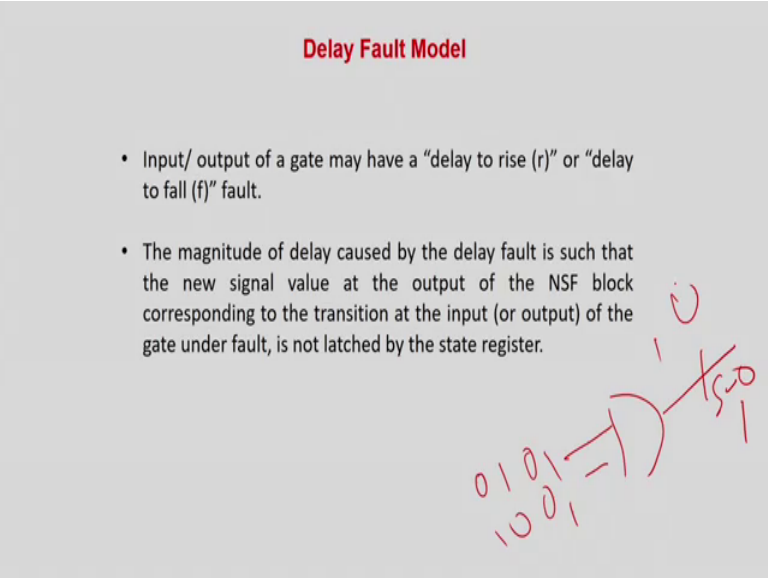
the nuclear reaction chamber everyone there are embedded system. So, there you cannot at all delay the answers by which time it should be ready. That is not only correctness, but absolutely before the redline, you should be made. There is something called software deadlines systems like microwave controller or washing machine controller where you can slightly delay the answers by which the delay the deadline by which the answer should be ready.

But mainly when you are talking about testing; today we are mainly concentrating on the mission critical applications because, there it is very very important that you do not miss the deadline. If you do that there is lot of catastrophic problems that can happen like for example, you are a applying a break in the in a AVS vehicle. Then basically if the break should be applied in your particular deadline after you press the break shoe, even misses the deadline there can be an accident; should be very very careful about the missing of the deadlines. So, we will look at that in this lecture.

So, we have already seen in the first unit the of this testing part that structural testing with fault model that is the stuck-at fault models in defect standard. So, what we have seen that we have we are testing the circuit structurally with fault models, which involve verifying each unit that is gate and flip flop is free of faults from that faults model. And generally we talk about stuck-at fault. So, what we have seen rather than testing the functionality of the circuit we have try to verify that none of the gates, or none of the flip flops have a stuck-at 0 fault. So, we have detail discuss this in details.

So, what is the fault model? So fault model is an abstraction of the real defects. And why we do that? Because they are really easy to represent, automatic test balance generation becomes simpler and the less number of test pattern to test those circuits so test tank is reduced. So, we have seen the stuck-at fault model for this purpose serves a very very nice very very nice job in that manner.

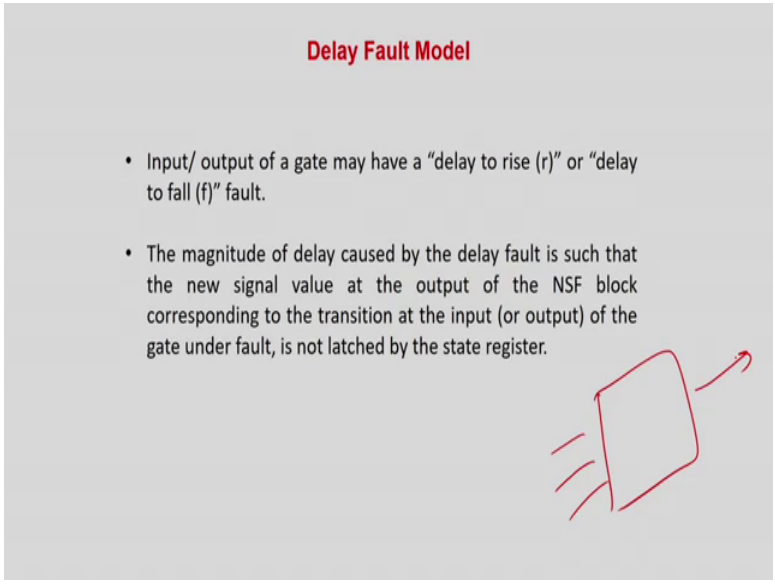
(Refer Slide Time: 04:17)



But there is a problem so, stuck-at fault model as we have seen that it tells that whether this line is stuck-at 0 or not. And actually verify that none of the line is at the stuck-at 0 or stuck-at 1; similarly we do it for all. But in only and it guarantees that if you can find out there are circuit does not have any stuck-at fault. Then generally it is 99.9 percent accurate this circuit does not have any kind of functional defects.

Functional meanings correctness of the answer; that is the if I apply 1, 1 you will get the answer 1, 1. If you get 0 0 0 1 0 or 0 1 you will also get the answer is a 0. So, that is the functional correctness I can guarantee, but what about delay. See I know you know that in real time systems what do you mean?

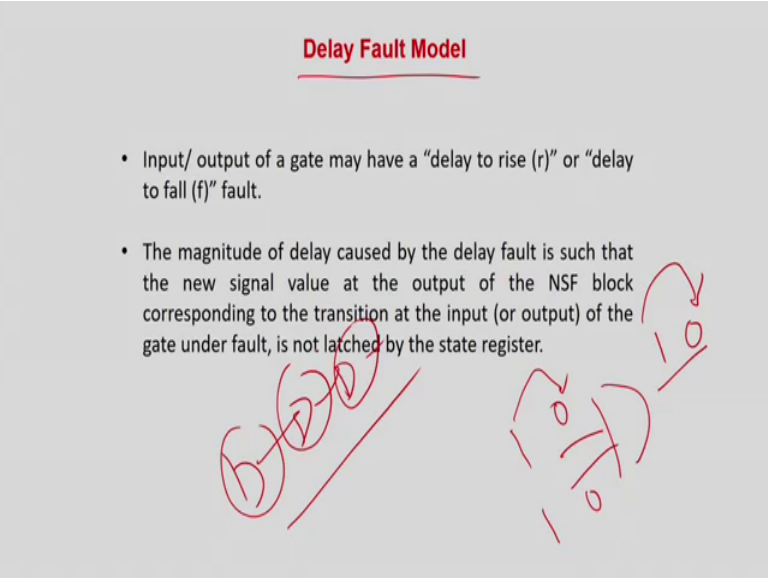
(Refer Slide Time: 05:01)



Real time system means basically; you have a black box and some inputs are coming to it and the output should be ready by a particular deadline. So, answer should be ready by a particular deadline means; your hardware after taking the input should give you the output is a specific deadline, it should not miss any of the deadlines. That means, your circuit should operate at the same frequency which you are expecting it to work. So, that, so you I think you will easily understand stuck-at fault will not serve the purpose. Stuck-at fault model will tell you that given a very large amount of time you are always guarantee than the output functionalities correct.

We will just test that none of the line is stuck-at 0 or stuck-at 1. That is, it is behaving properly as per logic 0 and 1, but time taken is does not verify. Therefore; to verify the temporal correctness or that it will definitely guarantee to meet all the deadlines. It is very very important that you should have another different kind of fault model which is called as the basically your delay fault model. So, what is your delay fault model?

(Refer Slide Time: 05:57)

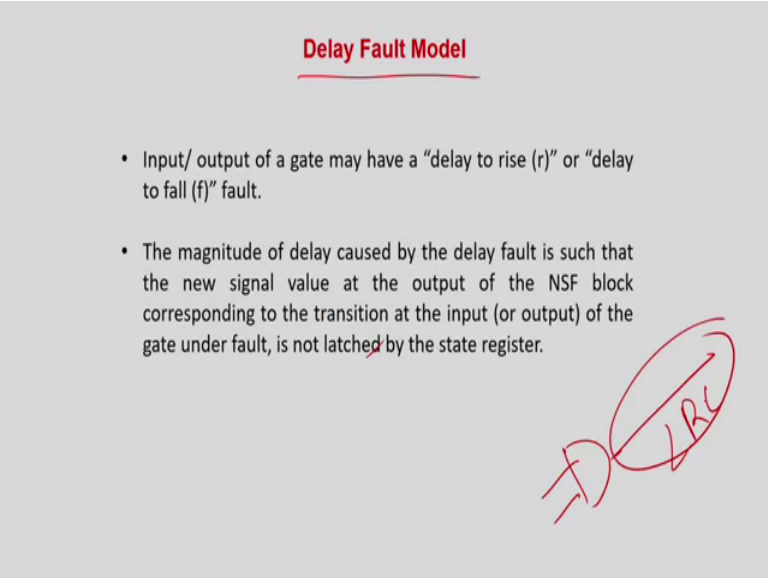


Delay fault model we will just tell you not only that that if you apply a 0 0 the answer will be 0. Not only the answer will be 0, but also it will be deliver at the proper time. Like for example, you have 1 1 the answer will be 1, you transit it from 0 to 0 0; then it will guarantee that this transit from 1 to 0 will be within a particular deadline, which is a delay of the gate. And as you know that circuit will be something like their lot of gates in chain.

So, if you assure that all the gates are meeting their deadlines or they will deliver the result in deadline. Then you can totally ensure that the circuit will also give the final answer within a deadline. So, that will allow that not only your devices would be functionally proper, but at the same time they will meet you delay and deadline sorry your deadline constrain so that is the idea.

So, what are the delay formula as I told you just like stuck-at fault model you cannot just like the modulation of stuck-at fault model you cannot thing of just testing for all the gates whether it will rise properly whether it will fall properly and so much transient delay. All these things is not possible to measure and test because of the huge amount of time that will be require. So, what we will do? Basically just like stuck-at fault model there is something called delay fault model which will take lumped delay of the gates whether.

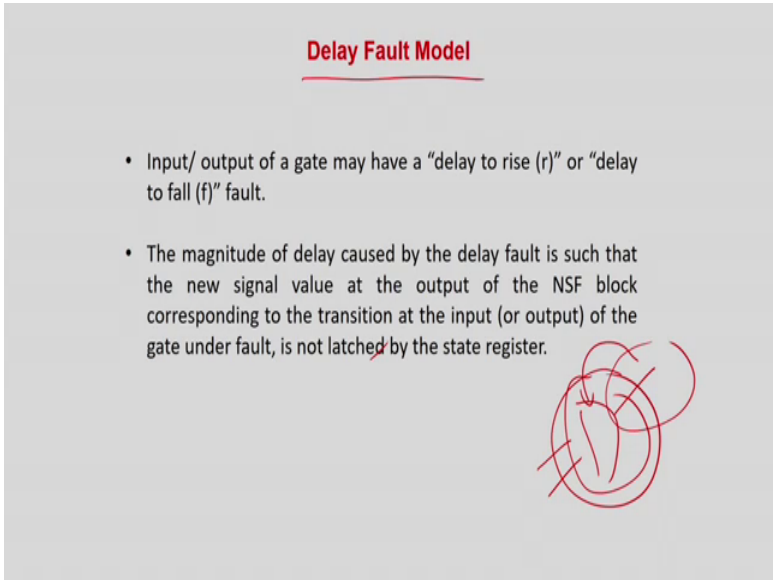
(Refer Slide Time: 07:07)



So you can understand that delay can be within the gates or delay can also be with the interconnects. So, if you starts thinking about all these because the, interconnect is nothing, but an RC line. Generally something it will call LRC or RC lines the Resisting Capacity and inductive lines. So, basically if you try to model all these resistance, capacitance, inductance, then find out the transmission line properties. Then it will actually led us into a very bad situation; where will take any amount large amount of time to testing.

Like the delay of a line dependence on it is capacitance, it is parasitic capacitance, inductance, resistance. So, when we start designing a circuit we sometimes assume a lumped delay model or sometimes we assume the there is no delay. But they as I as in the case of stuck-at fault model if you start testing about what is the capacity capacitance, what is the delay because of that you will land into a big problem because, we will have no time to test such rigorously and not required also.

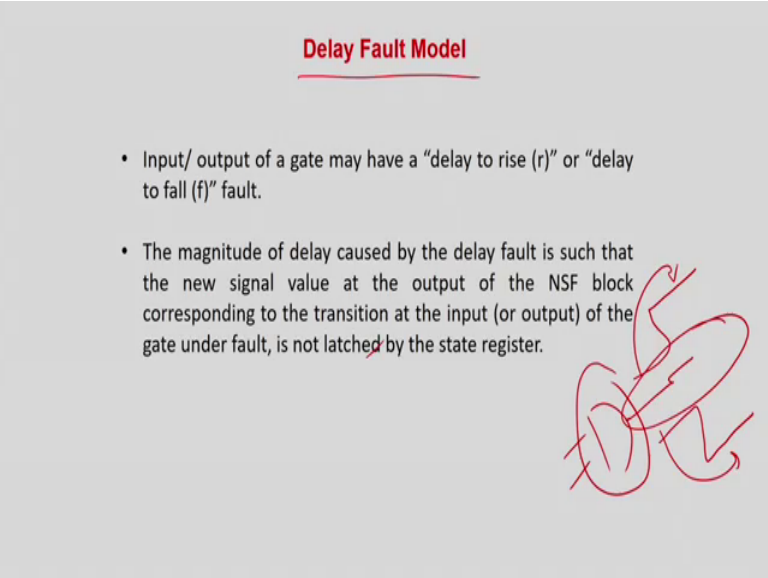
(Refer Slide Time: 07:59)



So, just like the stuck-at fault model people also try to thing that I will consider delay is lumped within the gate. That if the gate can delay if the gate can be deliver the answer in a proper manner, in a proper amount of time; then we assume that it is correct. T hat is the delay of these lines and the gates and all the transits etcetera are lumped into the gate delay itself, that is a model like a stuck-at fault.

Now, we will going into a more elaboration then will get the concept. For the time being is just enough for you to understand that instead of considering all sorts of physical delays involve in a gate transistors and the lines we lump it to a gate. And assume that there will be no delay to rise and no delay to fall.

(Refer Slide Time: 08:35)



That is being the output means if you want to go from 0 to 1 this rise will be within the proper deadline and if something some value has to fall it will fall within the required deadline; which is for the which is assumed deadline for the gate. So, if it have this properly you can take care that the delay of the gate as well as the delay of the interconnects are taken into picture. So, just like a stuck-at fault model instead of the looking for all kind of physical defects, we assume that none of the gates lines have a stuck-at 0 or stuck-at 1.

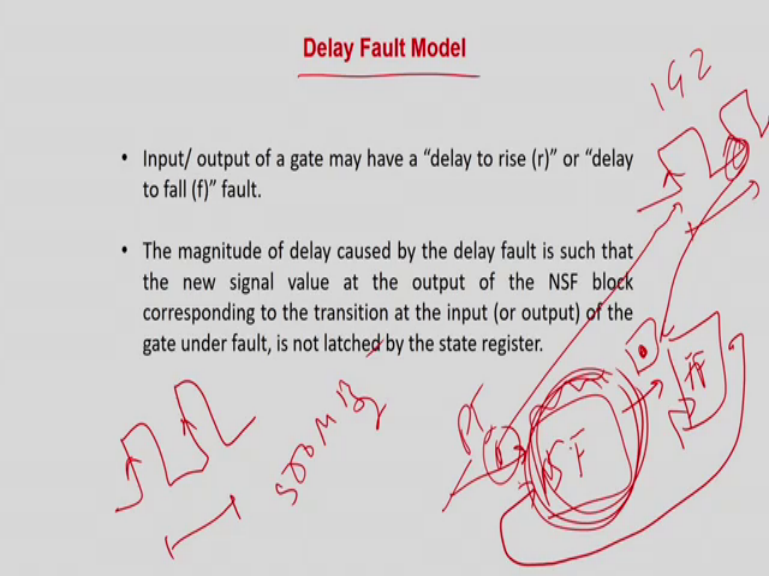
So, here in this case we try to verify that none of the gates we will have a delay to rise or delay to fall; that is what is the idea for delay fault model. So, if you look at it that is the input, output of a gate; have delay to rise or delay to fall. If none of the input lines or output lines have delay to rise and delay to fall fault, there we can assume that the circuit will be giving a solution in the real time within the deadline. And it has been found that more than 99.9 percent assurance can be given and if this is verified the circuit will have no delay faults or in other words you will be able to get the correct answer in the correct amount of time.

So, the magnitude of delay caused by the delay fault is such that the new signal value at the output of the NSF block corresponding to the transition at the input or output of the gate under fault is not latched at the state register. We will this very very important I will I will the meaning of the line has to be understood in a very rigorous manner to

understand the delay fault. Generally, if you have a combinational circuit there also there can be delay that is ok.

But mainly will be focusing on delay of a sequential circuit; because whenever we talk about circuits mostly they are sequential and also when you talk about frequency of a circuit 1 gigahertz, 10 gigahertz 100, megahertz that means there is a clock whenever there is a clock means the sequential circuit. So, whenever we are talking about delay faults and most of the circuits we are only talking about a sequential circuit.

(Refer Slide Time: 10:20)



So, what it says we know that in a sequential circuit there is a NSF block, next state function block it feeds the state registers, there is a feedback and there is the primary inputs, of course, the output function block is there. Already we have seen in the last few lectures that testing NSF is more difficult compared to NS output function block. Once the, whatever test technology we apply for receipt can be directly applied to the output function block; it is a combinational circuit.

So, I mean we are not much concerned about that because it more easier to test. Let us again that the same philosophy let us look at the NSF block primary input in the flop and it goes back. So, now, what happens some it is a combinational cloud. Whatever input you give from the primary input and feedback from the faults it will sit it sometime before the answer is delivered over here. And we know that there is a clock over here

that is very very important. The clock will occur at some period and that will decide you speed.

You can say that I have designed a 1 gigahertz machine or 1 gigahertz circuit, but then what the result should also be delivered keep on changing at the later level of 1 gate. Because say that some inputs apply over here at this clock pulse before the apply arrival of the second raising edge of the clock, this data should be available over here. Because then only whatever computation you start giving at this clock pulse it will be latched by this flip flop at the second clock edge.

So, before that this 1 gigahertz that is the time that is the frequency at that level these circuits should perform the job and the delay should be delivered. Maybe the circuit is slightly slower; then you can say now I cannot operate at 1 gigahertz maybe I will operate at 500 megahertz or something lower. Then in that case you will have a more delayed clock sorry (Refer Time: 11:57) percent we have a more delayed clock same thing.

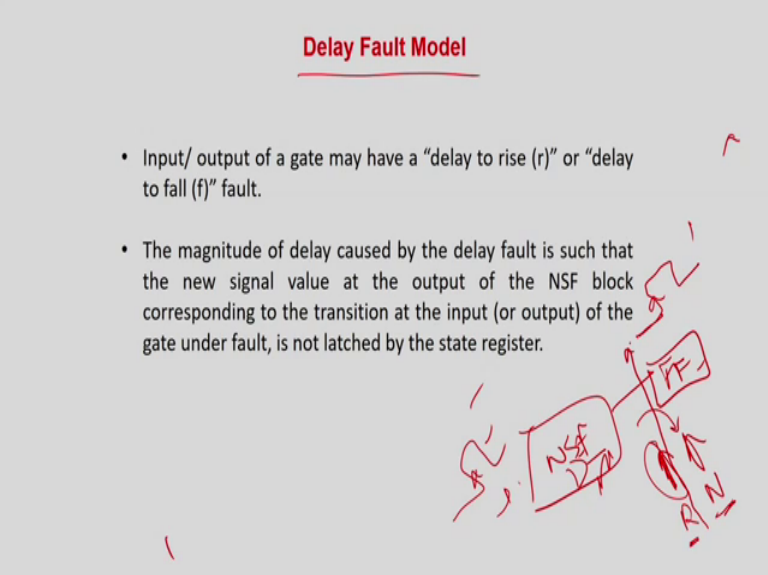
But in this case the time period will be in this case the time period will be larger, in this case the time period is corresponding to 1 gig maybe this is time this is corresponding to 500 megahertz. The frequency while reducing the frequency increasing the time period So, your NSF block will get more time to deliver the result so even if it is slower you can get the correct answer. But as a matter of fact you cannot always keep on delaying the clock because then we use you will not be able to get the correct answer at the correct time.

May be for certain application you will requite the answers to be coming at 1 gig level, then you cannot compromise on the frequency and made them say that I will be working at 500 megahertz; you cannot do that. In that case, you have to ensure that you have to take a proper type of gates designed properly so that your answers get operated at the proper time. Now, what is the fault causing the problem say you have verified everything, designed properly, you have all kind of static, and dynamic fault simulation, sorry.

Circuit, clock simulation, speed, timing simulation, everything you have done and you know that whatever data you give much before this 1 gig level you are also will be coming out of the NSF block. So, we will have enough time to get the data latched by the

flops, but due to certain delay faults in this the answer is getting delayed. So, you are not able to get the proper data in the flip flop at the next clock edge. So, those type of testing's we have to do in delay fault test. Now, let us look at the line the sentence again then it will make the stuff more clear to you.

(Refer Slide Time: 13:23)



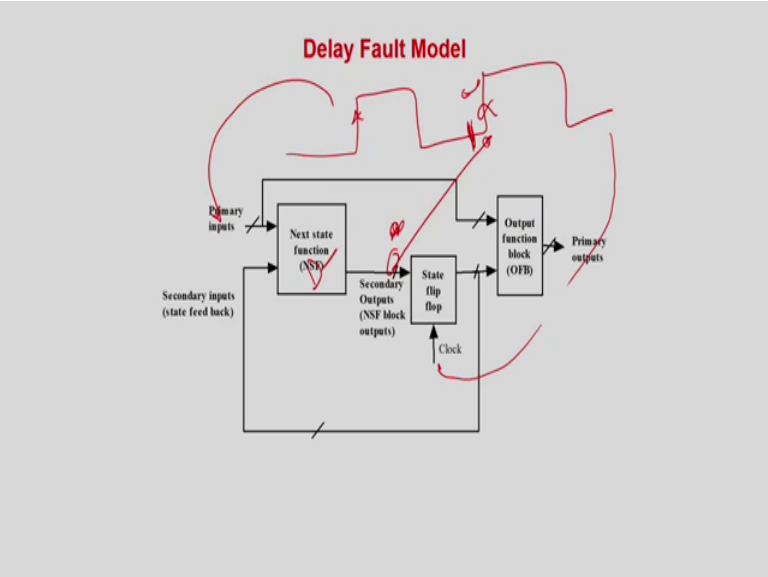
So, there is a NSF, it is feeding the flops, now what it reads? The magnitude of delay caused by the delay faults so there is a delay fault. So, the data should have been coming at this point, now it is delayed by this such that the new signal values at the output of the NSF block this is required and this is the new which is changed. NSF block corresponding to the transition at the input or output of the gate under fault; this may be this is the gate which is causing it, is not latched by the state register.

Some gate is causing a delay over here and it is reflected over here and your clock say arrives basically at this point of time, this is your arrival of the clock edge. This is your first clock edge in which case you are giving the data over here and the next clock is edge is coming over here which is after the required time, but much before the new value which is coming out because of the delay. Then what is going to happen had there been no fault of this gate your data would have been coming out of the NSF block at this point of time.

So, you could have easily latched at this point because your clock is coming at this point. But now due to the delay the result is not coming at this right time required time or the

right time it is coming here, but then the clock has passed and is not going to latched the new data, it is going to latched some stale data.

(Refer Slide Time: 14:35)



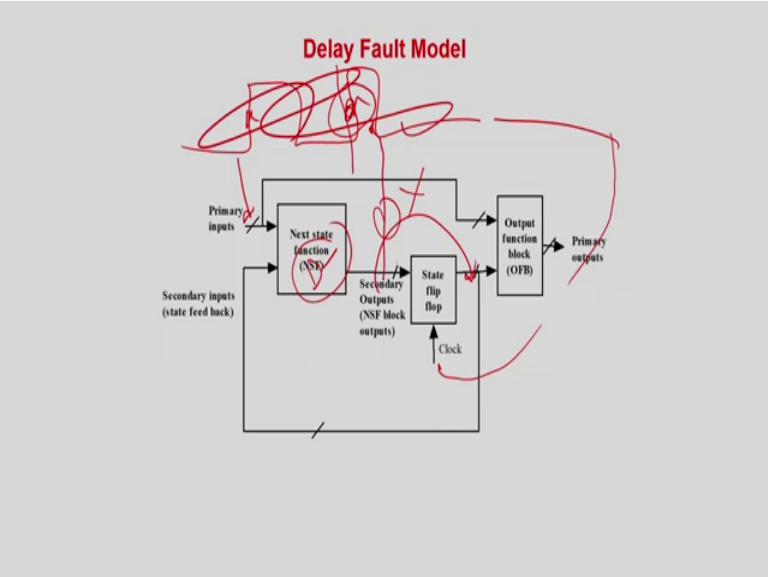
Now, let us look at it a more detailed manner or in a more block 1, main block using a block diagram model. So, if you look at it so this, the figure, this is the next state function block, this is the output function block and this is the flip flop set, this is the clock. So, assume in that there is some gate here which is causing the delay. So, now what happens since some at this clock is let me draw the circuit. Then the clock basically so there is a clock something like this so, this is your clock in this clock some data has come over here.

And we required that before the arrival of this clock edge your data should be available over here because this clock is basically this clock. So, at every positive edge because I think by digital design fundamentals you note that the flops will latch data other the positive is of the negative edge. In this course, you are all assuming that everything is happening at the positive edge of clock.

So, now new data has come over here so whatever was the value over it is processing. Now, before the arrival of this clock edge, your data new data by the new primary inputs and the state registers should be should be available at this point of time. And it should be available someone before that there is something called set up time.

So, before because if you are say giving some data right at this point of time the clock the flops may not be latching it properly. So, there is something called set up time so this is actually call the set up time. Before the set up time of the arrival of the new clock pulse your data should be ready over here. So, if it is ready then there is no problem whenever the clock comes this data will be latched and coming to this flip flop.

(Refer Slide Time: 15:58)

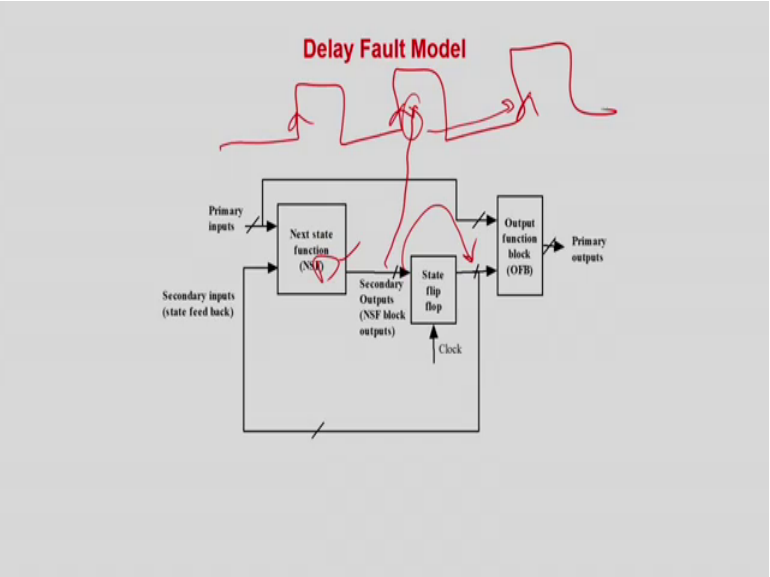


So, again draw the clock here so this is the, your clock over here again the same clock we are talking about. Now as I told you that before this setup time so your data should be ready over here. The data was given as input at this clock period. Now, let us have assumed that there is a gate which is the complete gate which is causing a delay. So, again I have to want the, emphasize that as a design you have chosen the proper gate by meeting proper timing. Like by the design fundamentals which Professor Arnab Sir has taught you have taken a proper gate and everything was matched, but because of the design fabrication problem, now this gate is having a more amount of delay.

So, more amount of delay means what? This is your set up time required, your data will not be ready by this time; you will have certain delay because this gate is having some kind of a delay due to manufacturing defects your data will be coming at this at this point of time. Now, what is going to happen the at this point of time your flap flop will latch some data.

But this is not going to be this data this is going to be some stale data which will be coming out and record at the flop. So, you will be delivering the answer at the next clock period. Now what will happen now if I extrapolate sorry? Now if I extrapolate so what is going to happen you are going to get the answer at the third clock edge, which will be the delay?

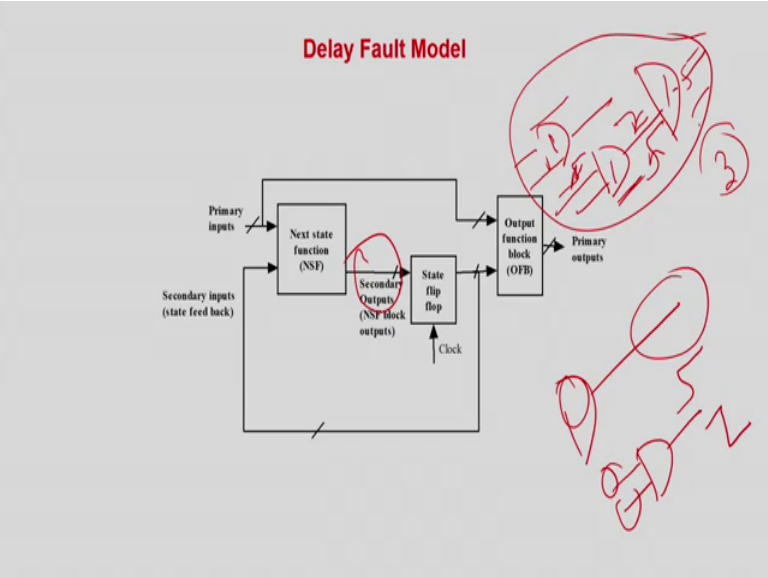
(Refer Slide Time: 17:06)



So, now what happens this is your first clock pulse, this is your second clock pulse and this is your third clock pulse. So, you have already seen because of your delay of this gate this data is coming at this point of time. So, at this point of time is some data is coming and the second edge this flip flop is not latching. I t this is latching this data address stale data, but this data will be latched by this clock period third clock period. So, at the third clock period your require data will come over here.

So, we will have a missing of 1 clock edge or 1 clock pulse. So, you will be delayed by 1 clock period for the data to come. So, because accurate the manner you will find out that your answers are all missing your deadlines So, therefore, this is actually the delay fault model and we have the ensure by that is by test planning as well by the test plan and application that there should not be any kind of delay. And all the delays has to be matched, but again I cannot do this at the functional level you will lead to enormous amount of requirement of time.

(Refer Slide Time: 17:59)



Secondly, as you have already seen I cannot go for like if it is a gate, I cannot go for starting the delay of the transistors, I cannot study the delay of the lines or interconnect lines, by modeling by the error and say. So, we are going to go to a delay fault model we simply text to the fact that none of the lines should have a delay to rise, delay to fall. Similarly for all the inputs of all the gates should rise in the proper time, and fall in the proper time.

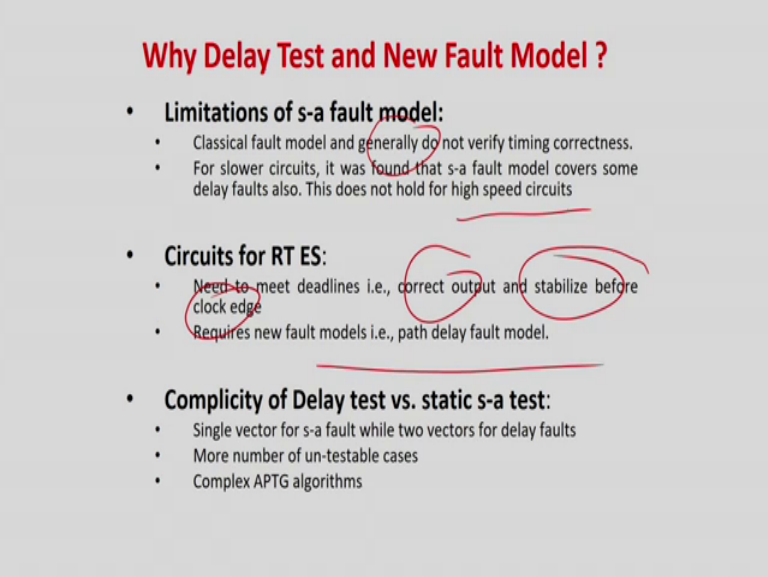
Of course if it is happens basically you will always going to get the answer the proper amount of time. Because by design so, whenever I say that say maybe this is the 2 gates required in chain maybe after there is 1 gate and you maybe want the answer by 3 nanoseconds. So, of course, I will have gates of delay 1 1 and 1 or maximum I can have a delay of 1.5 nanoseconds so, by 2.5 nanoseconds my answer will be ready.

So, when I am designing it I will take care that all the delays are made; so, that I can get the answer the proper time. But only of the, because the fabrication error this might have become from 1 to 2 so, it will lead to a missing of the deadline. So, we will just see by the fault model that all the all if they all the delay to that; that means, in this case the delay to rise sorry delay to fall and delay to rise all will happen within 1 nanoseconds corresponding to the inputs for these gate.

So, these things I will verify if this is tested then I know that you will always get the answer in 1 nanoseconds for this similarly for this and similarly for this. Of course, we

are always going to get the answer at the level of 2 2.5 nanoseconds or less; so, that you always meet the deadline of 3 nanoseconds so, that is what is the concept of delay for testing.

(Refer Slide Time: 19:25)



So, again what is the limitation of a single stuck-at fault model is the classical fault model and you do not generally verify the timing correctness. So, I should not called it generally it does not verify the timing correctness because if you want to go for timing correctness you have to apply stuck-at fault test patterns are a very high rate.

But in case of and again one more important thing I have to tell you and we are always talking about the tester cost. So, the testers are also low end and as well as high end for example, a low end tester will ability able to apply patterns here maximum 500 gigahertz megahertz and very modern tester will be able to apply test patterns at gigahertz level.

So, of course if you are if you want to higher or tester of a higher standard or higher speed you will have to pay more money. So, we if we want to do just stuck-at fault testing. So, what we do is then we hire a low end tester and apply test patterns as a slower rate. So, therefore, we say that we generate do not want to verify the timing. So, just stuck-at fault if you want test we will have a low speed tester. For slower speed circuit it was found the stuck-at fault models cover some of the delay faults also, but does not hold for high speed circuits.

So, if you say that my circuit will work at 1 megahertz 100 megahertz, then we need not worry much about your delay faults. Because anyway your whatever falls you have if it is not a permanent fault like stuck 0 and stuck-at 1 or the catastrophic fault. This delay fault etcetera will may be there, but it will not at all violet your deadline maybe you require a delay of you deserve design a circuit for 10 nanoseconds they get delay of 10 nanoseconds Even if the fault is there you will not overshoot 10 nanoseconds.

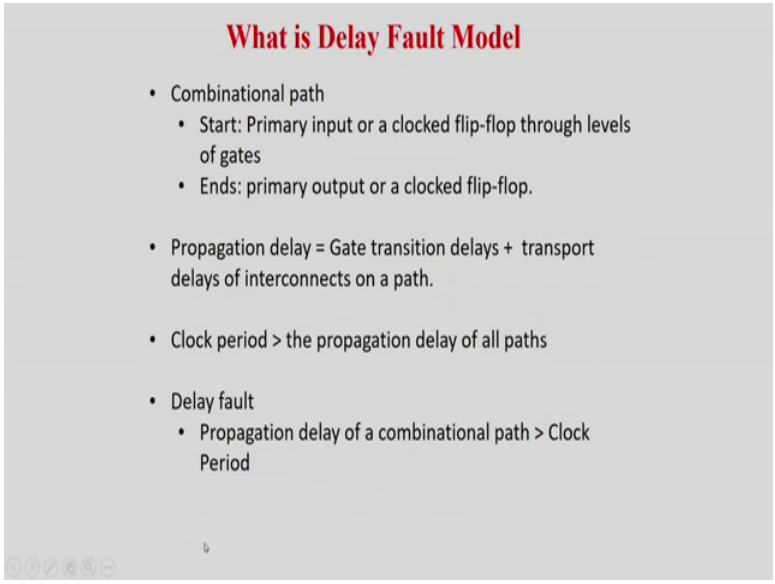
If, but if you have a gate which is delay your, we require to be 1 nanoseconds then due a failure it means easily become 1.5 nanoseconds or 2 nanoseconds. But if you have if you have design a very slow gate 10 nanoseconds then even due to failures of the this manufacturing process some delays will not increase. Because, last class we have discussed that why the faults are getting inserted during fabrication, because we are always trying to go for a very advanced manufacturing process even the technology is not matured.

So, therefore, if you try to fabricate gates with 1 nanoseconds delay 2 nanoseconds delay the incoming of the failures during manufacturing is more. Then if you are compare to a situation whether you are designing a very slow gate like 10 nanosecond 20 nanoseconds gates the for incoming probability of faults are very less. Because our technology is more or less mature to design such kind of gates.

So, therefore, stuck-at fault models are applicable for slow circuits, but you will not be able to test for delay fault for a high end high speed circuits. But circuits are real time embedded systems we require correct output and stabilize before the clock edge that is by before the set up time the output results will be stabilized. So, we require new fault module which is called the delay fault model is required. Now, again as we will find out that the complicity of delay for is this is of course, will be more higher than static stuck-at faults.

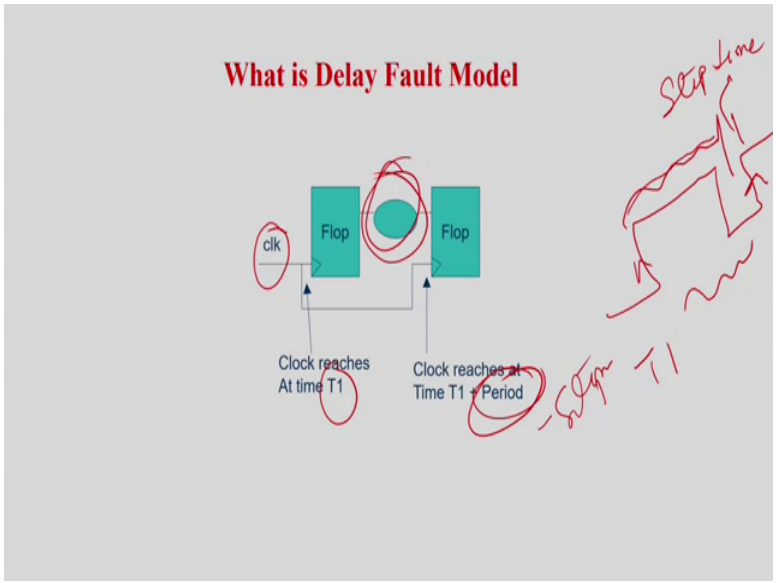
Because single test pattern can state a stuck-at fault as you have already seen. But in case of the delay fault models we will see that we will have two faults and also we have to do it at speed. And finally, APTG algorithms and will be more complex and of course, more number of untestable faults will be there. So, this will not going into the details, but mainly will be looking at how to generate test patterns and how to test stuck-at how to state delay faults. That is our main emphasis in the course.

(Refer Slide Time: 22:27)



So, basically certain ideas you should know that basically what happens I will take a figure.

(Refer Slide Time: 22:32)

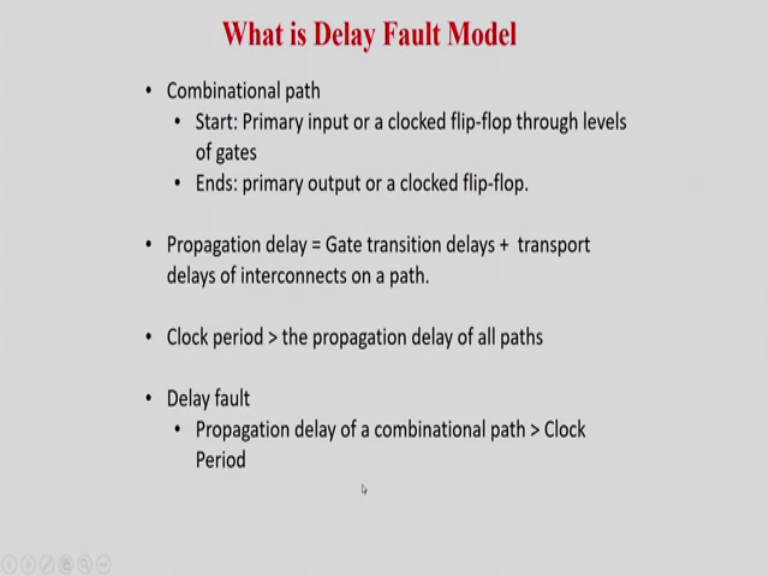


So, therefore, this a flip flop and this is a, another flip flop. So, a clock is coming over here and the next clock period; let us assume that the clock comes at T1. So, if you assume that the first clock edge comes at T1. Then within this next period that is your period the next clock edge will come. So, whatever computation you have to do within this period within this by the combinational cloud block with the NSF block should

always be completed by this duration. This is like something called the set up time this is something called the setup time.

So, everything you have to finish before by this time this time period. So, by period minus set up time period minus setup time that is the time given to this combinational block to complete it is job. If it overshoots then there is high chances that you will not be able to less the proper data in the proper time and there will be a dealing so, that is your faults.

(Refer Slide Time: 23:21)

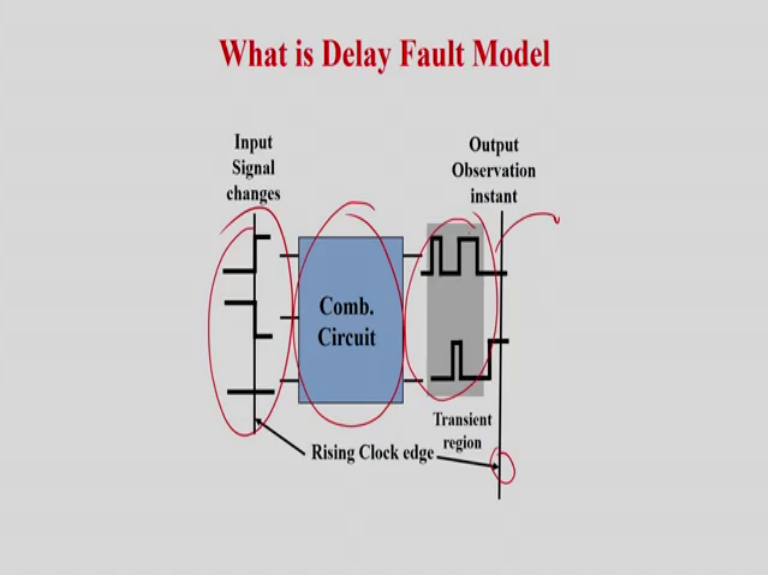


So, what is a combinational path so this is the actually your combinational path sorry so, this is your combinational path. So, what does the combinational path says primary input or a clock flip flop through levels or gates ends at the primary output or a clock flip flop. That means, it will start from the primary input or the output of one flip flop and it will end at the beginning of another flip flop. So, propagation delay if the gate transition delays, but transmission delays of the interconnects of a path. That is the time amount required for the gates to change their values as well as the delay path in the wires.

Clock period should always be greater than the propositional delay of the faults as we have seen because, if this is clock period you have to finish your completion of data completion by up time less than that. What is delay fault if the propagation delay of a combinational path is clear than the clock period? That is what; that means, you are not able to complete the completion by the set up time. So, you your data may be coming at

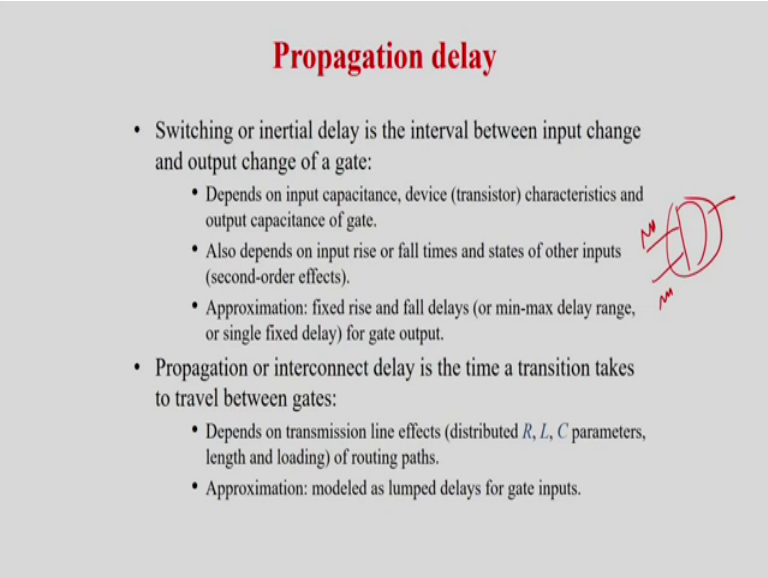
this step (Refer Time: 24:15) at this if your data is ready. So, we will not be able to less the proper answer so this is your delay fault.

(Refer Slide Time: 24:21)



So, this again representing the same thing so your signal changes at the input this is the combinational cloud and of course, this is what is your call transient period or I which you call the set up time or your values should be stabilized by that time and then the next edge comes. So, before that your data should be stable or correct data should have come. If you delay this this is your setup time. So, if you anything happens beyond this you may not be able to has the proper value that is actually a delay fault.

(Refer Slide Time: 24:45)

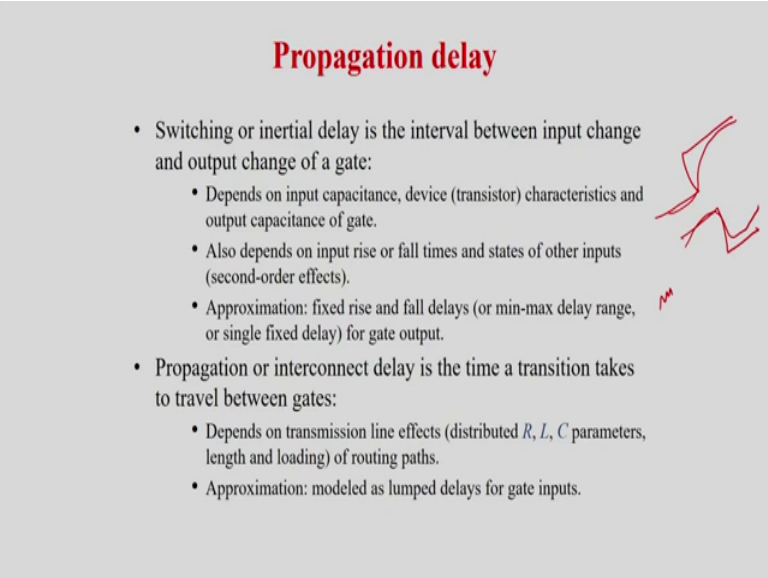


So, now why delay? There are lot of business delay happens, but we are not going to test for all those things because that will actually take enormous amount of test time. So, what they are switching or inertial delay is the interval between the input change and the output change of a gate. That is input to output change that is the gate delay. So, depends on the input capacitance device characteristics and output capacitance of a gate.

So, it will depend on the gate characteristics because we all know that the gates are designed by transistors. Transistors also have a inbuilt stray capacitance so that will cause the delay of the gates, would also depends on the rise and fall times and states of other inputs which is called the second order input because this gate will have been inertial delay because of the input capacitance.

But again output these are the inputs there can also be delay from the gate which is previous to that. Because there is, these may be into also driven by some primary input or some other gates so, there can be delay of that. So, those delays will also be having an impact on this gate because this gate will have it is own delay. And it will try to give the proper answer based on the fact that the other gates which is previous to that will give the inputs of the proper time. If that is also delayed then we can have some more delay of the gate.

(Refer Slide Time: 25:54)

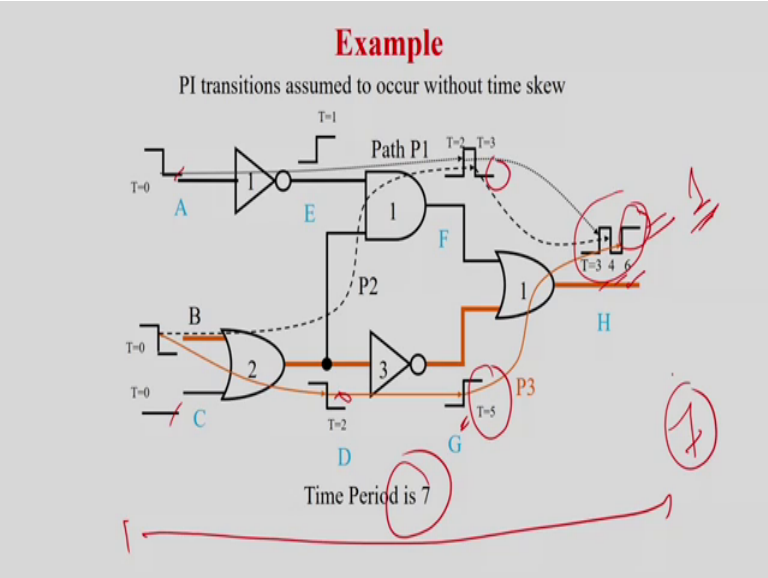


An approximation that is fixed rise and fixed fall time that is we always assume that there will be a steep rise and a steep fall, but actually in fact it is not. So, there are a lot of other capacitance and inductive physical effects which is there. So, that means, what I am trying to say in this line there is a lot of physical parameters involved in the causing the gate delays. Similarly, propagation or interconnect delay is the time taking between the gates. These are all gates are connected by interconnects and the basically they are something kind of a lines which have been LRC parameters.

That is resistance inductance capacitance parameters which will depend on the length, loading and different type of routing paths taken if there are fears if the routing path is long so it will have more delay and so forth. So, those will also basically come into play off the basically comprise of the delay of a path.

So, again if you take all these situations will be a very big problem of the time. So, we approximated saying there is a lumped delay for gate inputs. So, we assume that all the gates will have a proper rise time and a proper fall time. So, if it is maintained we assume that we are going to get the answers in the proper time similar to stuck-at fault model philosophy.

(Refer Slide Time: 26:26)



Now, a good example to show you all the fact; so for example, this is a circuit and basically we want to decide on the time let us not for daily. Let us just hide the track the time period is 7. Time period is 7 means something like this sorry time period of 7 means you will have a something like this this time period is 7.

So, whatever answer you have to compute by this cloud this is a flip flop over here they have not shown this and this is the primary inputs. So, by 7 clock period you have to get your answer ready. Now, let us see what happens so whatever number 1 2 1 3 1 whatever you are looking inside the gates are the say 1 nanosecond delay of the gates.

Now, say for example, if this 1 transit from 1 to 0, this 1 also transits from 1 to 0, and this one is c 1 is having the value of 0 basically. And the time period of 0 it starts so at 0 time period it becomes 0, B becomes also 0, C is any initially or this is not an issue it is always at the level of 0 and it continues to the flat time.

So, now you see so there is a fault of course, this is a and gate so if it is for if it is always 0. So, if these lines from 1 to 0 you will also get a fall in the output, but if it is happening at 0; so, this 1 is happening at 2 the fall is happening from 1 to 0 at 2 because the delay of this gate is 2. Similar the inversion gate so from 1 to 0 if it falls of course, there will be a rise inversion, but it will happen at time equal to 1 done.

So, now if you look at this gate so what is this this gate is very interesting to observe. So, finally, what will be the output of this gate this is the and gate so this is a 1 and this 1 of course, is 0 right. So, 1 and 0 the output of the and gate will be 0 that is the final value, but there will be some transients before this. So, what are the transients if you look at it so initially at time period of this has all happened at time period 1 sorry say at time period 2.

So, this 1 is rising at time period 1 and this 1 is falling at the time period 2. So, if you look at it this line at time period 2 what is going to be the case at time period 2; if you look at it the value is actually 1. And this 1 is falling at the time period of 2 right so, now, what happens and there is a delay of 1 by this gate. So, basically it will all happen at 3 plus. So, at so initially if you look at it the value of this one was a 0.

So, output value was 0 at this point because this one is 0. So, this one is also 1 sorry the answer was a 1 over here, but this was the 1 and initially this was a 0 so the output of the, and gate was a 0 over here. So, initially the, and gate is 0 sorry the, and gate was 0 initially because of this fact of this line being 0. Now, you see so this is initial value.

Now, what will be the value final value sorry what will be the final value. The final value will also be a 0 initial again let me just reiterate it. So, what is the initial value? So, the initial value this one is 1 that is stable value. So, initial value 1 means it will be a going to be a 0 of course, the value will be 0 over here. What will be the final value the final value by these cases this is rising to 1 no problem, but this will be again falling to 0. So, this will be 0 1 it will be again 0 so this is again be stable value, but in between there will be a transition.

So, what will the transition and why the transition will be there and of course, when the value will be coming and the level of 3 because delay of 2 this is delay of 1. So, maximum you have to take so this is 1 and 2 so maximum will be 2 plus 1 delay it will have. So, after time period 3 basically is going to be stabilized. So, 3 means I am basically 2 plus 1 and this basically is this is relative one.

So, basically if I again I mean to going to more complexities. So, if you see at certain point of time this at t equal to 1 this this value is equal to 1. And it is falling at time period equal to 2 so first certain amount of time if you look at it both of them will be how 1. For certain fixed period of time that is at 2 plus 1 delay. So, here if you look at it

so at the time period of 1 this 1 is 1 and the time period 1 this is also is 1 because only a 2 it is falling. So, a time period 1 this value is equal to 1 and of course, it will be reflected at 2 because our delay of 2.

So, we will observe that at time period to the delay is actually equal to 2 delay I am sorry the value is equal to 1 which is not a stable value after that what happen at time period 2 basically this 1 will become 0. So, at when this 1 is actually will become equal to 0; that means, again it will it will not be equal to 11 it will be becoming as 1 and a 0 so, it will again become 0. So, and so it is 2 plus 1 equal to 3 so, at time period 3 will find out that the value is again becoming 0 at this point which is the stable value.

So, if you just analyze this quickly then you can see for the time period of 2 to 3 it will get a get a it is getting a spike which is not a correct value it has that it will stabilizing after some point of time. And the stabilizing value is equal to 3 because the delay is equal to 1 and the worst case delay between the inputs to 2 plus 1 and 3 you are going to get the answer is a 3 over here.

Now, let us look at the other path so other path will be more interesting same way you can analyze. So, now, what happens the other path basically shows that this is 2 equal to 0 and the delay is equal to 3 so; that means, 2 plus 3 5. So, you are going to get the answer of a 5 equal to 1 the value of the gate this gate will be output this gate will be 1 only at the delay of 5 2 plus 3.

So, this final value is equal to 1 inversion because this is one. So, the answer should be sorry this answer is this is 0 that is the final value is 0 so it is an inversion. So, we will always going to get the 1 value 1 over here which is the correct value, but you will have certain delay for getting it because it is 2 plus 3 that is equal to 5 at time period 5 only you are going to get the answers of 1.

Now, in this case if you look at it. So, here also you are going to have a transition effect. So, what will be the value over here? So, it will be 3 1 so total delay will be 3 plus 1 so, you will start having delays incremented. So, it in this case the you are again take the worst case. So, this is 3 and plus 1 that is the 4 so, after 4 amount of time the answers will be the start coming in.

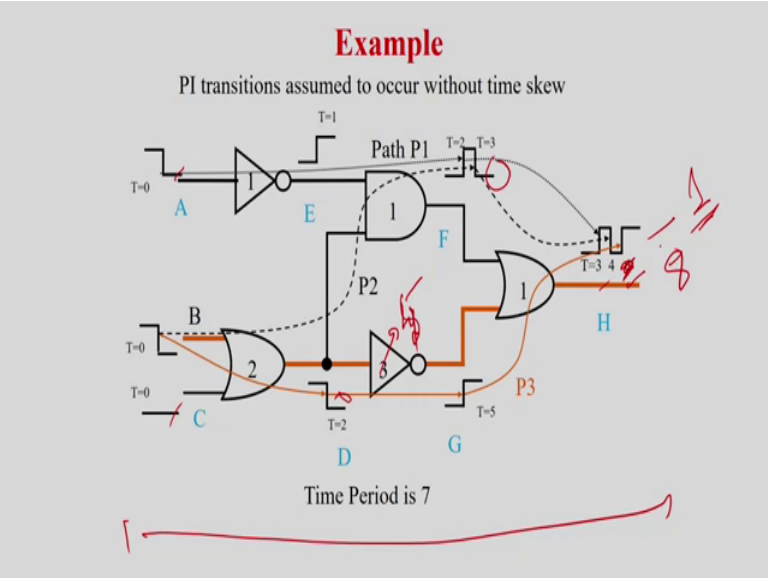
Now, if you look at it so in this case at the other parts I am not drawing this the stable part of it you can easily calculate. So, if you look at it is 3 at the level of 3 the value of this 1 is actually becoming 0. But at 3 this spurious value is also equal to 3 this is a long stable value. So, at 3 the value is equal to 1 so at 3 will find the value of equal to 1 and here it is 0 no not yet proper.

Now, after 3 at 3 it is falling down basically. So, what is going to happen in I will just spuriously 0. So, 3 plus 1 equal to 4 you will find that this value is falling because again 0 plus 0. But again after what happened at the level of 5 is a second become 1 this part is 1 and this part is always also 0 at that time, but this has now become a 1. So, if this has become a 1 then what is a that the case at t equal to 5 this is equal to 1, but this is 0 plus 1 delay at value of 6 it will become a 1 because this 1 is actually driving this.

So, if you look at this diagram it will may whatever I have told you will make this very clear. Then this I will going to just think about this at T equal to 5 this has become 1 at T equal to 5 this guy is 0 stabilized and therefore, this 1 is reflected here as 1 at the time 5 plus 1 equal to 6. So, at 6 you are going to get the value of 1 which is the proper and stable value and but here there are lot of transitions involved.

Now, what now you can see that if I have a time period of 7, I can easily class B proper value. But if I have got the value of time period as 6 4 3 or something other than I will going to get a wrong value as the output. So, you can think that this one is going to have a proper clock period of 7; I have designed a circuit for this. Now, by some means say we are all expecting that your answer should be properly latched by 6 time period. But due to a delay fault let us assume if you assume that this is not operating at 4 it has it become as 4 or it has become delayed as 5.

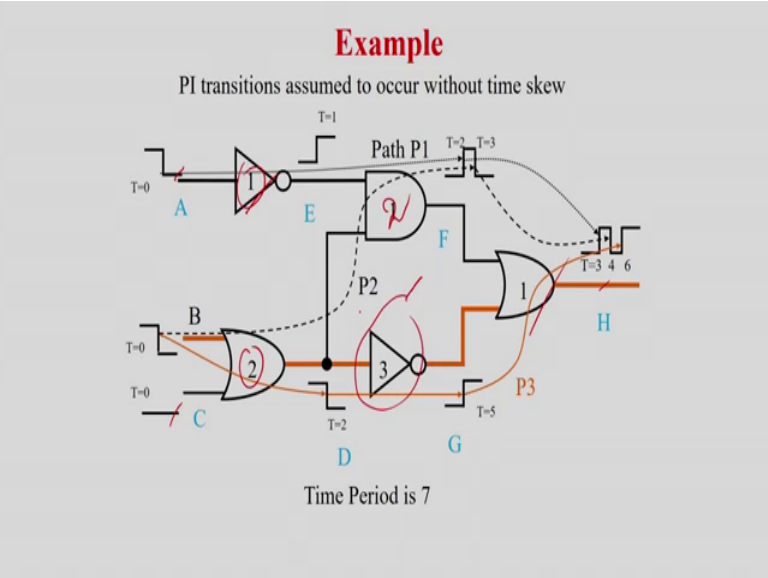
(Refer Slide Time: 34:58)



In this case what is going to happen? This value will be latched properly only after the time period of 8 only a time period of 8 to it will increase because of this path. So, you are going to get the, you are going to get the wrong result. So, what this example has shown you have to analyze this properly. Because, I have just told you also need to do your pen and paper work then only you will be able to understand properly what is happening. But you can easily figure out that there is always an intact of the clock because all the outputs are shifted by the delay involved in this.

But properly we are because while designing we know this is guarantee in 1 nanosecond is guarantee in 1 nanosecond, 3 nanosecond, 1 nanosecond and that only you will we have calculated the output will be stabilized by 6 6 nanoseconds. So, we have designed the circuit with a period delay of 7, but due to manufacturing problem this may have becoming 5 this, but of course, if you think that just you have to you can also analyze what happens if this becomes true.

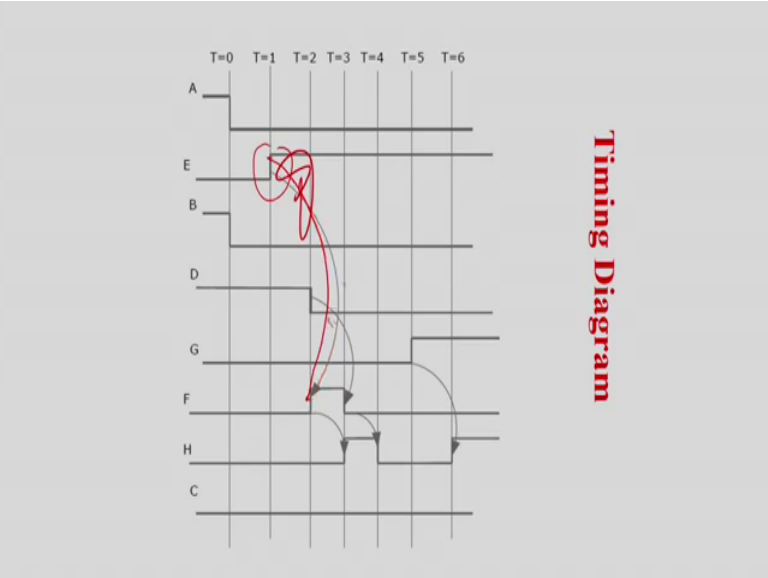
(Refer Slide Time: 35:54)



When it when it affects my output I think it will not because we are always we have to always consider the worst case those things we have to analyzed. So, I am not I mean just you need to analyze then find out what is going to happen that if this becomes delay to. But definitely if this 3 becomes 5 then basically; obviously, you are going to get a wrong answer if you latch at 7.

So, we have to find out that all the gates are having a proper rise to delay and fall delay. So, that always you are going to get the answer at the time second for which your circuit is design. Because all these transient analysis all these delayed analysis will be done based on the guarantee which is given by the gates. But after manufactory certain gates may have more delays which we have to find out the testing.

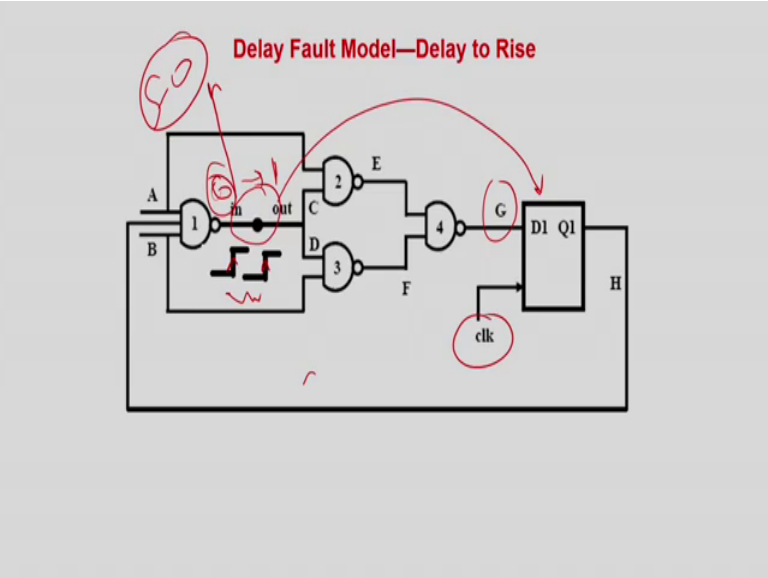
(Refer Slide Time: 36:34)



So, this is what is the example this is the entire timing diagram which corresponds to this circuit. So, already I have explained you this timing diagram in my in words and by making proper mappings and trying to explanation. Trying to explanation of the what time it is falling, and what time it is rising, one of this spuriouer values etcetera. It is your job now to take a print and match this 2 that whether they are matching properly or not just I mean giving you idea that whenever I say that E is having an impact on the rising edge of F.

So, basically E is having a effect on the rising edge of F. So, if there is a rise over here so there will be actually also rise over here that is actually the impact. So, you can see that this rise is having a impact over here by this way. So, this way the whole time being diagram can be analyzed so you can just sorry. So, this 1 is having this rise is having an impact on this rise. So, this way we have to analyze the whole time diagram anyway I will explain you this.

(Refer Slide Time: 37:28)



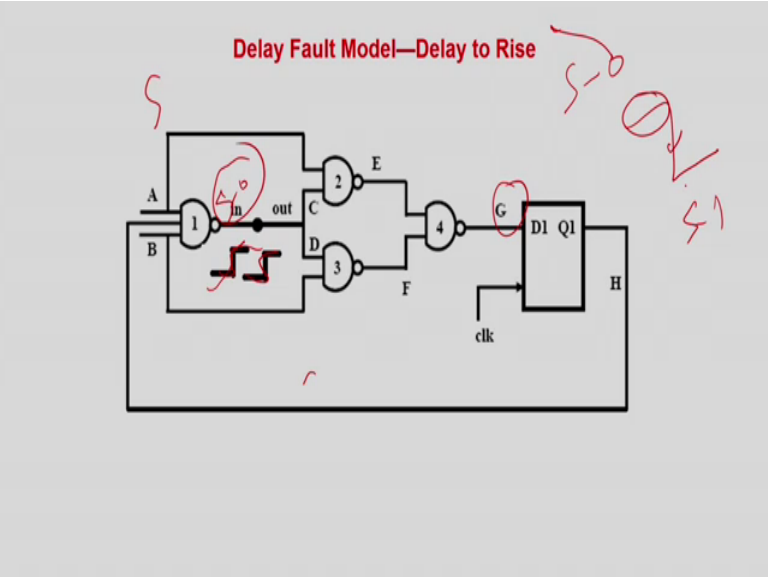
Now, we are coming to our main job that is how to test and how to generate test patterns for delay fault model. So, we will taking a simple example like faults can be in all the lines like all the lines you take over here you have to consider delay to rise and delay to fall. So, we are going to take an single fault and try to see how test patterns can be generated and what you have to do etcetera.

So, let us assume that in this gate we are going to assume a delay to rise fault. So, this is the rising time we assume that it will be delayed so that the value will not be latched. So, we have to assume that this delay is so, much that it will not be able to properly latch over here. So, it will latches stale data not the proper data so; that means, what very simple to analyze that you would have been probably gone from 0 to 1, so; the value would have been 1. But due to this delay to rise for due to the still having a 0 so this is very synonymous to stuck-at fault.

But we have to do this testing at a very very high speed. That is proper frequency of the clock we have to do it, we have to execute it in a very fast manner. So, that this reflection that whether it is still stuck-at 0 or whether it is gone to 1 has to be reaching over this in the proper frequency then only you will be able to know that there is no stuck-at fault. That means, there is no delay, but that has to be done at an extremely fast rate. Again reiterating this is very important so if you just look at this gate so, what we are saying the delay to rise means some rise delay is there. So, that you are not going to latch in a

proper fashion that means what do you mean by delay to rise; that means, this line is stuck-at fault.

(Refer Slide Time: 38:57)

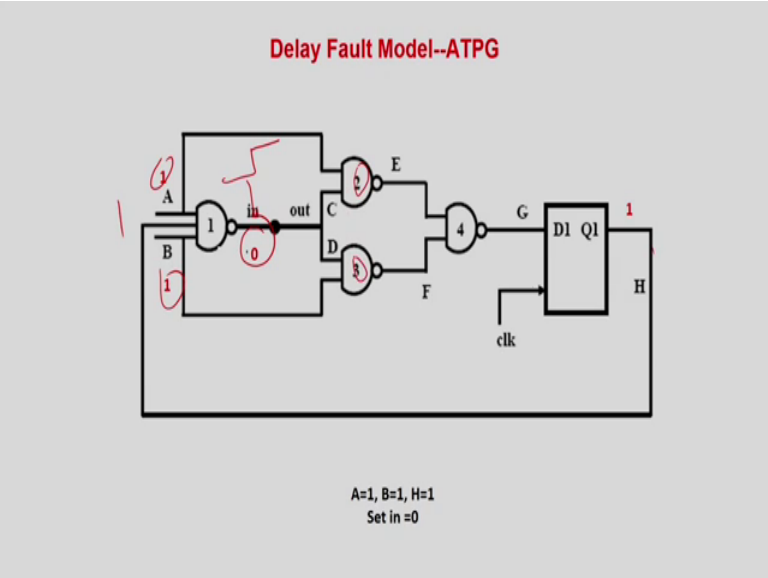


It is not rise properly so I will be doing stuck-at fault testing only, but I have to do it a very very fast rate because this is not a permanent stuck-at 0 fault given proper time it will; obviously, rise and get the normal value. So, I have to do this stuck-at fault is very very fast because unlike a proper stuck-at fault it will not remain stuck-at fault for ever. It will remain stuck-at 0 only for a small amount of time, but the small amount time is enough to means this clock edge over here.

So, what do you have to do we have to do this stuck-at fault is that are very very fast rate or the required clock rate. So, then only you will be able to delayed. So, testing philosophy of delay faults is very similar to stuck-at fault delay fault is delay to rise fault is stuck-at 0 delay to fall test is stuck-at 1 because we will be stuck over here.

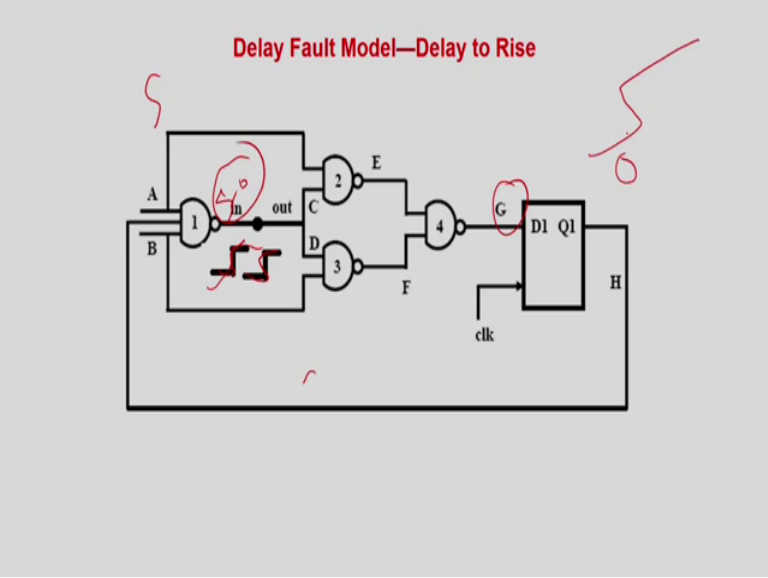
But that is this is delay to rise and this is delay to fall. So, delay to fall is nothing, but your stuck-at 1 and delay to rise fault is nothing, but your stuck-at 0 fall I think you will easily appreciate the fact. But, you have to very very careful very very careful that we have to do it in the very fast manner. Because proper stuck-at fault means it will be remaining at 0 or at 1 forever, but do to delay only for a short amount of time which is, but more to means the clock edge did that delay is there, but after that it will become a normal value o we have to be very careful when testing it.

(Refer Slide Time: 40:12)



Now, example so, assume that this is the stuck-at fault we have to test it. So, basically what you have to do of course, you have to apply a means basically what you have to delay to rise so, it is a delay to rise.

(Refer Slide Time: 40:20)



So, initiate will be of course and one very important one when you are testing delay fault it will be never a single pattern it will be a 2 pattern. Because first you have to ensure that 0 and then you have to quickly make it rise. So, there is; obviously, of course, so, first that first thing is that you have to set in the value because a stuck-at fault testing is

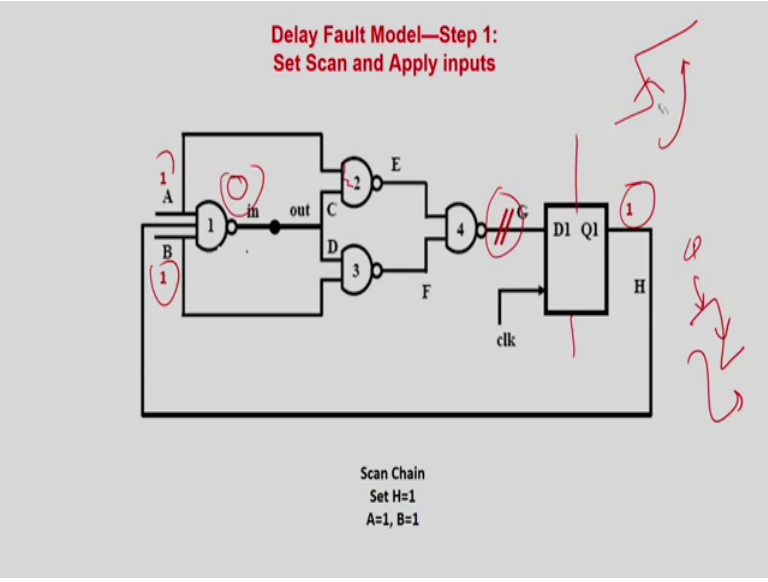
always ensuring that basically you have to a you know single pattern test. If the scan chain there will be scan value and the input pattern to test it.

But in case of delay fault very; obviously, it will get 2 patterns because I first stack 0 then you quickly make it rise and see whether it is happened delay to fall fault means you have to make it 1 and quickly make it answer to be 0 and see that whether it can be done in a proper amount of time so, of course, 2 patterns are required.

So, here we are testing the delay to rise fault so, of course, first you have to make this line is 0. How we can do it A equal to 1, B equal to 1 and this is the feedback from the flip flop. So, this also (Refer Time: 41:09) 1 NAND gate so 1 is a NAND gate. So, in this case do not think that these are the delays of these are this is corresponds to delay of this is not like that. Because this figure and this figure slightly no integer difference is there. Here the these numbers present decay delays, but here I am just naming the gates.

So, if you think all the gates basically a NAND gates so to make the NAND gate output equal to 0 you have to put all 1 1 and 1. So, applying A equal to 1 B equal to 1 is very simple, but again, but again the this feedback line I have to make as a 1 which we can done by a scan chain.

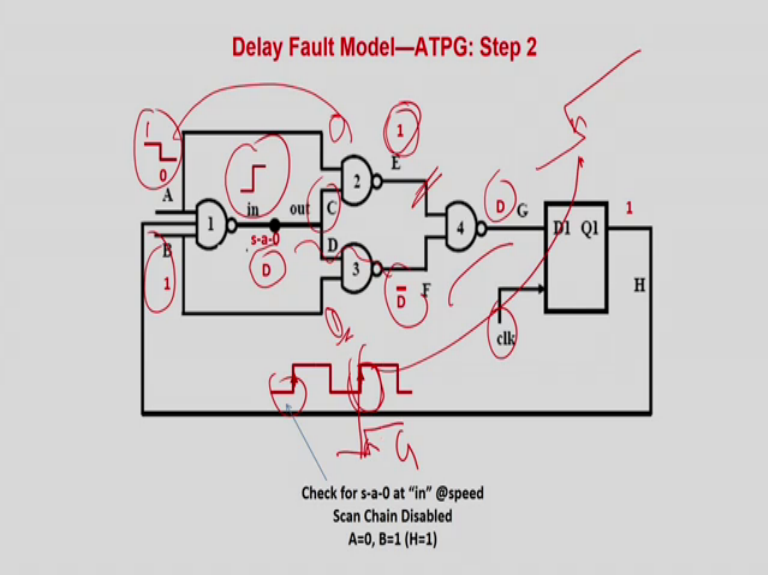
(Refer Slide Time: 41:39)



So, I am first decoupling the clock then I am making decoupling the circuit from the flip flops; that is by a scan chain by scan chain actually I will give a 1 over here and I will

apply a 1 over here and I will apply a 1 over here there is the primary inputs. Feedback can be easily done by scan so this is I am not shown over here this is scan which I can easily do that I can make it as a 1. So, you are going to get the answer as a 0 1 this is very simple.

(Refer Slide Time: 42:02)



Now, the concept of delay is coming so initially this is very slow where I am sending this value to stuck-at 0 by means I am going to test this rise fault. So, I have to make this is a 0 for this I do not require any speed. The speed will be only important for here if it is a delay fall a delay to fall fault. So, setting a 1 is very simple, but this fall we have to do it the at speed testing this is actually called at speed test. But setting the initial value setting is not required to be at speed; so I am apply the scan value in a very simple manner.

Next so, this is already 0 because I have put a 0 over here. Now, I have to assume that this is stuck-at 0 fault over here because if there is a stuck-at 0 fault then only the rise with not happen properly, but again as I told you this is a stuck-at fault maybe in case of a delay fault is a temporary stuck-at fault given enough amount of time it will become one, but I have to do this at a very fast manner. So, what I have to do so this is a stuck-at 0 fault over here.

Quickly I have to make this right right at a very very fast rate of course, after applying this scan value I will actually again stop the scan. So, it is normal output over there as you can see this is again connected to the circuit. So, I am going to give a lower edge this

one. This one is actually happening at this clock edge and it is happening very very fast and at speed testing. So, if it is at the circuit they are promise to work at 1 gigahertz this clock is also be at 1 gig and this pattern is also applied at the 1 gig.

So, in 1 gig whatever is adequate, I am making this line from 1 to 0. So, this line becomes ones to 0 it as you know that it is a NAND gate. So, it will it has to be quickly rise from 0 to 1 and if it rises within the proper amount of time then of course, your answer will be correct. Just like again a single stuck-at fault model we assume only 1 gate output or input as the fault at 1 point of time So, when I am testing for gate number 1 2 3 and 4 and assume to have no delay or stuck-at faults. So, I am making this from 1 to 0 immediately it should rise from sorry 0 to 1 within the proper delay.

So, now that is simply I am going to apply a 1 over here. So, you just test that there is no stuck-at delay for, but I am doing at the extremely fast rate and again I am going to get less this value at the next clock edge. Like an unlike a simple stuck-at fault I will not wait for indefinite amount of time to wait for the results. Because if I wait for indefinite amount of time this will definitely maybe if not fast definitely it will take a very long time and it will rise. So, I am not be able to catch this delay fault.

So, I will immediately use this clock edge to capture the value over here. So, that if the stuck-at fault stuck-at 0 is there and it does not give the correct answer by this amount of time I will say that these are delay to rise fault. Maybe if I will be allow 10 more clock pulses to go this will of course, become a 1 and I will going to get the correct answer. So, there is no not a stuck-at proper fault that is static stuck-at fault is not there but it is a delay fault.

So, similarly it will stuck-at 0 fault, but it is for a very small amount of time when I have to hold detected at that small amount of time. So again so whatever I am making a very fast rate 1 rate this at this time period and so I have to expect the that will going to be write gig from 0 to 1 at a very fast rate. So, stuck-at 0 so, this answer is the that algebra is said that normal case it is a 1 fault case it will be a 0.

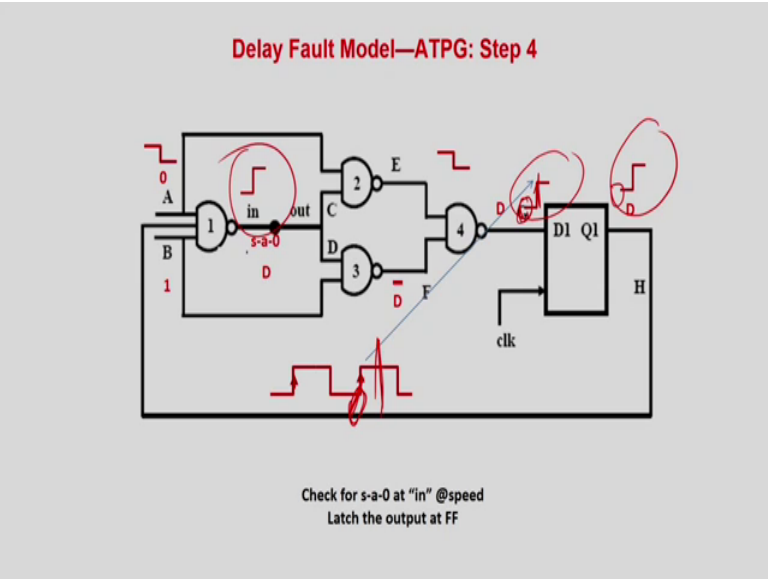
Behind again keeping it as 1 so this is basically your output propagation fault and of course, if this is a 0. So, this is also a 0 by this fact because I am assuming that the gate 2 or this line denotes a any delay faults. So, basically this 0 is immediately reflected over here and of course, so, whenever any of the inputs of the NAND gate is a 0 you will

know that it is a 1. So, even if this path is having a delay fault, but basically because of we are assuming that gate 2 is fine no delay and this 0 basically will be making it 1 at the proper amount of time.

So, now basically we are now again this gate number sorry. So, if you look at it so, this is the value of a 1 over here and this path basically is falling as a D prime. So, because of this controllability path so this is I am putting a 1 so, D prime will be coming over here Similarly, this is a 1 over here and this is a D prime so this is the propagation path because of the inversion it will become a D. So, D means what? This is a normal 0 and fault equal to 1. So, at this clock edge I am going to take this diagram. So, what should happen before the arrival of this clock pulse this D should appear at G.

That means, what so this output if you think of this clock edge then. Then if you think this is the clock edge then the output at G is D that is it should become 1 much before this set up time of this. If this is the case that your G becomes 1 much before the arrival of this clock edge which is just 1 gig that is this clock edge so, you are know that there is no delay for that this. But, if there is a delay that is if this edge is not coming over here this is a delay. So, it may be delayed by this amount of time then you know that I will not be able to capture a 1 I will capture a d 0 at G then you know that this was definitely a delay fault.

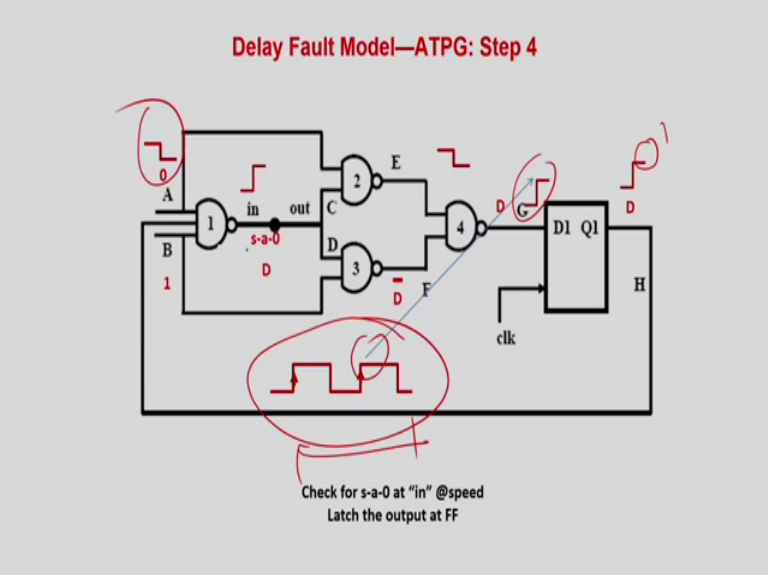
(Refer Slide Time: 46:50)



So, next what we do we try this picture shows this. So, basically we are trying to capture the D over here and absolutely at this clock period and then your value should be coming over here as properly latched. So, if these phenomena this rise actually happened someone before this is the time then there is no delay over here and you would have captures the right thing.

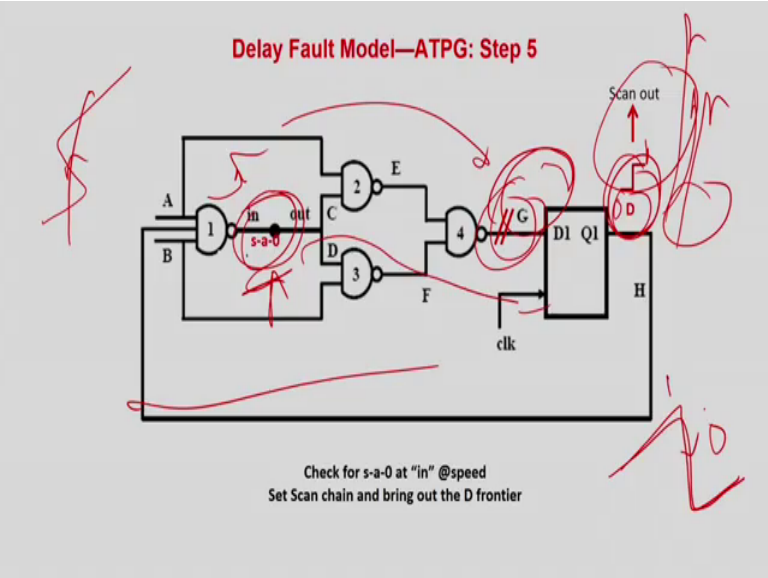
But, if there is a delay over this you are going to get a, you are going to capture this 0. Because rise may be happening over here this may be happening over here at this point of time you will be basically catching a 0 over here and in this case you are going to get a 0 over here. So, in that case there will be a delay fault, but if of course, everything is proper there is no delay fault, then you are going to get this proper latching you are going to get the answer at this point of time which is one. And you are know the there is a no delay fault and this test has to proper.

(Refer Slide Time: 47:35)



Now, this has been latched so only these 2 clock, I have to applying proper frequency first you have to give at 1 gig answer you have to capture at. The next clock this is actually a proper 1 gig clock which is the clock frequency of the testing test or the design of the circuit.

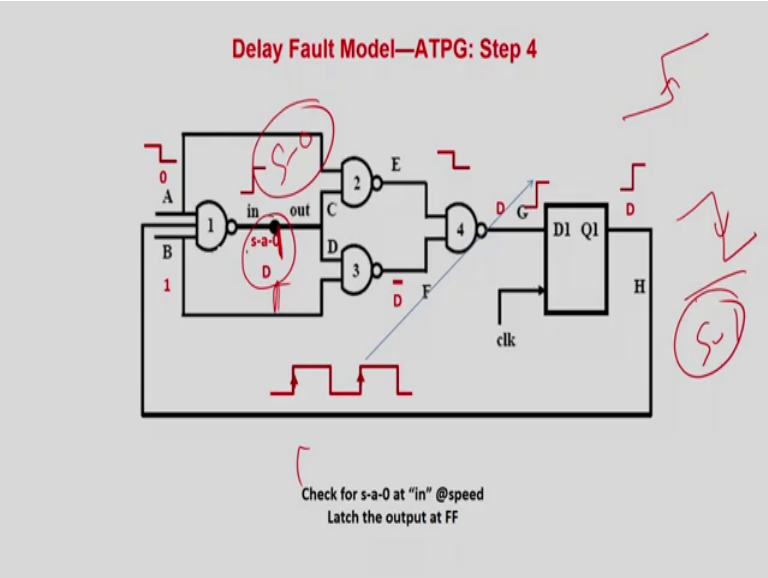
(Refer Slide Time: 47:51)



And then when this value has been latched again you have to be couple the flip flops on the circuit and you have to that put out the output whatever you have latched through the scan. This can be again at a slow speed so, after you have checked for this you said this scan chain and bring out this value this you can write a very slow speed of time because anywhere this value has to be read. So, if you are getting a value of 1 so, it is no problem no delay was there and if you get the value of 0 over here. So, it is a at the output will 0 so, know that there was a delay to rise fault over here.

This is how basically we go for testing of delay faults in case of circuits. So, what is it is very similar the algorithm is very simple. So, what you do just like this is very simple just like you know stuck-at this is ATPG is very simple just like for our normal combinational circuits so if it is a delay to rise fault.

(Refer Slide Time: 48:40)



First you have to set these 2 value set this value to 0 that is the first test pattern. Second pattern is that you have to just test for a stuck-at 0 fault over here. But this you have to do at a high speed of clock. So, this is just like combinational circuit testing, but only one more step is extra is required that you have to if it is a 0 rise delay to rise then you have to first set it to 0 and then basically you have to check for a stuck-at faults.

If it is a delay to fall fault then basically you have to apply a 1 over here and in the next clock pulse at speed you have to find out whether it is actually falling from 1 to 0 that is you have to take stack for a stuck-at faults so, two test patterns are required. So, the complexities just somewhat doubled then the normal combinational or sequential circuit testing.

But only thing is that you have to use a high speed tester because you have to apply these 2 clock pulse at a very high frequency. But again you have to note that the first phase in which case your setting this value and as it is same any time when you are pulling this value of the output of the scan chain the reading the value setting this value and reading the final value they can be done at the (Refer Time: 49:45) speed only this change and this latching of this change over here has to be done at speed. So, 2 clock; 2 clock 2 test vectors have to be applied at a very high speed and you have to check the output.

But setting the value here and after latching the value moving it to the output can be done a slower speed. So, algorithm is more or less very similar to a normal scan chain based

testing. That is, but only one more step is required when you have said the lines properly. So, that is very simple setting the line properly means it is just sensitizing the line and justifying it. Say the say the absolute test pattern is sensitizing proper propagating at basically it is justified.

So, the same ATPG algorithms we have done can be very easily with very slight modification can be applied over here. But only 1 thing you have to observe is a 2 test patterns are required and you require really require high speed tester to test it that actually increases this cost. Because algorithmic complexity and test whether a generation complexity is not much higher, but of course, it is just double. Because in case of stuck-at faults basically you have to test for a single pattern and here you have to apply for 2 patterns.

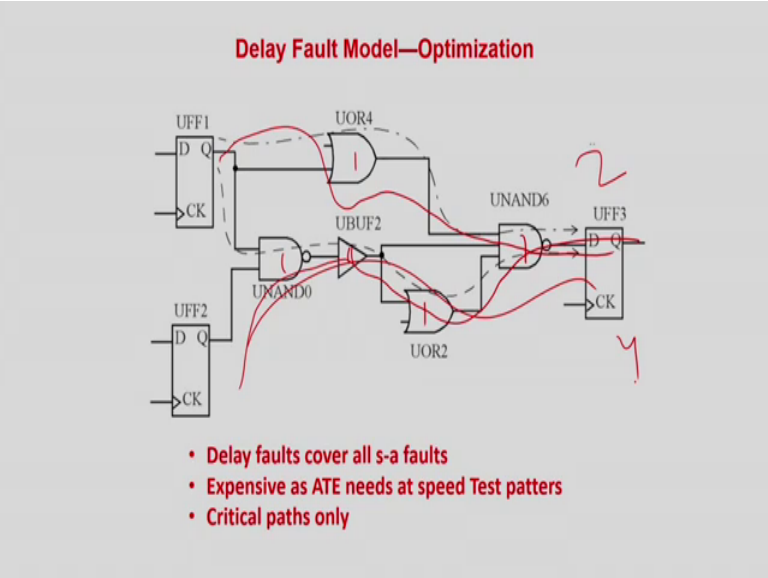
So, may be twice the amount of time we required to apply the test patterns. So, more time in the tester and of course, slight increase in complexity of the ATPG algorithms. So, you may tell me that I require a more time in the tester that is true you have to play slightly higher cost, but more of a concern is then you require a high speed tester, that annually actually making it more complex.

But one thing you should also understand that when I am testing for a delay to rise fault. Of course, by default you are testing this stuck-at you need not required to test this stuck-at fault again because of course, if your line becomes from moves from 0 to 1 at a very high speed and of course, definitely there is no stuck-at 0 fault.

Similarly, if you can assume that it generally at very and required speed it goes from 1 to 0. So, definitely this line does not have any stuck-at 0 fault so you can suck at 1 fault sorry. So, you can easily think that I will do bridging I will I will go for delay fault testing of all the gates and stuck-at faults are automatically cover that. In fact, you are true so in fact, if you go for delay fault testing of all the gates stuck-at fault models stuck-at faults are not required to be tested.

So, in that way you can say that I have saves applying some test patterns for the stuck-at faults. But actually we will not go by this philosophy we always try to go for stuck-at fault testing first and then we only look at something what is called the critical lines which has to be tested for daily faults.

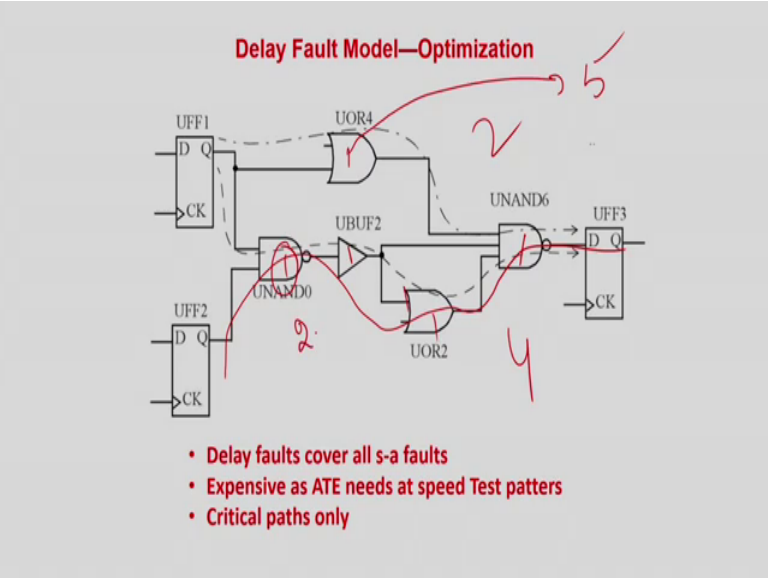
(Refer Slide Time: 52:01)



Like for example, if you look at this circuit this is 1 line which is taking 1 to 2 2 gates get the answer the may be the all the gates having a delay of 1 you assume. So, the delay here is 2; another part if you look at it is 1 2 3 4 1 2 1 2 3 4 so, by this path the delay is equal to 4. Now some people always see that as delay fault testing is very expensive by the tester because you have to apply many higher frequency patterns because again we have to understand that a tester have lot of channels and it is used by multiple partners. So, if you tell that I have I will just require high speed test patterns for some this amount of time for other parts I will be doing. So, pattern time it will be a less expensive option.

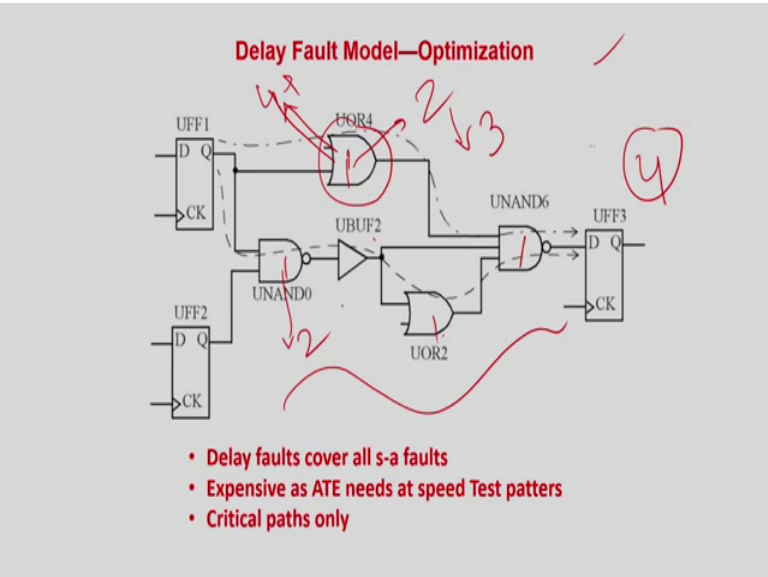
And, but if you always say that no I always want to use the high speed channel so it will be a more expensive of set. So, what we will try to do is that with in the same tester also people want to use the high test pattern high speed channels less compared the slow speed channels. So, I will in this case I will only like to test the delay faults for these gates because, if this gate is 1 and 1 this is again 1 1 1 and 1 this example.

(Refer Slide Time: 53:03)



So, this is 4 and this is 2 so, this is very very rare case does this gate delay will be made may become from 1 to 5 or something like that. But, if any of this gate delay becomes from 1 to 2 because, anyhow in this example you are not going to get the answer before how much 1, 2, 3, 4.

(Refer Slide Time: 53:20)



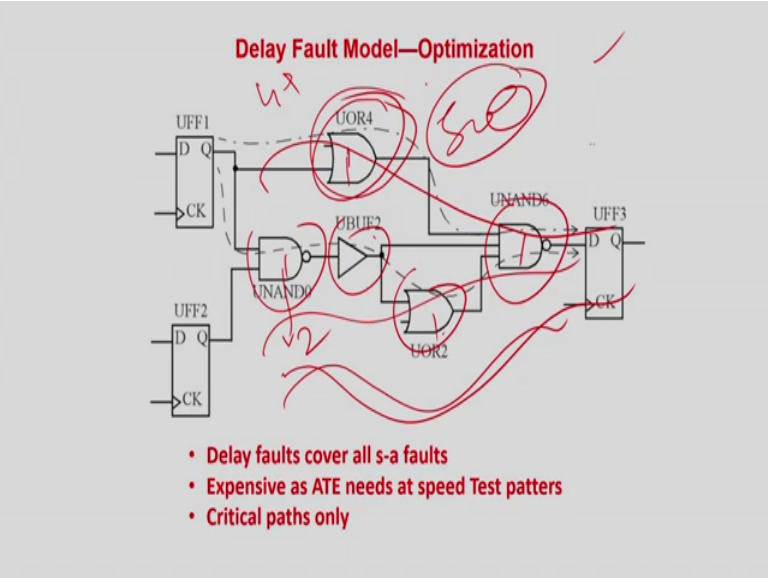
Before 4 unit of time you are not going to get the answer from here because this is something call the critical path. Even if this gate becomes so, 1 to 2 is the delays is there I do not bother. Even it becomes 3 I do not bother because 3 plus 1 equal to 4 of course,

if it becomes 4 plus then I have to be worried about it, but in a very. If you look at the statistical sense such a stray case that is some delay from 1 to 4 will not happen basically, but it is very likely that it may become from 1 to 2.

So, basically what people will more likely to test is that they will not test this gate or they will not test this path. They will actually test this path. So, this is something called a critical path. So, whichever is the most time taking path because everything will be decided by the most critical path of the most path of the most amount of B A. So, people will find out selectively some kind of these critical paths and they will go for delay fault testing only on these paths.

Because delay fault testing is expensive because you have to apply test patterns in a very high speed, but again for those paths where you have done the delay path testing you need not go for any kind of stuck-at fault testing because as you have shown delay fault testing automatically cover stuck-at faults So, we not we are going to going for any kind of test a stuck-at fault in this lines.

(Refer Slide Time: 54:25)



Of course for this gates we will going for the stuck-at fault testing and we will not go for any kind of delay fault testing. So, even if as we have seen so even if you have seen that basically delay fault testing can come out stuck-at faults, but we do not mean that we will totally through away the stuck-at fault model and applied delay fault and all the gates.

Rather we will apply delay fault is only at the critical path for all other cases we will go for the stuck-at fault testing only.

So, the good analysis we have seen over is that a delay fault 1 is a very powerful model covers many other 4 models. But generate it is more expensive to apply so we will take care of very critical path very important paths which are more dependent on your speed of your operation of your embedded systems will be mainly testing those gets in those lines only and for others the stuck-at fault always is there.

So, with this curve with this we come to the end of this lecture on delay fault model which is very very important for real time embedded systems. In the next class we will try to again discuss on some more important testing methods which have arise for modern day beeps submicron embedded system design like built in self-test, fault tolerance, etcetera.

Thank you.