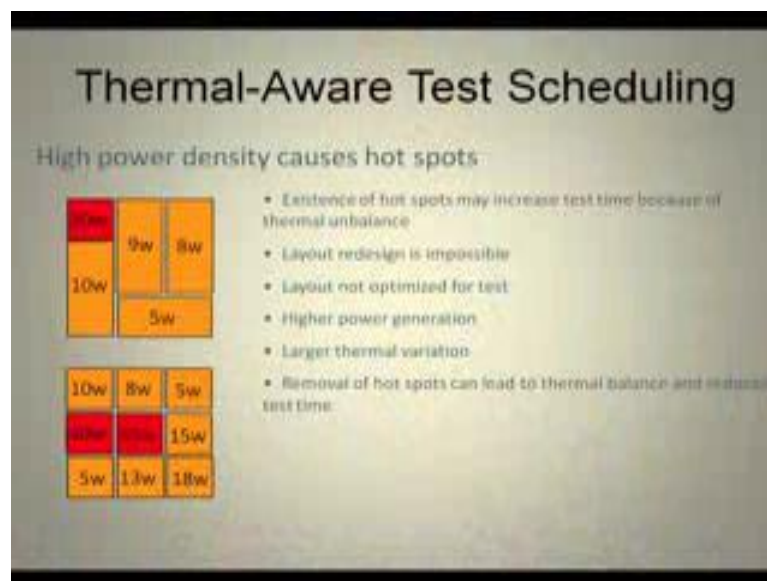


**Digital VLSI Testing**  
**Prof. Santanu Chattopadhyay**  
**Department of Electronics and EC Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 55**  
**System/Network – On - Chip Test (Contd.)**

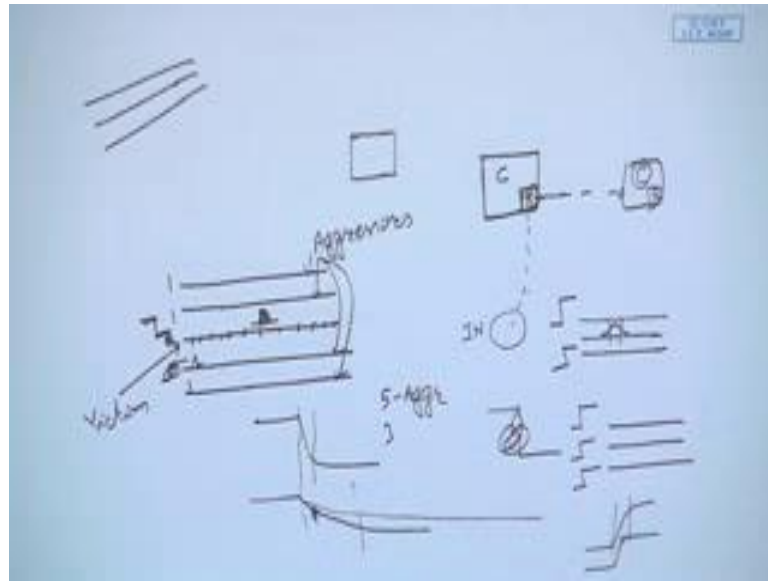
After this power aware test scheduling, so for NOC also this thermal our test scheduling becomes a concern.

(Refer Slide Time: 00:24)



Because, there are two sources of power consumption and in a NOC like this individual cores they are consuming some power in test session. And also the routers are going to communicate traffic through them or the this packets through them as a result they also consume power.

(Refer Slide Time: 00:54)



And the problem that we have is in a SOC based system, if this particular core is not active for some time we can do some we can apply some power management technique to reduce the power consumption there.

But in case of NOC what happens is that this router and core they are often integrated. So, we make a tile and this tile this may be at one corner of the tile we put the router. So, this is the router and this is the core, so that way this entire thing. The power management module so that comes for the entire tile, so it is not for the simple router. Now we have find that this core is idle because this core is not being tested it is not a part of our testing process it is not I O pair core and all that.

So, this can be switched off or if we can do something so that power consumption will be not be there. But what will happen is that these routers may be taking part in communicating some pattern that is coming from the input of from the input port or the input core it may be going through this router to the destination core. So, the router there it may be connected there, so this core is being tested so patterns are going like this. So, this router needs to be kept on. As a result this entire tile is kept on, so that way the power consumption of this individual cores may be mode in the testing mode not because of the fact that it is being tested, but because of the fact that it is becoming a part of the whole process; it is becoming a part of this testing process that we have for the system.

So, when we have this type of high power consumption. So, there also may be high power density positions in the NOC and that way that may start consuming with the produce lot of heat. Maybe, it may create some hotspots like this is say consume a 30 watt of power and this is say 40 watt 55 watt. So, power density may be high as result there may be some thermal imbalance.

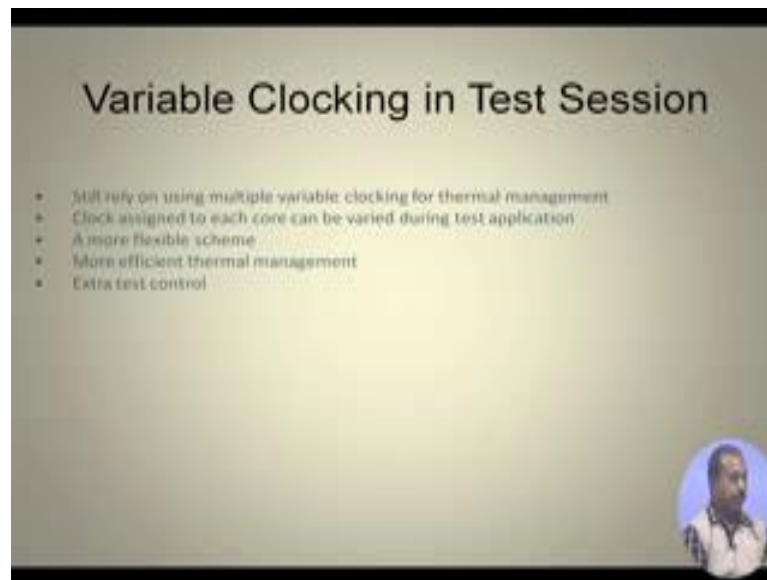
So, existence of hotspots may increase test time because of thermal unbalance. So, this delay of the paths will be not be correct and also we cannot test portion if it is very much heated. So, what will be required is that in the entire test session we need to put some gaps, we need to put some time gap at which these paths will get cooled down the routers and cores get cooled down; and after that only we continue with the testing process so that way introduction of these gaps or the idle times in the test session; so that gives rise to in the increase in the test time.

Now for the testing purpose we cannot redesign the layout. So, layout is fixed from the design angle, we cannot change that layout. And naturally layout is not optimized for test; layout is optimized for the functional operation of the system; so whichever modules were required to communicate mode then this layout editor; so it has put those modules close to each other. But for the testing angle we want to put the modules which are not consuming lot of power to close together, because they to high power consuming modules if they are close. So, the power density of the region will go up.

So, to avoid it we may like to do layout in a different fashion, but that is not possible for the test engineer. So, that leads to higher power generation. And if the higher power generation the temperature increases so that again increases the leakage power and that acts as a positive feedback. So, this power consumption increases significantly.

So, there will be larger thermal variation. So, across the chip if we see then the thermal variation will be more and that way the delay variation are more; so that is the problem. And we have to do something so that this hotspot can be removed. So, this is the high temperature region they can be removed and they can get thermal balance and as a result the test time can get reduced.

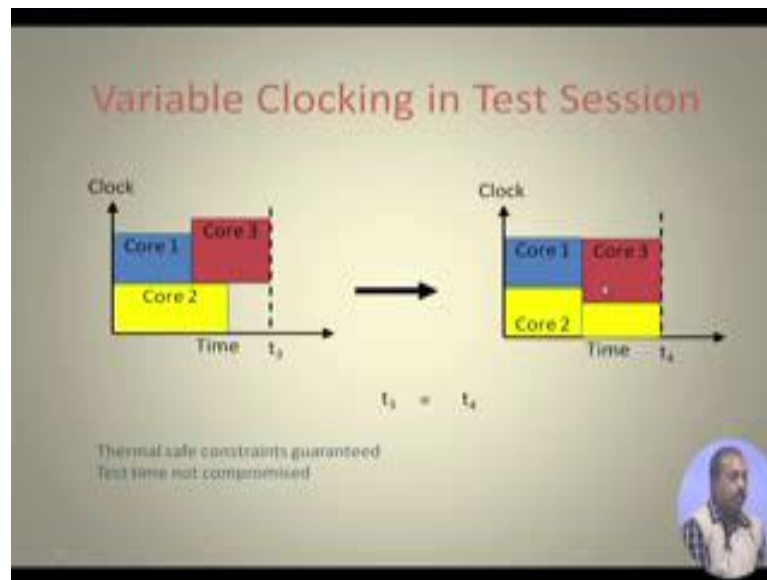
(Refer Slide Time: 05:19)



So, one possibility for doing this power and thermal reduction is that variable clocking in the test session. So, we use multiple variable clocking for thermal management: so if the idea that if this clock frequency is low power consumption will be low as a result this temperature of the region will also go down, and if you do it as to be done in a region wise fashion. So, it is not just reduced the test clock the clock frequency of a single core because of it also depends upon neighbours; for the neighbours also we have to do the something.

So, we have to assign clock, we have to do this variable clocking for a region. And then the clock assigned to each core can be varied during test applications. So, that is what we are talking about. So, naturally this needs a more flexible scheme. I must have the flexibility of varying frequency of these cores when the test patterns are being applied. So, naturally it becomes more complex; so handling such situation becomes more complex. So, more efficient thermal management policies are required. And the extra test control mechanisms are to be incorporated so that these techniques can be utilized.

(Refer Slide Time: 06:46)



So, this is a variable clock session. So, what is happening is that; here we are testing for core 2 so it is this clock frequency. So, it is core 2 is tested at some clock. And in the second case core 2 is tested at a higher clock. So, as a result its test time reduces and after testing. After testing upto this much we increase the clock frequency of core 2. As a result it is being tested at a higher clock here.

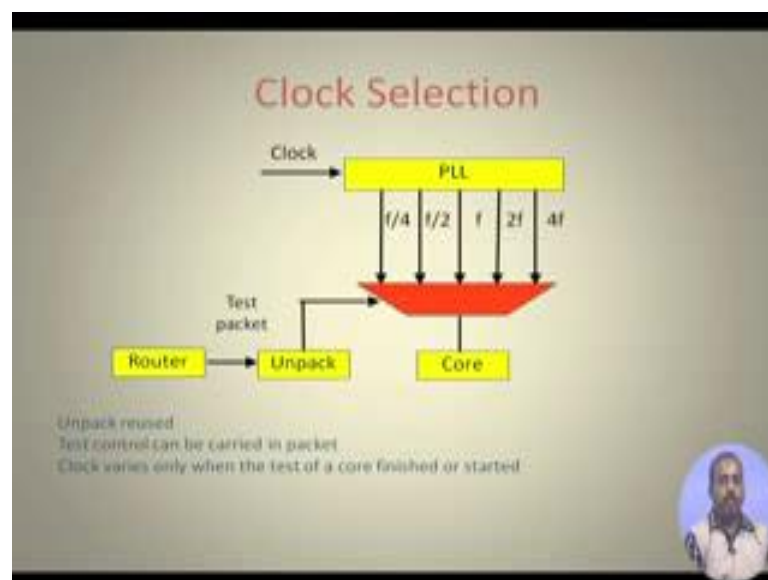
So, what happened is the core 1, so after this point we see the core 1 testing is over only core 3 testing is going on and it is not reaching the power limit that we have so we can increase the clock for core 2 so that this part can be reduced. Instead of taking time from this point to this point now takes time from here. And this also becomes the additional testing region for the core 2. So, this rectangle is also getting added into the schedule. Core 1 remains and core 3 remains and we see that the test time may be reduced by this fashion.

Now, thermal safe constrains are not violated test time reduced. So, if we assume that this is the thermal level that we have, so this is not violated here, so this is fine. However, in the second case we have got the situation like; this core 1 core 2 and core 3 the original schedule was like this. And then what we tried to do is that we try to increase the we reduce the frequency of this core 2 after it as done upto this much, we have gone upto this much, we reduce the frequency of core 2. As a result its time increases, but I get some facility in the power part. So, power part may be here there may be thermal

violation. So, thermal safe constraints are guaranteed time is test time is not compromised.

So, here may be that there is a violation in the thermal limit, because this core 3 is being tested at a higher frequency. And in the second solution the frequency of core 3 has been reduced, so it is going to consume less power. And possibly is going to ensure the thermal safety. And time is not compromised. So,  $t_3$  is equal to  $t_4$ . So, time is not compromised only thermal safety as been brought into by changing the frequency operation of core 2 and core 3.

(Refer Slide Time: 09:31)



So, we can have this type of clock selection mechanism. So, the clock that is coming from the tester it may be given to a phase lock loop and this phase lock loop can be designed to generate different clocks by  $f$  by 4,  $f$  by 2,  $f$ ,  $2f$  and  $4f$ . Now from the router when test packets are coming so this is unpack and test packet itself will tell at which frequency this core will be tested. So, this is unpacked and then the frequency is fed to the core. So, unpacked module of this network interface module so that is again going to be reused.

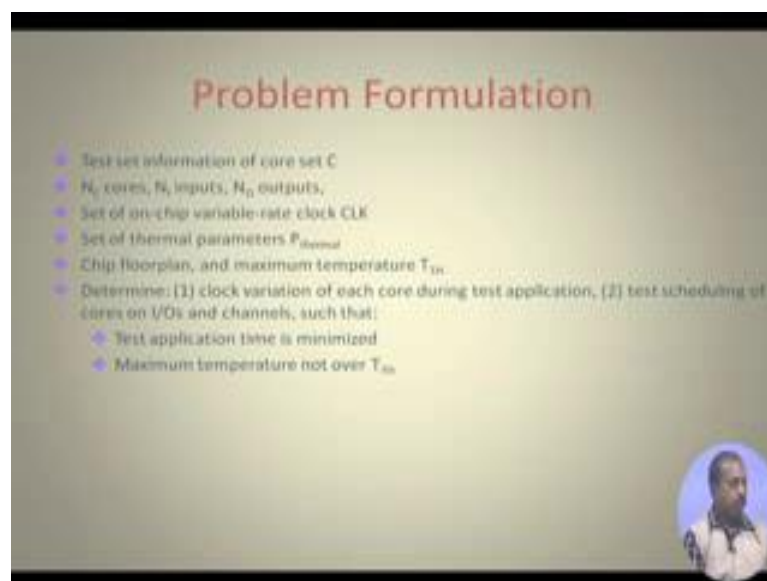
So, this is in general network interface module is converting or unpacking a message packet and forming the message and giving it to the core. The same network interface modules, unpackaging sub module it can be used to identify for the test packet what is the frequency part and it can be used to feed this multiplexer at to select the frequency of

operation for this core test; so unpack is reused. So, test control can be carried in the packet. So, this is the thing that I was talking about.

So, the test engineer might have decided that core that test patterns this core should be applied on this particular clock. And of course, for variable clocking it may decide for test patterns, so  $t_1$  to  $t_{100}$  it will be done at some frequency; and  $t_{101}$  to  $t_{200}$  will be done at other frequency. So, that can also be done, as the packets are being transmitted. So, there may be special packet which will identify that this is a packet for clock selection.

So, when that packet comes the N I, so it gives input to that multiplexers so that the appropriate frequency selection part changes. That way this clock rates can be selected and the test time and the power consumption and temperature that can change. So, test controlled carried in packet and clock varies only when the test of a core is finished or started. So, this can be done. When a particular test pattern has been applied then only we can do the thing or we can do it like this that we divide the whole testing of a core into several sessions and for we can define the session boundaries and session boundaries only this clock change over take place. Of course, it otherwise there will be problem in the synchronization.

(Refer Slide Time: 12:14)



**Problem Formulation**

- Test set information of core set C
- $N_c$  cores,  $N_i$  inputs,  $N_o$  outputs,
- Set of on-chip variable-rate clock CLK
- Set of thermal parameters  $P_{thermal}$
- Chip floorplan, and maximum temperature  $T_{th}$
- Determine: (1) clock variation of each core during test application, (2) test scheduling of cores on I/Os and channels, such that:
  - Test application time is minimized
  - Maximum temperature not over  $T_{th}$

So, we have got test set for a information for a core set C there are  $N_c$  number of cores,  $N_i$  number of inputs and  $N_o$  number of outputs; we have got a set of on chip variable

rate clock, so CLK signals we have got a set of variable clock rates. A set of thermal parameters like how for thermal simulations so what are the information needed. So, they are actually the thermal parameters this dielectric thickness, then this heat spreader then the size of the heat spreader, this size of heat seeing so that we can do thermal simulation.

Then the chip floor plan and the maximum temperature  $T$  threshold; as I have said most of the systems they have got a dynamic thermal management policy and they have got a threshold there. So, whenever that threshold is exceeded the system is automatically shut down. So, we do not want that in testing each that particular threshold so we need to control our operation below that threshold.

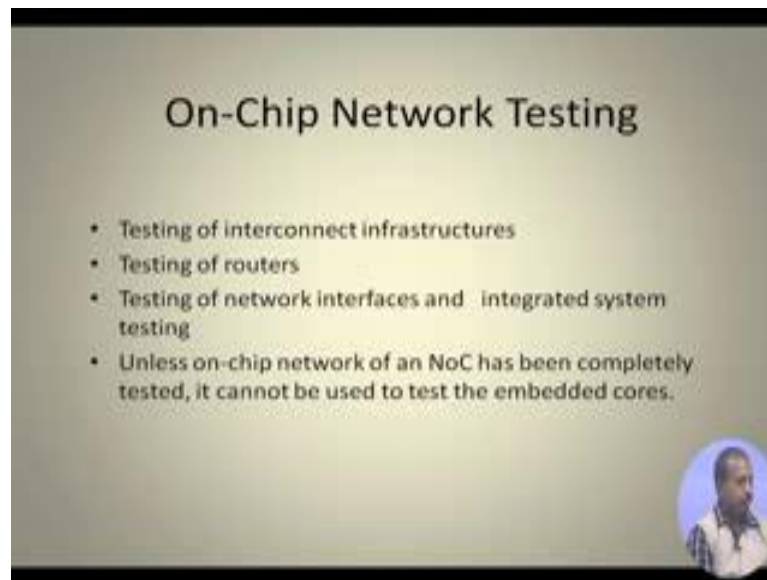
So, that is the maximum temperature  $T$  threshold may be that maximum threshold is say 85 degree centigrade, so we will be operating at within say 75 degree centigrade. We will try to operate within 75 degree centigrade. Being pessimistic that in actual operations, in actual testing session it may become about 80 or so, but still in that case we have got a guard of 10 degrees so that this DTM is invoked and the test session is not talked between.

So under this given this temperature threshold and all that, so it determines clock variation of each core during test application; how this clock rates will be varied. Test scheduling of cores on I/Os and channels; so this how I/O, which I/O core is going to be tested, on which I/O channel and at what time they are going to be tested. So, those parameters are to be taken.

And the test application time objective is to minimize the test application time. So, this determine this clock variation and test scheduling of I/Os and channels such that this test application time is minimized and the maximum temperature is not over this  $T$  threshold. So, as I said that is a DTM related threshold. So, this is the problem to be solved.



(Refer Slide Time: 14:40)

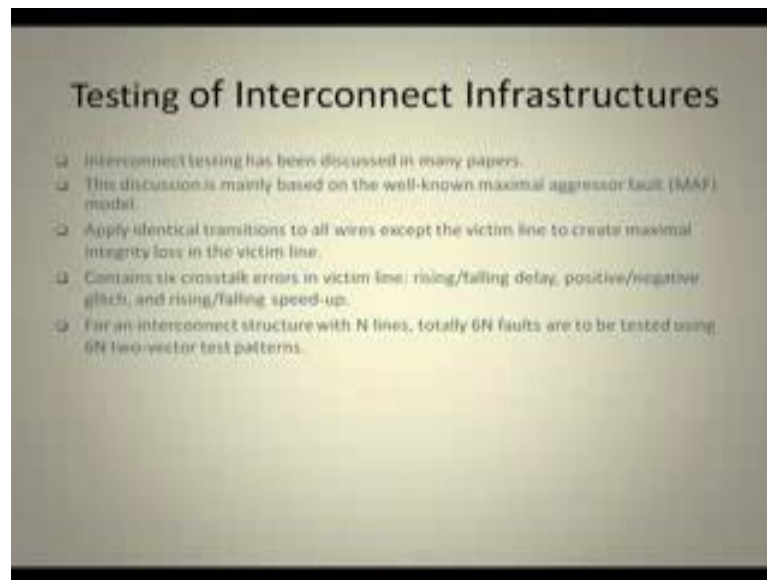


And now the rest of thing how do we solve this problem. So, we have seen many such heuristic techniques, there are many heuristic methods that have been proposed. You can also makes some exact problem formulation by means of some ILP, you can use some meta search techniques. So all those techniques have been reported in the literature and we can take help of those methods to get this problem solved. So, we will not going to the detail.

Rather we will be looking into the other part of it; like how are you going check the on chip network part that is the components of the network that is routers and links how are you going to test them. So, testing of interconnects infrastructures, testing of routers, testing of network interfaces and integrated system testing. So, unless on chip network of an NOC has been completely tested it cannot be used to test the embedded cores; so this is definitely true.

So, first we need to test the individual routers and once we get the confidence that the routers and links are working fine then only we can transport the test patterns those routers and links for to the test the individual cores; that as to be done.

(Refer Slide Time: 15:58)



So, how can we test interconnect? So, interconnecting testing has been discussed in the literatures many documents are there, they are based on a one model which is known as maximal aggressor fault model or MAF. Here what happens is that, in case of NOC we have got a set of parallel lines. Now these parallel lines if they are running, now out of these n lines we take some of one line as weak team. So, say it may be if I got five lines may be this is the victim line. And what we try to see that if there are transitions; so if there are transitions in these lines what is the effect on this line?

So, this line is called a victim line, and all these lines they are called aggressors; they are called aggressors. So, the effect of these aggressors will come to this victim and it will change the victim line. So, may be that if this line is say 0 if this line is at 0, but there is a transition in aggressors. Since this links are operating at a high frequency we have got this capacity coupling between them. And also there are if it is gigahertz range then there may be inductive coupling as well. So those coupling effect; so this will have effect that will affect the logic level that you will get here.

So, if this line there is a rising pulse here, so it may so happen that this line will also be it was 0 it increases to some value. And after some time when it settles to one then it comes down to 0 and it continues to 0. As a result there is a glitch is seen here. So, that way the logic level that we get at this line is not 0 for some time, but it is something else. So, this

is called maximal aggressor fault model. Maximal aggressor fault model we have got one victim line and a number of fault lines.

So, there are 5 aggressor line models; this is that thing. Then 5 to 1 aggressor line then there is a 3 aggressor line, there are only. So, situation is there are one victim line and two aggressor lines. So, that is also there. This type of models are available this is effecting; this is creating a cross talk and due to this cross talk between the lines the signals levels may be inconsistent at some point it some glitch. And also this transition time may reduce, like if this line is say permanently high and this line is a permanently high and this line tries to make a transition from 1 to 0.

Now, this transition will not occur immediately because of the coupling between these two lines. So, it will take some time, it may be it is supposed to be like this let us say- it is in normal case may be transition should be like this, so this is the time which it decays to logic low level. But due to the presence of high signal on the aggressor line it may take more line, so it may be decays over sequence like this. So, what will happen is that if the system is operated at a very high frequency then it may sample the line and at this point and it at this logic level is still 1. So, at destination it may be sampled as a wrong value.

So, that is the problem of this is cross talk and coupling between the lines. So, that is captured by the maximal aggressor fault model. So, we apply identical transitions to all wires except the victim line to create maximal integrity loss in the victim line. So, it is maximal, so in the example that we are discussing we have taken transition in only 1 line. So, if you take transition in all the lines; so if all these lines are say at logic 1 and you have to trying to get from go to from 1 to 0 for the victim line. Then you can understand that this effect will even spread over long longer time so that will be creating the problem further. So, we apply identical transitions to all wires except the victim to create maximal integrity loss in the victim line.

Contains 6 cross talk errors in the victim line; like rising-falling delay, positive-negative glitch, and rising-falling speed-up. So, basically rising falling delay that we have seen that one of the all the aggressors are some voltage level. So, logic high and this victim is going to make a transition from 1 to 0 so it takes time. Similarly it is falling delay. So, it

is a rising delay can also occurs rising lines all the aggressors are at level 0 and the victim is trying to make a transition from 0 to 1. So, it will take some rising delay.

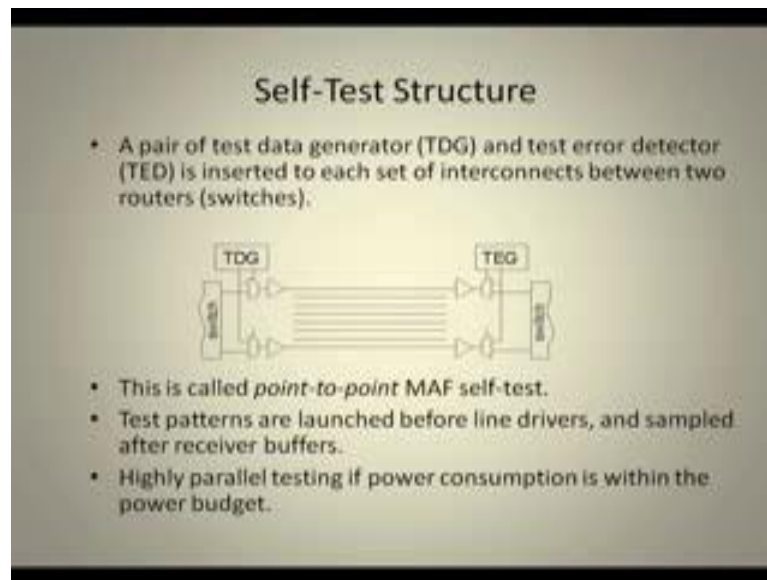
Similarly we have got positive or negative glitch, as it was telling if all this lines are making transition. So, we have got this situation and in a three wire model, now if both of these lines are going from low to high. So, as a result this line will also be temporary if this line is 0, so for temporarily it will be pulled high and after that it again go back to low. So, this duration is determined by the amount of coupling that we have between this lines so that is also going to occur; so that positive glitch.

Similarly, we can have a negative glitch. And rising falling speed-up, so this is just the reverse case because here may we have got say this a three model two aggressors and victim at the middle. Now all of them are making transitions like this- 0 to 1 transition. Now ideally this transition on the victim line should takes place in this fashion, so this as the delay rising delay. But due to the fact that these lines are also making transition, so it may be speeded up may it will be do it much faster; it will do the transition much faster So, that way that is the speed-up that can occur.

So, speed-up apparently it seems it is not a problem, but in some cases it may create some difficulty. So, for an interconnect structure with  $N$  lines totally  $6N$  faults are to be tested using  $6N$  two-vector test pattern. So, this is for the maximal aggressor fault model it has been established. You see there are six faults here; rising falling delay; positive negative glitch and rising falling speed-up. So, there are six possible faults. And since there are  $n$  number of lines so we have to have  $6N$  number of faults that we have to test.

And for each of these tests required two-vector test patterns, because first of all to put all the lines that some particular voltage level then you have to apply our transition or change it. So, naturally the change aggressors and then see the victim point. So, that way it is going to take time, so it is total  $6N$  two-vector test patterns are to be applied for interconnects.

(Refer Slide Time: 23:48)



Now how can we do this testing for interconnect? First of all one possibility as self test structures. So, a pair of test data generator and test error detector, so TDG and TED are inserted to each set of interconnects between two routers. So, this is one router and this is another router, these are the buffers that we have. And then this in normal operation this for the data from these switches will go into this links, but in the test mode this test data generator so that will generate some test data that will be coming to this line.

So, this way this pair of TDG and TED they can be used for this applying this test pattern to the links. So, this is called a point to point maximal aggressor fault self test structure. So, test patterns are launched before line drivers. So, these are actually the line drivers these buffers they are actually called the line drivers. And they are sampled after the receiver buffers. So, they are sampled as after receiver at this point it is sampled at this point.

Highly parallel testing is possible, because you see that for every link we have got this TDG, TED pair so all those tests can go parallel. So, this is a huge parallelism that we can have. But of course with the restriction that it the link power consumption so that will go up significantly and we can plus this TDG TED they will also consume power. Normally they use some sort of finite state machine to generate the test patterns for testing this fault model. So, under that it will be taking some, it will consume huge amount of power.

So, if the power consumption can be sacrifice; power consumption can be tolerated then this all the interconnects in the NOC can be tested parallel. However, that may not be the case and we have to figure out, we have to do that in groups. Maybe we do testing for a part of the interconnects and then again we repeat it for another part. So that way it can be done.