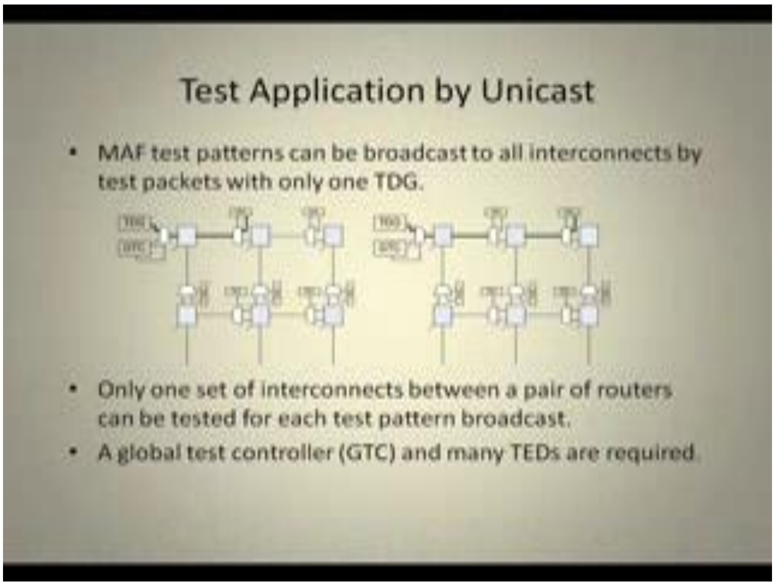
Digital VLSI Testing

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

Lecture – 56
System/Network - On – Chip Test (Contd.)

(Refer Slide Time: 00:23)



Test Application by Unicast

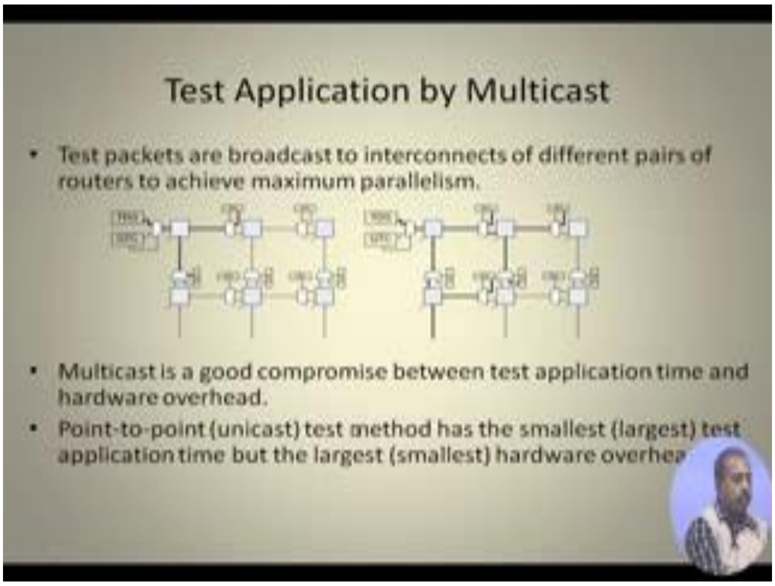
- MAF test patterns can be broadcast to all interconnects by test packets with only one TDG.
- Only one set of interconnects between a pair of routers can be tested for each test pattern broadcast.
- A global test controller (GTC) and many TEDs are required.

So, this test application process for interconnect testing, the problem that we have is that we for individual links, so we are have having this test data generator and this test error detector TDG and TED, which is lot of over rate. In the sense that all of those as I have said that they are normally having some finite state machine implemented there, so the over rate may become large if there is a large NoC with large number of links. So, other possibility that we have is to have some global test controller. So, this global test controller, so it can generate some, it will have this TDG test data generator, so this TDG, so this is generated all by at one TDG and it will be transferred to all the links. So, MAF test pattern the multiple access false module test pattern, maximal false module test patterns, they are broadcast to all interconnects by test packets with only one TDG, so I got only one TDG here, so it will be broadcasted to all of them.

And there is a global test controller, so which will be controlling this broadcasting process. So, here see the first this is from TDG, so it is connected to this, it will get connected to this TED. So, as a result it will be coming to this, this interconnect and this

TED will do the analysis and it will send the information of back to the global test controller about the status of this one. And then it can also, this will also get forwarded to the next link, so that way the next link can be tested and that will be giving the information back to the global test control. So, only one set of interconnect between a pair of routers can be tested for each test pattern that is broadcast, because this will test this one after that, so it will be assumed that this is ok. Now, it will test this one; so that way one set of interconnect can be tested. And a global test controller and many these TEDs, so test error detectors, so they are required, so that is the situation of applying this unicast method.

(Refer Slide Time: 02:37)



Test Application by Multicast

- Test packets are broadcast to interconnects of different pairs of routers to achieve maximum parallelism.
- Multicast is a good compromise between test application time and hardware overhead.
- Point-to-point (unicast) test method has the smallest (largest) test application time but the largest (smallest) hardware overhea

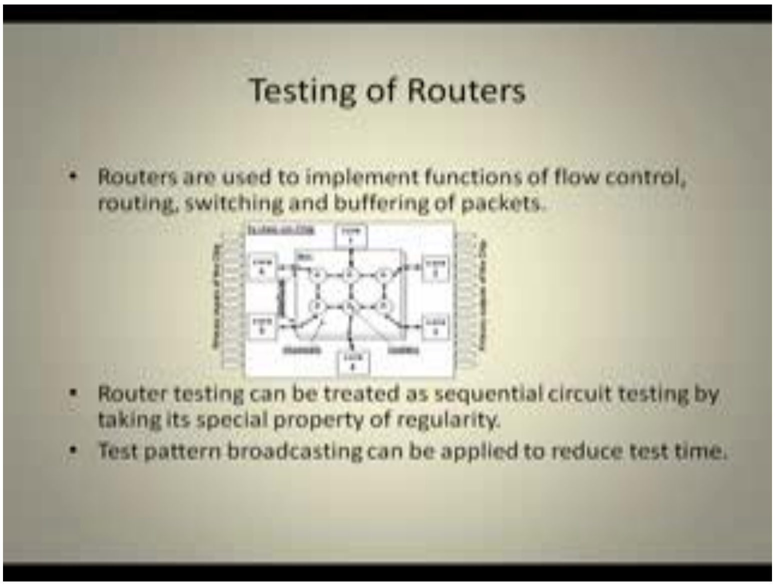
Under a multicast mechanism, so these test packets are broadcast to interconnect of different pairs of routers to achieve maximum parallelism. So, here what is done from this test data generator, so this is broadcast to this link and this link, so both of them are being tested simultaneously, and then this TED, so error detector, so they detect the response and they if they find some errors, so they will tell the global test controller that some error has been detected. Similarly then since it is a broadcasting mode, once this link is correct, so this router will get the test pattern, and it will send this test pattern to the next link, so that way it will be going to the next link, so that part will be tested.

And similarly this, so this links will also be tested, so this link will be tested in a broadcast fashion, this link will be tested, so this is way very fast to all the links will get the test patterns they will get tested. The TEDs will store the status about their correctness and all and all TEDs will transport those information back to the global test controller. So, test packets are broadcast to interconnects of different pairs of routers to achieve maximum parallelism. Multicast is a good compromise between test application time and hardware over it, because multicast is going to help in this reducing this transporting the test application pattern, test patterns to the interconnects.

Point-to-point or unicast test method has the smallest test point-to-point that has got the smallest test application time, but the largest hardware over it. Because in point-to-point case, so we have got this dedicated test data generator and test error detectors, so those modules are there, so there the testing is the fastest, but the hardware over it will be maximum.

Because, so many are TDGs are TEDs are required. On the other hand, we have got this multicast based approach, so that is going to have faster test time, and overhead also reasonably low and the other option that we have seen the unicast method, so it has got the largest test application time, but the hardware overhead is smallest. Because individual routers, they need not have the broadcast facility. So, as I have said already that this testing can use the broadcast mechanism only when we have got this part taking care of that is this broadcast facility is available with the router; if it is not, we cannot do this thing.

(Refer Slide Time: 05:37)



Testing of Routers

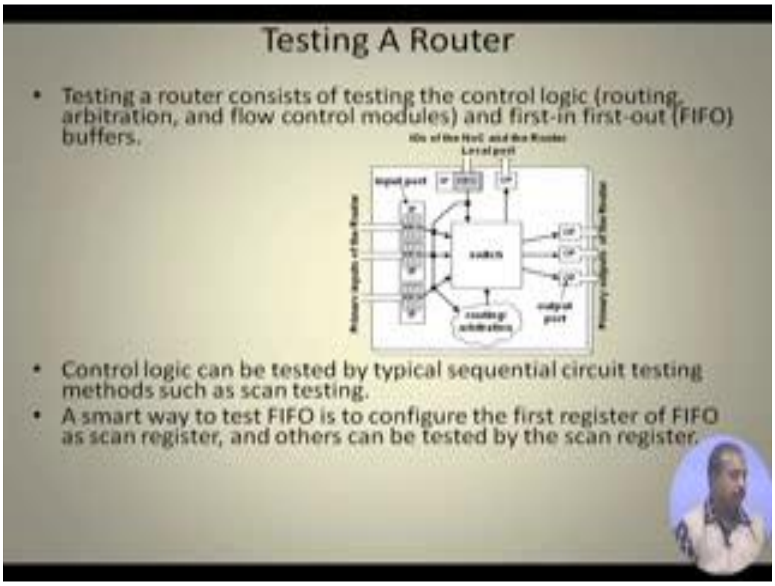
- Routers are used to implement functions of flow control, routing, switching and buffering of packets.
- Router testing can be treated as sequential circuit testing by taking its special property of regularity.
- Test pattern broadcasting can be applied to reduce test time.

So, apart from testing, the interconnects you have talked about testing of cores, testing of interconnects; now what remains is the testing of routers. So, routers are used to implement functions of flow control, routing, switching, buffering of packets etcetera. So, it has got the good amount of logic in it and there are different types of structures like see if this is a system on chips, so we have got this individual cores, they are connected to the routers. So, these you know these cores are connected to the routers, and there is an interface which is basically the network interface NI that is there, so through which these cores are connected to the routers.

And we have got this channels, so these are these lines are the, this links are the channels and this is also the channel, so the channel we have seen how to test the channels, but by means of that interconnect testing. We have seen how to test this cores, because core test patterns are provided by the vendors; and by test scheduling approach, so we could apply those test patterns to the cores for testing. What remains to be seen is how to test these routers. And router testing is difficult, because router testing can be treated sequential circuit testing by taking its property of regularity. So, what happens is that all these routers they are similar in their structure, so that is one good thing. But at the same time, these routers have got within them some sequential element as well as some combinational element. And as soon as we have got sequential element and there we have got discrete flip-flops as well as we will have some memory components there.

So, memory testing will see that that is another very challenging task to be done and those memory testing part also has to be a part of this router testing. So, since the routers are all similar in nature, so it is desirable that we do a test pattern broadcast. So, by doing a broadcast, so we can the same test pattern set can be send to all the routers for testing, the routers can do a self-test and come up with they are may be some error detector which will be determining may be based on some signature register or something like that. It will combine the responses and see we are finally come up with the answer whether the router is working correctly or not, so that way we can have this if the broadcasting features are available then we can use it for this testing purpose of the router.

(Refer Slide Time: 08:11)



Testing A Router

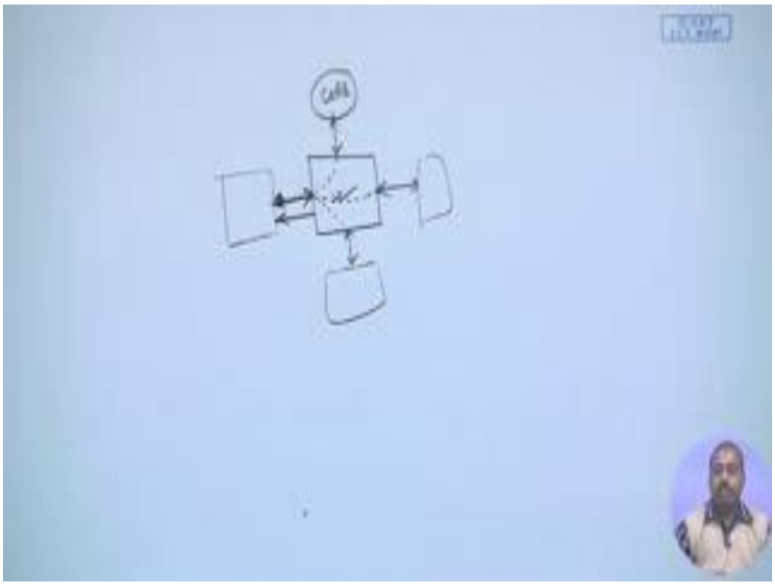
- Testing a router consists of testing the control logic (routing, arbitration, and flow control modules) and first-in first-out (FIFO) buffers.

IDs of the NoC and the Router
Local port
input port
switch
routing/ arbitration
output port
Primary inputs of the Router
Primary outputs of the Router

- Control logic can be tested by typical sequential circuit testing methods such as scan testing.
- A smart way to test FIFO is to configure the first register of FIFO as scan register, and others can be tested by the scan register.

So, before going into this router testing part, so we need to understand what how does a router look like. So, you see a typical switch or a router it has got a number of ports, so it has got a primary inputs, it has got this input ports coming from different, different neighbors. So, in this case, it is assumed that there are three input port, global input port and one local port, so this router that is discussed here, so it has got three puts.

(Refer Slide Time: 08:57)



So, this router, so there is a local port, yes, this side I will connect to one core, so this is one port, and there are three other ports to which these neighbors will be connected. So, another router will be connected here, another router will be connected here, another router will be connected here. So, when we are talking about so this router. Then what is the structure of it, so it has got for every site, so every site, so this is bidirectional link, so I can break it up in two links; one is coming to the router and one is going out of the router. So, a traffic which is coming from this port, so it may be required to be directed to this port or it may be required to directed to that port, or it may be to the local port. So, there has to be a decision module which will decide which will look into the destination address of the packet that has arrived and accordingly take a decision to which port this incoming packet should be put into.

So, we have got a number of input ports, every we channel that we have or every input primary input that we have it goes into one input port or you call it IP. Now, input port it contains buffers because the path may not be immediately available, so that it can be switched to the destination port, so there is a buffer of in this particular example, there is a buffer of depth four in each of the input channels. So, if the buffer depth is large then it is good, because we can buffer a may be the entire packet into this buffer, but that will increase the complexity of individual routers. And as we have said that the individual routers are going to be very, very simple, so this buffer depth cannot be large. So, normally it is kept as 4 to 6 not more than that.

And as a result a packet gets distributed over flits and each flit some of the flits of a packet will be available at a buffer of a of a router. Now, out of these flits there is one hidden flit which identifies the destination address, source address and all that, so the destination address has to be analyzed, and it has to find out like what is the routing to be done, so there is a module which is called routing and arbitration module. So, this will decide like it will look into the hidden flit and it will decide how this packet has to be transferred to the output port, so which output port it should go. Depending upon the destination address, so this routing function will run and it will find out the destination port, so it may be x y routing, it may be some other routing policy, but whatever it is ultimately it identifies the output port to which the packet will go.

Now, also there is a arbitration. Like if there are say multiple packets which are asking for same port then I have to tell like which port which packet will which packet will go first and other will be waiting. Similarly, there maybe the through this switch at one point of time only one packet will go, so we have to see like which one will be picked up, maybe you follow a round robin policy, so one of the packets are picked up one of the packet is giving green signal to proceed through this switch. And this switch part, so this is actually some sort of cross butt type of connection where from each of this input port that is a local input and the three neighboring routers input port, so they will be routed and they will be connected to one of this output ports. So, this is the arbitration the arbitration part that is done, so the switching part will be these routing and arbitration modules, so it controls the switch, so this switch is configured like this.

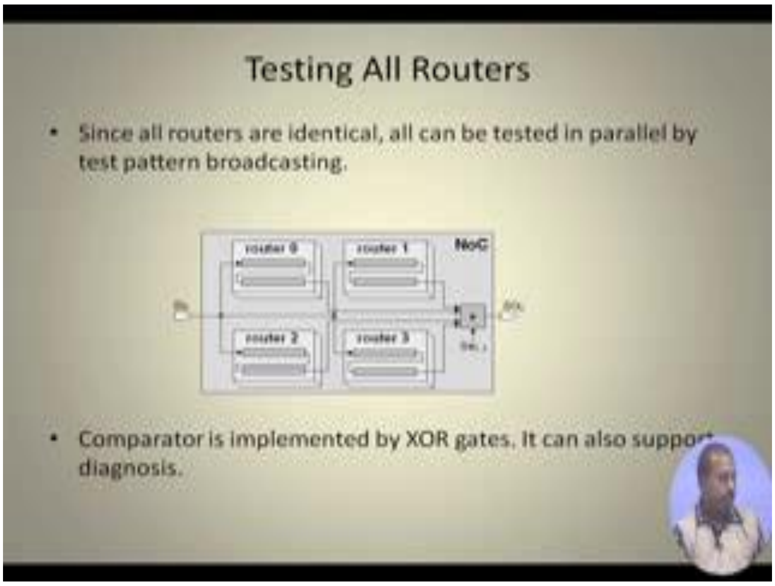
So, you can understand that this FIFO the first-in-first-out buffer that we have, so that definitely has got the memory elements in it, so that testing of this part should be done using the memory testing policies. Similarly, we have got this switching part, so this switch is designed it is a crossbar, so it is a combinational logic, so this should be a combinational testing. And this routing arbitration part, so these require some sequential element and some combinational element.

Like if we are going to do say round robin arbitration, then we have to remember like which module which channel got chance last times, so it need some amount of memory, so the sequential elements will be needed, so this way we have got this sequential and combinational elements in the routing arbitration module. And we have got this switch part which is combinational and we have got this memory part in the FIFO. So, that way this testing of this router is a heterogeneous task, so it is not very simple. Only thing is that the router itself being very simple, so the maybe the magnitude of the problems may not be that much severe.

So, this is the thing that this testing a router consists of testing the control logic, routing arbitrational and flow control module and the first-in-first-out buffers. So, flow control module is basically what happens is whenever this FIFO some locations are fed then only from the previous router some part of the packet some of the feds were you brought into, so that is controlling the that is with the flow control. Otherwise, there it may lose some of the flits the flow control has to be the flow control module is not shown here basically, but that has to be there.

So, this control logic has to do this thing and the FIFO buffers, so all of them are to be tested. So, control logic can be tested by typical sequential circuit testing method such as scan testing. So, this routing arbitration and control logic, so I has said that they have got sequential elements in them, so they can be tested using some scan type of philosophy, so all those flip-flops are put in the scan chain, and then they are tested they are put into input patterns can be applied to that. And for the FIFO part, so it maybe better it may be a smart way to test the FIFO is to configure, the first register of FIFO as scan register, so this FIFO has got four register. So, first one is configured as a scan register, so that we can put any desired pattern into this, and then the others are can be tested by the scan register. Now, this since it is a FIFO there is a facility provided in this module, so that the content of the first location can be shifted to the second one to the third one to the fourth one. So, if you load a particular scan pattern here and then transfer it by shifting to the lower cell, lower FIFO cells then that way all the FIFO location can be tested. So, this is the FIFO operation that, so this is the overall class of the router testing.

(Refer Slide Time: 16:07)



Testing All Routers

- Since all routers are identical, all can be tested in parallel by test pattern broadcasting.

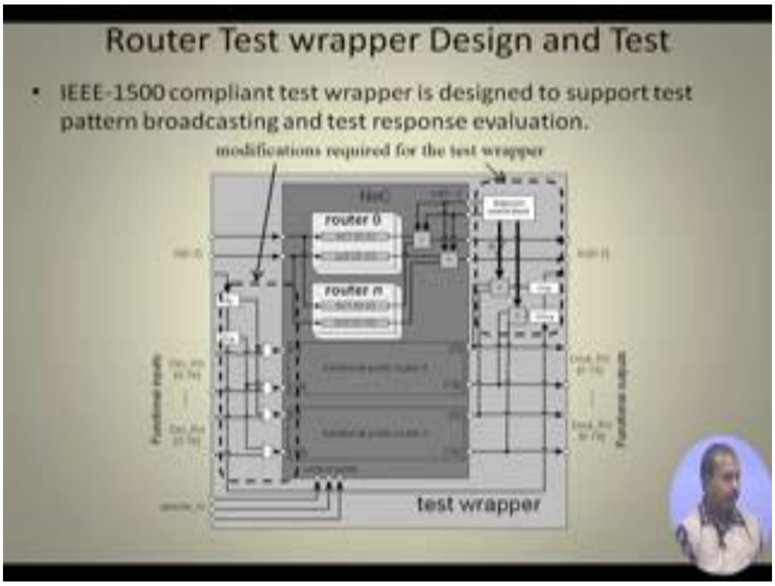
- Comparator is implemented by XOR gates. It can also support diagnosis.

So, since all routers are identical, all can be tested in parallel by test pattern broadcasting, so this is the thing that I have already said that we can say that ok whichever test pattern, so same test pattern is sent to all of them. So, from the scanning input, we can send the test pattern to all the routers and assuming that we have got the scan chains for configuring for the control logic, and as the first location of FIFO configured as a scan chain, so through that scan chain, so you can send in the test patterns. And then it will do after the test pattern has been sent in a broadcast mode, so it is a going to all the routers, then they will do their own testing. And once was the testing is done then the response will be coming, and it is putting onto equality checker. And this equality checker it will do holding the correct responses for all the test pattern that will apply, and it will do a equality check.

So, there is a select function, so there are there are four routers here 0, 1, 2, 3. Select function will select the input from here, and then it will do the comparison, accordingly it will tell the scan out line whether the particular router was or not. So, comparator is simply xor gates, and naturally it can also do some sort of diagnosis, because now this comparator it will know like if there is a mismatch, it will know for which router there is a mismatch. And it will try to see like for which pattern it has failed and all that, so they are may be intelligence like clubbed on to this.

And what happens is that in case of NoC design, so we always keep some spares. So, if some router is determined to be faulty, so we can take help of some other router, and the core that was attached to the previous router is transferred to the to another good router, so the core that is faulty router gets transferred to a to a functionally correct router. So, this way we can take there are the many works on this fault diagnosis and for the fault tolerant NoC designs and all that, so they actually relay on this type of principle.

(Refer Slide Time: 18:31)



Router Test wrapper Design and Test

- IEEE-1500 compliant test wrapper is designed to support test pattern broadcasting and test response evaluation.

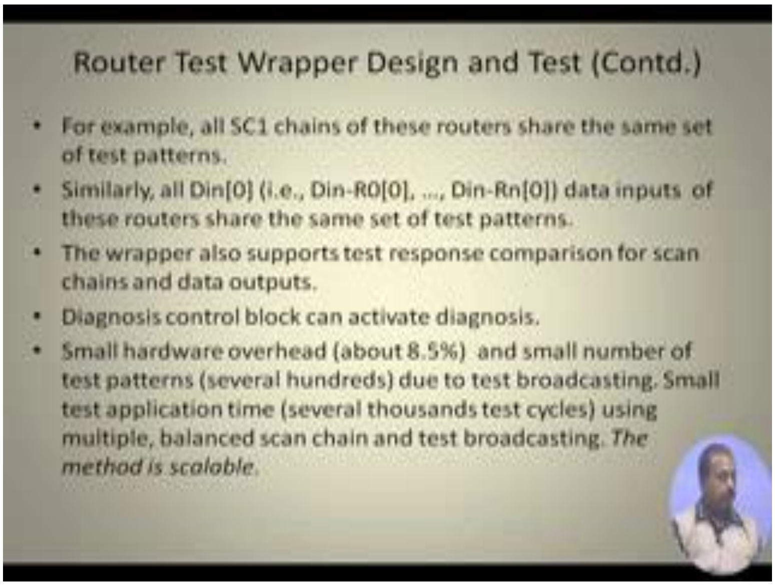
So, for the router, we have to design the test wrapper. So, again the same thing that is IEEE 1500 complaint wrapper is designed to support these test pattern broad casting and test response we have evaluation. So, you see that in the test wrapper is designed like this, so we have got the functional inputs, we have got this special inputs for this control ports and this scanning puts. So, this is this scan input, they are feeding this scan chain 1 and scan chain 0. Then this scan chain serial input 0 is feeding these functional inputs for the functional ports of the routers. So, they are all those routers are they all those functional ports they are put on a chain, so they are being feed by this scan input 0. And then scan input one is feeding this scan changing zero of all the routers, and the scan input is feeding the scanning scan chain one of all the routers.

So, this way all the routers are fed with the test pattern required for a particular one particular test pattern can be loaded to all the routers. So, they do the operation and then there the equality checkers are there; so equality checker for this scan chain outputs of this scan chain output and the correct one, so there that will be there, so that is evaluated. So, scan select that is a select function output of this R router, it will be out of N routers that I have. So, it will be selecting some of the router, it will select one router and do the comparison check. And after that comparison, so we have got so many comparison check, so the comparator module, so their results are available at this point and through the this scan outline. So, there will be available on the scan out lines.

Similarly, this functional output line, so they will also be available on this scan chain. So, the three scan chains are there, so SI lines they will be giving the test patterns, and this is SO line they will be giving the responses. So, we can have a diagnosis control block, so that is optional. So, this is actually trying to see if some router has failed. So, if some router has failed then it can be initiative some fault tolerance mechanism by which it can just see like what sort of patter, for which router has give gone faulty and accordingly it can try to which router has the gone faulty, and it can arrange for a replacement of it.

Now, extra things that are needed is this part and this part. So, you see all the routers are put onto this test wrapper, so that is a big challenge, so putting all the routers onto this thing. So, you see the delta NoC is getting wrapped by a wrapper like this, so that is the modification that is there. So, it is for SoC testing, this was not required, but for NoC testing, if we have going to test all the routers together, so we have to have designed this type of wrappers.

(Refer Slide Time: 21:45)

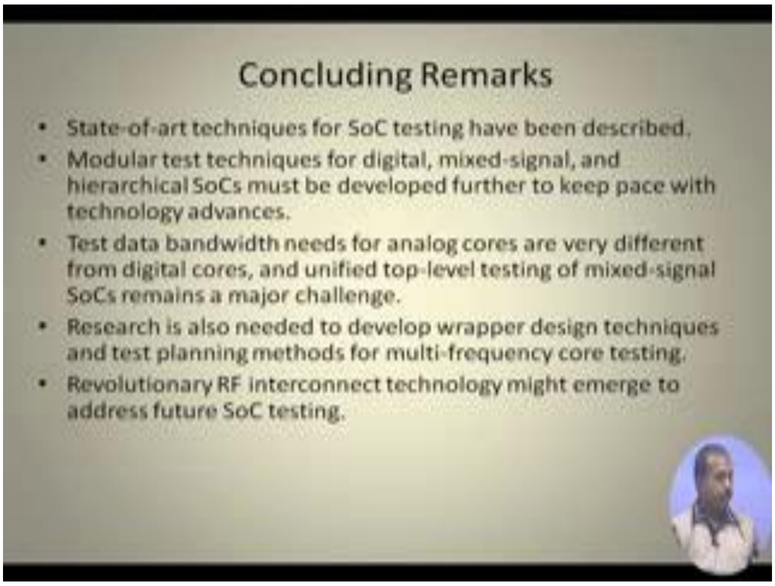


Router Test Wrapper Design and Test (Contd.)

- For example, all SC1 chains of these routers share the same set of test patterns.
- Similarly, all Din[0] (i.e., Din-R0[0], ..., Din-Rn[0]) data inputs of these routers share the same set of test patterns.
- The wrapper also supports test response comparison for scan chains and data outputs.
- Diagnosis control block can activate diagnosis.
- Small hardware overhead (about 8.5%) and small number of test patterns (several hundreds) due to test broadcasting. Small test application time (several thousands test cycles) using multiple, balanced scan chain and test broadcasting. *The method is scalable.*

So, this is the all SC 1 chains of these routers share the same test patterns, so D in 0 data inputs of all these routers share same test pattern so that we have already seen. So, diagnosis control block can activate diagnosis. Small hardware overhead about 8.5 percent and small number of testing patterns some hundreds due to testing broadcasting. So, this will be required. So, overhead reduce a significantly compared to the overall chain. Small test application time using multiple balanced scan chain and test broadcasting, so that can also be done, so the method becomes scalable; so with the increasing number of routers in the NoC, it is not that this time requirement of the hardware requirement increases exponentially.

(Refer Slide Time: 22:33)



Concluding Remarks

- State-of-art techniques for SoC testing have been described.
- Modular test techniques for digital, mixed-signal, and hierarchical SoCs must be developed further to keep pace with technology advances.
- Test data bandwidth needs for analog cores are very different from digital cores, and unified top-level testing of mixed-signal SoCs remains a major challenge.
- Research is also needed to develop wrapper design techniques and test planning methods for multi-frequency core testing.
- Revolutionary RF interconnect technology might emerge to address future SoC testing.

So, coming to the concluding remarks, so state-of-art techniques for SoC testing, we have seen, so we have seen the modular test techniques for digital, so they can be developed, so this modular techniques can be developed, so that they can be integrated with this SoC testing mechanism. Test data bandwidth needs for analog cores are very different from digital core, and unified top level testing of mixed signal SoCs, so that remains a major challenge. So, this as I said the analog testing is a challenge, so that continues here also. Research is needed to develop wrapper design techniques and test planning methods for multi frequency core testing. So, these wrappers are not capable of handling multiple frequencies, so may be something can be done there.

RF inter connecting technologies are coming up, so they can also be and when the inter connects are operating in the gigahertz length, then this RF will come into picture, so they are to taken care of. Advances in testing up NoC based system we have seen, so how to utilize the on-chip network as it can without compromising fault coverage of test time, so that is the testing that we are going to that we have addressed in the NoC testing.

So, NoC testing research is premature and naturally so in fact, NoC itself is going to come up and then it is in a future it will be developed and this research methodologies are going to be developed which will solve this NoC testing problems significantly. Wrapper design techniques for SoC testing can be adopted for NoC based systems and we have seen, but we have also seen that this 1500 wrapper needs to be modified for NoC testing, so that is to be there.