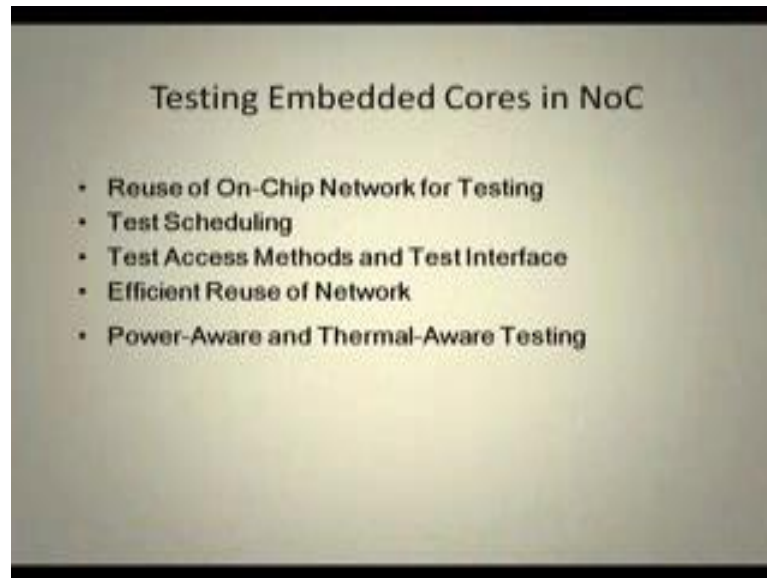


Digital VLSI Testing
Prof. Santanu Chattopadhyay
Department of Electronics and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 52
System/Network – On-Chip Test (Contd.)

(Refer Slide Time: 00:22)

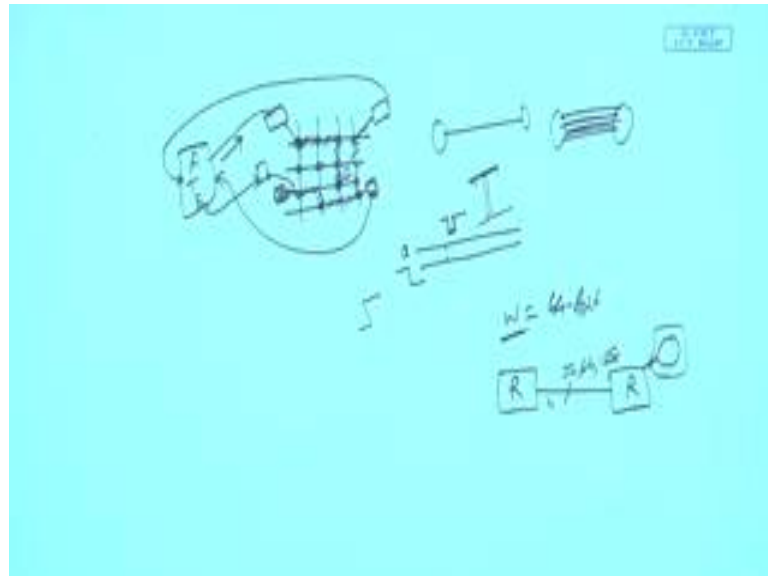


So, while testing network on chip, we have got certain issues to be considered. First of all instead of providing a test access mechanism separately in terms of TAM lines and test buses, so it will be better if we can reuse the on chip network, because if you can reuse it then this extra TAM lines are not necessary, and while we are designing a NoC based system, so the customer is anyway paying for this extra interfacing that we are putting, this routers and links that we are putting in to the system, so they are paying for that. But in fact, what they should pay for is the functionality of the system the functions or the features that this system has. So, as a result, so those are seen as overhead from the designer or customer's angle.

So, if we can save something at it at the testing point, so it is better. So, instead of putting separate test access mechanism, so it is better that we try to reuse the infrastructure that is already existing in the NOC. So, reusing on chip network for testing so this is one of the important features; then we have to do test scheduling, so scheduling has to be done just like the SoC testing. So, we say it at which time which core is going to be tested. So,

here also same thing we have to talk in terms of the time at which individual cores will be tested.

(Refer Slide Time: 01:58)



Now, you see that this may lead to some test resource conflict. In the sense, that if we have got an NoC where say one mesh based NoC wire we have got this. So, suppose this is a four by four NoC where at the junctions we have got the routers and say with the routers, the cores are attached, so here one core is attached. Now, with this similarly for with other routers the cores are attached. Now, suppose if this router, some core is attached and we want to test that core.

Now, how can we test it? So the ATE that we have in the system, so ATE will communicate to one of these cores. So, normally what is done is that ATE channel is connected to one of this core as input point, and similarly ATE channel is connected to another core as the output point. So, this is the input towards ATE and this is output from the this is input from the ATE and this is output to the ATE. And this ATE is interfaced to top with these cores and these cores in the test mode, so they will transfer the test pattern to this router to be destined to this core under test. And similarly the core under tests, so it will send the response to this core from here it will be transferred to ATE.

Now, while doing this, so assuming that it follows an x y routing policy, so during the entire tests session, so these resources are all busy, all these resources are busy. So, if suppose yeah this ATE has got multiple channels, so there is another channel that uses

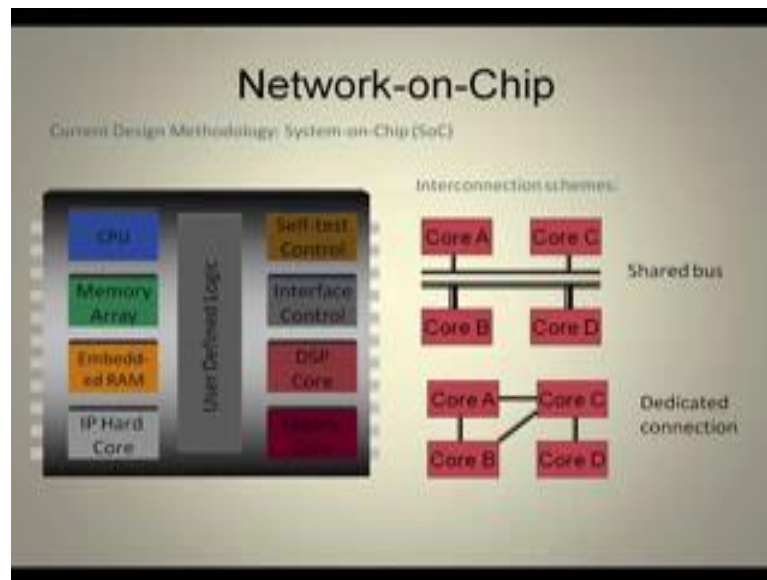
say this core as source, and the core attached here as source, and the core this is another source and say this is another sink. So, that way the input is fed to this core and output is collected from this core. So, two parallel checking testing can go on like, one test can do this, now if another test is say testing say this core, then say it is testing say this core then it is possible because the test patterns can go like this, it can go this through this channel reach here. And the response can be send like this through this channel. So, that way this path are non-overlapping, for the two channels the paths that are coming for transferring the test pattern and response they are non-overlapping in nature.

So, if they are overlapping, then naturally the two test sessions cannot go on parallelly. So, this scheduling is important. So, scheduling will tell us like at what time we can do the testing. And the resources that are needed for testing in terms of this intermediate routers and links, they must be available for this entire test session. So, this scheduling has to be done carefully. Then this test access methods and test interface. So, how are you going to use the test?

In the sense, that whether it is going to as I was telling the in this example that we assume that these core and these core, so these two are acting as source and sink. Similarly these two cores are acting as source and sink, so like that we have assumed, so that that is the access mechanism for the testing. So, and the interface like how this interface is there like individual cores that are attached, so they must have their wrappers and all that, so this test access method and test interface, so they will be talking about that then efficient reuse of the network.

So, we have to identify potential parallelism that can be achieved by doing this scheduling of testing of various cores to by different ATE channels. So, we have to do some efficient reuse of the network. Power-aware and thermal-aware testing, so these are two important issues that we have seeing seen a NoC. So, it is a more regular in nature. So, it should be able to do justice to the power and thermal issues. So, though many of them are same as the SoC testing, but here we have got the added overhead in terms of these routers to be tested then this links to be tested and all that. So, this is the extra thing that has to be done. So, after which was not present in SoC, where the TAM lines might be tested initially by some flash patterns, but here we need to test this individual routers and links separately, so that actually adds to some overhead.

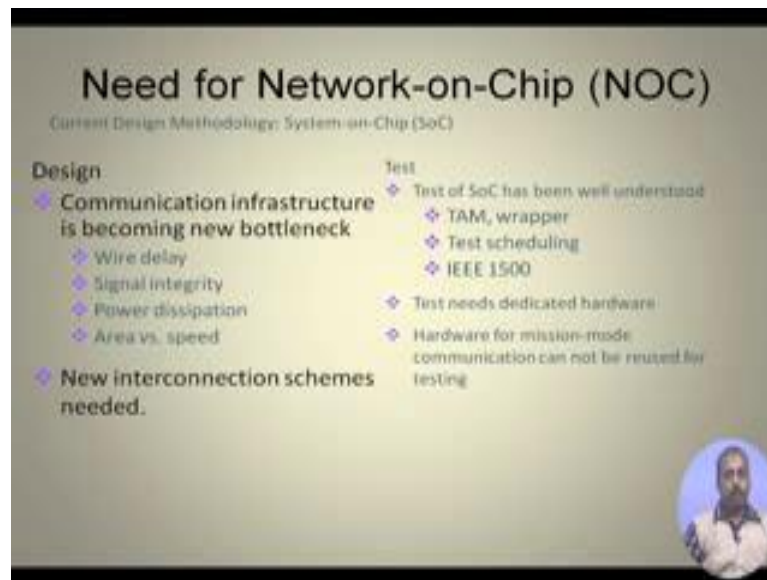
(Refer Slide Time: 06:32)



So, to go to the network on chips, assuming that the current design methodology is a system on chip based design. So, we have got CPU, memory, embedded RAM, IP hard core then this self-test control, so all this modules are there and there is user define logic. So, interconnection scheme is like this. So, shared bus, so there are buses are may be core A and core C are put on first bus; and core B and core D they are put on another bus, so that may be the shared bus architecture

Now, you can have dedicated connection like say this cores, so they are they are point-to-point connection. So, core A needs to talk to B and C as a result it has interconnection to B and C; similarly, B needs to talk to C, so it has got connection; C needs to talk to A and D, so it has got connection to A and D. So, this way you can have some point to point type of connection. But as we have discussed this point-to-point connection is going to be costly, because large number of interconnect line needs to run, and this shared bus is costly because performance will be at a premium. So, performance will suffer when we are connecting more number of cores to a bus.

(Refer Slide Time: 07:46)



So, this need for network on chip; so it design community, so the communication infrastructure is becoming the new bottleneck. So, individual cores they are made more and more fast they are made faster and faster, so they are actually operating at a much higher frequency, they can produce data at a very high rate, but this needs to be transferred to the destination, and this transferring, so that becomes a problem. So, if you assume a pure serial communication between two nodes, then this serial communication takes time. Now, we will go for parallel communication, so parallel links number of lines that will pass will is a problem. So, routing of all those lines is a problem.

Now, apart from that this compare to a serial communication, so when we have got this parallel communication, so this parallel communication is faster because you can transfer multiple bits simultaneously. But this also has got limitation, because we cannot transfer data at a very high rate because of the cross talk between this parallel lines. And if one of the line makes a transition it at a high frequency, so it affects the values that are there the logic values that are there on the adjacent lines. So, this is known as aggressors, and aggressors and victim type of situation, so that way this signal delay becomes a problem.

So, communication infrastructure is becoming the new bottleneck. Wire delay, so that is one issue the delay from source to destination serial communication is out of question. If we have point to point parallel connection then also this establishing connection between all the cores, so that takes lot of wire delays. In signal integrities, signal integrity means

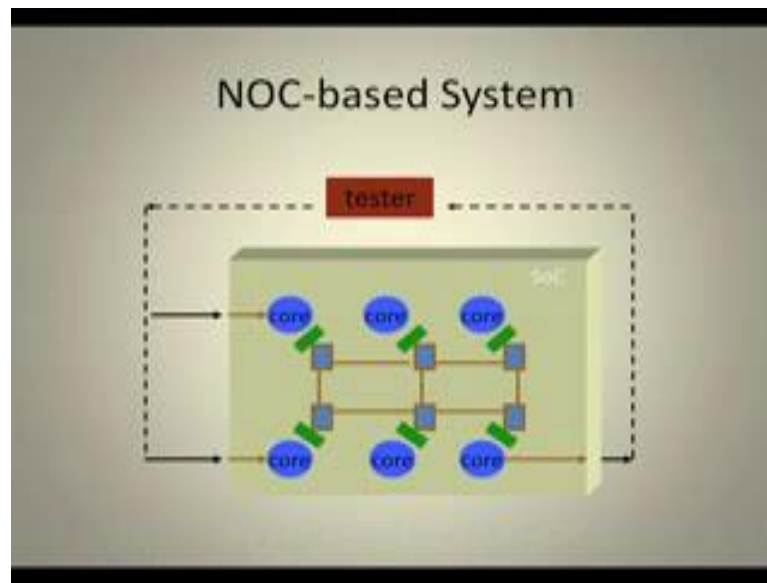
due to as I was telling that due to transition in one line, so the other line may be in some other line value may be change may be this line is constant to at logic one, but this lines makes a transition to low. When it makes a transition to low, so due to the capacity when inductive coupling between these lines, so this line also may be for some time it will show a glitch and it will be something like this.

Now, if this glitch is if this voltage level drops significantly, then there is a possibility that at the receiving end, we are sampling the line at this time, as a result we get a wrong logic value rate. Or if there is a positive glitch, so this for this line is making a transition from low to high, as a result this line may also makes a transition it shows a glitch and then it comes back. Then this glitch peak value that will tell whether this can cause any harm; and also the duration of the glitch that is also another issue, so that way the signal integrity may be in question, so that has to be addressed.

Power dissipation, so power dissipation of this bus and this interconnects, so they are going to be high, so we need some regular way to handle this interconnection. Area versus speed; so area is definitely is the area in terms of this bus, so we can we can do it on by running a single bus. So, as a result, the area requirement is low, but speed will also be low. We can have multiple buses, so that way this area requirement goes high, but speed also goes high. And finally, so if we are going for some other mechanism, so we can try to have a better tradeoff between area and speed, so new interconnections schemes are needed.

So, for testing purpose, so this is for the design side. For the testing side, so test of SoC, so that is well understood. So, we have seen the how it can be done like the TAM design, the wrapper design, then the test scheduling problem and the IEEE 1500 standard, so that has told the standard interface of this core wrappers, so that way this testing part SoC testing is understood. But in case of NoC testing, this network environment, so we need some dedicated hardware. So, the test will require some dedicated hardware, and hardware for mission mode communication cannot be reused for testing. So, the hardware that is used for communication between the module, so when they are doing the function, so they cannot be used for testing purpose as per as the SoC is concerned. So, we need some solution to that.

(Refer Slide Time: 12:27)

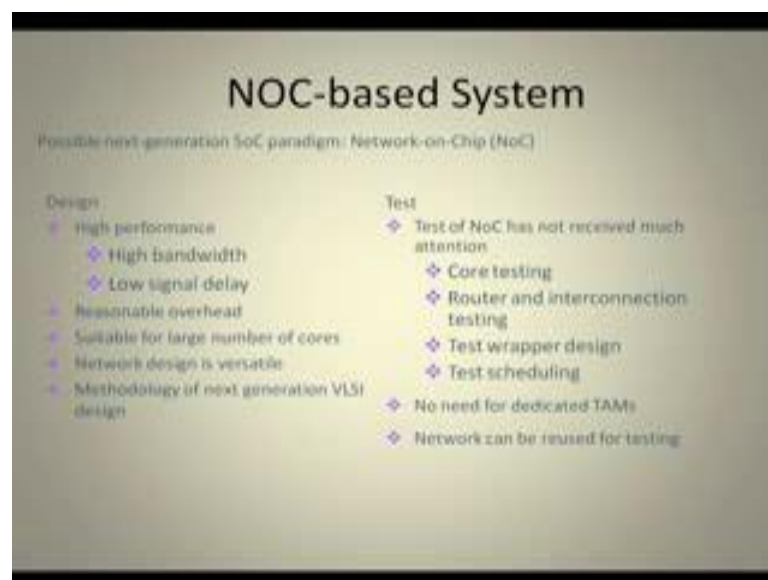


So, in a NoC based system, as per as testing is concerned, so what is done, so as I was telling, so this is SoC implemented as an NoC where these boxes are the routers. So, we have got this cores and between them this green boxes, they are actually the network interface. So, what is this? So this particular router network, it will follow some networking protocol, it will have some packet length, it will have some a packet structure and all that may be in terms of start beat, stop beat, package size, header, trailer and all that, so for every communication has to be demarcated by those special symbols; so the message that this core generates to be transmitted. So this needs to be divided in to packets which can be transmitted over the network. So, this packetization has to be done; and in the packetization process, the proper formatting has to be done.

Then second thing is that this at the receiving end, so this is at the sending end, at the receiving end, this packets have to be combined together and a message has to be formulated and that message has to be delivered to the destination core. So, this green boxes, so they are known as network interface or NI in short, so they are actually doing this packetization and depacketization of messages into packets and vice versa. So, what happens is that whatever communication is going from this core it goes to the corresponding NI, and this NI converts it into a format which is acceptable to the underline network. So, through the underlying network packets travels to the destination; and at the destination they are combined again to get back the message and deliver to the destination core.

Now, from the tester, as I was telling there maybe multiple channels like say the tester may decide that ok, so it will feed test pattern via two cores, so this one and this one; may be this cores are equipped with some special feature by which it can accept input from the tester. Basically the cores which have got interface to the system pins, so they are dedicated for this purpose because of through that we can access individual cores. And they can be put in some sort of bypass mode, so that this test patterns that are transferred will be given to the NI to be transmitted to the destination. So, some modification has to be done, some extra logic has to be added with the core, so that this tester can be interfaced with these cores. So, they are sending the patterns, and this test patterns are from this point, it will be send to the destination using the underlying network.

(Refer Slide Time: 15:19)



NOC based system, so they are actually the possible next generation SoC paradigm which is known as network on chip. Design angle, so high performance and because of high bandwidth and low signal delay. High bandwidth effectively the bandwidth increases, because in an NoC what happens is that several communications can go on simultaneously, unlike SoC say bus based SoCs, so where only one communication can go on through the bus at a time. In case of NoC, there can be multiple communications going on parallelly throughout the NoC; as a result the effective bandwidth increases.

Signal delay is low because these individual links are much shorter. So, if you look in to a mesh type of structure, then between the links are running between the successive routers only. And these links are of some fixed length, so as a result this required some fixed amount of time to travel through the link. Unlike say one SoC, where there may be it may be put on to a bus or it may be a one-to-one connection, point-to-point connection, but that connection length the length of the wire may be too large. So, those problems are there, so that way the frequency of operation gets restricted.

So, in case of NoC based system, so high performance can be achieved by this high bandwidth effective high bandwidth and the low signal delay between the successive stages. This is overhead in most of the cases, it is reasonable. In the sense that this routers that we have, so the routers are made very, very small, so that they supports just a basic routing algorithm and so that the overhead is router overhead is much, much low compared to the core. So, it may be 0.1 percent or 0.01 percent of the total SoC area may be dedicated for the network part including the routers network interface and the links. So, it is suitable for large number of cores so that is another point because for low less number of cores maybe we can do a point-to-point connection or bus based connections, so they do not cause any harm; however, when we have got large number of cores. So, taking to SoC is difficult because the communication between them becomes a problem.

So, NoC based system can do a better communication, so we can take help of this mini parallel communication that may occur among these cores. So, if you have large number of cores possibility of this parallel communication is more, as a result we can exploit the NoC infrastructure for sending large amount of information from sources to destinations. Network design is versatile because that is design has to be versatile it should be common for any design. And it is the methodology for next generation VLSI design, so that is what is coming up, so we are started getting designs around NoC and industries they are doing the designs.

From the test angle, test of NoC has not received that much attention like this test this design part that uses NoC. So, the challenges are again increased. So, test engineers are again at a negative side or a uncomfortable side because now the job of this testing process, so it has increase the amount of job to be done. Core testing - so it was there in SoC design, so it is there in the NoC design as well. So, the test data volume transfer time, so those things are not going to reduce.

Apart from that, we need to test the routers and interconnects. So, this is the additional thing that has come up because all the test patterns will be transported through the network. So, if the router or interconnect is faulty, then this test patterns cannot be transported. So, core testing does not have any meaning, if we do not perform this router and interconnection testing. And for the correct operation of the system, this router and interconnect, so they must be working properly. So, even if not for the sake of transporting test patterns, so we need to ensure that this router and interconnection network, so that is there are all working fine, so that has to be done.

Test wrapper design definitely there, because we have to we have to do this you have to wrap all the cores, so that this NoC channel width like between the routers the links that are running, so links are not of arbitrary size. Unlike say this TAM based design where may be I have got 64-bit TAM, W equal to 64, and there we had some trade off. So, we gave different amount of TAM lines to individual cores, and try to see what is the corresponding test time.

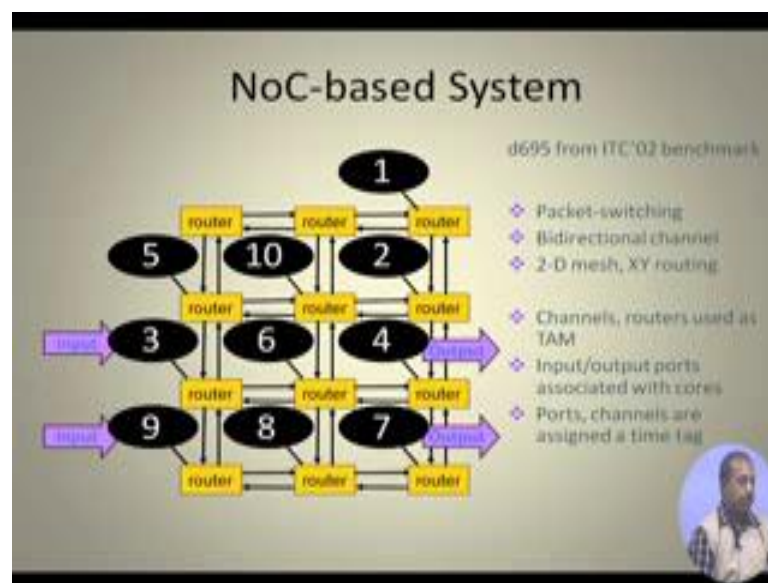
But, in this case, what happens is that say this is one router and this is another router, so between them we have got a link. So, this link has got. So, this is a router this is a router and this is a link. Now, this link has got a particular size may be this link is say 32-bit or link is say 64 bit or 128-bit, so like that link may have some number of bits in it. So, this w, so I do not have any option, so I cannot give less number of bits to be transferred to this core. So, this core that I have, so core this local channel that is there. So, that is also of size this 32, 64 or 128. So, this core it gets input 32-bit inputs, 64 bit input or 128-bit input, the other options do not exists, so that has to be done. So, that has to be taken care of by means of this by means of this wrapper design that we have here.

So, if I have if this core has got more number of inputs than the channel that is connecting it, so the wrapper has to take care of it and it has to it has to connect it has to do the wrapper design part, so that the interface becomes 64 bit only. So that way depending upon this link width we have to see like how many bits we can how many bits we have for the TAM.

So, we can do this test wrapper design, so which is now more specific unlike SoC where it was more flexible we could have allocated different test with TAM width and see different types of wrappers here that is not the situation. Then the test scheduling. So,

this problem remains, so anyway you have to do the test scheduling. So, at which time which core is going to be tested or which router is going to be tested and how the test patterns are going to be transported to the core or the router, so that way the scheduling has to be done. On the plus side we have no need for dedicated TAM, so because the underline network itself will be used for transporting the test patterns and responses. So, there is no need for dedicated TAM. And network can be reused for testing the say the network that we have underline network in the NoC that we can use for testing purpose as well.

(Refer Slide Time: 22:56)



So, this is a typical situation. So, this is a particular SoC d 695 from the ITC 02 benchmark suite. So, it has got some 10 cores in it. So, cores are numbered as 1, 2, 3, 4 up to 10. Now, suppose a particular NoC based realization of this d 695 system is like this that is no core is attached to first two routers, third router has got core one attached to it, fourth router has got core five attached to it. So, this way we have got a four cross three mesh network to which this 10 cores have been attached. Now, we are trying to address how can we test this entire NoC.

So, from the designer's angle, from the designer side, so they have decided on how this routing will take place. So, these routers have been populated with the routing algorithms and then what to do, how to do the communication the packet structure and all that. So, it is assumed that this particular design is a packet switching network that

has been realized. So, every packet may go through different paths. So, for every packet may go through different path, so that the switching is at the packet level; then it is a bidirectional channel every channel is bidirectional.

So, wherever we have got a connection from left to right, there is a connection from right to left. It is a two-dimensional mesh and the routing algorithm followed is x y routing. So, x y routing means first the communication will go in the x-direction then it will go in the y-direction. So, if there is a communication from say 5 to 8 core 5 to core 8, so core 5 will give the packet to this router and this router finding that the destination is 8, it will go in the x-direction first and reach the column where this core 8 is located and then it will go in the y-direction. So, first it goes in the x-direction then it goes in the y-direction, so that is that is why it is called x y routing. So, x y routing has got many advantage it is deadlock free, then it is shortest path and all that. So, those features are there.

Now, for the testing purpose this channels and routers they are to be used as test access mechanism, whereas this input output ports associated with the cores and they are input output ports, they are associated with cores. So, this say from the ATE the test pattern will test patterns will come. So, core three acts as one input point and corresponding output point is say core 4; and core 9 acts as one input point, and core 7 is the corresponding output point. So, we have got this test patterns to be transported from the ATE. So, it comes to the to this to input say core 3.

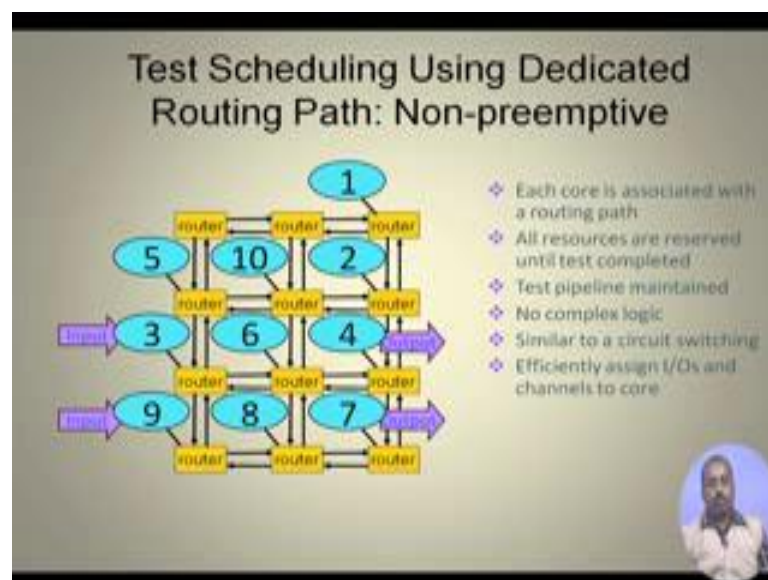
Now, if it is for say testing core number 2, then it will come to this routers it will follow the x y routing. So, it will come it will proceed to this router and then it will go in this direction. So, the test pattern will reach core 2. So, core 2 the patterns will be applied and the responses from the core are to be transfer to the output. So, again it is transferred to this router and it is destined to core number 4 now because core number four is the corresponding output port. So, it is sent to this router from here it comes to this routers knowing that the destination is core 4. So, then it goes to core 4 and that is connected to the output. So, this way this x y routing will be utilized for transporting test patterns and responses between the cores between this input core to the core under test to the output.

So, you can see that we can do parallel testing because one part so for this entire operation the routers that were needed were this one this, this, this, this one then this

router and again for response, so it was so only this four routers were utilized and these links were utilized for doing the testing.

So, at that time, if we were say trying to test core number 8 through the other channel, so that is very much possible because from nine we can send the test patterns. So, it will come to this router to here it will be delivered to this core 8; and from the core 8, the response will be again loaded in to this router and that will be destined to core seven via this router and that will connect to the output of the ATE. So, this way the two test two testes can go on parallely, two test sessions can go on parallely.

(Refer Slide Time: 27:46)



So, here it is assumed that it is a non-preemptive path because the test once it starts it does not stop in between. So, it completes and then only it can go to another test session. So, each core has got an associated routing path from input to that core and from that core to the output. All resources are reserved until the test is completed; unlike the functional mode of operation of the NoC where since the switching is done at the packet level, so once a packet has been transferred, so it is not mandatory that the remaining packets will also go in the same direction same direction. So, resource reservation is not needed, but in this case, while test in this case we are thinking about non-preemptive testing. So, test patterns will be applied all the test patterns for a core will be applied in one shot. So, they will be forming non preemptive test testing. So, we have to keep all this resources reserved till the test ends.

So, test pipeline has can be maintained, because once this test patterns are going through this routers, so the test patterns can be send in a pipelined fashion. So, since it individual routers, they introduce something called a wormhole routing, so where a packet traverse it through the network in a serial fashion. So, as the buffers are becoming free in successive routers, so the remaining part of the packet will be drawn from the previous routers, so that way at any point of time you can find that a test packet is distributed over a number of routers buffers in the path. So, anyway so that gives rise to pipelining. So, we can send all this packets in a pipeline fashion from the source to destination.

No complex logic is required because this is a simple transfer and this routing is already implemented by the functional operation. So, in the functional operation of the NOC, the routing is already implemented. So, no complex logic is necessary. So, it is similar to circuit switching, because in circuit switching, we know that the at the beginning of this message transfer the circuit gets established. And for the end then the entire message goes through that circuit only.

So, similarly here also the same thing is happening. So, we have got this source and destination between which the test packets are going in an order fashion, and there is no interruption in between. So, any two efficiently design assign I Os and channels to the cores, so that is very important one to be addressed.