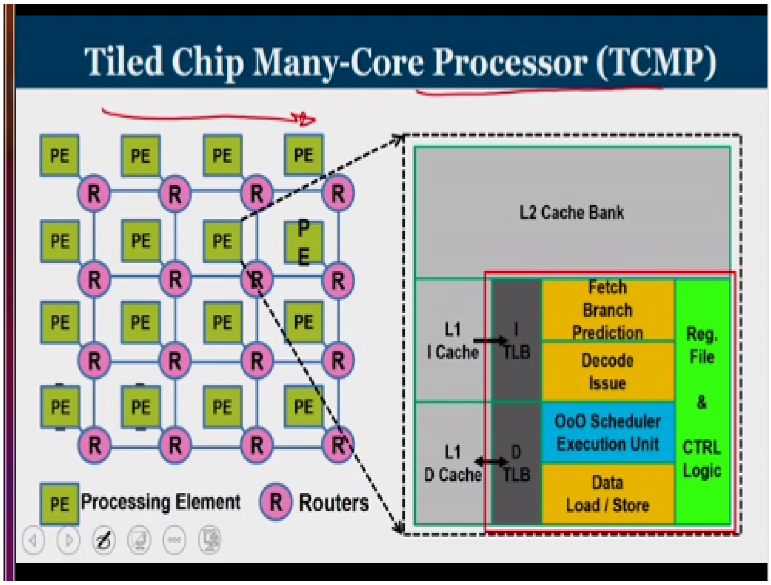
Advanced Computer Architecture
Prof. Dr. John Jose
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology-Guwahati

Lecture-30
Routing Techniques in Network on-Chip

Welcome all of you to lecture number 21, in this we are going to learn about network on-chip concept. In the last lecture when I introduce to tiled chip multi core processors, we quickly touched upon the interconnection mechanism. Today we will learn deeper into this, this is a relatively new concept that was evolved after 2010. So this is something new which you cannot generally find out in textbooks of computer architecture.

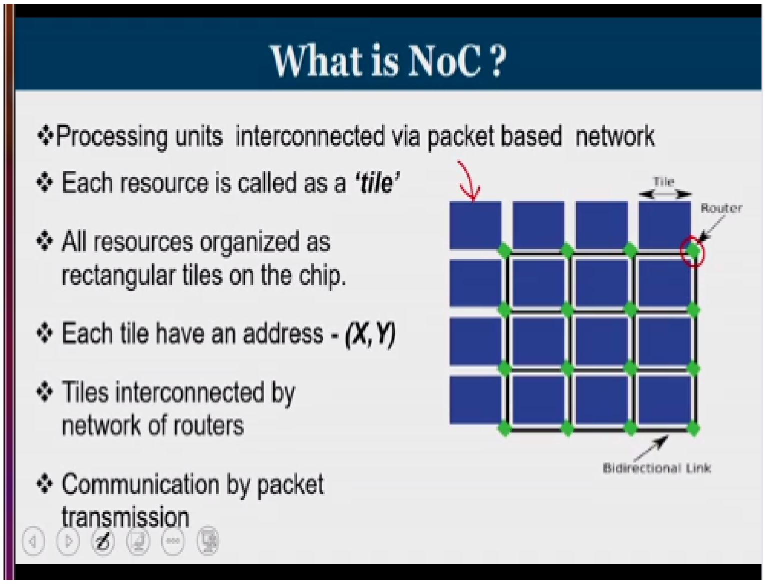
(Refer Slide Time: 00:52)



Tiled Chip Many-Core Processor (TCMP)

So in our last class discussion we were trying to see what is there inside a processing element which we have seen that everything pertaining to your instruction pipeline plus the registers and control files. And 1 or 2 levels of caches both private and shared caches are incorporated in the tiles. Your processing elements are organized in tiles fashion, both row wise and column wise then you get a feel of organized tiles that is why it is known as tiled chip multi core processors.

(Refer Slide Time: 01:25)



What is NoC ?

- Processing units interconnected via packet based network
- Each resource is called as a ***'tile'***
- All resources organized as rectangular tiles on the chip.
- Each tile have an address - ***(X,Y)***
- Tiles interconnected by network of routers
- Communication by packet transmission

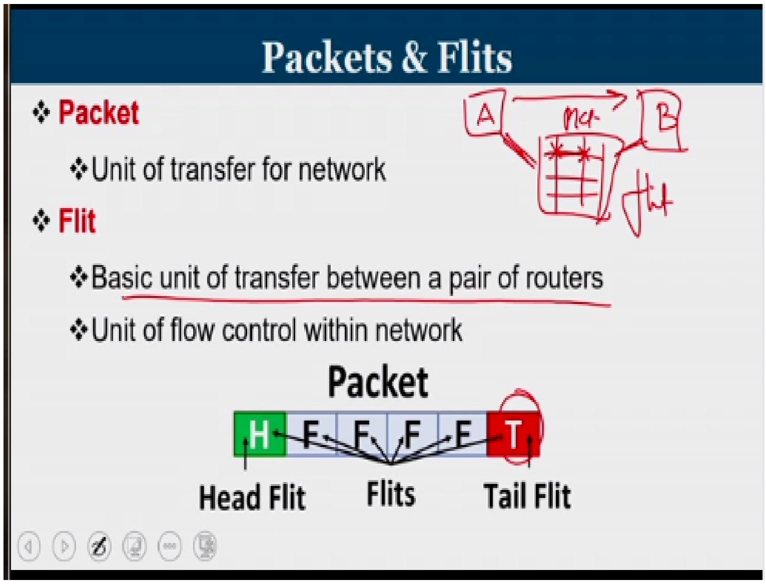
So now we try to understand what is network on-chip, this is the layout of 16 tile multi core processor, where the blue portion houses the processor and the memory hierarchy that is a core and the cache and the green portion represents the routers, which are the interconnect fabric. These routers are interconnected with bidirectional links, so your tiles routers and links form your modern multi core processor.

These processing units are interconnected via a packet based network, each of this resource that is basically your tile is called as a tile. And resources are organize as rectangular tiles on the chip and each of these tiles are having addresses X and Y and tiles are interconnected by network of routers, and communication is basically in the form of packets. So we are no longer sending any signals there like a BUS communication, where somebody is a source and then send some data and it goes all the way to the receiver it is not like that.

We have packet based communication traditionally whatever you see in computer networks, where computers are being connected a computer will create a packet add a header and this is passing through the shared medium at each of the intermediate junctions or routers. This header is being examined, find out what is the source and destination and appropriate intelligent switching or routing decisions are being done by these routers.

The same concept what you see in computer network is inherited into on-chip networks, only thing is we do not have a layered hierarchy everything is physical layer and you have intelligent devices called routers, which is a crucial component of your interconnect fabric. So the traditional BUS based communication is been replaced with packet switched interconnection mechanism or network on-chip communication infrastructure.

(Refer Slide Time: 03:18)



Packets & Flits

- Packet
  - Unit of transfer for network
- Flit
  - Basic unit of transfer between a pair of routers
  - Unit of flow control within network

Packet

H F F F F T

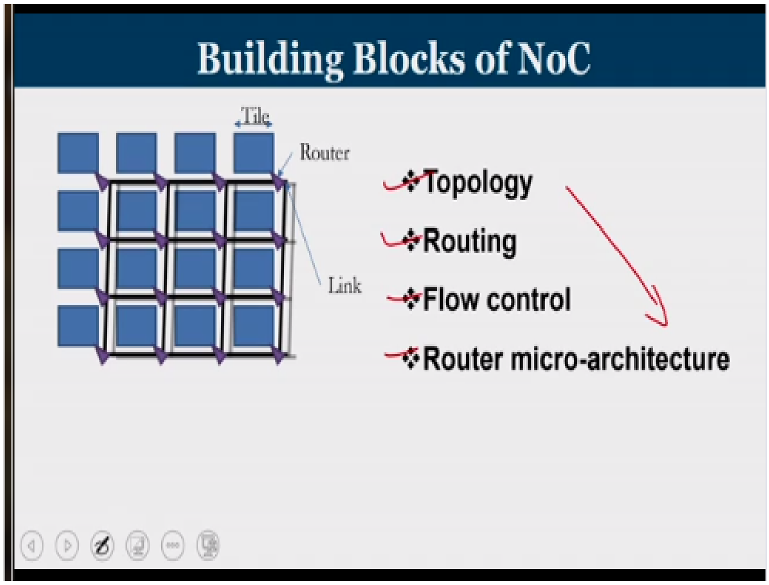
Head Flit Flits Tail Flit

Now packet is a unit of transfer for the network but flit is the basic unit of transfer between a pair of routers. So let us say A and B are the 2 tiles which wanted to communicate and they are going to communicate through a network, what do you send from A to B that is a packet. But they underline bandwidth that connects from A to B which passes through multiple smaller routers cannot take the packet in one go.

So there divide the packet into smaller units, so the basic unit that can be exchanged between a pair of routers is known as a flit. So flit is basic unit of flow control within a network and generally the packet is divided into flits you have a head flit which contains the control information, the routing information source address, destination address, sequence numbers or and then we have a sequence of body flits.

And then at the end we have a tail flit which will release the resources that are been blocked for us upon the travelling of the head flit.

(Refer Slide Time: 04:24)

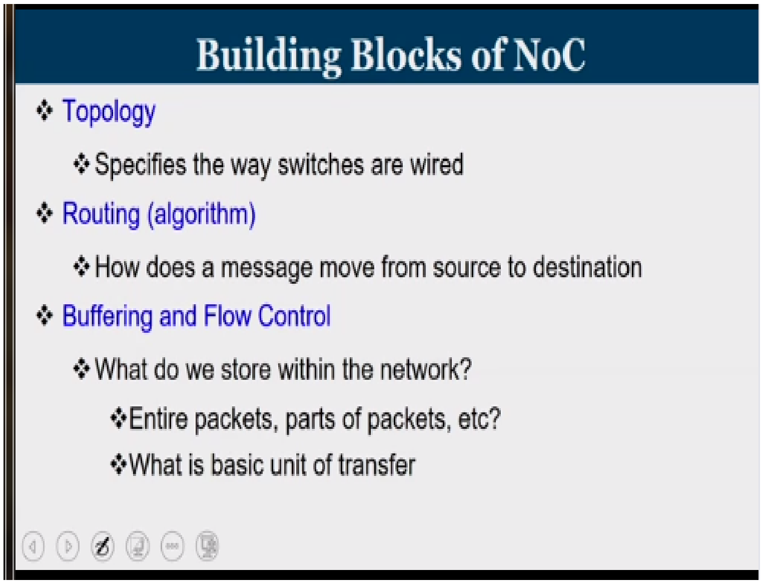


Building Blocks of NoC

- ❖Topology
- ❖Routing
- ❖Flow control
- ❖Router micro-architecture

Now looking into what are the building blocks of network on-chip, basically we have a topology and then the routing will tell you how to send the packets and then we have flow control and router micro architecture. In this course will be just briefly touching upon all these aspects just to get a flavor of how interconnect mechanism work, we are not going deeper into the flow control and router micro architecture studies. So in today's lecture we will work on what is topology and what are the basic routing principles.

(Refer Slide Time: 04:57)

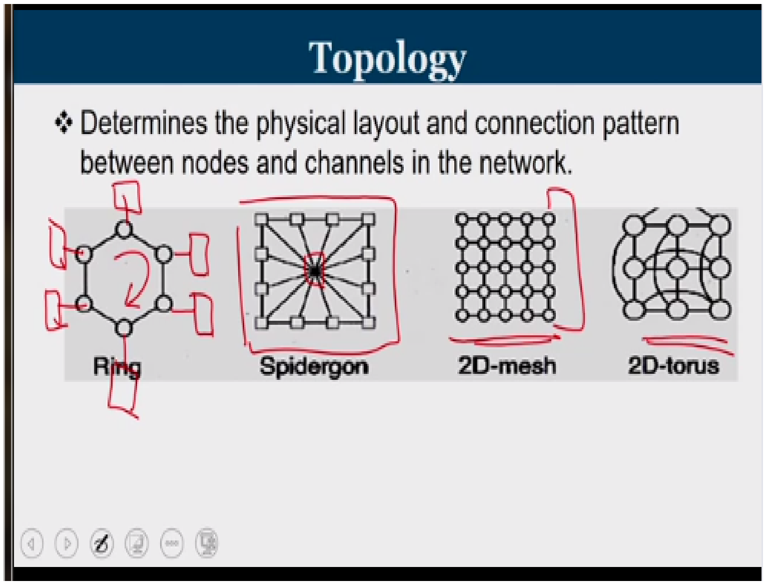


Building Blocks of NoC

- ❖ Topology
  - ❖Specifies the way switches are wired
- ❖ Routing (algorithm)
  - ❖How does a message move from source to destination
- ❖ Buffering and Flow Control
  - ❖What do we store within the network?
    - ❖Entire packets, parts of packets, etc?
    - ❖What is basic unit of transfer

So topology is one which specifies the way how the routers are being connected and routing is basically an algorithm that is implemented on the routers which tells how does a message move from source to destination. And then we have this buffering and flow control will take care of like how packets are moving, what is a basic unit of transfer, how you divide packets into flits etc.

(Refer Slide Time: 05:21)



Topology

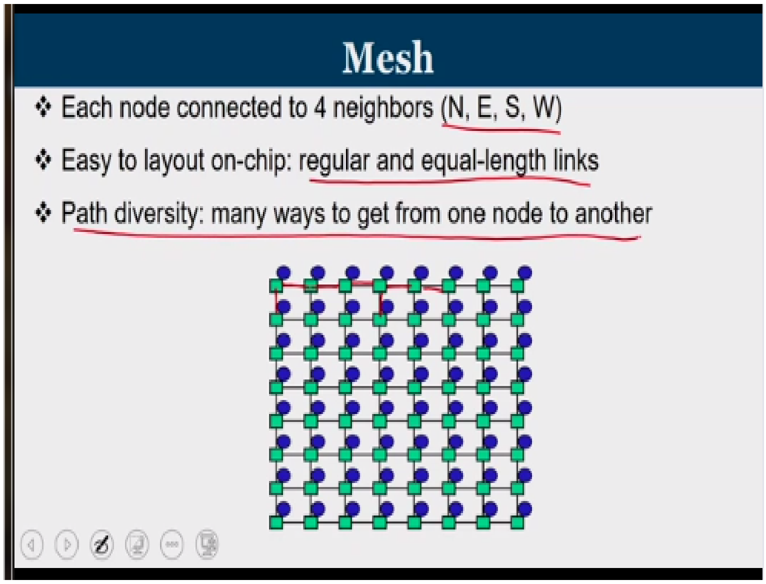
- ❖ Determines the physical layout and connection pattern between nodes and channels in the network.

Coming into topology, topology determines a physical layout and connection pattern between nodes and the channels in the network. If the routers are connected in a ring fashion, what is been shown like this you have to imagine that to each of the routers there is a tie and the tile holes, the processors and the cache. So we have processors and cache and each one of them is connecting to a router, so the circular ones that you see around the ring, they are the routers.

And if the routers are connected in a ring fashion either unidirectional or a bidirectional, we call it as a ring topology. Now the routers can be connected like this now onwards I am going to talk only about the routers, the processor and the cache the tile portions implied. If you connect like this with the central special hub then it is called as spider gun topology this is called a 2D-mesh topology and this is called 2D torus topology.

The difference between mesh and torus is, if you connect a link from one end of the mesh to another end direct high express link then it is called a torus. So in torus the edge nodes are being connected together.

(Refer Slide Time: 06:31)

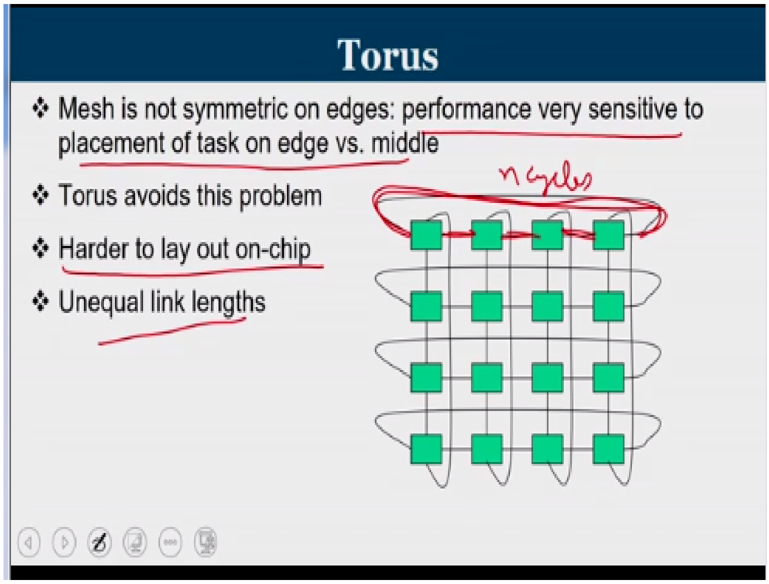


Mesh

- Each node connected to 4 neighbors (N, E, S, W)
- Easy to layout on-chip: regular and equal-length links
- Path diversity: many ways to get from one node to another

And mesh is the most popular one Intel Xeon-phi and Tilera-Tiles-64, they are built with mesh NoC topology, each node is connected to 4 of it is neighbors in north, east, south and west. It is an easy to layout chip, regular equal length links are there. So these are all the links all links are of uniform same length and there is a path diversity There are many ways to get from one node to another.

(Refer Slide Time: 06:57)



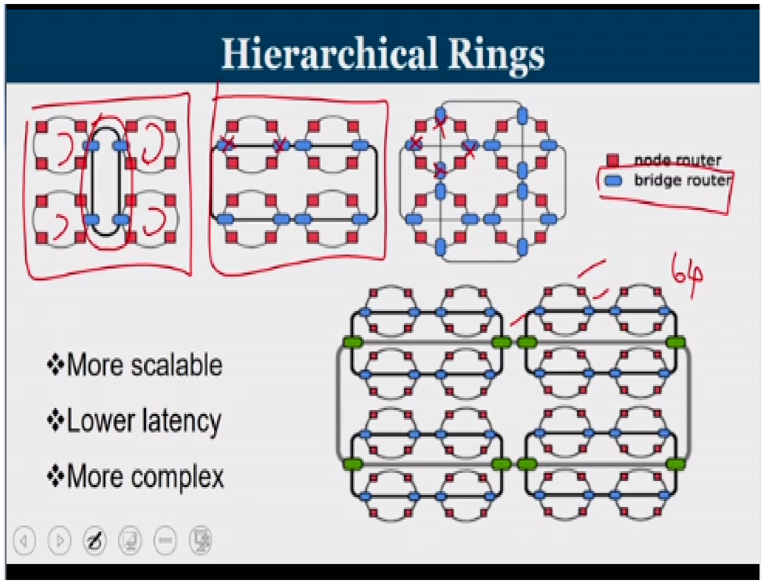
Torus

- Mesh is not symmetric on edges: performance very sensitive to placement of task on edge vs. middle
- Torus avoids this problem
- Harder to lay out on-chip
- Unequal link lengths

Now coming to torus the mesh is not symmetric on edges because those who are in the edge and corner they do not have 4 neighbors. If you make sure that everybody has 4 neighbors then the nodes that are on edge and the corner are additionally being connected to the other end. So it is performance very sensitive to placement of task whether the task is running on edge or in the middle. So this torus is little hard to lay out on the chip because of this special link that goes to the next layer and come back you unequal links.

So some links will take short time and some link, so obviously these links are going to take more time whereas this links will take short time maybe one cycle, this can be n cycles. So handling of this variable length links is yet another design issue in torus.

(Refer Slide Time: 07:47)



Hierarchical Rings

- ❖More scalable
- ❖Lower latency
- ❖More complex

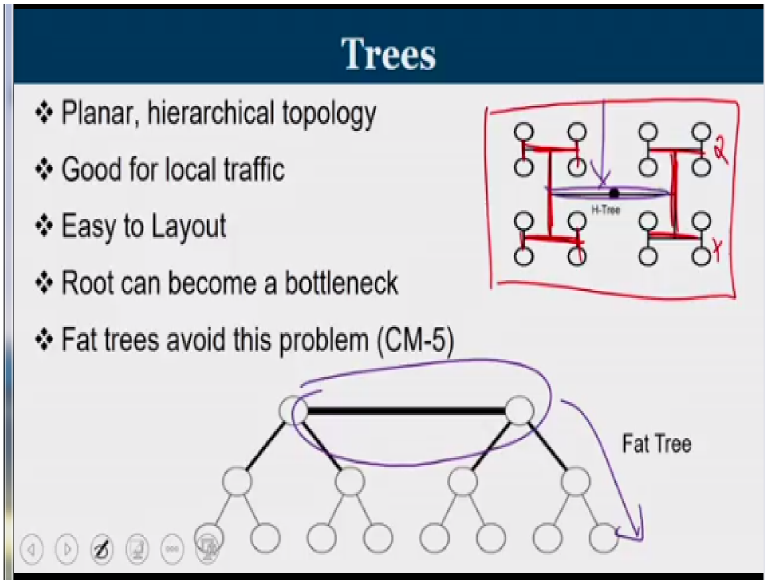
And then we have hierarchical rings, where think of a case that you have 16 routers, so this is 16 routers, wherever you see the red color and each of this router is in turn connected to the tiles that with which houses the processor and cache. Now if the routers in the quadrant are connected using a ring, so you have an 4 rings there. They form the first level ring and each ring itself is connected to the second level of ring that is what is being shown in the blue color they are basically known as bridge routers.

So you have an underlying ring topology and this multiple rings are being connected, so if the source and destination is part of the same ring then you do not go to the bridge router. If they belong to different rings then it is through the bridge router only we will be able to move it further. So this is yet another policy of hierarchical ring connection, so rather than just only one bridge router per ring, we have now 2 bridge routers per ring and here we have 4 bridge routers per ring.

So the concept is basically saying you have 2 levels of rings, one is the first layer which connects the tiles and these ring is in turn connected to the next layer which is called the bridge routers. I can have 3D level where you have 3 different types of bridge routers the red one, which in turn connects to the blue one and the blue one in turn connected to the green one. So this is basically your 64 core tile where I have 64 routers, each of them are part of 3 hierarchical level rings.

Level 1 ring consisting of the red ones, level 2 rings consisting of the blue ones and level 3 ring which consists of the green one. So depending on source and destination, sometimes we travel only through level 1 ring, sometimes we may take level 1 as well as level 2 and sometimes we may have to take all the 3 levels in order to send the packet.

(Refer Slide Time: 09:46)



Trees

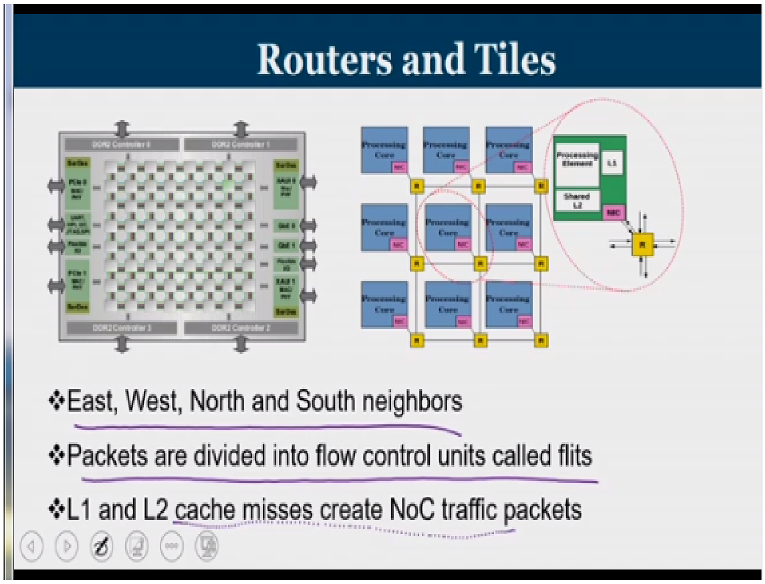
- ❖ Planar, hierarchical topology
- ❖ Good for local traffic
- ❖ Easy to Layout
- ❖ Root can become a bottleneck
- ❖ Fat trees avoid this problem (CM-5)

We have even 3 topologies as well with the planar hierarchical topology easy layout. So this is the general way how a tree layer topology looks like you have 16 routers that is being shown in circles and which are the routers that are being directly connected. So these are the direct connections and if at all they need to be a communication then they take this path, ok and if at all there is a communication between this sector and this sector then I am using this.

And then if at all it is to be connected to through the route, so and then route will have lot of bottle necks. So avoid that I can increase the bandwidth of this route that is called a fat tree. So as you go closer to the leaves the bandwidth reduces because the amount of traffic there is also very less and the routes are handling lot of traffic. So the routes between the routes they should have fat tree mechanism kind of a structure.

So general topologies that is being followed here in the processors that are there they are the mesh topology. Then this spider gun topology, the ring topology row and then torus and last one what we have just heard it is the fat tree topology. So based upon the application if it is an application specific integrated circuit we can have case specific topology or general if it is a general purpose, generally the mesh topology is been use and that is the most popular type of topology.

(Refer Slide Time: 11:22)

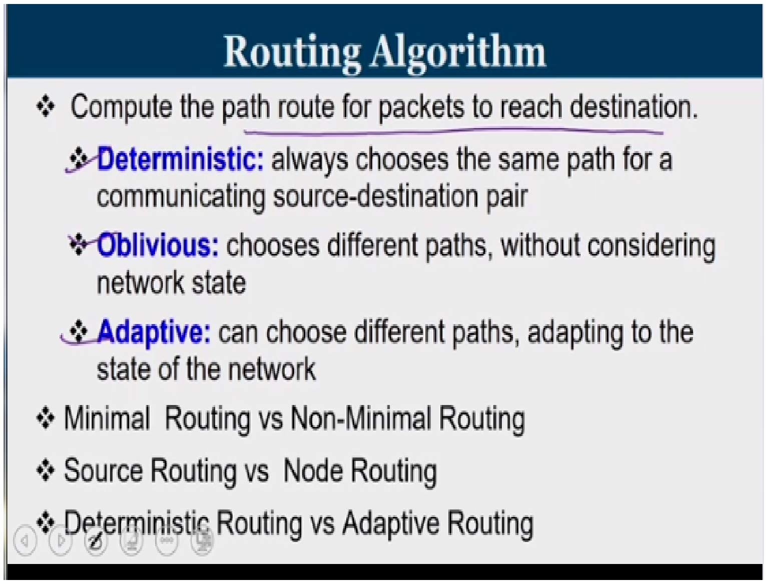


Routers and Tiles

- East, West, North and South neighbors
- Packets are divided into flow control units called flits
- L1 and L2 cache misses create NoC traffic packets

So we have seen that routers and tiles are the basic building blocks of modern multi core communication infrastructure system and each router has an east, west, north and south neighbors. Then packets are divided into flow control units called flits and L1, L2 cache misses create NoC traffic.

(Refer Slide Time: 11:44)



Routing Algorithm

- Compute the path route for packets to reach destination.
  - **Deterministic:** always chooses the same path for a communicating source-destination pair
  - **Oblivious:** chooses different paths, without considering network state
  - **Adaptive:** can choose different paths, adapting to the state of the network
- Minimal Routing vs Non-Minimal Routing
- Source Routing vs Node Routing
- Deterministic Routing vs Adaptive Routing

Now coming into routing algorithm, routing algorithm compute the path for a packet to reach the destination, 3 types of routing deterministic routing, oblivious routing and adaptive routing. In the case of a deterministic routing whenever there is a packet from A to B all packets from A to B travel in the same route, irrespective of the current topology or traffic conditions. So there is a prefixed path between A and B that is called deterministic routing.

The second one called oblivious routing, where packets from A to B can take different paths but the path choosing is not based upon adaptive runtime condition. So let us say the first packet from A to B will go through r1 route 1, second packet goes through r2, third packet goes through r3, so this prefixed r1, r2, r3 like that. But I am not looking into where should I choose r1 now, should I choose r2, now depending on some runtime input, no.

It is completely taken or the decision is completely taken based upon a random choice that is called oblivious routing. The third category is known as adaptive routing which takes care of the run time input the present traffic topology systems are been studied and based upon that appropriate decisions are taken on runtime that is what is known as adaptive routing. So that is one level of classification similarly we have 3 more other levels of classification.

The first one is called minimal routing versus non minimal routing, minimal routing means if a packet at every step is coming closer to the destination then that is minimal routing, at every hope my packet is reaching one step close to destination. If this minimality is not imposed then it is called non minimal routing. The second category of routing deficient is called source routing versus node routing.

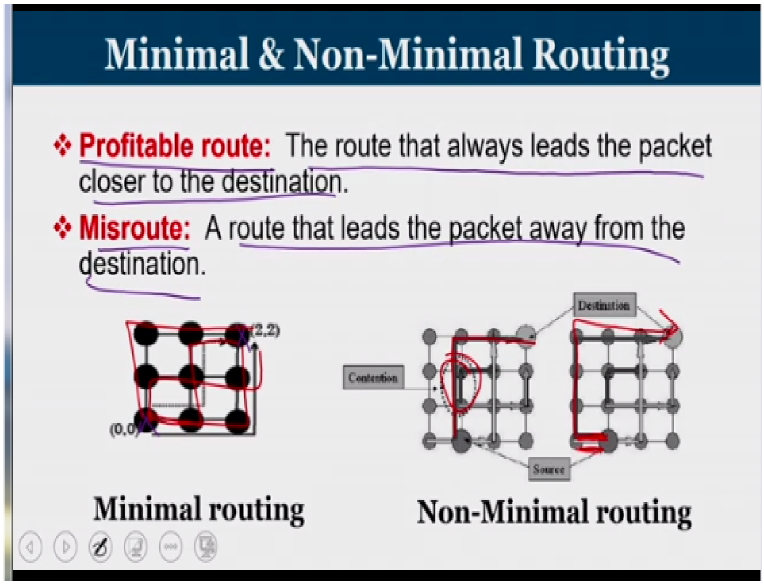
In the case of a source routing a packet before it is starting from this source node the entire route it has to take is embedded on the packet header. So at the source itself the entire route is studied and the route is stored, now at each intermediate junction the control logic only looks into what is there in the header and based upon that take up the appropriate junction like east, west, north and south.

So this is not a scalable approach when you go to large networks the route information itself is very big and that will require a substantial amount of your header space. The second aspect is called node routing, so it is a distributed routing where at the source I am just forwarded to my immediate neighbor, upon reaching that neighbor you tell what is the destination. From the destination information the control logic that is embedded in the router will find out the appropriate output port.

So at every junction each of this routers are intelligently using their logic in find out the best outgoing port to these incoming packets. So in source routing, everything is determined at the source and you are just making use of those information in the intermediate junctions. In the case of node routing, every router will tell you what is a next immediate neighbor or next outgoing port and depending upon that at every point it is a decision that is being taken by considering the logic of the router also into it.

So any change in topology that happens a source routing requires lot of effort in order to reconfigure lot of changes are to be incorporated. But in the case of a node routing even though there is a change because my decisions are taken run time it is not going to impact much. The third category is called deterministic routing and adaptive routing that we have just seen.

(Refer Slide Time: 15:33)



Minimal & Non-Minimal Routing

- **Profitable route:** The route that always leads the packet closer to the destination.
- **Misroute:** A route that leads the packet away from the destination.

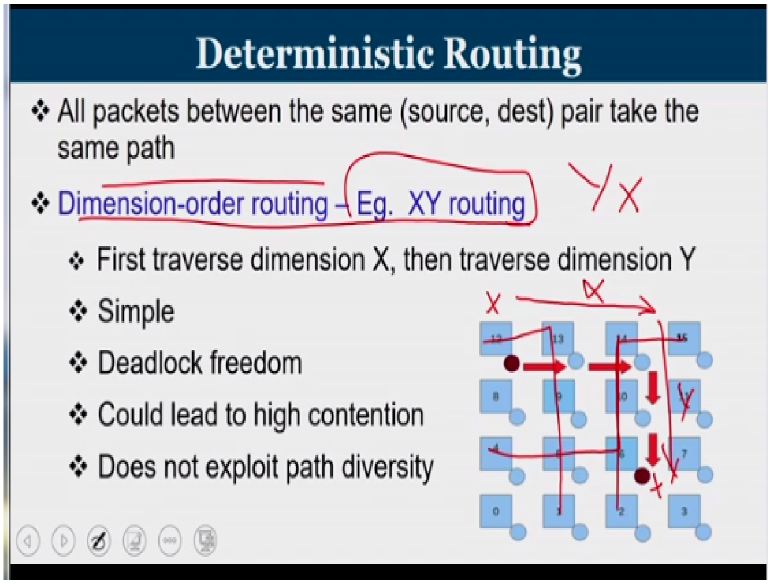
Minimal routing

Non-Minimal routing

Now one small illustration of minimal versus non minimal routing, so minimal routing is also known as profitable route, the route that takes packet closer to the destination and a misroute is also known as non minimal route, a route that leads the packet away from the destination. So consider the case a packet from 00 wanted to go all the way up to 22 this is minimal path because every step it is taking me closer to destination this is also minimal, this is also minimal, this is also minimal, so there are many minimal paths.

Now in this case you just see if I wanted to move from source to destination that is a minimal path, imagine that there is a congestion here. So if I determine to bypass that, so whatever route I am taking now I am not going in the minimal path because a moment I take up this path then I am moving away from the minimal route, so we have profitable route and misroute.

(Refer Slide Time: 16:35)



Deterministic Routing

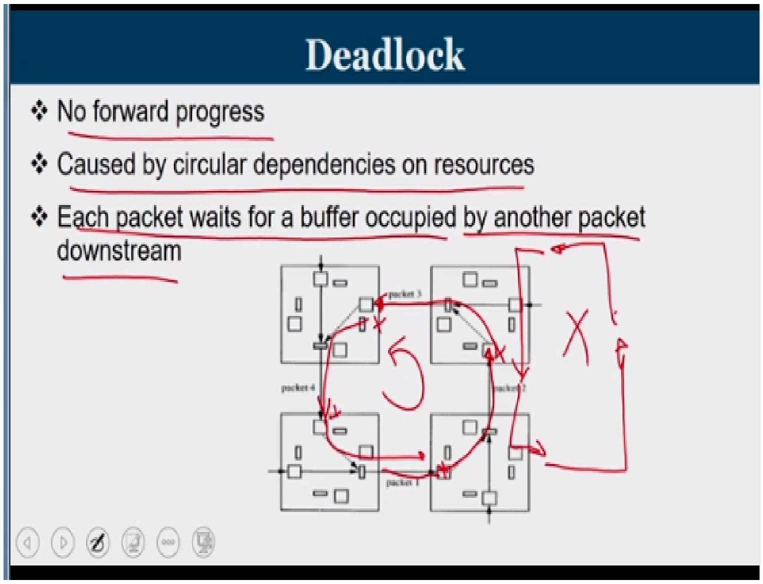
- ❖ All packets between the same (source, dest) pair take the same path
- ❖ Dimension-order routing – Eg. XY routing
  - ❖ First traverse dimension X, then traverse dimension Y
  - ❖ Simple
  - ❖ Deadlock freedom
  - ❖ Could lead to high contention
  - ❖ Does not exploit path diversity

In deterministic routing all packets between the same source and destination pair take the same path, dimension order routing is a classical deterministic routing. So consider router number 12 let us say there is a packet which I want to send from router number 12 or from tile to l2 to tile 16. And the squares represent the tiles, the circles represent the routers from the diagram we can see that. The packets are first travelled in the X direction and then it takes a 90 degree turn in Y direction this is called dimension order routing.

First I complete one dimension X then I take a 90 degree turn and then move to the other dimension. So XY routing is an example similarly all routers can follow YX routing also, so if you have a packet from 12 to 1 this is the path X and Y. If a packet is there from 4 to 14 this is the path, similarly if a packet is just travelling from 15 all the way to then this is the path XY routing.

It is a more simplest approach it is ensuring deadlock freedom and then but the problem is every packet is travelling through the same path, so that could lead to high contention, it does not exploit path diversity.

(Refer Slide Time: 17:57)



Deadlock

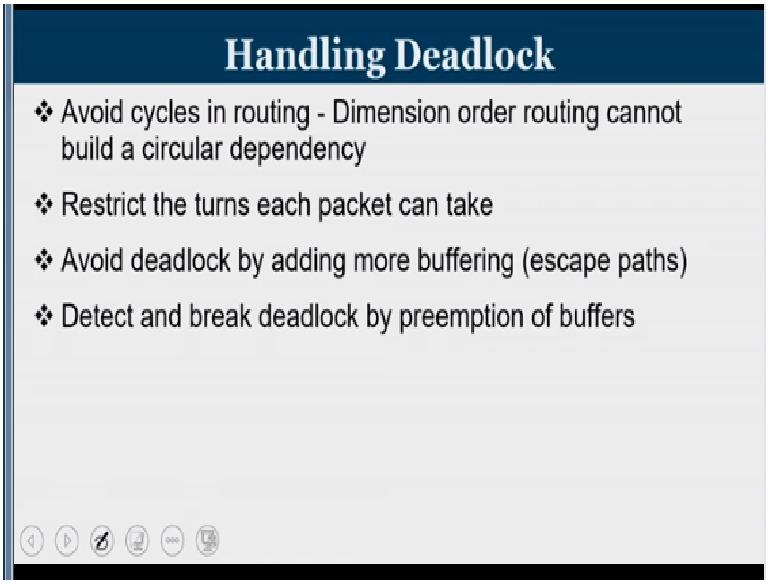
- ❖ No forward progress
- ❖ Caused by circular dependencies on resources
- ❖ Each packet waits for a buffer occupied by another packet downstream

Now let us try to understand how deadlock happens, deadlock is a scenario where there is no forward progress, it is caused by circular dependencies on their resources. And it is happening when each packet waits for a buffer occupied by another packet in the downstream. Consider the case that you have a packet that is coming from here it is sitting in one of the buffer, now the aim of the packet is to go to the north.

So it is trying to get into this buffer but that buffer is not available because somebody else is sitting there and they are trying to look for a buffer here that is also not available, somebody else is sitting there they are looking for this buffer that is also not available because they are looking for this. So this is called a circular dependency, a packet is looking to get a buffer which is in turn hold by some other packet that in turn is looking for another buffer like that you have a sequence of circular dependency.

This happens because a packet which is travelling in X direction trying to go to Y direction and some other packet travelling in Y direction wanted to go to X direction like that it forms a cycle, so this kind of movements has to be broken.

(Refer Slide Time: 19:17)



Handling Deadlock

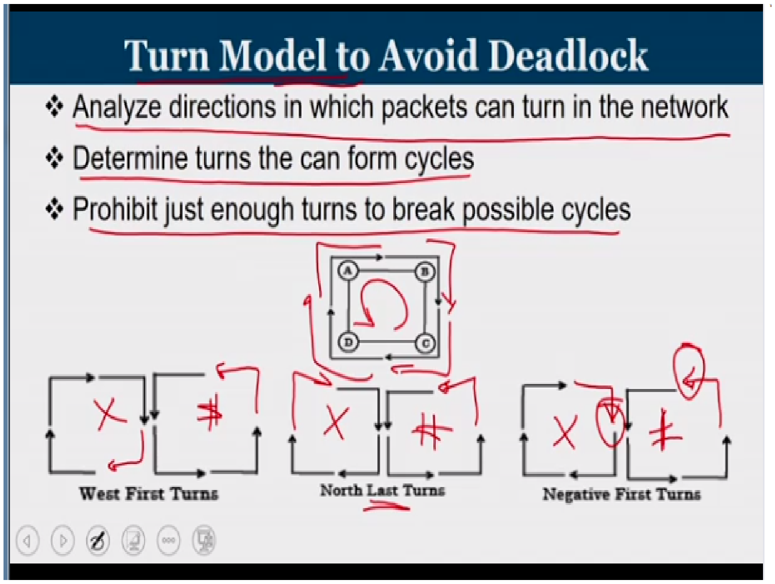
- Avoid cycles in routing - Dimension order routing cannot build a circular dependency
- Restrict the turns each packet can take
- Avoid deadlock by adding more buffering (escape paths)
- Detect and break deadlock by preemption of buffers

So how are you going to handle deadlock, you have to avoid cycles in the routing, in the case of a dimension order routing like XY routing, there is no cyclic dependency or circular dependency. Because in the case of dimension order routing packets will take only this path travel in X direction and then take a Y direction. With these kind of turns with these kind of permitted turns you will never get a cycle, you will never have a scenario where you do movements like this because these are all YX transition.

So when I implement XY routing in all the routers all of them are following XY routing, you will never get a case where these kind of movements happen. So avoid cycles in the routing that is a best ways it is an dimension order routing is already taking care of it. But dimension order routing is highly restrictive it do not give you flexibility in dealing in congestion. So restrict the turns each packet can take that is only one solution, avoid deadlock by adding more buffer.

So that you have escape paths when there is deadlock happening , so we will learn a couple of routing algorithms, which are been implemented in NoC or if deadlock happens we need to have mechanisms by which we can detect it and once it is detected then we can break deadlock by preemption of buffers.

(Refer Slide Time: 20:45)



Turn Model to Avoid Deadlock

- ❖ Analyze directions in which packets can turn in the network
- ❖ Determine turns the can form cycles
- ❖ Prohibit just enough turns to break possible cycles

One such classical method by which we can come up with deadlock free adaptive routing is called turn model. In this turn model, you analyze the direction in which packet can turn in the network, determine turns that can form cycles and prohibit just enough turns to break the possible way. So we have our aim is to come up with routing algorithms which are better than XY routing XY routing is very restrictive but it is a deadlock free routing.

In this turn routing it is permitting you few more other turns sometimes you can take YX transitions as well but at certain places certain kind of turns are been prohibited. The whole purpose of prohibiting the turn is to prevent formation of deadlocks. So like what I mentioned it is a circular dependency that is going to create the deadlock this is one kind of dependency that can happen in clockwise manner.

Similarly you can have one more that will happen in the anticlockwise manner as well, the first category of routing is called west first turn. In west first turn if at all a packet wanted to travel towards west direction then right from the source travel only in west direction. If you take any non westward movement then no more west turning is been permitted. So there would not be any case of you travelling through north and take a west, there would not be any case where you travel in south and take a west.

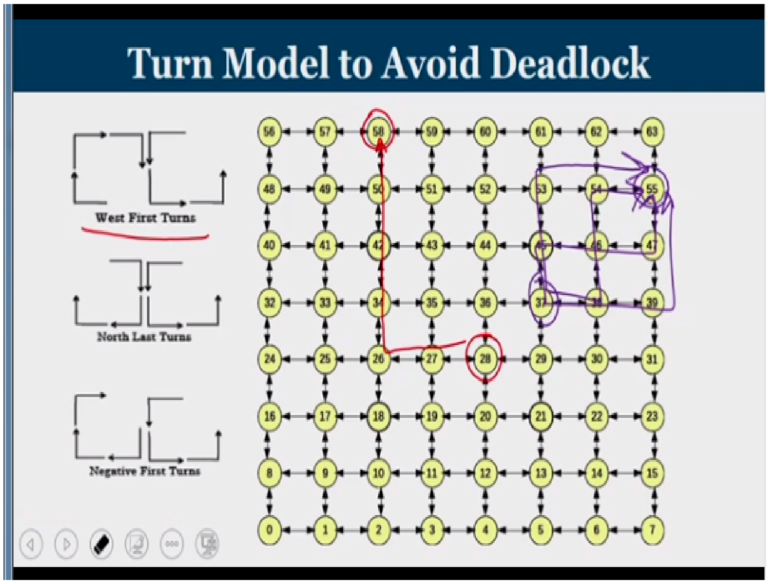
So southward moment followed by westward deviation, northward moment followed by westward deviation, they both are prohibited. If you look at this the one that is missing is the red arrow that is travelling in southward direction and then moving to west and travelling in northward direction and then moving to west both are been prohibited in the case of west first turn. So any turn or any movement towards west it has to be done first.

So once you take a non west turn then trying to move towards west direction is prohibited hence the name west first turn similarly we can define east first turn, north first turn and south first turn etc. The next category is called north last turn, so here the term as the name indicates it is last if if all you wanted to move to north then do it at the end. So you would not be having a case where you are travelling in the north direction and then moving to east, any movement towards east you should have done earlier and north is the one that you take last.

So north east turns are not there, similarly north west turns are also not there if at all you travel in north, so if you travel in north then westward and eastward turns are been prohibited. Similarly we can define south last turn, east last turn and west last turn, now we come to negative first turns algorithm, in negative first turns algorithm the generally the movement towards west and movement toward south are considered to be negative.

So while moving in the negative direction, so movement towards west and movement towards south is been prohibited. So with that we can actually get rid of deadlocks, so in all these cases we can see that we are breaking one of the turn in clockwise direction and one of the turn in anticlockwise direction, one turn in clockwise and one turn in anti clockwise direction that is the way in which it is being balanced.

(Refer Slide Time: 24:18)



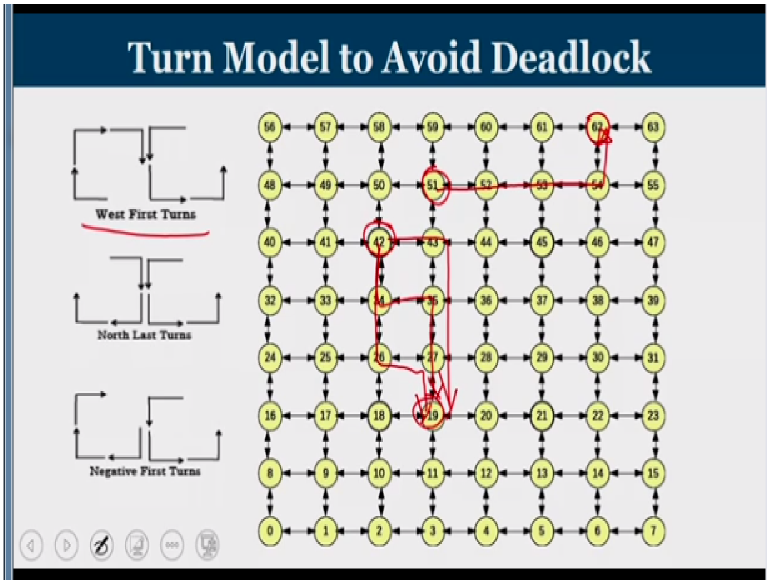
Turn Model to Avoid Deadlock

Let us now consider few examples in which we are going to work hard, take the case of west first turn. Let us say there is a packet that wanted to move from 28 to 58 west first, so 28 is a source and 58 is a destination since it is west first I have to travel to west. Because my destination is on west of source, so travel in the west direction and then you come like this. Now you may wonder this is same as XY routing.

For this particular source destination pair XY routing in west first routing is same, now let us consider my source as 37 and destination is 55, I am applying west first routing. Actually there is no need to travel to west because my destination is towards east of 37. In this case there is no restriction if at all I travel to west it should be first, so I can travel like this that is route number 1, this is route number 2, this is route number 3, this is route number 4.

Like that all options are there because I am not violating the west first turns. So if the destination is towards the east of a router then west first routing will give you the very best performance, similarly we can try with north last routing also.

(Refer Slide Time: 25:42)

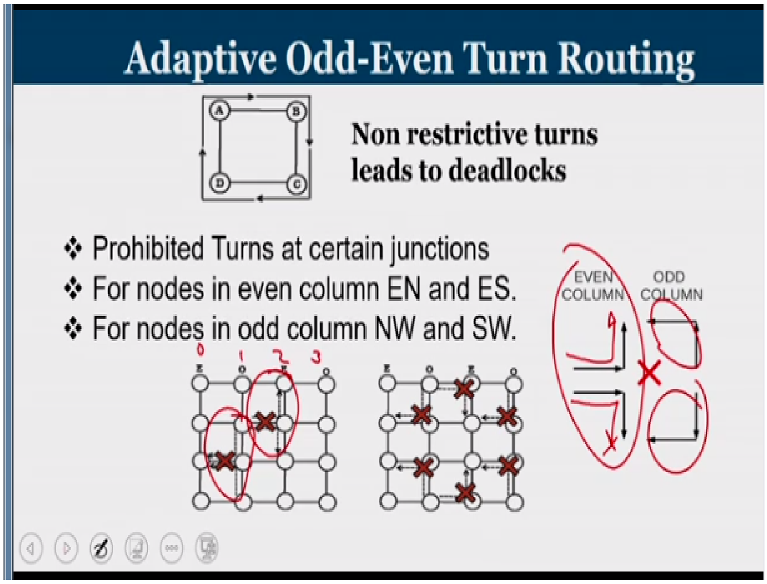


Turn Model to Avoid Deadlock

So consider the case where we have a packet that is moving from 51 to 62 which it is north last I have to travel in the east direction and then take north at the end if there is something from 42 to 19. So here destination is since out of 42, so there is no restriction at all there is if at all you travel to north then travel last, since I am not travelling to north, I can take this path. So different packets starting from 42 which is travelling all 19.

They can take XY and Y and X all combinations there is restriction in which column you can take Y which column you can take X as far as north last it is been not violated. The whole point we have to understand from here is every router has to incorporate the same routing, it is not like some router will do east last and some will do north first no that is not possible, to ensure that certain turns are been prohibited, every router has to be implemented with one of these routing.

(Refer Slide Time: 26:45)



Adaptive Odd-Even Turn Routing

Non restrictive turns leads to deadlocks

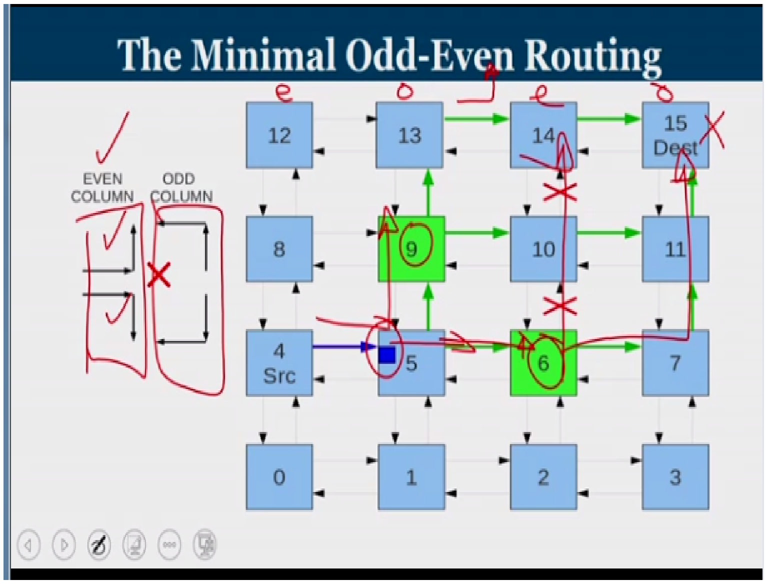
- Prohibited Turns at certain junctions
- For nodes in even column EN and ES.
- For nodes in odd column NW and SW.

Moving on further let us try to understand odd even routing, what are the problems of the turn routing that we have seen west first turn the north last routings. They are restrictive on certain junctions, in certain junctions I cannot take certain turns. So adding little bit more flexibility into the system that is called odd even routing, in odd even routing we divide the entire chip into odd columns and even columns.

So in odd columns certain turns are prohibited, in even columns certain other turns are been prohibited let us see that. In even columns east north and east south this is called east north travelling in east and moving to north, travelling in east and moving to south are prohibited. When it comes to odd columns northwest and southwest are been prohibited, so if you do this then also we can break the deadlock.

So in a 16 core processor what is been shown zeroth column, first column, second column and third column 0 and 3 is even. So even columns these turns whatever is been shown in the diagram these turns are not allowed and in odd columns these turns are also not allowed similarly the same thing happens in the other one as well.

(Refer Slide Time: 28:07)



The Minimal Odd-Even Routing

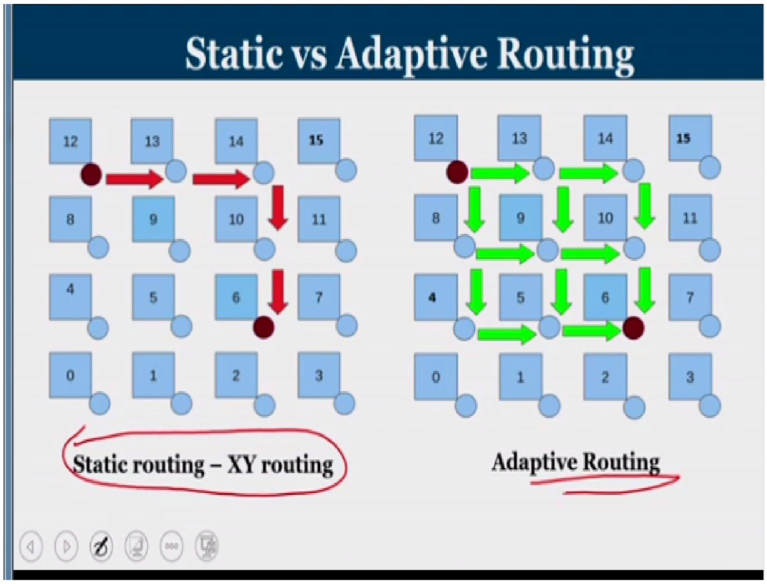
Now consider the case where we have a 16 tile multi core processor let us consider the routing in this, imagine that a packet has already reached tile number 5 or router number 5. Now this is an even column this is odd column, even column and old column. So now we know that in even columns these are the turns that are prohibited and in odd columns these are the turns that are prohibited.

Now you have a packet that is already reached 5 my destination is 15, it is minimum odd even routing. Now a packet which is come like this, nobody is restricting mean going up and nobody is restricting mean going straight. Because it is an old column only in even column we have this restriction, so I can take a path to 9 as well as a path to 6. So basically at router number 5 you have 2 options packet can be forwarded to router 6 and packet can be forwarded to router 9.

Now let us imagine that the packet has taken 9, in 9 it has 2 options it can try either 13 or 10. So wherever you have seen the green path that is a path through which the packet can travel. Now imagine the packet has reached 6, let us say the packet is moving and this reaching is 6, to reach 15 generally if you look in both north and east looks a possibility. But I cannot travel to 10, I cannot take a north turn.

Because in an even column these turns are not allowed this been actually mentioned here even columns . So travelling from 5 reaching 6 and then travelling north is prohibited, so this movement is not allowed whereas if it is reaches 7 then that is allowed. So you are odd even routing takes care of the column at which certain turns are happening and that is the way how it is been proceeded.

(Refer Slide Time: 30:03)



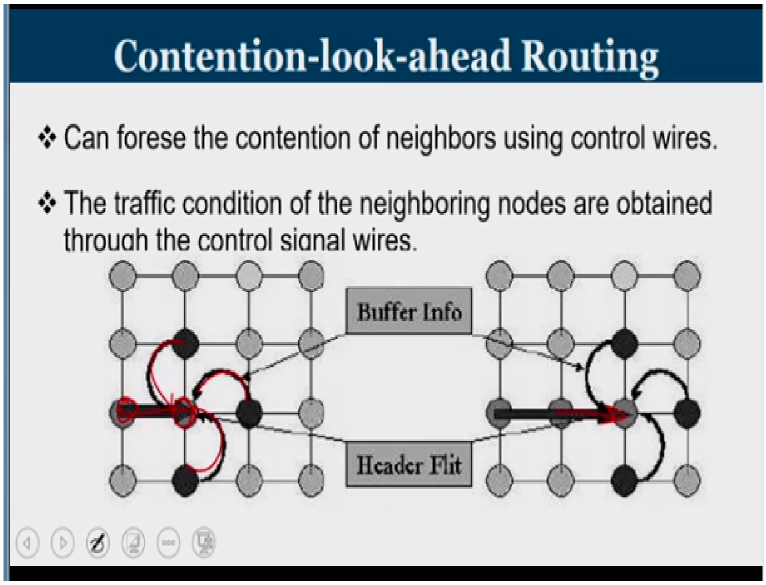
Static vs Adaptive Routing

Static routing – XY routing

Adaptive Routing

So we have this compared in a static XY routing they are exist some restrictions in which you do and you are bringing more adaptivity into XY routing by combining the turn model and the minimum odd even model. And adaptive routing is the technique by which packets from a source to destination can travel in whichever direction provided that takes care of the deadlock issues. So any adaptive routing that is been designed on network on-chip has to be free from deadlock.

(Refer Slide Time: 30:36)



Contention-look-ahead Routing

- Can forese the contention of neighbors using control wires.
- The traffic condition of the neighboring nodes are obtained through the control signal wires.

And contention-look-ahead routing means we should be in a position to oversee the contention of neighbors using special wires. The traffic condition of neighboring nodes are obtained through the control signal wires, so think of a case that you have a packet that is here. Now it is going to come to the next router, now in order to go further through special control wires from all the neighbors I am trying to get real traffic updates and based upon that I move further.

So in the general architecture every router is connected to it is neighbor and there is a flit channel through which your data flows, there is an opposite direction through which the credit flows through feedback flows. So before sending a data every router will in turn check the suitability of forwarding the data over there. So we are getting feedback information it can be related to the number of buffer occupancy, it can be related to congestion information, everything is being shared through another channel which is known as the feedback channel or the credit channel.

We will learn more about them in the next lecture, where we focus ourselves on execution of this task. So with that we come to the end of the basic principles of network on-chip and it is topology and routing aspects, remember network on chip is the new communication paradigm traditional computer network principles are embedded into modern microprocessor chips. So that your communication from one end of the chip to another is been taken and care of, with that we conclude this lecture thank you.