

## Multi-core Computer Architecture – Storage and Interconnects

Dr. John Jose

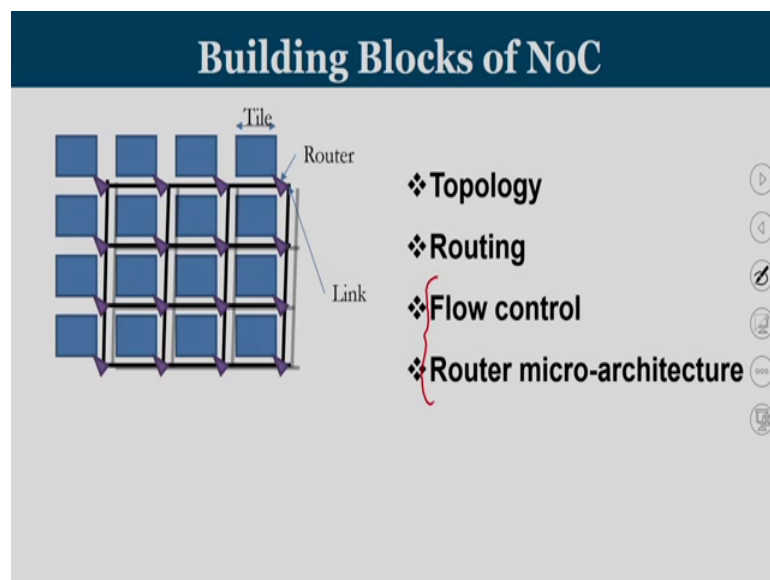
Department of Computer Science and Engineering  
Indian Institute of Technology, Guwahati

### Lecture – 14

#### Network On Chip Router Micro-architecture

Welcome to the 14th lecture. Today, our focus will be more on in understanding the Internal Architecture of an NOC Router. In the last 2 lectures, we were trying to understand, what is the role of NOC and what is routing and the different types of topologies that we use in chip Multi-core processors. We have seen this 4 Building Blocks of NOC over the last 2 lectures.

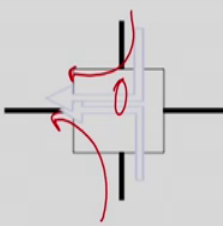
(Refer Slide Time: 00:58)



Today, our focus is more on trying to understand this Flow control aspect and the Router micro-architecture aspect.

(Refer Slide Time: 01:10)

## Handling Contention



- ❖ When two packets trying to use the same link at the same time ?
  - ❖ Buffer one ( widely used mechanism)
  - ❖ Drop one
  - ❖ Misroute one (deflection)

So, how are you going to handle contention when two packets are reaching a same router and two packets one coming from the north and other coming from the south; both wanted to go in the same west direction.

So, when two packets trying to use the same link at the same time, what we can do? One approach is since we use Buffers inside the routers we (Refer Time: 01:41) buffer one of them and permit the other one to take the output channel or we can drop one of the packet or we can misroute one packet.

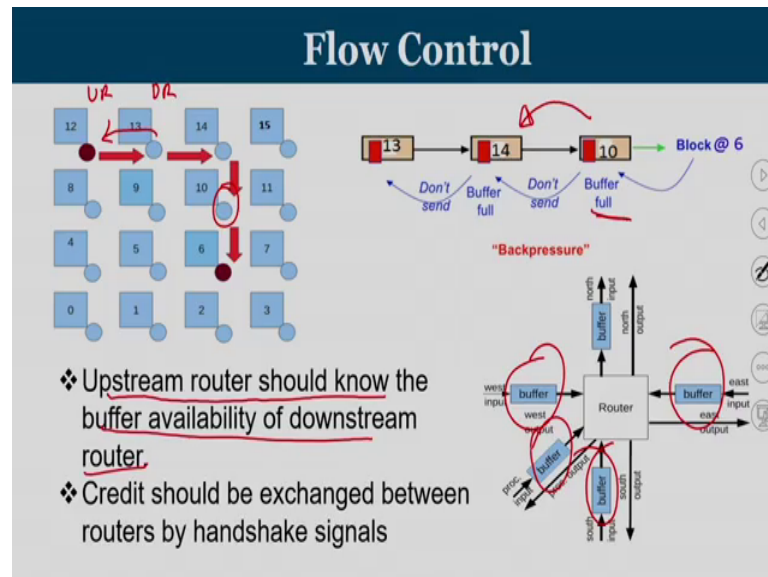
So, these are the 3 different strategies that are used in order to handle contention. So, contention is a scenario where multiple packets are looking for same output port. Since, the bandwidth is limited only one packet can go through that decide output port; the other packet we have to handle. 3 approaches; one is Buffering, second one is Dropping and the third one is Deflecting.

Buffering is keeping the packet, retaining the packet for the next cycle and try it chance in the next cycle. Then, the second approach of dropping means we have to again restart these packets activities back from it source. The third one is trying to rerouted through some other port.

We will try to understand, what is the concept of flow control? So, we are going to discuss about network on chip which are using the buffering concept. So, packets will be

buffered in the routers still it get a productive port. There can be scenario where you want get a productive port in the current cycle due to conflict between other packets or due to lack of buffer availability in the next router.

(Refer Slide Time: 02:57)



So, consider a case, we have a packet that is moving from 12 all the way to 16 by using your x y routing.

So, Upstream router should know buffer availability of a downstream router. Let us try to understand what do you mean by upstream and downstream? So, 12 is the upstream router and 13 is the downstream router as per as the packet is concerned. Let say the packet is a 12. 12 should know upstream router should know the buffer availability of a downstream router. So, before 12 is going to sent a packet to 13; 12 should know whether there is a buffer that is available in 13.

So, how it is been done? Credit should be exchanged between the routers. So, 13 should send some feedback signal back to 12. This feedback mechanism is there between every router to its neighbors because packets can come from any one of its neighbors. So, basically flow control is the process by which an upstream router that is one router will know about whether there is buffer available in the next router or not.

So, likewise if the buffer is available, a packet can be forwarded to the next router. If the buffer is not available, then we have to retain this packet and try the same process in this

subsequence cycles. Let us try to understand see consider the case that in 6, the buffer is full. So, Buffer full scenario.

So, over a period of time buffers will come to 10 and slowly will get accumulated in in the buffers of 10 and over a period of time, 10 will be full. So, 10 is going to communicate back to 14 that buffer is full Don't send and this mechanism is totally known as "Backpressure".

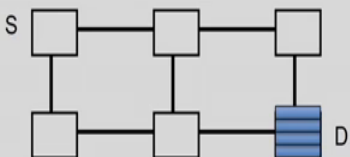
So, we know that every routers has a set of buffers in its input. The packets will come and it will be the residing there, you have this buffers that is coming and once you are in the buffer, you try to see whether there is a buffer that is available in the next router and based upon that we are going to ensure the smooth flow of packets. So, flow control is basically a technique that will help packets to smoothly flow and that is being facilitated by a proper handshaking mechanism between adjacent routers.

Now, we will try to understand what are the different kinds of flow control mechanism that is been adapted? Since, network on chip concept is inherited from the traditional computer networks. Let us try to see can we use the same flow control mechanism that is used in macro network. So, traditional computer networks, can we inherit them in network on chip?

(Refer Slide Time: 05:42)

### Store and Forward Flow Control

- ❖ Store and Forward Packet based flow control
  - ❖ Packet copied entirely into network router before moving to the next node
  - ❖ Flow control unit is the entire packet
- ❖ Leads to high per-packet latency
- ❖ Requires buffering for entire packet in each node



Can we do better?

So, the normal flow control mechanism that is adopted in traditional computer network is store and forward packet based flow control. So, what happens is a packet is copied entirely into the network router before it is moving to the next node.

So, let us say we have a router at S. Now, from S you are moving to the next router let us say it is A and B, then it is going to D. So, the entire packets, these are all flits of the packets what you see the flits of the packet it is getting completely copy to A and once it reaches A, you perform some error detection and correction. Once a packet is fully received you move to B. So, the same process that is why it is mentioned that packet is copied entirely into the network router before moving into the next node.

So, these make sure that once the packet moves from one router to another, then there is no need to hold for this packet. The next point is this will lead to high per packet latency. So, a packet cannot move further until all of the current packet, that means flits of this particular packet is getting stored in the current router.

This will lead to a scenario such that even if I can move further, since my tail has not reached the packets forward progress is blocked. This will surely lead to high per-packet latency.

And moreover we requires buffering to the entire packet. Every node should be able to accommodate a full packet because until if the full package comes I cannot make a forward progress. So, in traditional computer networks where buffering is not that a problem, buffering is not cause is that much a costly kind of an issue it is no longer practice.

So, they basically use store and forward switching. But such a kind of a costly buffering is not possible once it comes to network on chip, where in the available chip space that much of buffering is not possible and moreover it is going to incur high latency.

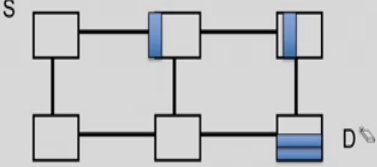
So, this is how it does if we can see the first packet is going to come and that the first flit has come, second flit has come. We can see the third flit is coming and the final flit is coming. Now, the entire flits reached the next router. So, based upon that the packet is going to advance to the next one, advancing from one router to another happens fully in terms of packets; all the flits get copied and then, only it is moving to the next destination.

So, you require a total buffering inside each of the router. Can you do better than this that is what is a design issue.

(Refer Slide Time: 08:30)

### Cut through Flow Control

- ❖ Another form of packet based flow control
- ❖ Start forwarding as soon as header is received and resources (buffer, channel, etc) allocated
- ❖ Dramatic reduction in latency



❖ What if packets are large?

Another aspect of flow control that is been practiced in macro networks is called virtual cut through switching. It is another form of packet based flow control. So, here what we do is start forwarding as soon as the header is received and you have buffers available.

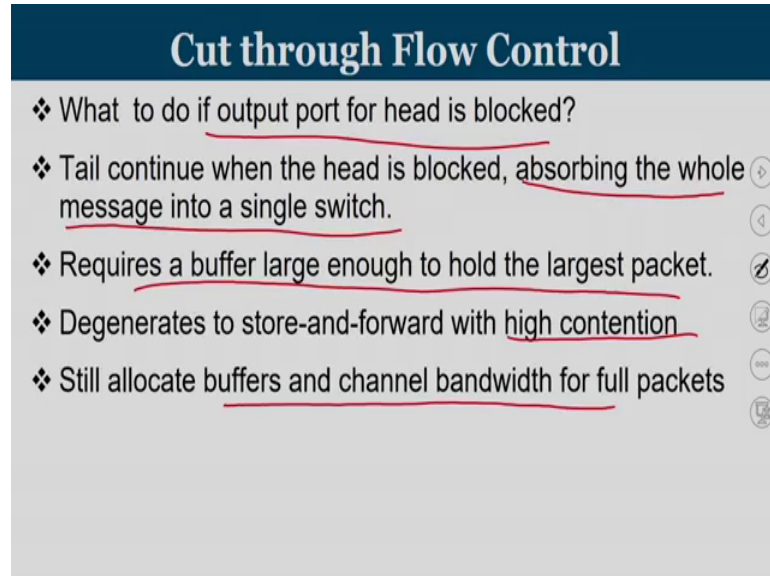
So, the first flit is going in the head flit is going to reach the next router, we do not wait for the entire flit to reach meaning the entire packet to reach that is different flits of the packet to reach, whenever the head flit can move further forward the head flit will advance. So, you can see that this will dramatically reduce the latency because we are not waiting for the entire packet to reach, store and forward concept is not there.

So, let us try to see what happens here. What is the packets are very large, we will see that issue. First flit is moving; second flit will reach. Now, you can see that the head flit has moved on from here the head flit has moved on, the tail flit has not yet reached there. The second flit is also moving.

So, the head flit now if you look at the snapshot, you can see that the flit is there and many routers; flits of a single packet is occupying in many routers, but this has its own problem that is associated. In worst case sometimes a head flit may not be able to move.

So, in that case, the tail flits are going to move further and the buffering requirement is still maximum.

(Refer Slide Time: 10:07)



### Cut through Flow Control

- ❖ What to do if output port for head is blocked?
- ❖ Tail continue when the head is blocked, absorbing the whole message into a single switch.
- ❖ Requires a buffer large enough to hold the largest packet.
- ❖ Degenerates to store-and-forward with high contention
- ❖ Still allocate buffers and channel bandwidth for full packets

So, what to do if output port for the head flit, if the output port for the head is blocked? So, the tail will continue as, but still the head flit is blocked. So, that will lead to absorbing the whole message in a single switch. So, the advantage of virtual cut through switching is sometimes you could reduce latency.

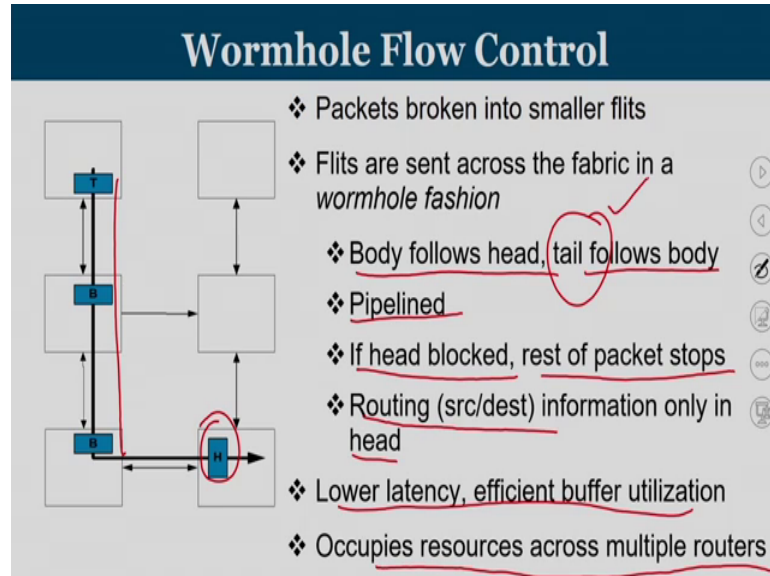
The head flit can advance as long as buffers and channels is available for him to make a forward progress. So, similarly every flit will advance as long as there is a space available in the next router and it get the resources. But sometimes the head flit maybe blocked, but tail flit can still advance.

So, the buffering that you need to keep inside a router is still maximum. There should be buffering for the entire packet, but when compared to store and forward switching this is going to have little bit of reduction in the latency, but there is no reduction in terms of power and area conception because still you need to have maximum space permitted for a packet.

So, it requires a buffer large enough to hold the largest packet. So, when there is high contention, then the head flit cannot move; the tail flit will fully come. Then, it is kind of

a process like store and forward switching. Still we have to allocate buffers and channels bandwidth for the full packet.

(Refer Slide Time: 11:30)



Another category that is been to typically followed in the network on chip is known as Wormhole Flow Control. So, packets are divided into smaller units called flits that we have seen. Flits are sent across the fabric in a wormhole fashion. So, body flit follows the head flit; tail follow the body flit and that happens in pipelined. Even if the head is blocked at the rest of the packet, if the head is blocked; then, the rest of the packet is also going to stop.

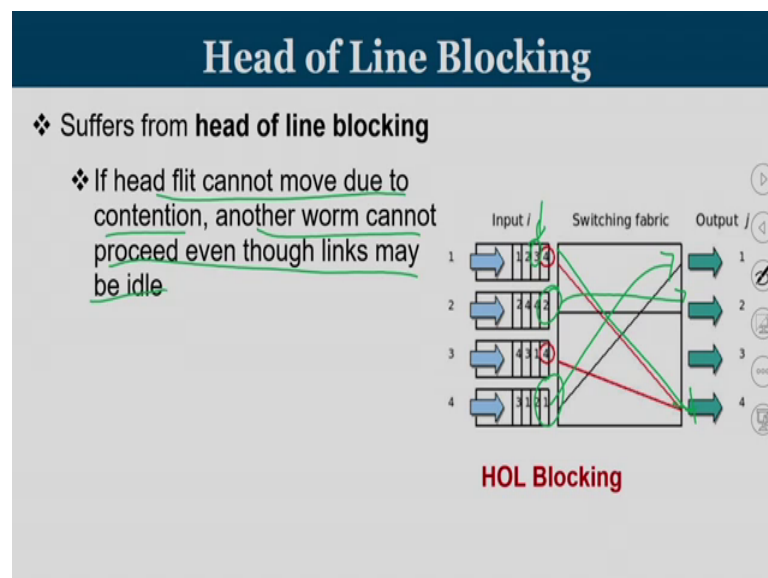
So, what happens is the routing information is available only in the head. Once the head comes the body flit and other things will slowly come. Let us try to understand this in a bit deeper. Now to the head is here and the body flits are here and the tail flit is there.

So, the flits of a same packet is catered across multiple routers. Now why this is different or what is the main difference between Wormhole routing and virtual Cut Through switching? In the case of virtual cut through switching, the buffering still maximum; but in the case of Wormhole, I can still keep a smaller buffer. It may not be completely able to accommodate the entire packet. So, when the head flit is not moving all others are also stop.

A packets only 2 flits; for example, consider the case that you have a packet with 4 flits and the buffering maybe only for 2 flits in a router. So, at any given point of time a router can accommodate at most 2 flits of a packet. So, when this is 2, it is stopped. Then, no other flit can come. Once one of the flit advances that permits, one more flit can come.

So, even without having space for the entire packet, still I can ensure that movement of flits across multiple routers is streamlined. So, this is the idea of wormhole switching and its having lower latency efficient buffer because we are not reserving the buffer for the entire packet and here also we can see that it occupies a packet occupies resources across multiple routers and tail flit whenever you get the tail flit, this tail duty is to de-allocate the resources that is been given.

(Refer Slide Time: 13:45)



Now, we have seen that store and forward routing has its own problem and slight improvement was virtual cut through routing and the third one that is been practiced in modern network on chip is the wormhole routing. Now let us try to understand that is a another important problem what this wormhole routing is having is called Head of Line Blocking.

So, consider the case now you have buffers. Let say these are the input ports where flits are residing and we have a switching fabric. Consider these are the output ports that you have and we are giving different numbers, output port number 1 2 3 and 4; different numbers are there.

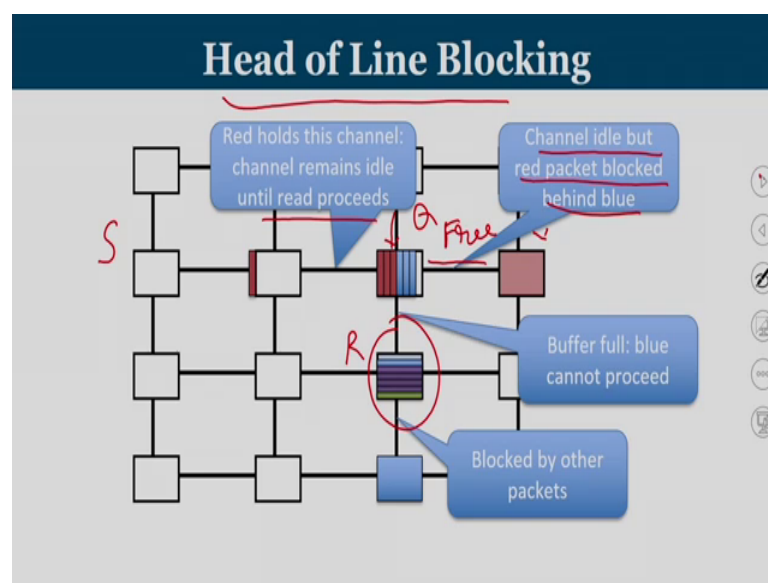
Now, the number that is written in each of the flit indicates which output port advance. So, currently we require this is the first flit; in input port 1 requires 4 of the output port; it is trying to look to 4. This requires 2; this requires also 4; this requires 1. What we can see that the flit that is taking 1 can easily move into.

So, this flit which is looking for 1 will surely get it; this 2 will surely get it there are 2 candidates for 4. So, one will get 4; if somebody gets 4, you can see that there is a flit, there is looking for 3. Had this 4 being moving back, this flit since it is waiting outside or waiting after 3 is not going to get.

So, we have a scenario where 3 is idle. Flits are there inside this Q which are looking for the third output port and you cannot grant this and this problem is called Head of Line Blocking. If a head flit cannot move due to contention; here you have a scenario where the head flit cannot move due to contention. Another worm cannot proceed even though the links may be idle.

So, here we have a scenario where one of the link is idle. There is a flit that is looking for that particular link, but because the buffers the wormhole buffers are in a Q structure, in a FIFO structure, the head flit cannot advance because of contention in its decide (Refer Time: 16:18). Everybody after the head flit cannot move. This is called Head of Line Blocking.

(Refer Slide Time: 16:24)



Let us now try to understand, what is head of line blocking. We have seen that wormhole routing is the most commonly used routing technique. Now, wormhole routing has a small problem and that is called head of line blocking.

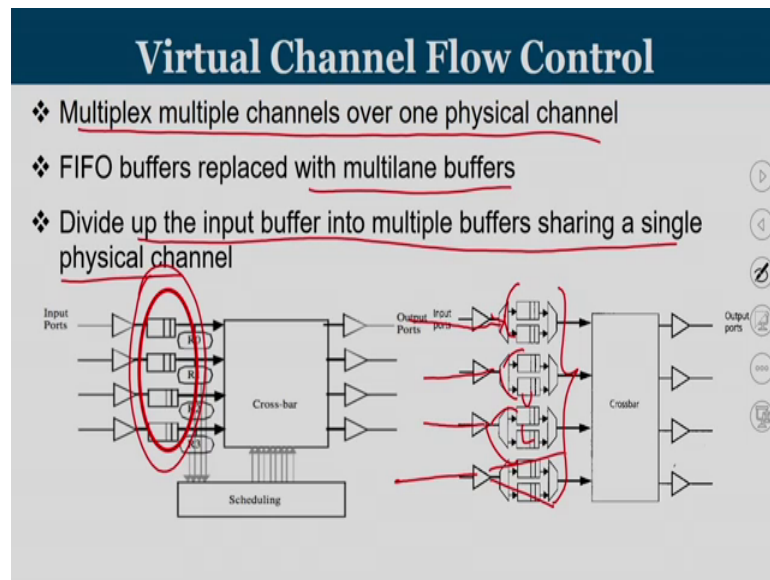
Consider the case that this particular link, what we have at the bottom end, this is now blocked; that means, no more flits can reach this router. Now, we have 2 flits that are travelling from this point S. We have a blue packet consisting of 4 flits and we have a red packet consisting of another 4 flits. The red packet is looking for this as the destination and the blue is looking for this as the destination. So, these are the 2 destinations that we are trying to work on.

Now, for the time being since because of the blocking blue packet cannot reach here, but blue can make some partial progress. Let us see how blue moves. The blue is now slightly getting advance. The blue flits reaches here. Now because of buffer that is full, here the buffer is full. So, further no more blue is allowed to reach this router R. So, blue is stopped by back pressure mechanism blue is stopped in the previous routers. Let us say Q is the previous router; in the previous router blue is been temporarily stopped.

So, the flits will reach to Q. But all the other flits that is the red flits that is following blue, they are also waiting in the buffers of Q. Now you have to see that this channel is free this is a free channel which could have been used by the red these red flits. But since the red is waiting after blue in the Q, even though the channel is idle; red packets are blocked behind the blue packets.

So, blue is in the head of the Q; blue cannot advance because of that red packets which are waiting after blue is blocked and that is called head of line blocking. So, packets will reach. So, red is holding the channel it remains idle until it is been progress. So, this is the problem of head of line blocking.

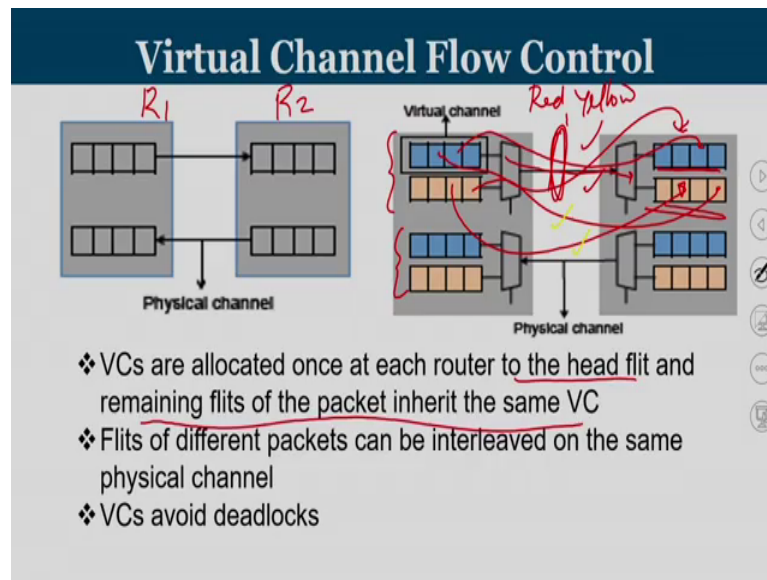
(Refer Slide Time: 19:05)



So, what is the solution for it? We are propose this is the basic problem of head of line blocking is being shown there because we have FIFO structures. So, we are going to multiplex multiple channels over one physical channel. So, FIFO buffers are replaced with multilane buffers divide up the input buffer into multiple buffer sharing a same physical channel. So, this is called a physical channel.

The packets are going to reach like this. Once it reaches a router rather than having a single FIFO, now you have different buffers available. You can see that in each of the port packets can either go to the upper buffer or it can go to the lower buffer and this concept is known as virtual channels. You have a single physical channel; a single physical channel is terminating at multiple buffers and these buffers are called virtual channel.

(Refer Slide Time: 20:03)



And the whole idea is called virtual channel flow control. So, this was our previous structure where each router has this is router R 1; this is router R 2. So, each router is having its own FIFO structure and we know this is going to create head of line blocking. The solution is rather than having a single Q structure, now you are going to have parallel Q's called Virtual Channels.

So, virtual channels are allocated once at each router to the head flit and the remaining flit that is a body flit and tail flits are going to inherit the same VC; that is why even the head flit when it moves, the other flits, the body flits and tail flits how we did know in which way the head flit has gone.

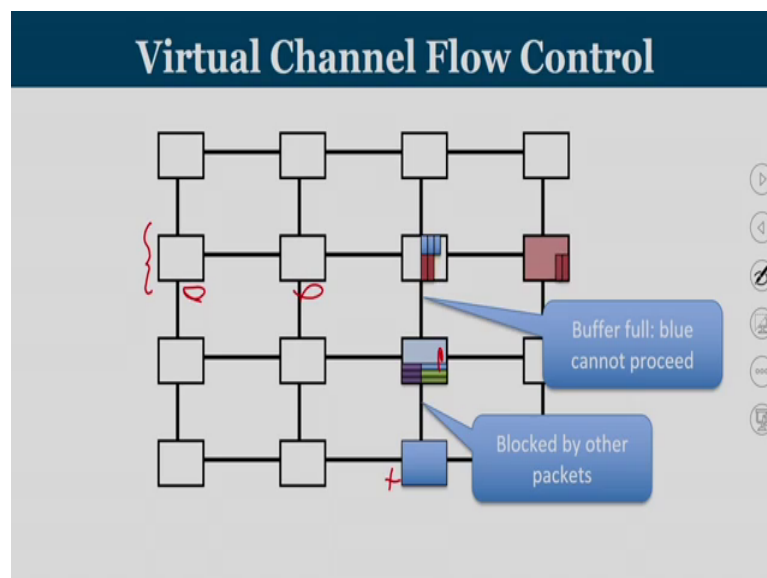
There is a process of inheriting a VC head flit when it reaches once the routing is done, its going to find out which is the neighbor. So, in that neighbor I have to get a buffer that is called flow control. Once I get the buffer; essentially in this case it is a virtual channel number,. The virtual channel number is been shared by all the body flits and tail flits.

So, even when the head flit moves advances to the next router, the body flits and tail flits are going to inherit the same virtual channel number that was used by the head flit. So, flits of different packet can be interleaved on the same physical channel. So, consider the case that in this case there is a blue one.

So, this blue can travel through this; blue may travel through this. Next cycle the second blue may travelling; third cycle it can be in yellow. So, yellow goes here. Then, it can be a blue; then, blue will go here. Then, it can be an yellow; yellow goes here. So, we know that these 2 R by the yellow flits and the others are basically the red flits.

So, if you look at a link you can see that both the red as well as yellow are travelling in an interleaved manner, but we make sure that the green sorry the blue flits will come and occupy only in this virtual channel and yellow flits will occupy only in the corresponding virtual channel. So, virtual channels also avoid deadlocks. Since, we have multiple buffers available, I am not waiting for one particular buffer; there are pool of buffers. So, that will actually break the hold and wait condition there by eliminating deadlock.

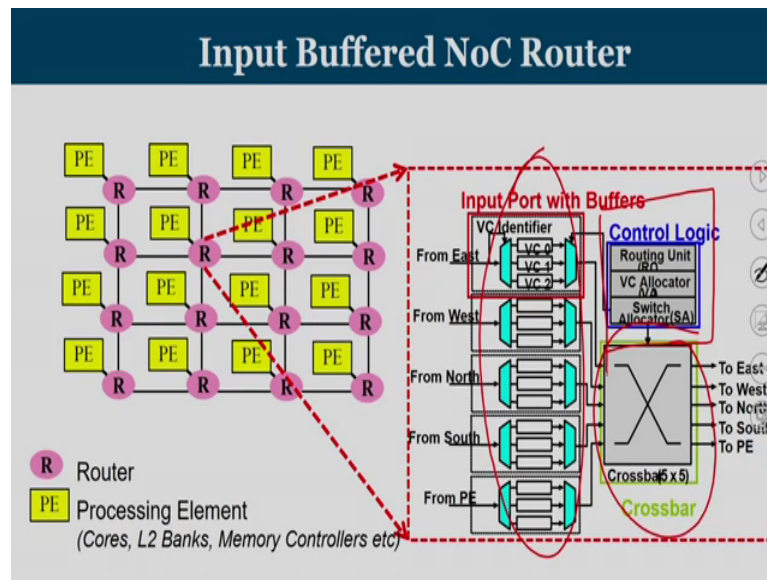
(Refer Slide Time: 22:47)



Now, we will see how virtual channel flow control happens. You can see that rather than having a single Q. Now you have 2 Q's; one which is hold held by the red flits, the other one which is held by the blue flits.

Now here as usual our previous problem, the blue wanted to reach at this point, but it is blocked here. So, blue will close reach closer to that. So, blue is going to be consumed. Now assume here it is full; but blue is held in this router. Even though now blue is held, now you try to see what happens to these packets, they will come and reside in the adjacent one and slowly they are moving into the destination.

(Refer Slide Time: 23:28)

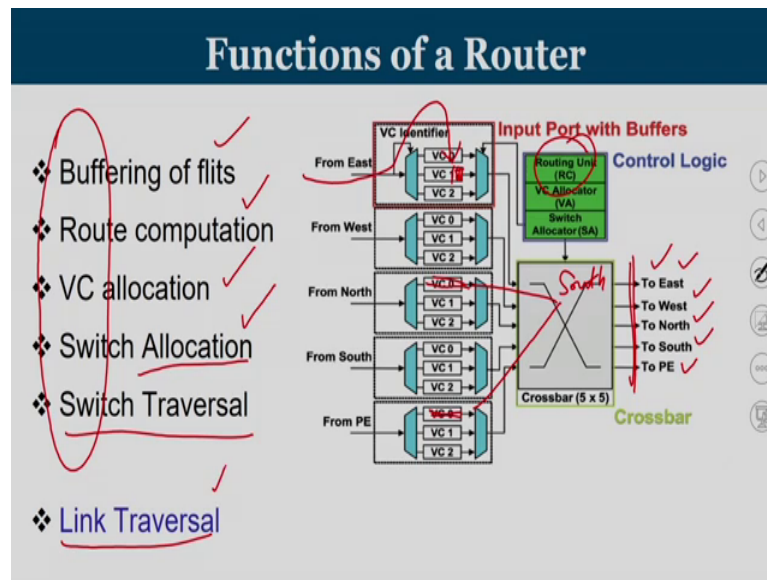


So, we can see that the red reach destination, even though blue is blocked here, there is no head of line blocking. So, by virtue of the parallel tracks called Virtual Channels that is available, even though one of the packet which reached this router early, that is a blue packet. Blue is blocked because I cannot make a forward progress here, due to back pressure, but the red which was after blue can make forward progress to its destination.

So, we were trying to understand how flow control is happening and this is your tail chip multi-core processors; in the last two lectures, we were trying to see what was routing; what is topology and what is flow control. And this is how the input side of the routers look like; you have buffers in the input and they are known as Virtual Channels.

And then, there is a cross bar the crossbar is going to connect the input to the corresponding output and this input output is been facilitated by a control logic. So, we have buffers in the input; we have a crossbar that connects the input to the output and we have a control logic that facilitate the smooth flow of packets from the input side to output side. So, essentially the NOC router consists of this many components.

(Refer Slide Time: 24:50)



Now, we will try to understand, what are the functions of a router. So, the first and foremost function is Buffering of a flit. Whenever a flit is coming through a channel, the flit occupies a buffer. The second task is Route computation for a flit that is residing inside a buffer, the route computation unit will find out which is going to be the output port that is to be assigned.

So, the process of finding an output port for an incoming packet is called Routing or Route computation and route computation is done for the head flit and the body flits and tail flit will follow the same route as been assigned to the head flit.

The third task is called Virtual Channel allocation. The process of reserving a buffer in the downstream router is called Virtual Channel allocation. We know that to ensure flow control, it is based upon handshaking between adjacent routers. So, one router has to tell the adjacent router, I have buffer available you can send me a packet. So, sitting in current router, a packet that already got I am going through north output port.

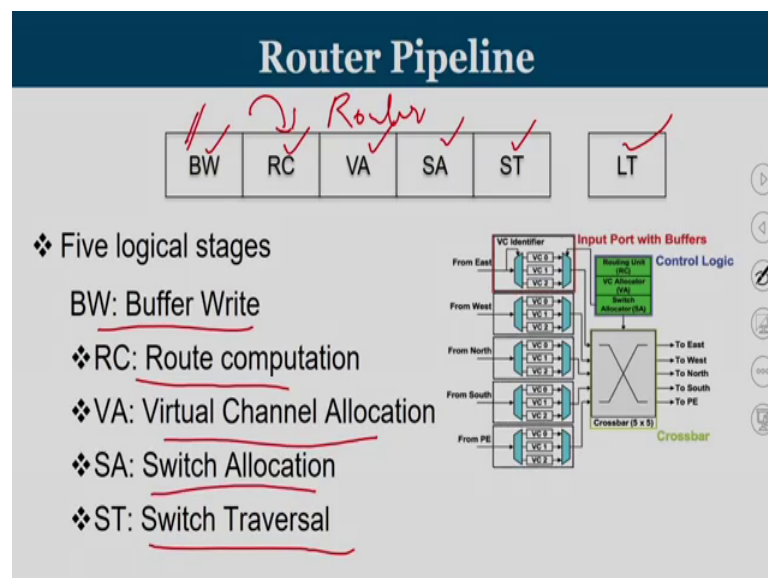
So, contact my north neighbor or get an update from my north neighbor and reserve a buffer in the north neighbor. I will repeat once again the process of reserving a buffer in the downstream router is known as Virtual Channel allocation.

Next is called the Switch Allocation. Whenever I have multiple let us say have a flit here and another flit here; let us say it is growth or going to look for the same output port let

us say south. So, when multiple flits are competing for the same output port which of the flit has to be chosen. It is an arbitration process and that is known as Switch Allocation and once arbitration is over the flits are going to travel through the switch. So, how switch traversal.

So, packets will travel through the switch at any given clock cycle; at most 5 flits can travel through the switch one going to east output port; one from west; one from north; one from south and one can do the processing element. So, based upon switch allocation, switch traversal takes place and then we have the link traversal. So, this many operations happen inside the router and once switch traversal happens switch is connected to the link. So, you have the links here. These are the links and it is called the link Traversal.

(Refer Slide Time: 27:41)



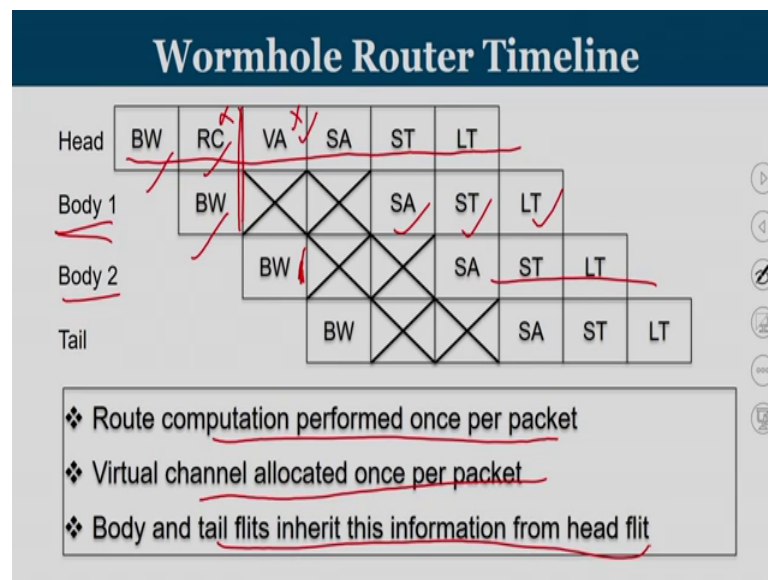
And the router is pipelined. So, whatever function we have seen, buffer write followed by route computation, followed by virtual channel allocation, switch allocation, switch traversal; this much happens inside the router pipeline and then you have the link traversal. So, this is the traditional router that we have seen. So, Buffer writing is there 5 logical stages; then, you have Route computation, Virtual Channel allocation, Switch Allocation and Switch Traversal that is going to happen.

Now, traditionally since it is pipeline. So, whenever I am performing route computation for one of the flit, I can perform buffer writing for other flit. Similar to the instruction pipeline that we have learnt, here also we could do it in a pipelined manner. So, a router

may multiple stages are there; whenever I am working on a couple of flits in, the switch allocation stage parallelly; the other set of flits maybe in the route computation; some other set of flits maybe in buffer writing.

This shows that you require close to 5 cycles to complete its operation for a packet inside a router 5 is little to on the higher side. Can we optimize it? Some of these units can be merged together that is what we are going to see.

(Refer Slide Time: 28:59)



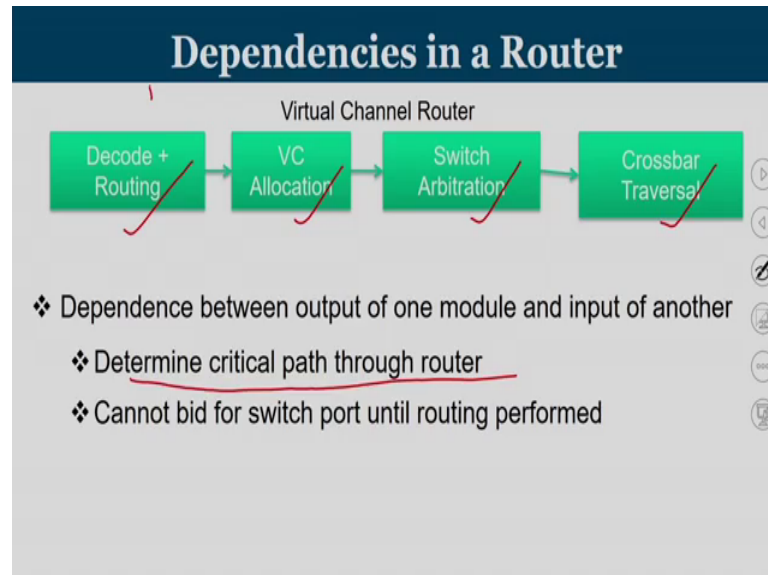
First, we try to understand what is wormhole routing timeline? So, buffer writing, route computation and also when you look at this point, I am performing route computation for the first flit; the same time buffer writing for the second flit. When I am performing virtual channel allocation for the first flit, I can perform buffer writing for the next one.

So, when you have head flits and body flits we can see that. Head flit has route computation, virtual channel allocation, switch allocation, switch traversal and link traversal. But for body flits, we would not perform route computation and virtual channel allocation. So, it has only buffer writing, switch allocation, switch traversal and link traversal. Second body flit also follows the same thing.

So, route computation and virtual channel allocation is done only for the head flits. The body flits and tail flits will only inherit the route and the virtual channel, that is allocated to the head flit. Route competition is performed per packet only once. Virtual channel is

allocated only per packet and body flits and tail flits inherit this information from the head flit to make the forward progress.

(Refer Slide Time: 30:04)



We will try to see, what are the dependencies.

Now, what do you mean by dependency? I can do a task only after some other task is over. We can see that routing should be done to perform virtual channel allocation. Only virtual channel allocation is done, then only I can perform switch arbitration and then only I can travel through the crossbar. So, dependencies between output of one module and input of another module.

So, how are you going to design? You need to have one unit, only if the task in that unit is over, then only I can move to the next one. So, this determines the critical path of the router. How much time you need for routing plus how much time you need for virtual channel allocation plus switch allocation plus the crossbar traversal the whole thing combines together (Refer Time: 30:50) way that time that a packet takes inside a router.

Now, we have something call look ahead routing.

(Refer Slide Time: 31:02)

## Lookahead Routing

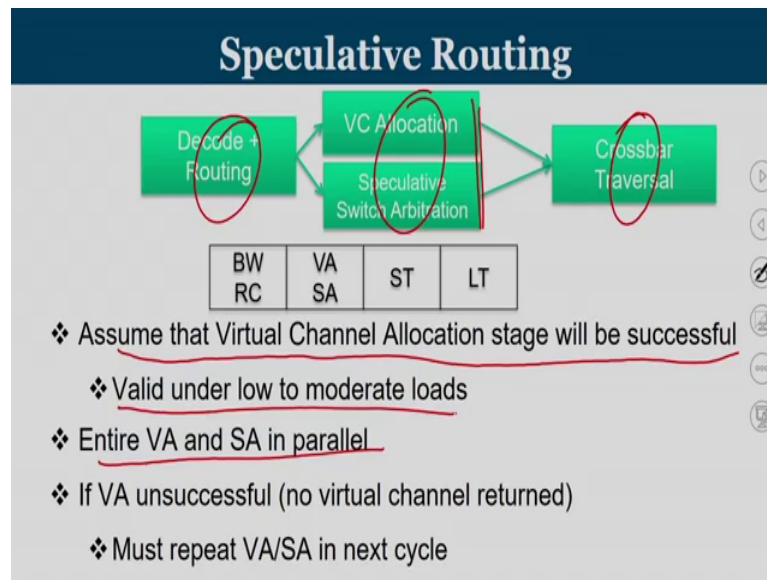
- ❖ At current router perform routing computation for next router
- ❖ Overlap with BW

BW	VA	SA	ST	LT
RC				
- ❖ Pre-computing route allows flits to compete for VCs immediately after BW
- ❖ RC decodes route header
- ❖ Routing computation needed at next hop
  - ❖ Can be computed in parallel with VA

At current router, we perform the route computation for the next router. So, since we already know I am going to travel to the next router and aspect topology that particular router after reaching the particular router, what is going to be my next neighbor; can I perform that that is called route computation. So, Pre-computing a route allow flit to compete for VC's immediately after buffer write. So, once you reach the adjacent router, buffer writing is been done. Immediately because I know what is a route, because a route is already pre-computed in the previous router.

So, my route is north. I wanted to take north. So, virtual channel allocation the north neighbor we will happen. So, routing computation needed at the very next hop can be computed parallel with virtual channel. So, whenever I am allocating virtual channel, I may compute the route and this is the way how it is being done. These are the ways by which a 5 cycled initial router was now cut down into 4 cycles. This is the way how you optimize the router pipeline.

(Refer Slide Time: 32:02)



Let us see one more level of optimization, it is called Speculative Routing. Virtual channel allocation and switch allocation, speculative switch allocation can happen parallelly. So, routing and decoding stage 1, stage 2 is VC allocation and Switch Allocation and stage 3 is called Crossbar Traversal. So, what we do is we will assume that virtual channel allocation stage will be successful.

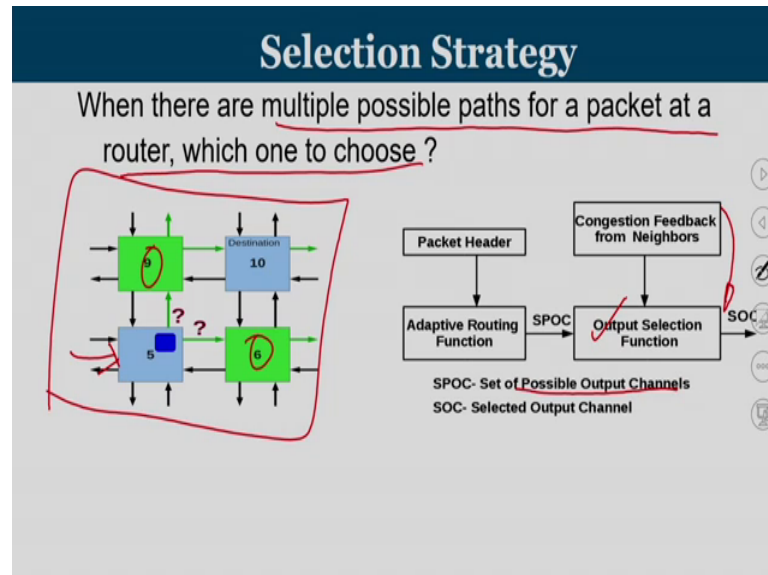
How it is possible? What is virtual channel allocation? The process of reserving a buffer in the downstream router. Yes, if that is success, then I compete with other flits also who got buffer in the downstream that is called Switch allocation. So, this is happening one after another.

We could do some optimization here and that is being done by the process of Speculation. Speculation means I assume that virtual channel allocation will happen if so I am going to perform the switch allocation. So, the entire virtual channel allocation and switch allocation is done in parallel. So, when the speculation is going to be successful, it is normally valid only under low to moderate load. But there can be cases where virtual channel allocation may not happen.

So, in that case you have to repeat the cycle. So, the whole idea of speculative switch allocation is we assume that virtual channel allocation is be successful. So, I should not wait for the process to get over with the hope that it will be successful, I perform switch

allocation and that is called Speculative Routing. Now, we will try to learn something about the Selection Strategies.

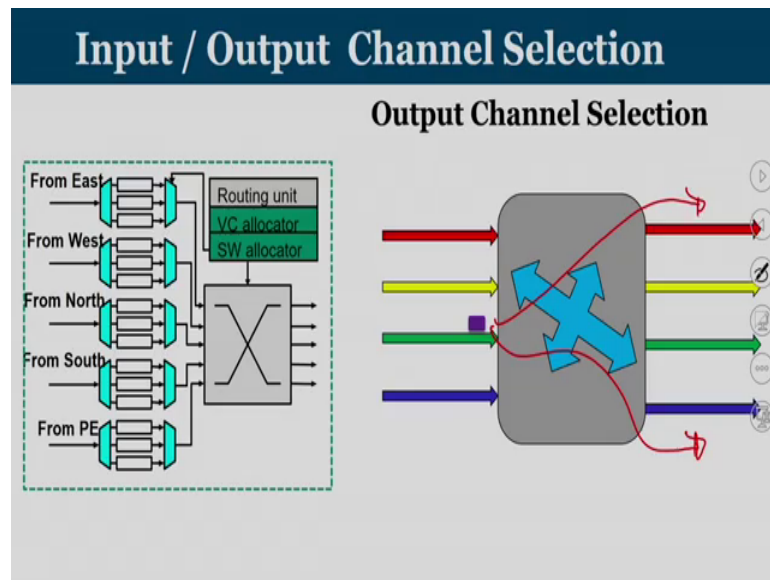
(Refer Slide Time: 33:41)



Consider the case where inside a mesh, you are now currently your focus is on router number 5 and destination is 10. I could travel through either through 9 or through 6. So, this is a small segment of a 4 by 4 mesh NOC. Your routers are from 0 to 15 and I am taking a small segment of it. Let us say there is a packet at 5 wanted to go to 10. It can either travel through 9 or it can travel through 6.

So, when there are multiple possible path for a packet in a router which one to choose and that is what is known as Selection Strategy. So, your adaptive route function sometimes will return like in this case it is returning 6 as well as 9. The adaptive routing function will return a set of possible channels and we collect feedback from neighbors. So, congestion feedback is collected from neighbors and based upon that one of the output is been chosen. This is called output Selection Strategy.

(Refer Slide Time: 34:46)



So, input and output selection that is a 2 different mechanisms by which you make a router adaptive. So, adaptivity is the process by which from many I am choosing one. So, I am trying to introduce you to 2 concepts; the first concept is called Input Channel Selection, second concept is called Output Channel Selection. Let us try to understand what is Input Channel Selection.

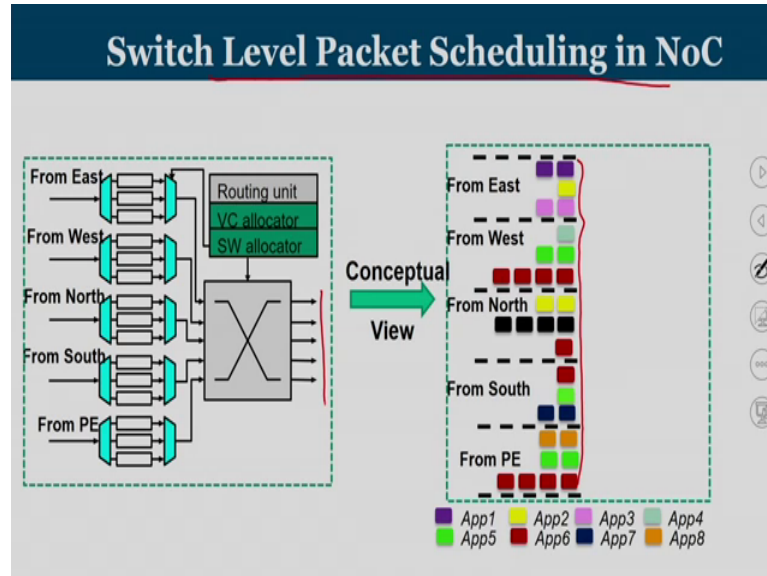
Consider the case you have 3 flits that is shown in green color; 3 flits are reaching this router. Let us say you can assume these are north, east, west and south input ports of a router. You are getting 3 flits. Now, all the 3 wanted to travel through this output port that is why it is been shown as green color. This 3 flits wanted to go in that particular direction. So, flits coming from north, east as well as south wanted to go to west at the same cycle. So, which one to pick and that process is called Input Channel Selection.

Now, we have to see another scenario, where a different set of packets are going to come with a different problem that is called Output Channel Selection. So, we have 1 flit that is coming and the peculiarity of this is this flit can be either router through this output port or it can be router through this output port. Whenever the destination is having a different row number and column number than the current router, then from the current router it can have sometimes more than 1 output ports.

We have seen that sometimes in west first routing, in north last routing, in odd even routing, in certain routers, certain packets can have more than 1 output ports. So, when

you have 1 packet with more than possible more than 1 possible output ports which one to choose that is called Output Channel Selection.

(Refer Slide Time: 36:50)

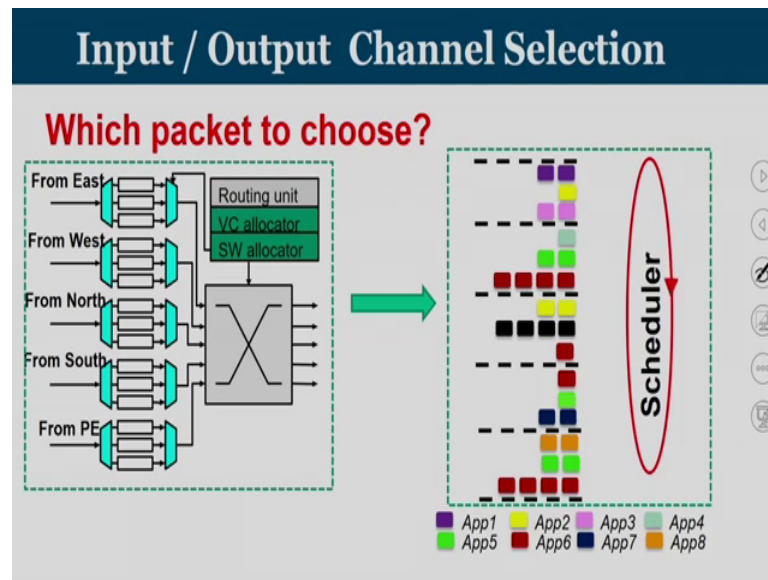


So, we will focus something on switch level packet scheduling; it is also known as Input Channel Selection. It is a conceptual view of a router. These are the virtual channels. Let us say you have different applications that we are there and we can see that one virtual channel contains only one color; that means they are flits from the same packet.

So, this is the instantaneous snapshot of the input buffer of a router. We have flits of the same packet here that is being shown by the same color. These are all 4 flits of the same packet; 4 flits of the same packet. 2 flits of the same packet like that we can see that the entire channels are the entire virtual channels with respect to one router are full.

Now, if you look at the head of all these channels, you have many flits and you have only maximum 5 outputs. So, these many flits are competing for 5 outputs.

(Refer Slide Time: 37:53)

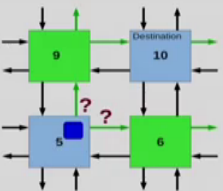


So, which is the one that you are going to choose and that is what is known as Switch Scheduling, which packet to choose from among many. It is also known as Input Channel Selection.

Switch scheduling plays an important role like which packet will make a forward progress; some packages may be very critical packets; some packet may not be critical; some packets will be very old packets. So, we are using different types of switch scheduling or input channel selection in order to make sure that which of this packets has to move further.

(Refer Slide Time: 38:26)

### Why selection strategy important ?



- ❖ Cache miss / coherence packets need to be serviced fast.
- ❖ To service them fast we need to reduce latency of packets.
- ❖ Congestion increases average latency of packets.
- ❖ Selection strategy chooses paths with less congestion.
- ❖ A good selection strategy reduces average packet latency.

So, why this selection strategy is very important? So, what is the source of NOC packets? It is typically is Cache misses or coherence packets. So, whenever there is a Cache miss, whenever there is a coherence update that has to be given in a tiled chip multi-core system, they are going to create NOC packets. Now, these packets have to be serviced very fast to service these packet very fast, essentially we how to reduce the latency of those packets.

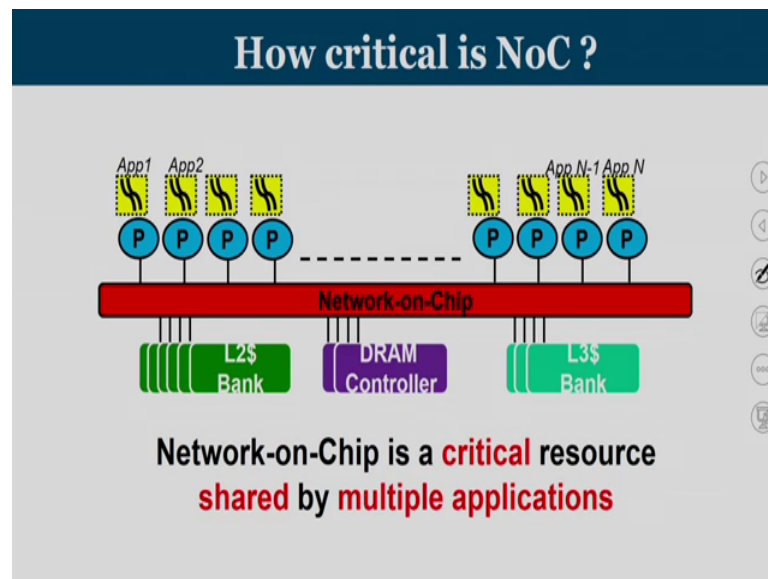
But whenever there is congestion in the network, congestion is going to increase the average packet latency of the packets. So, if you use a good selection strategy that is going to pick the best path. So, good selection strategy chooses the path with less condition and a good selection strategy reduce the average packet latency.

So, in short, we are going to work on an NOC which is been shared by multiple processors and there are applications that is running on these processors. These applications will work from the L 1 Cache. Whenever you are not able to get the instruction or data from the L 1 Cache; it is going to incur an L 1 miss and based upon the address mapping that is where we have familiarize in the last lecture.

This misses will be triggered as NOC packets to various other course. These packets are going to travel through the network and we how to assume that multiple such packets are going to enter the network from different-different nodes which are belonging to different-different applications.

These packets are going to interfere each other, meet each other, compete each other and content each other at different-different routers. Now, the router when they get packet, they have to take a call, which one to be prioritized. So, working on NOC routers, what are the kind of scheduling algorithms; what are the kind of adaptivity that you have to bring in? It is very very important.

(Refer Slide Time: 40:22)



No, how critical is NOC? We have different applications; that is running on processors and it also your network on chip is also connected to L 2 Cache. DRAM controllers and L 3 Cache if any; so, NOC is highly critical resource. So, we have now completely covered up the background of network on chip with routing, with topology, flow control and router micro architecture we have seen a how router pipeline works.

Now, we will see the big picture in the next couple of lectures; we will see the big picture of how things can be put together in a TCMP Architecture in a multi-core architecture, where we will see what is the role that network on chip is going to play in improving the performance of a tiled chip multi-core system.

So, with that we complete today's lecture. We are putting some small tutorial exercises also. I request you to familiarize with them so that you get an easy grip over it.

Thank you.