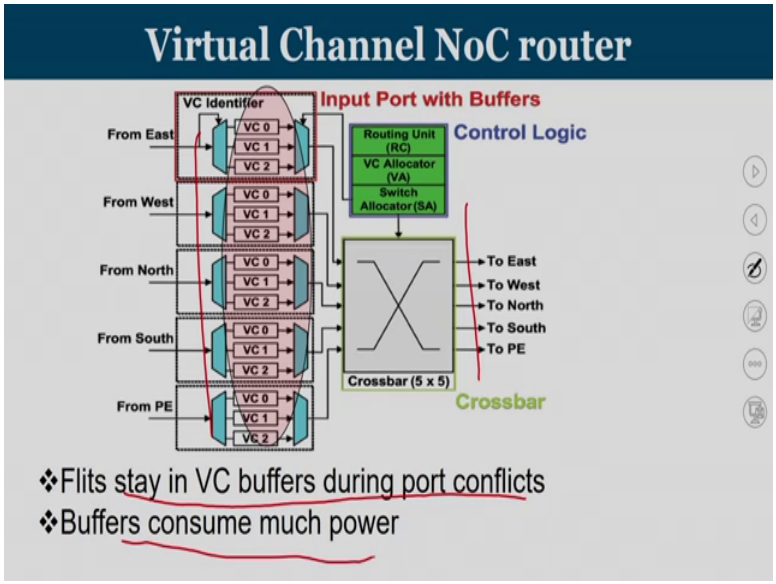
Multi-core Computer Architecture-Storage and Interconnects
Dr.John Jose
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Lecture - 15
Energy Efficient Bufferless NoC Routers

Welcome to the 15th lecture. Today, we are going to see a special category of a NoC Router Micro Architectures. So, far we have seen NoC routers with input buffers. And, buffers are going to consume power. Our today's discussion is can we have an alternative router design where we could eliminate buffers.

So, next 2 lectures are basically on design of energy efficient NoC routers. This lecture it is on Bufferless NoC Routers and the next lecture is on minimally buffered NoC routers.

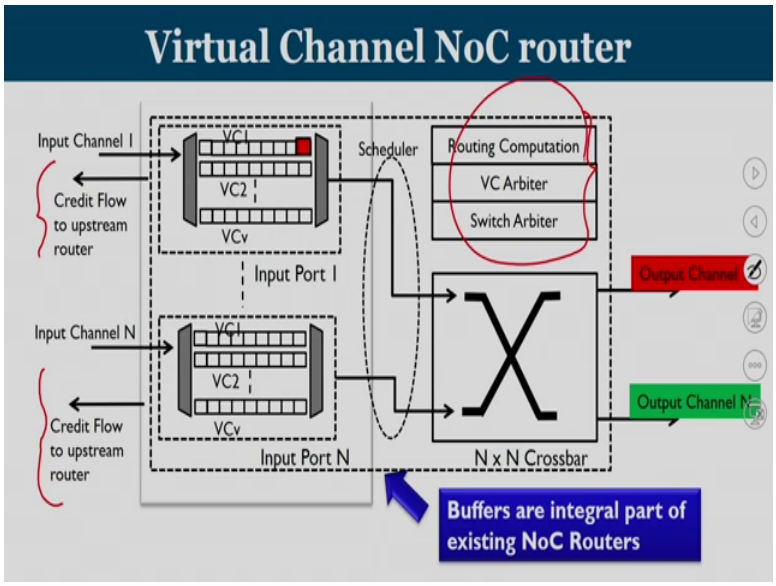
(Refer Slide Time: 01:16)



We know that this is the general structure of an NoC router, where we are going to have buffers here, which we call it as virtual channels and we have a crossbar through which it goes.

So, flits stay in virtual channel buffers, whenever there are port conflict and this buffers are going to consume much of your power. So, these buffers are buffer hungry, buffers are power hungry circuits.

(Refer Slide Time: 01:50)

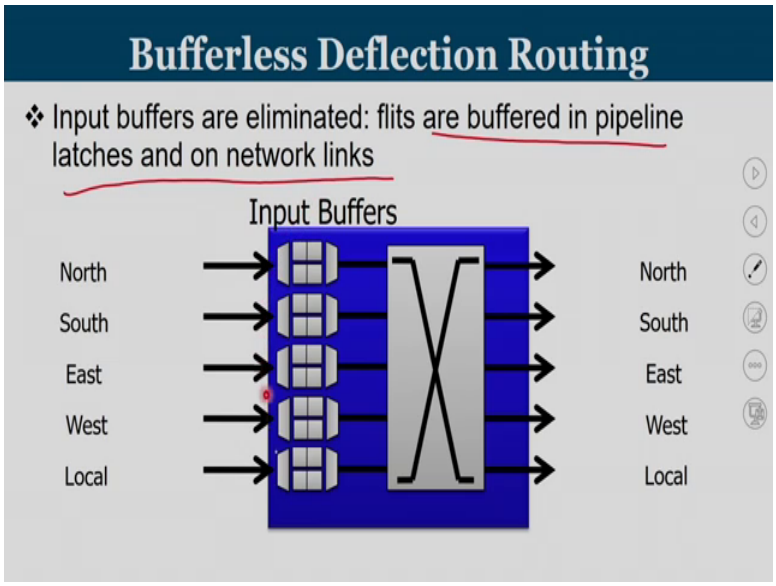


If, you look at from a different perspective, from the buffers the credit flows to the upstream.

So, credit basically means number of free buffers inside your virtual channels. And, once you pass the credit to upstream, your packets are going to come and reside inside the buffers. And, then these packets will move to the corresponding output channel based upon the control logic that is implemented in the router.

So, buffers are an integral part of existing NoC routers, because buffers are going to hold your packets.

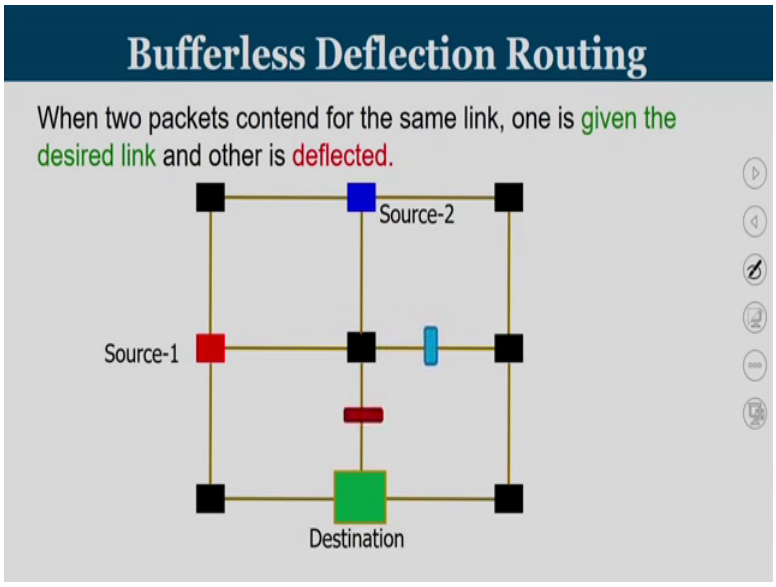
(Refer Slide Time: 02:34)



Now, Bufferless Deflection Routing is a concept where input buffers are eliminated. So, how is it possible? Flits are buffered in pipeline latches and network links. This is the conventional structure where we have we can see that buffers are there, you can see that buffers are there and these are called the virtual channels.

Now, with bufferless design we are going to eliminate the input buffers and going to put latches and which is what is known as deflection routing logic.

(Refer Slide Time: 03:11)



Now, how the deflection routing logic is basically going to work? When you have 2 packets, they are contending for the same link; one is given the desired link and other packet is deflected. Consider the diagram where we could see that there are 2 sources; one is shown by the red color node, other one is shown by the blue color node, both the red and the blue has the destination, which is marked as green.

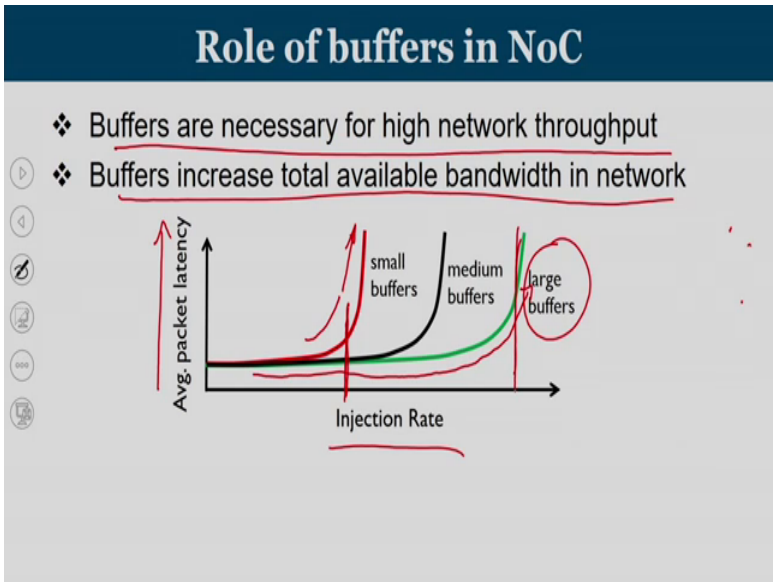
So, when you apply x y routing, our conventional routing algorithm. We can see that, the packet from the red is going to travel like this and packet from blue also is going to travel like this. And at this junction point they both are competing for the same output, the same output is south. When you have a scenario like this, then if buffers are not there out of the 2 flits; one flit is going to get the desired port, the other one will not get.

In conventional NoC routers, the flit that did not get the desired output port has to wait there in the buffer until you get the desired output port probably in the subsequent cycles. Let us see what happens here, let us say there are 2 packets the red packet and the blue packet both are travelling together. At one particular clock cycle they both reach the central router from which by applying some priority mechanism let us say red was selected as the winner.

So, red will get the productive port, eventually blue is deflected away and because of the peculiar structure blue will also reach the destination. So, one of the packet will get productive port and the other packet has to be deflected away. Deflection means we are forwarding the packet in a non-productive direction.

Since, it is a mesh topology even though you are going away from your destination maybe at the adjacent routers are the next routers you can still come back in the productive part. So, the idea of bufferless deflection router is when 2 packets are competing for the same link; one of the packet is given the desired link and the second packet is deflected away.

(Refer Slide Time: 05:39)



What is the role of buffers in NoC? So, we know that buffers are really necessary for high network throughput, because this your buffers are going to hold your packets. So, when you have more number of buffers that is available inside a network we can accommodate more packets in the network. More the number of buffers more will be the throughput of the network, but buffers are going to increase the total available bandwidth in the network.

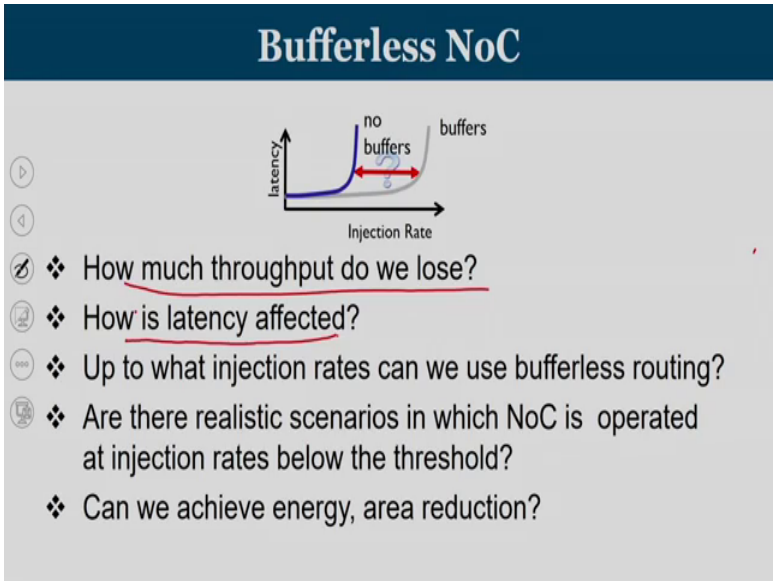
Now, you could see that this is a graph where the x axis is injection rate; injection rate means the number of packets that are entering into the network per cycle. And the y axis is called average packet latency. When you increase the injection rate, naturally the number of packets that are going to enter into the router is going to increase, when you increase the injection rate naturally the load will increase packets will compete each other. So, beyond a point you can see that the latency also increases. When you have larger number of buffers that is been pointed out by the green chart, even if you increase packet injection rate, we are not going to see any increase in the latency.

But, after some time you could see that latency is also going to increase, but look at the case if you do not have any buffers, then you are going to saturate early, the point where latency is going into an exponential decrease or a sudden increase is known as injection rate. So, we can see that whenever we have very less number of buffers the network is

saturating early, when you have more number of buffers the network is going to saturate late; that means, network can accommodate more number of packets.

So, buffers are going to play a significant role in the performance of an NoC. More number of buffers, more packets can be accommodated; more packets can make forward progress, when you do not how buffer network will get saturated early?

(Refer Slide Time: 07:47)

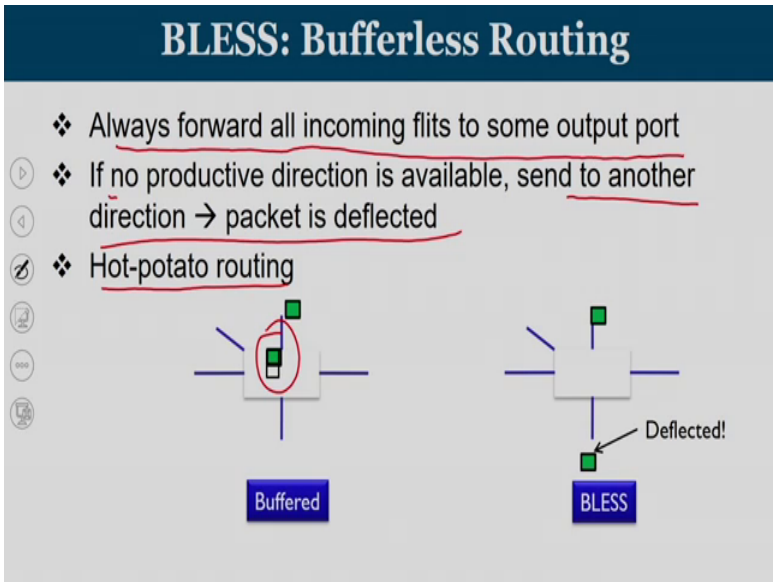


Now, this is the diagram which we have seen so, far where if you have buffers you can see that this get stretched little bit. Now, we will try to understand, how much throughput we will lose, if you are going for a buffered design versus a bufferless design? How much throughput we will lose, if you are shifting to bufferless? How much is the latency affected is there is a drastic decrease in the latency or up to what injection rates can we use bufferless routing.

Are there realistic scenarios in which NoC is operated at injection rates below the threshold? So, we wanted to know in a real time application, what are going to be the injection rate. Are, they suitable for bufferless or can still were we have to go for buffer to design. And, once you get rid of buffers we are going to have saving in terms of energy area, how much area or energy savings that we get?

So, these are the typical design questions that we have when we talk about bufferless network on chip.

(Refer Slide Time: 09:02)

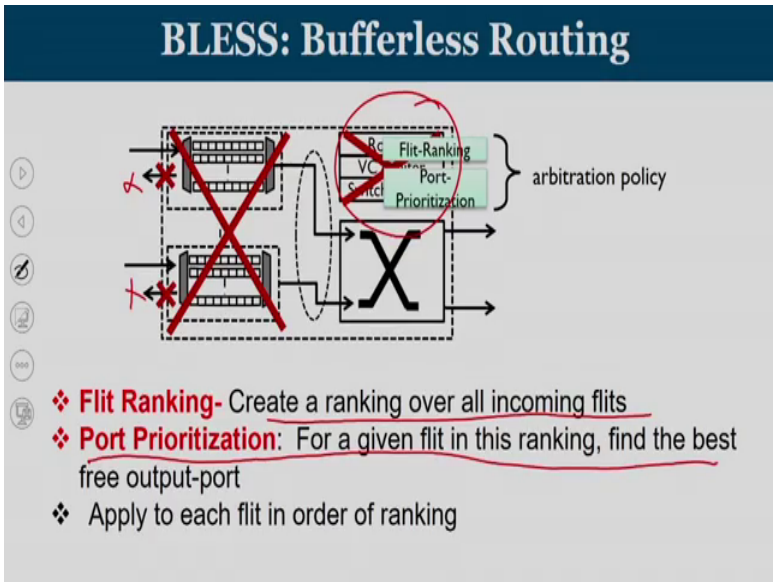


Let us try to understand bufferless routing from a different perspective. So, what happens always forward an incoming flit to some outgoing port that is the idea. Whatever flits you are going to receive these, flits have to be forwarded to some output port. If no productive direction is available then send that flit to some other direction that is what the flit is called deflected.

And, this concept is also known as Hot-potato routing. On the left side of the slide, you can see that there is a buffered router where I can accommodate flits and you can see various direction, consider the case that you are going to have 2 flits reaching the router exactly at the same time.

Now, when 2 flits are reaching one of them will get a productive port. Let us say we wanted to go to north direction. So, that moves to the north the other one is buffered here. So, the flit is buffered, that is a advantage of buffered. Now, this is what it is happens? Look on the right hand side, where we do not have buffers. Now, 2 flits are going to come. One will get the north port, since there is no buffer to accommodate the second flit it is getting deflected away. So, the flit is going to get deflected, that is the basic difference. So, as long as you have buffer those flits which are not getting productive port are going to be deflected away.

(Refer Slide Time: 10:31)

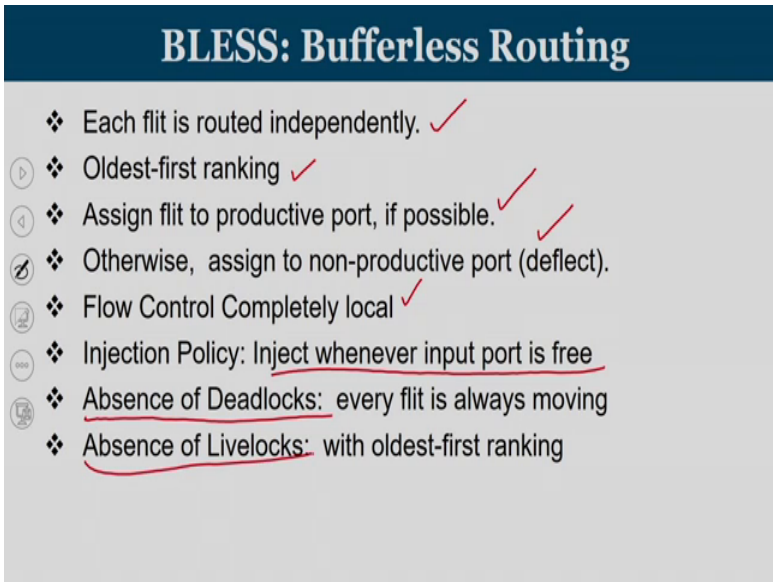


Now, how this concept is going to work? This is your traditional input buffer virtual channel router, we are trying to get rid of the virtual channels, and if the virtual channel is not there, the credit outflow is also not required and we can get rid of this control logic the traditional one and that has to be replaced with a flit ranking policy and a port allocation policy. So, what do you mean by flit ranking? We have to create a ranking over all the incoming flits. Whatever flits we are going to receive inside a router, we have to order them. This is the highest rank flit, this one is the next highest ranked flit like that we are going to order the entire flits in some order.

Now, after the ordering the next thing is called port prioritization for a given flit of a specific rank. We are going to find out what is the best free output port. So, there are 2 stages first is called the flit ranking, rank the flits based upon the order. If from among the flit that is going to have the highest rank pick that flit and try to assign a port to that flit based upon it is preference.

So, it is a sequential ranking order you have to apply this for each of the flits. So, we are trying to get rid of buffers, we are trying to get rid of the credit system that tells from one router to another how many buffers are available that credit system is also eliminated there is no need for virtual channel allocation and switch allocation. And, what we need is we have to rank the flits based on the ranking we have to pick each flit from the best ranking onwards and try to assign them ports.

(Refer Slide Time: 12:18)



In bufferless routing each flit is routed independently, because we do not have any guarantee that the flits are going the flits of a same packet may get the productive port at the same time. So, flits are independently routed and we have to take care of all the design issues and implementation issues, in making sure that the flits are reaching the destination.

Second one is oldest flit the ranking priority is based on the oldest flit assign flit to productive port if possible. So, wherever possible once you are going to assign the port we have to give a port that is productive to the flit. And, the next one if you cannot get a productive port, then also we are going to give the flit a port and that is what is called the deflection port. And, flow control is completely local.

So, we are not going to look at adjacent routers, whether do we have buffer available because since the whole router is going to work without routers there is no need to have a handshake mechanism between the adjacent router. So, flow control is completely a local thing. Now, when you inject we have to make sure that there exists. So, injection is possible only when one of the input port is free.

So, when you have a router with the 4 neighboring directions connected and there is a local flit that is going to be injected. And, the locally injecting flit, that injection is possible because for him anyway it should get one of the and 4 neighboring direction. So, when you have a scenario where there are 4 flits coming from 4 of it is neighbors,

and then you have a local flit that is to be injected then it is not possible, because the local injected flit has to get one among the orthogonal directions north, south, east or west.

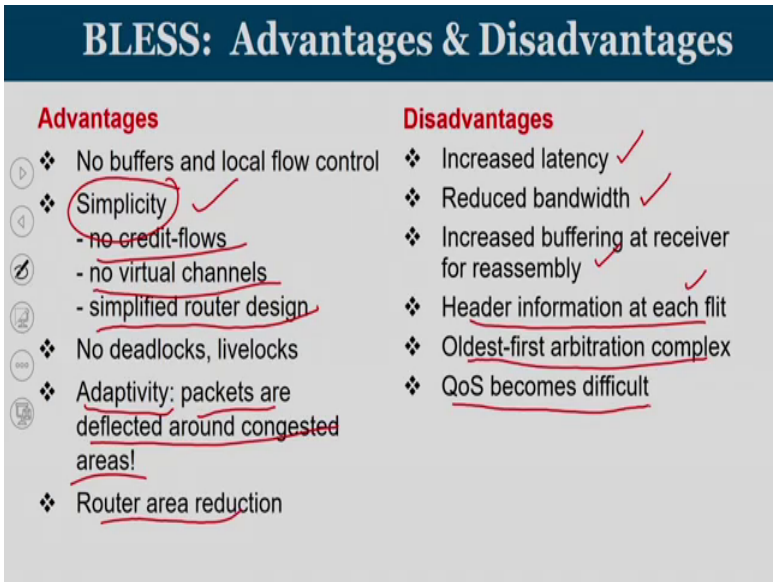
So, this local injection is possible only if one of the input channel is idle. Now, this make sure that in the case of a bufferless router known as bless Deadlocks are not there, because every flit is moving there is no dependency on any buffers I am not looking for a buffer to be available and we have absence of Livelocks also.

So, how will you make sure that livelock is not happening? We are always prioritizing one flit and that is the oldest to flit. So, once you have the oldest flit if the oldest flit is moving or it is making progress. At some point of time every flit will become the oldest in the network and once you are oldest in the network you are no longer deflected.

So, I wanted to just emphasize on 2 aspect of bufferless routing. First one is it would not create any deadlock. Deadlock is happening because there exists a cyclic dependency for resources. One router has to wait for other router to get a buffer. Since, the concept of buffer itself is totally eliminated, in the case of bufferless reflection routing, a router, a packet inside a router is not waiting for availability of a buffer in the downstream router. So, there is no deadlock.

And, livelock is eliminated by virtue of oldest flit highest priority routing. Oldest flit will always get the productive port and since everybody will become the oldest flit in the system at some point of time. And, once you become the oldest flit, then there are no more deflections at all.

(Refer Slide Time: 15:54)



What are the advantages and disadvantages of bless, were blessed and so, bufferless routing no buffers and it is strictly local flow control, so no kind of handshake mechanism is required, that is no credit flow, no virtual, channel designs, and the router is fairly simple.

So, this make sure that it is a simple router design no deadlocks and livelocks what we have seen and we have a bit of adaptivity because packets are deflected around congested areas. So, when you see that certain port is congested the flits are not going to move in that, because of the deflection mechanism I could bend around the point of deflection. And that gives you a bit of adaptivity and surely since we are getting rid of buffers routers area is being reduced. Nothing, we are going to gain without sacrificing something.

So, let us try to see what are the disadvantages. One is surely the packets are going to have a slightly higher average packet latency. This is because of the fact that sometimes packet may how to get deflected. So, they will take more number of hops to reach the destination.

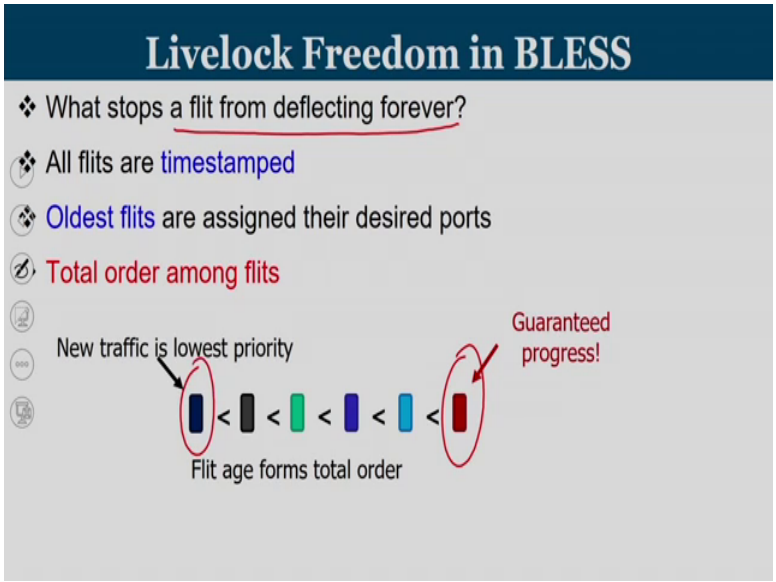
So, then the latency of the packet is going to increase. And, second one is a reduced bandwidth, since you do not have buffers number of packets that can be accommodated in the network is going to be less, that will reduce the bandwidth. And, increase the buffering at the receiver for reassembly. Since, different flit inside a same packet are

independently router, we need to have some kind of routing information, basic routing information available in each of this flit.

So, the flits are going to reach the receiver in out of order. So, some kind of a reassembly mechanism is needed in the receiver such that I can regroup with the packet together. And, you have how to find out the header information at each flit, that is what I mentioned since, each of the flit is independently router there should be a mechanism, which will help us to ensure that some kind of basic information is available regarding routing.

That means basic header information should be available in each of the flit. And, this whole setup works with finding out who is oldest flit in the system and he is being given guaranteed progress in the productive port, but finding out the oldest flit is going to be a bit of complex, because in it towards a sorting operation. We will see that and quality of service become difficult, because a lot of packets will get deflected sometimes you may be a very critical packet, but you are the oldest flit in the system, you may not be an older flit in the system then it gets deflected.

(Refer Slide Time: 18:34)



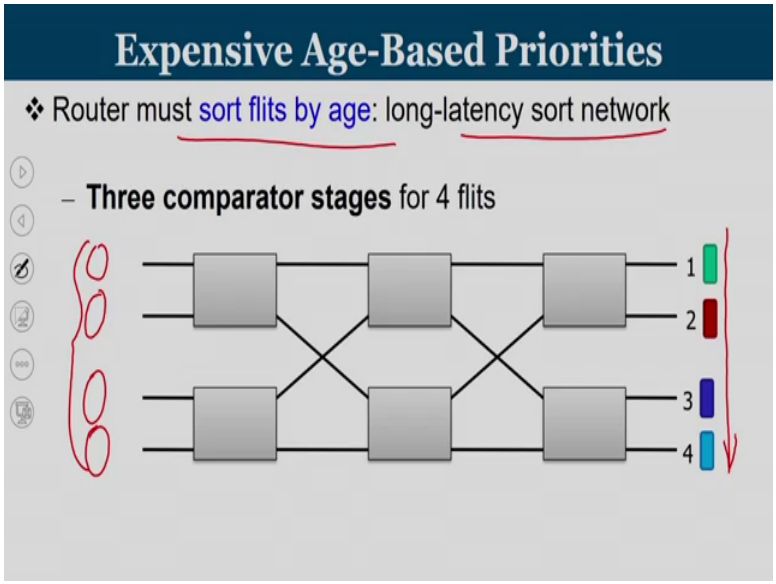
Now, livelock freedom in bless how are you going to guarantee that? So, one stop a flit from deflecting forever, so we when we are moving around the system forever, that is what is called it is livelock.

So, how can you ensure that if you get rid of buffers that is going to save from livelock issue also. All flits has to be time stamped. So, there need to be a total ordering older flits are assigned to the directive ports and total order among the flits. So, let us say these are the flits that is there. So, the oldest flit in the system he will surely get guaranteed progress. And, whoever is newly injected into the system they may be having the lowest priority. As time progresses each of the newly created flits will become the oldest one in the system and then onwards it is going to have guaranteed progress.

But, what is going to be the cost associated with this operation that is what we are going to see. So, the idea of livelock freedom is done with there should be a total ordering of the flits based upon their age and the oldest flit is going to get productive port always.

Now, in order to sort 4 flits, so, let us say these are the 4 flits that we are going to have this blue, green, red and this dark blue the 4 different fits that you can see it on the slide.

(Refer Slide Time: 19:53)



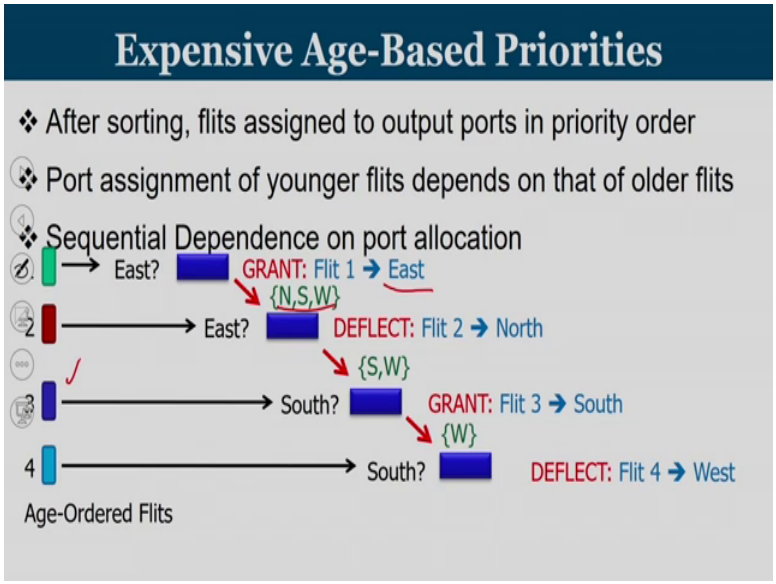
And routers must sort flits by age. So, you need to have a long latency sort network. So, in order to sort 4 flits so, essentially we are having 4 values. Let us say the number that is represented here represents the time at which these flits are generated into the packet. So, you can see that this blue flit the first one it is having time stamp creation of 4; that means, it was created at the time unit 4 that is green was created at a time unit 1, I red at 2, and this blue it as 3.

So, lower this number; that means they are the oldest one. So, it is a 3 stage sort network the first 2 are been combined together the larger one is going to come down whereas, in the case of second one the larger one is going up, and then this second level of comparison is also going to do a similar process like this, and then we have the third level where in the larger numbers will come to one entrance, smaller numbers are going to see at the other end.

So, by virtue of this you can see that now it is totally ordered. So, we had a different set of totally an unsorted set on 1 end of the sorting network it is a 3 stage sorting network. Once you move through all these 3 stages, then at the end what we can see that the flits are already in the sorter order.

But, this is an expensive process. Age based priority is going to be expensive because of this following factor.

(Refer Slide Time: 21:28)



So, after the sorting flits assigned to output ports in priority order, port assignment of an younger flit is always dependent on what is the port that is assigned to the older flits. Because, we have only 5 ports that is available inside a router; north, south, east, west, and local, you take the highest priority flit and give them the desired port once you have given one of the port to them then only 4 more ports are available.

So, the next highest priority the youngest flit or the next younger flit can choose only from this 4 he may be given one among that. So, as you come to the youngest flits they will be available only with some set of ports very few number of ports. And, then they may not won those ports there may be non-productive ports leading to deflection.

So, assignment of younger flits always depend on that of older flit. So, what are all the ports that are left behind after the older flits are been allocated and this can be done only with a sequential dependence on port allocation. So, consider this case let us say you have your blue flit that is highest in the age and blue flit is going to ask for the green flit is going to ask for east, since he is the highest priority flit the green flit is going to get the east. So, it is granted.

Now, we have the second flit that is red. So, whatever is available since east is already been assigned. Now, we have left with only north, south and west that is what has been shown there, but the red flit is going to ask for east that is his highest priority flit, highest priority direction.

So, after the green is assigned with east if anybody ask for east is not possible. So, we have your red flit requesting for east is cannot be given let us say arbitrarily north is been given. So, red is assigned with the north. Now, whatever is pending it is pending is only south and west is pending.

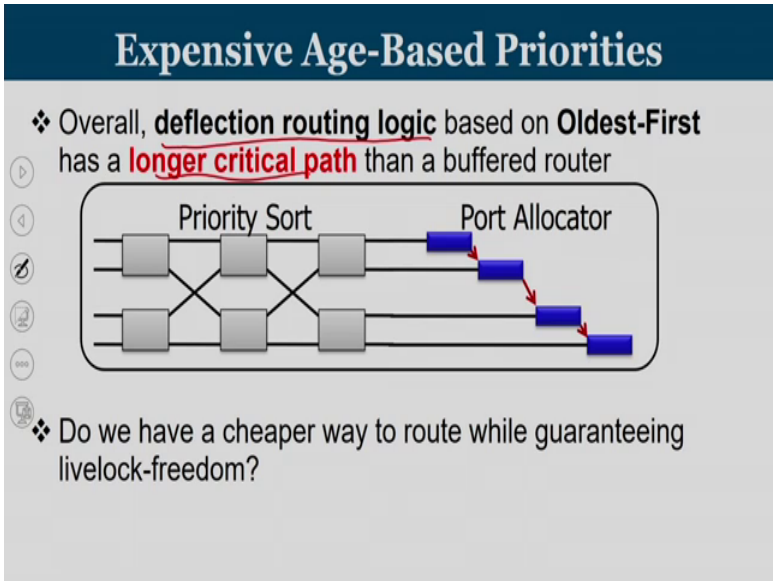
Now, we are going to consider the next highest priority this is going to be the next highest priority, that is looking for south yes south is available. So, that is been granted flit 3 is given with south and whatever his balance, balance is west. And, the last one is looking for south, but no chance south is no longer available. So, what I can give I can give it only on the west.

So, if you look at this flow that is there a very first flit is asking for a port it will get that. For after port allocation whatever is the ports that is available that only is passed on to the next flit. So, the processing of this second flit is possible only after the first flit is given with a port.

Similarly, the processing of the third flit can be done only after port allocation is done for the first flit and the second flit in terms of priority order; that means, there exists a

sequential dependence, because the output of the nth priority flit is to be done only after knowing what are the output ports that are available after satisfying the first N flits.

(Refer Slide Time: 24:49)

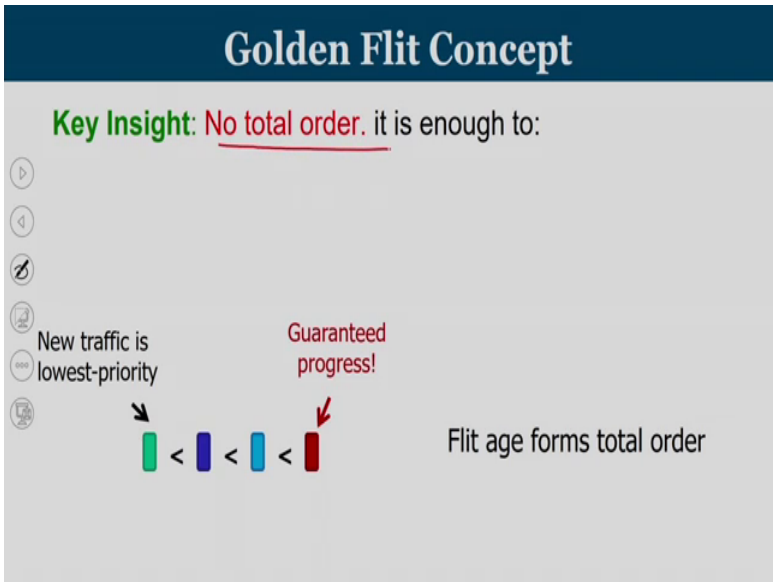


So, how it is been done? You have a priority sorting circuit that we have sun, which will sort the flits a followed by you have a port allocator which is sequentially that is being connected.

So, overall deflection routing logic what we have seen, which includes a flit ranking circuit which is known as a priority sort and the port allocator is based on oldest flit. So, overall this is going to take a longer critical path than your conventional buffered router.

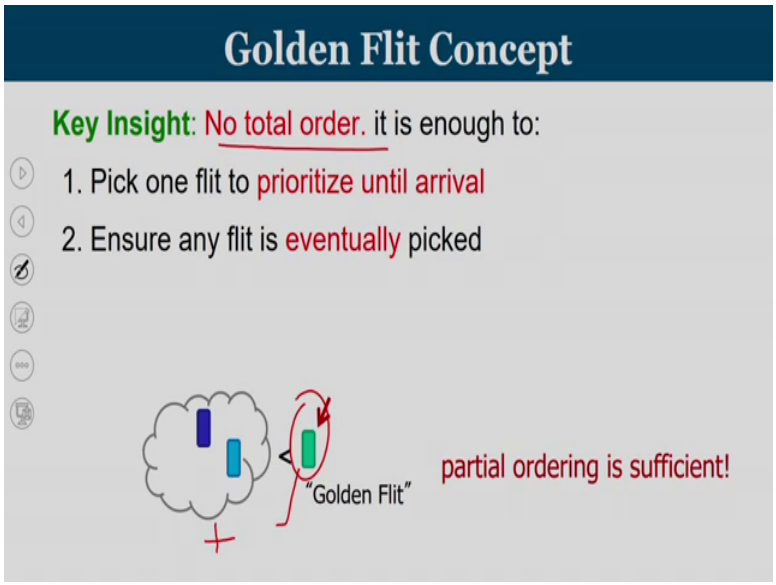
So, can we make it cheaper by guaranteeing the livelock freedom that is what we are going to see next? So, this is a work that is already in been published some 5 to 6 years before and we will give you the link for this published work also it is a golden flit concept is trying to be introduced here.

(Refer Slide Time: 25:42)



So, what do you have to do this is typically what has been done newer traffic are going to get lower priority, and older flits are going to have the highest priority and we forms a total order.

(Refer Slide Time: 25:58)

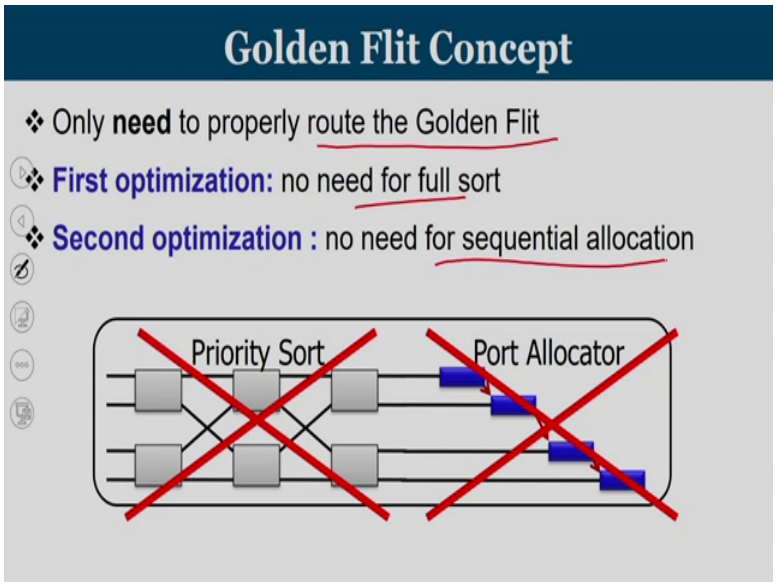


So, do you really want a total order, no total order is needed. It is enough to pick one of the flit that is what we are going to do pick one of this flit and call him as golden. And, we have to make sure that is there and all others are rest. So, it is golden words us a rest. So, we are not going to sort the entire flits, we are going to pick one of the flit as golden

all others are not golden. This is not such a complex kind of an operation like what we have seen in a 3 stage sorting network. A 3 stage sorting network is not required in this context. So, one flit is being picked ensure that after some time all the other guys are also going to be pick.

So, now we can see that the first one has reached destination. Now, from among the remaining flits that is available you are going to pick this next flit. So, the next flit is eventually that also will reach destination. Once that becomes golden you are making progress. Now, you choose a next flit to become golden. So, ensure that any flit is eventually picked up.

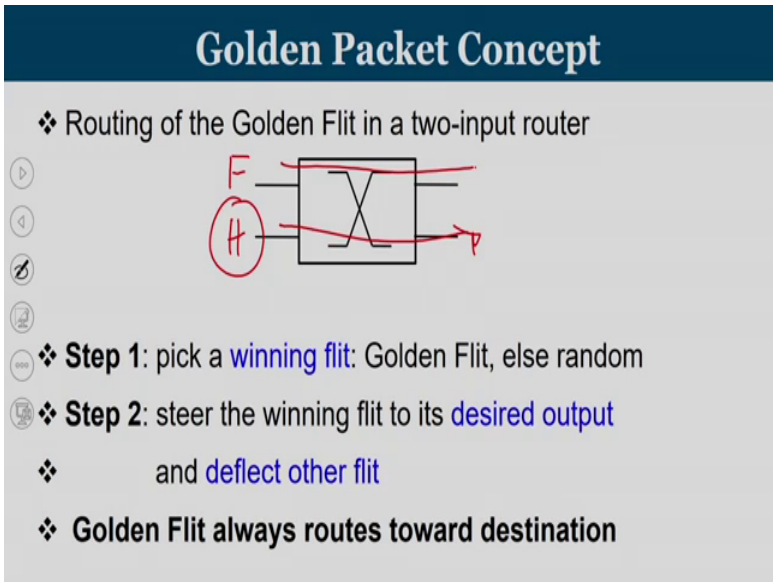
(Refer Slide Time: 26:58)



What is the golden fit concept? We only need to properly route the golden flit, the golden flit is now not going to be deflected. And so, we do not want a priority sorting circuit and a port allocator no need for a full sorting circuit.

Second optimization no need for a sequential port allocation we can go for a parallel port allocation. So, these are the 2 changes that we are going to bring on this.

(Refer Slide Time: 27:22)

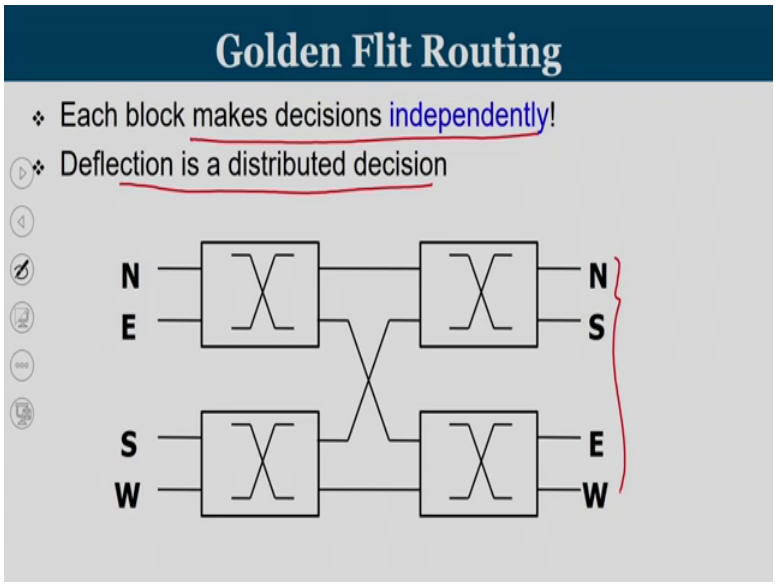


So, how are you going to work with a golden packet concept?. So, how will you route a golden flit if you have a 2 input router. So, step 1 from among the 2 flits that is coming to a router picked the winning flit, winning flit is typically golden. So, if you are one of the flit is golden that is going to be the winning flit, if none of the flits are golden then pick anybody by a random. Step 2 whoever is a winning flit give them the desired output port, the other one has to be deflected away and golden flits will always make a route that is going to be progressing.

So, consider the case that we have a flit F and a flit G that is going to come to this router; G is the golden flit and F is the non-golden flit. So, once you have 2 flits that is coming a router. The golden flit is given the productive port. Let us say the golden flit wants this as the output port; the golden flit is granted that automatically the other flit is being given the other output port.

Now, let us see a scenario where 1 of the flit or both the flits are not golden. So, consider 2 flits F and H. Let us say both are not golden, let us say I am randomly picking H as the winning flit. So, H is getting the productive direction whatever H wants and you have feel automatically assign to the other one. So, picking a winning flit if both the flits are non-golden pick somebody as a winner by random, if one of the flit is golden the golden flit is going to be winner, winner always get a productive port, the loser always get the deflected port.

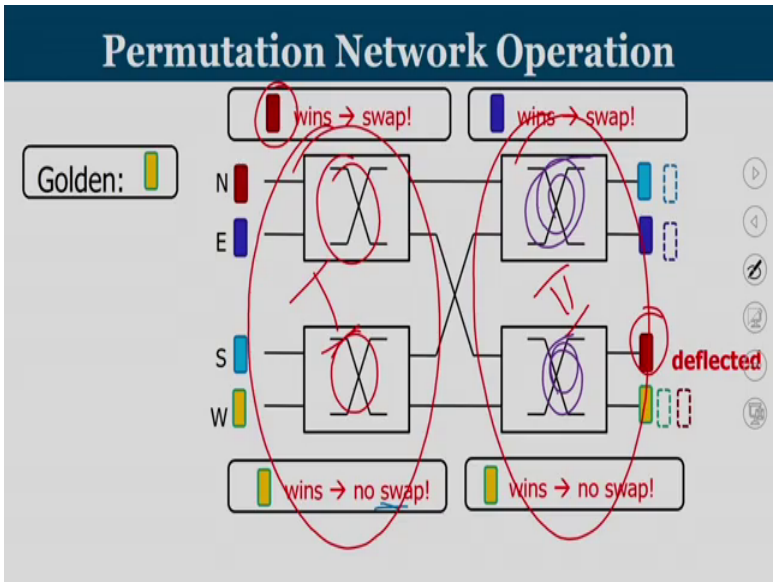
(Refer Slide Time: 29:03)



Now, golden flits are routing. So, what we are going to do is we are going to have 4 blocks, which is called a permuter block which called a permutation deflection network; deflection is taken as a distributed decision. So, each block make decisions independently.

So, consider this is the parallel port allocation scheme the sequential port allocator is eliminated with a parallel port allocation scheme, these are the flits, that is coming from north east south and west input direction. And, what you see on the right side are the flits that are coming from north, south, east and west of the output ports.

(Refer Slide Time: 29:43)



Now, look here how we are going to have this. Let us say this is the golden flit that we are going to talk about and the golden flits wanted to go to this output port. Let us say the red flit wanted to go to the same output port there. So, the golden flit and the red fit they both are looking for the same output port, this blue flit is essentially trying for this output port and this light blue is looking for this output port.

So, what you see on the right hand side is basically the desired direction for each of this port. And, the flit that is coming from the west input is the golden. Now, consider let us say what happens? In the first permuter since both the flits are non-golden one is been chosen by random. So, red is the winner and red wanted to go to this is the direction that the red is looking for. Since red wanted to go to an output port that is connected to the bottom end of the permuter. So, red wanted to go to this output port. So, as far as the first router is concerned the red should move into this block. So, when red is looking for that block blue automatically goes to this direction.

So, I will repeat once again as this is a bit tricky to understand. In the first permuter that is what we are going to see now, this is the first permuter in the permuter there are 2 flits that are coming both are non-golden flits. And, by random let us say red is the winner, we look for what is a desired output port for red. Red output lies in this port red actually wants this as the output.

In order to reach that the flit has to reach this permuter block. So, eventually red has to come down. Once the red is coming down, then the other flit that is a blue flit do not have any other choice, because the red is going to take the path this path is going to be used by a red. So, blue has no other choice blue has to move through this path.

Now, when you consider the second one? So, the red is going to come like that, now consider the second permuter that is there down here we have one of the flit that is golden. So, golden flit wanted to go to this direction that is the direction that is requested by the golden. So, this golden flit will travel straight there is no swapping that is required. The other blue flit is going to take the upward direction.

So, what I have mentioned here is the golden flit is going to take up a path like this, this is the goldens path. So, the flit will move straight like this. Now, after that the appropriate flit will reach the second stage of the permuter.

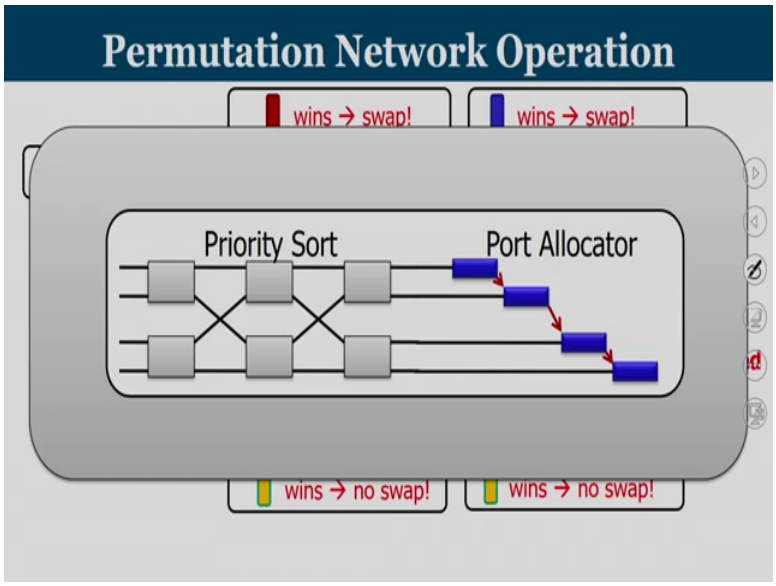
Now, in the second stage let us say blue is going to be because both the input flits both are non-golden. Now, we will assume that from among the available one this blue is the winner and blue wanted to come down to this direction.

So, once blue is going to give me given that direction automatically the light blue has to go up there is a swap that happens and the flit gets themselves exchanged. Now, what happens at the bottom end where the golden flit is involved. Golden flit to wanted to get this output port. So, that is output port what the golden fit is requesting for. So, golden fit will get it, there is no swap the red fit is going to get deflected.

So, this is the flit that gets deflected. So, now, what we have to understand this the permutation deflection network will consist of 2 permuter stage one parallely, they will work parallely, after that we have 2 more permuters that is stage 2. So, this whole thing is going to be stage 1 and this whole thing is going to be stage 2; both the units of stage 1 are going to work parallely. Similarly, both the units at stage 2 is going to work parallely so, rather than a sequential port allocation.

Now, we have parallel port allocation that is happening. This parallel port allocation can works, because we are not looking for a total order of flits we are only looking for a golden flit and the rest.

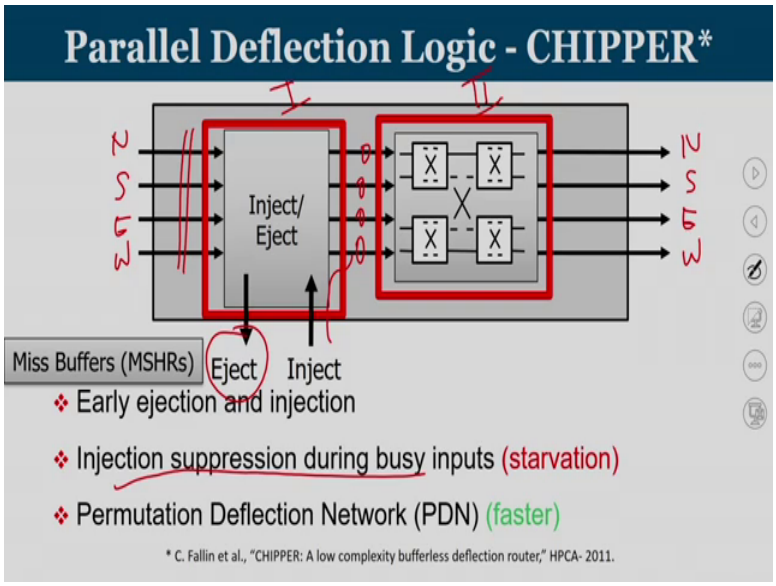
(Refer Slide Time: 34:39)



So, what we are going to achieve with the whole thing is whatever we are having. So, what we have we are going to replace the priority sorting and the port allocator with a permutation, deflection network. So, a work was published by Chris Fallin Et Al in high performance computer architecture conference in 2011, which was trying to get rid of the issues that is associated with the bless router. The sorting network and the sequential port allocation, they have completely been removing them and coming up with a routers whose name is called chipper.

So, let us try to understand how this chipper router works?

(Refer Slide Time: 35:19)



These are the 4 directions that you see it is the, north, south, east and west are the 4 input directions. And, then we have the north south east and west as the output directions. So, the very fastest unit inside the router pipeline is the eject and the inject unit. We have seen that is the conventional NoC router you have buffers in the input. So, the packets will come and stay in the buffers and then you have the routing logic a virtual channel allocator logic and the switch allocator logic followed by the crossbar.

So, here it is totally different thing we are going to have an early eject state. The first operation that a router does when it gets a couple of flits is try to see whether any of the flit is to be ejected to the local port. If so, remove that flit from the router pipeline and that is known as early ejection.

Similarly, we have an injection also to be done. So, once the ejection is over then you are going to inject the newly created packets. In a tiled chip multi core processors packets are injected whenever you have to send a flit into the router and that happens whenever there are cache misses. So, cache misses are recorded in this miss buffers MSHR, Miss Service Holding. Miss Status Holding Registers and injection suppression happens whenever the input port is busy.

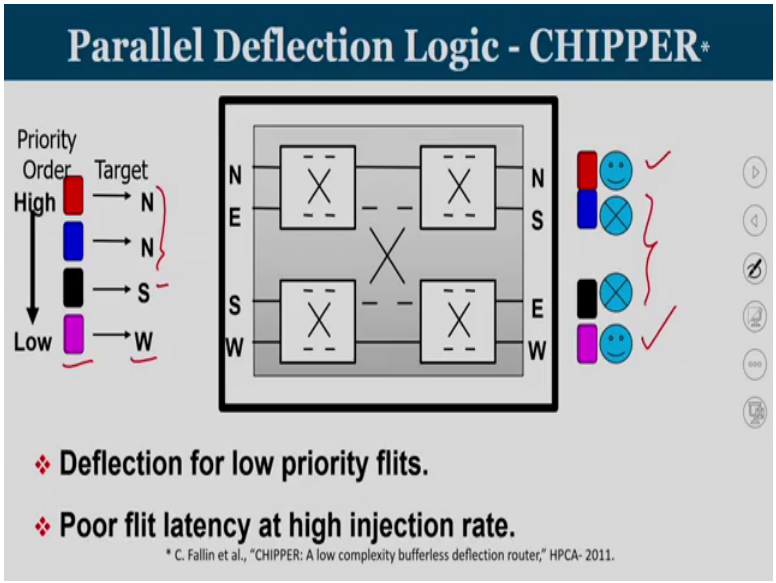
So, here the idea is I cannot inject any flit, if all the input buffers are full we have to see that a flit that is going to be injected has to occupy one among this channel. So, if all these channels are going to be full or containing some valid flits, then I cannot inject.

So, injection is possible in a bufferless deflection router only if one of the input channel is idle. So, if you get 4 flits through all the 4 direction injection is not possible or if at least one of the channel is empty, then injection is possible. So, that is why we are keeping ejection at very early in the pipeline. So, if there is a flit to be ejected surely that channel from which the flit is ejected is empty which gives space for the newly injected flits.

So, injection suppression happens during busy inputs and that will lead to starvation and the right side is the second stage. So, this is basically your stage one the first cycle of the router and what you see second that is a permutation deflection network. And, this is acting as a very fast unit that is your stage 2.

So, it is basically a 2 cycle router, when you work in chipper; chipper is going to forward all the packets in 2 cycle. In cycle number 1 you perform the ejection and injection, in cycle number 2 you are going to perform permutation deflection network or basically it is a port allocation.

(Refer Slide Time: 39:11)



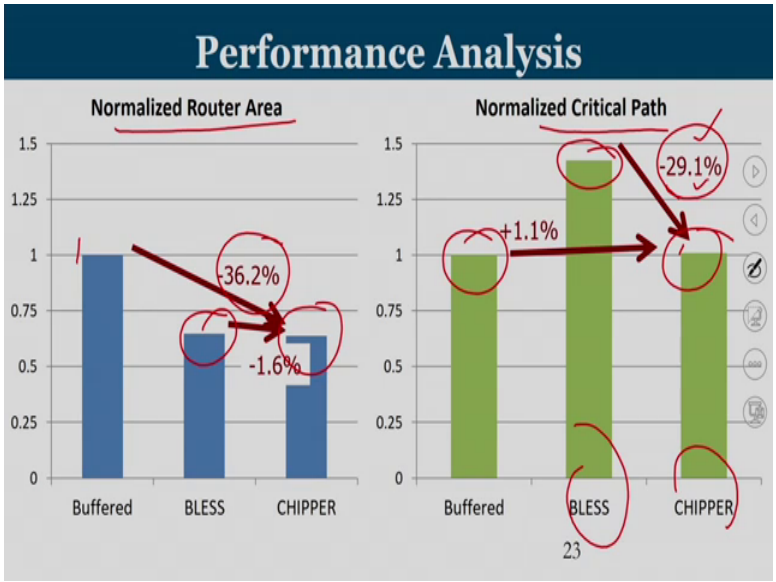
So, this parallel deflection logic it is what we have seen in the chipper. Let us say these are going to be the priority that is being shown we have highest priority for red and going to the lowest priority. And, red and blue they wanted to go to north, the black wanted to go to south output port and this magenta I wanted to go to the west output port. We have seen what is a logic 2 flits are coming together based upon the priority, some of them are

getting swap and they reach the second stage and now you see the red got the desired ports or red is happy this magenta also got the desired one. So, they are also happy.

Whereas a blue and black they did not get the desired output port. So, that lead to a scenario that some of the flits make it deflected. So, the main problem that is associated with the chipper design is lot of flits will may get deflected, whenever all the input ports are busy with packet. So, when you have too many packets, they may compete for same output port and some of the flits may get deflected.

So, this problem is very severe when you go to higher injection rates, when you have higher injection rates means more packets are coming into the network and more flits are going to travel through network more of them will compete and leading to higher deflection. And, when you have higher deflection a flit is going to travel more distance. It is going to take more number of hopes before it is reaching the destination. So, in that context the average latency of the flit is also going to be higher.

(Refer Slide Time: 39:42)



Now, this is the results that is been coated by the authors of the chipper paper. The left hand side where the blue graphs are showing what is it normalized router area.

So, in architecture research, especially in the case of network on chip how will you know that if you wanted to propose a new router micro architecture. How will you prove that, that new router micro architecture is going to safe more area or space or power whatever.

The concept that has been use is we how to implement the router, in terms of a hardware description language like V H D L or Verilog.

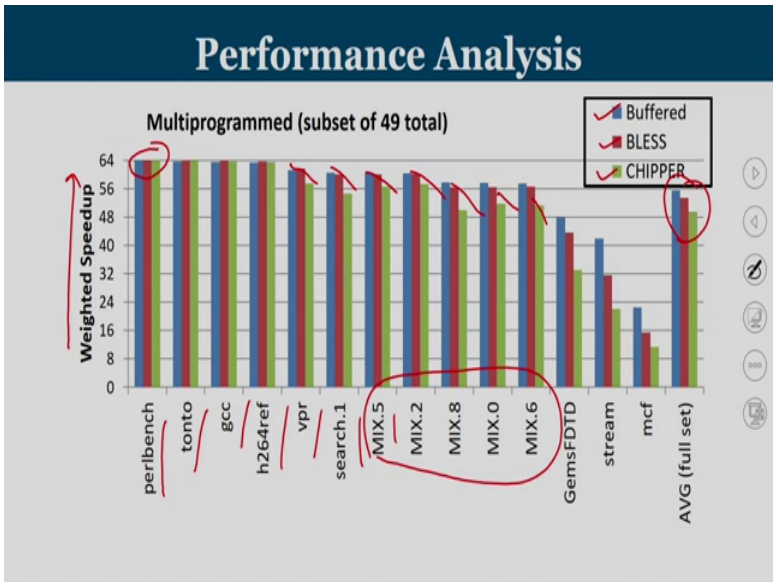
So, once you design your router in this hardware description language do the synthesis and get the hardware synthesis reports and they are going to normalize. So, if you are using a virtual channel router and we let us say that is going to take 1 unit area, then bless is going to consume roughly up to 0.6 round at area. Even chipper is also going to consume more or less the same area.

So, in terms of area of a router, we can see that bless and chipper is able to reduce substantially close to 36 percent is the savings, what the authors of this proposed work of chipper has achieved. But now you have to understand that how complex is a circuit and that is what is known as critical path. How will you obtain critical path? Here also once we implement our design, this design has to be synthesized and we have to get the maximum combinational delay. The number of gates that come in the maximum combinational delay path, it is what is known as the critical path. So, how much time it takes for this router to process it is input.

So, this is also normalized to buffer if the buffer is going to take one unit of time then your bless is taking roughly up to 1.4 above, but chipper is comparable chipper is slightly above one chipper is comparable to that of the buffered router. So, this is only 1.6 less percent direction, but here is a critical path is reduced by 29 percent. What does it mean? It is more or less same that of buffered. So, how much reduction we are or how are we going to get reduction? We are eliminating the parallel sorting that the sorting circuit the 3 stage sorting circuit there is no total ordering that is required, and that the second operation is rather than going for a sequential port allocation we are coming up with a parallel port allocation called permutation reflection logic.

So, the golden flit concept together with permutation deflection logic is responsible for the reduction of 29 percent of routers areas critical path. The meaning is a router or an NoC with a chipper router can operate 29.1 percent times faster clock than an NoC that is working with the bless router.

(Refer Slide Time: 42:48)



They have shown a couple of results what they have got they have taken a 64 core tiled chipper multicore processors were some benchmarks are on the X axis shows, what are the various benchmarks and some of the benchmarks are mixes. So, they ran probably 32 of them with 1 application the remaining 32 core is with other application.

So, these are all the workloads that they are going to consider and the X Y axis is basically weighted speed up. So, if you have 64 cores the maximum weighted speed up that you are going to get is 64 and you can see that the red one or the brown one rather it is a bless router, the green one is chipper and the blue one is buffered.

So, the speed up that you are going to achieve is slowly going to come down that is what you can see that there existed then and this is the average that is going to be there. So, the speed up that you achieve the performance in terms of the number of packets that is reaching the destination, the number of instructions that completes. Once you employee a bless router or a chipper router is going to be compared with this.

So, the performance lightly comes down, but you are going to have lot of area savings. So, there is to summarize what we had in today's lecture. We were trying to understand that buffers, that is a very crucial component in giving performance of an NoC routers are really power hungry circuits.

So, if we can get rid of buffers we are going to gain lot of power and area savings, but since buffers are an important component that governs the performance of NoC we we have to compromise on certain aspects. So, at very low injection rates, this much buffers are not going to play a significant role. So, traditional input buffer routers are going to be replaced with buffer less reflection routers, where we have pipeline latches only.

We have seen 2 types of deflection routers today. The first one is completely bufferless and we have a total ordering sorting circuit followed by a sequential port allocation that is called bless router and, then the second one this total ordering is replaced with a flit which called a golden flit followed by a parallel sorting circuit that is called permutation deflection network. So, these are the 2 standard bufferless reflection routers, that is proposed a half a decade before still it is a very active research domain.

Our next lecture we will try to see what are the problems of bufferless deflection router? Of course, the main problem is you are going to have high deflection rate. Can we reduce a deflection rate by incorporating some kind of a storage inside. So, with this we complete this lecture. We will continue with lower power NoC designs especially energy efficient NoC designs, by reducing or by minimizing the number of buffers. So, with this I conclude.

Thank you.