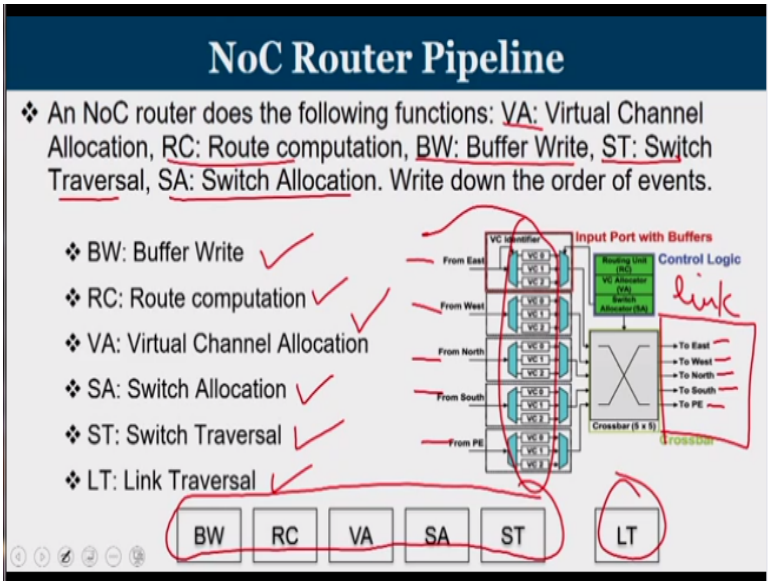
Advanced Computer Architecture
Prof. Dr. John Jose
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology-Guwahati

Lecture-33
TCMP and NoC Design Principles

Welcome to tutorial number 10, in this tutorial, the final tutorial of this course our discussion is on tiled chip multicore processors and NoC design principles. We will work out few problems related to these topics which will help you in getting a deeper clarity on the concepts learned during the lecture videos.

(Refer Slide Time: 00:54)



The first question is on an NoC router does the following functions virtual channel allocation which is represented by VA. Then route computation, buffer write, switch traversal, switch allocation, so these are some of the functions that happen in an NoC router. Now we have to write the order of events in which order these are happening, so this is the internal architectural diagram of an NoC router in a 2 dimensional mesh NoC each router is connected to 4 of it is neighboring routers and there is a link that is connecting to the local processing element.

The flits will travel through this ports and reach these virtual channels these are called buffers. So buffer write is the first operation that happens in every cycles the flit is that are residing inside

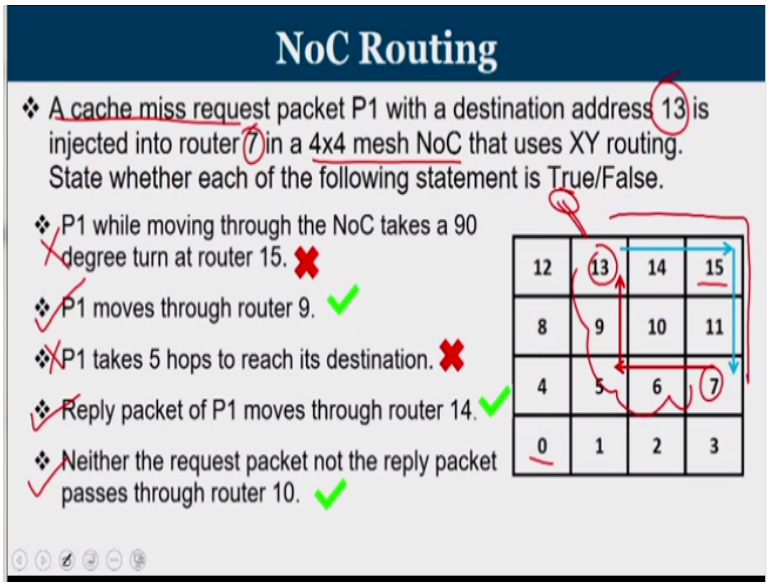
the buffers they are been routed. So you apply the corresponding routing algorithm to find out what is the desired output port. So routing computation is the second operation once route computation is over the very next process is flow control.

The process by which you reserve a buffer in the downstream channel, let us say if a packet get east as the output then in the east neighbor I have to reserve a buffer for this packet and that reservation is known as virtual channel allocation. For all packets who got virtual channels, now we have to assign them the output ports. So there maybe cases that multiple packets might have got the route same route or different packets are competing for the same port.

And in that case only one can be permitted per port, so we can permit only one port through east, one through west, north, south and east utmost. So switch allocation is a process by which whenever there are multiple packets competing for the same output port which is the winner that is decided by the switch allocation algorithm. And then the flit travel through the switch and then at the end it is travelling through the link.

So the order in which it is been done is buffer write, route computation, virtual channel allocation, switch allocation and switch travel. This many things are happening inside the router and this is the link traversal. So the moment it comes out of router then you are already in the link, so link traversal is the final stage.

(Refer Slide Time: 03:19)



We move onto the next question a cache miss request packet P1 with a destination address 13 is injected into router 7 in a 4 by 4 mesh network on-chip that uses XY routing. State whether each of the following statements is true or false. So these are some statements that is been given, so we are talking about a 4 by 4 mesh NoC. So we have to be familiar with the numbering let us say this is the 16 core processor that they are talking about in which the routers are organized in a mesh fashion in a 4 by 4 mesh fashion.

And always numbering is from bottom left all the way to the top right, now here we have a packet whose destination is 13 is injected into router 7. So from router 7 there is a packet moving into 13 and when you apply XY routing, this is the way in which the packet goes from 7 to 13, it is a cache miss packet. So there is a cache memory that is associated to 13 and you go and search in the cache get the data and then a reply packet is created that will travel all the way from 13 to 7.

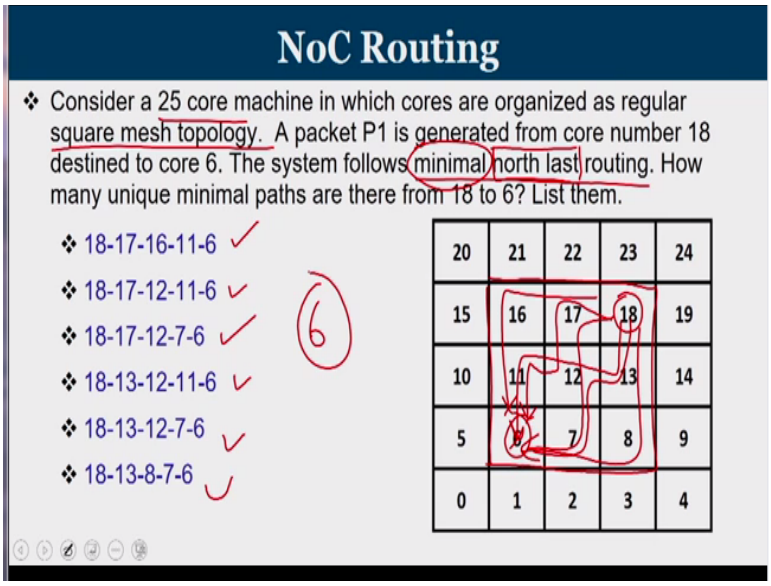
So your request packet is going through the red line what is been shown as per a XY routing and the reply packet is coming through the blue line as showed in the diagram. Now there are few statements given from which we have to say whether they are true or false, P1 that s a packet we are talking about while moving through the NoC takes a 90 degree turn at router 15, so the P1 is moving through the red, so it never travels through router 15.

So question of whether it takes a 90 degree turn at router 15 that means it is actually wrong, it is false. Now P1 passes through router 9. We can see that it is passing through router 9, so that is true, P1 takes 5 hops to reach the destination. So from 7 it takes 1 hop to reach 6, 2, 3 and 4 it takes only 4 hops to reach destination, so P1 takes 5 hops to reach destination that is also false.

The reply packet of P1 moves through router 14, you can see that the reply packet is travelling through the blue arrow. And the reply packet is moving through 14, so this is true neither the request packet nor the reply packet passes through router 10. We can see that the request is passing through 7, 6, 5, 9, 13 and the reply is coming from 13, 14, 15, 11, 7, so none of them are touching 10.

So neither the request packet nor the reply packet passes through router 10 that is actually true. So we have first sentence false, second one is true, third is false, 4th is true and fifth is true.

(Refer Slide Time: 05:59)



We now move onto the next question, consider 25 core machine in which cores are organized as regular square mesh topology. A packet P1 is generated from core number 18, this is core number 18 destined to core number 6. The system follows minimal north last routing, how many unique minimal paths are there from 18 to 6. So I am talking about a packet that start from 18 to 6 in a 5 by 5 mesh interconnects.

So how will you know it is 5 by 5 it is a 25 core machine and they are organized as square mesh topology. So 25 has to be organized as square and this is the way how it is organize you have 5 row and 5 column format and the numbering is 0 is there in bottom left corner and 24 is there on the top right corner. Now we have to understand what do you mean by it is north last routing, minimal north last routing .

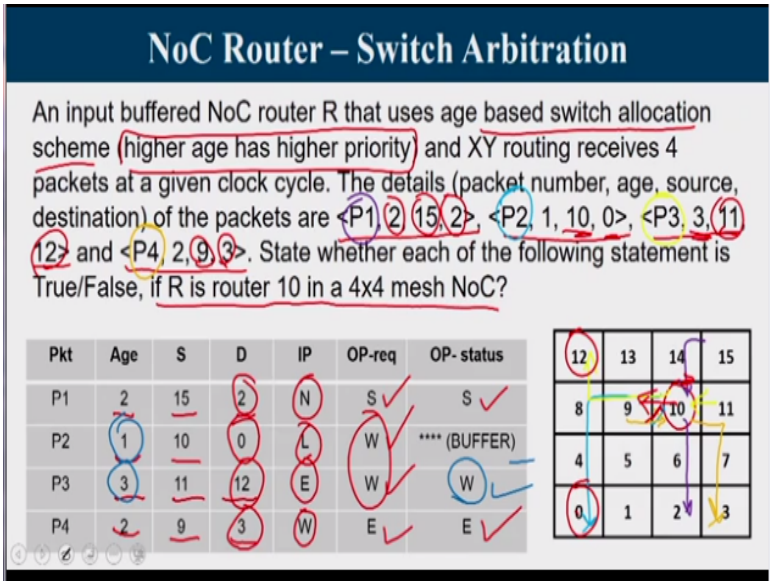
So minimal north last routing say that first of all the minimality condition at every hop the packet should reach 1 hop closer to the destination that is called minimality. If that condition is ensured everywhere we can call the algorithm as minimal. Now the second one is called north last, so if a packet wander to move to the north direction at any point of time. Then it has to be done at the end, that means transition towards south, east and west has to be taken.

And then only you can take a north, the moment you take a northward direction movement then no other turns are been permitted. So first thing is here you are talking about minimal that means my packet can travel only in this quadrant from beginning. So it cannot think of the routers which are there in the edge and corner, so from 18 it has to go either to 17 or to 13. Now north last, so since my destination is on south of me there is no need to go to the northward direction.

So there is no restriction that is been applied, had the destination be in 20 for example then I can travel from 18, 17, 16, 15 and then only I can take north transition, that is called north last. So in this case we are not talking about 20, anyway I am just giving an example of how it works. So from 18 we have possibilities, 1 is I can take a possibility like this that is my first possibility, second possibility is I can take an option like this.

Because there is no restriction in taking an east south or southeast whatever order it is, we have restriction only in moving towards north that is called north last routing. And the third possibility is like this then we have a possibility like this and then the possibility is like this, and the last possibility is like this. So we have 6 options 18- 17- 16- 11- 6, 18- 17- 12- 11- 6, 18- 17- 12- 7- 6, 18- 13- 12- 11- 6, 18- 13- 12- 7- 6, 18- 13- 8- 7- 6. So we have total of 6 unique path from 18 to 6 with minimal north last routing implemented.

(Refer Slide Time: 09:09)



Now the next question is on switch arbitration, an input buffered NoC router R that uses age based switch allocation scheme, so age based means higher age has higher priority and the XY routing receives 4 packets at a given clock cycles. The details that is packet number, age, source and destination of the packet are let us say we have a packet P1, 2, 5, 15, 2 P2 is 1, 10, 0, P3 is 3, 7, 11, 12 and P4 is 2, 9, 3.

Let us try to understand what does representation is, you are talking about a router, state whether each of the following statement is true or false, if R is a router 10 in a 4 by 4 mesh interconnect. So to a router 10 in a 4 by 4 mesh NoC we are getting 4 packets in a given clock cycle, packet P1 it has an age of 2, it is coming from 15 it is in the process of going to 2. So 15 and 2 are the source destination similarly P2 is coming from 10 going to 0, P3 is from 3 going to sorry P3 is from 11 going to 12.

And P4 is from 9 going to 3, so the age of the packets are also given, now let us try to understand how these packets are coming. So we are talking about router number 10 these are the packets, now from where P1 coming, P1 is coming from 15 it is going to 2. So P1 coming from 15 it is going to 2, so this is the process by which P1 comes and it wants to go to 2, that is all about P1. Now P2 is coming form 10 and it is going to 0, so P2 is a newly created packet coming from 10 and it is going to 0, that is a condition of P2.

Now P3 is coming from 11 and it is going to 12, so P3 is coming like this and it is going to 12 that is called P3s case. And finally we have P4 which is coming from 9 and going to 3, so the packet coming from 9 and it is going to 3, that is all about P4. So these are the **2** 4 packets that is coming, so P1 is entering router 10 through north input port that is what is been shown there north input port.

Now P2 is coming from local input port and then it is going towards or which is trying to go towards western direction, P3 that is coming from east direction and trying for west and P4 is coming from 9. That is 9 is connected to west, so it is coming from west input port and trying to move towards east. Let us try to write all these. So packet P1 has an age of 2 whatever data that is been given source is 15, destination is 2 it is coming through north input port, P2 is having an age of 1.

It is coming from 10 it iself that is it is coming from the local port and P3 age is 3, source is 11 destination is 12 that means it is coming from east input port. And P4 age is 2, source is 9 and destination is 3, so it is coming through the west input port. Now when you apply XY routing at router 10 if you have a packet that is looking for destination 2, then south is going to be it is output port.

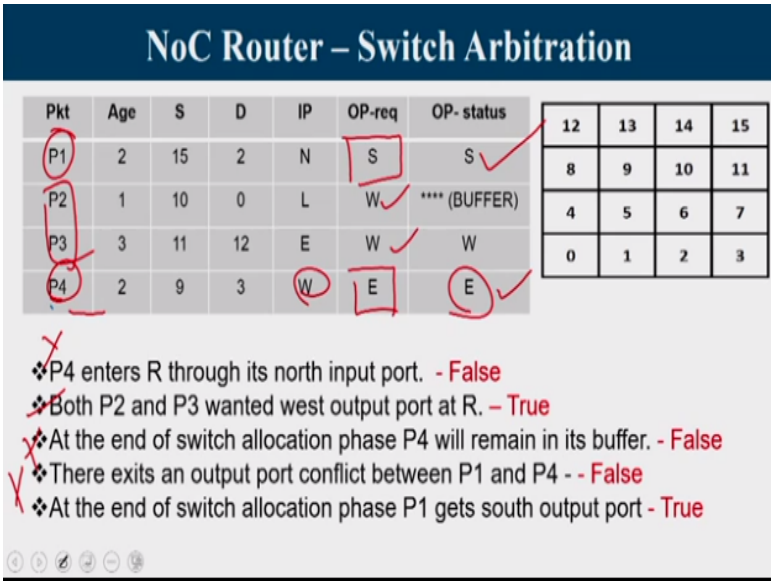
Now in 10 if you wanted to go to 0, 0 is here but if you apply XY routing first I have to travel in X direction, so west is what I want. Similarly if I wanted to go to 12 that also will be possible only through traveling westward direction. So a travel to 0 and travel to 12 anyway from 10 it is a westward movement as per XY routing. And the last packet P4 destination is 3 and to reach 3 I have to take an eastward movement.

Now if you look at we have 4 packets out of which the first one is looking for south, second and third is looking for west and the 4th packet is looking for east. So since there is nobody for south, south will be granted for P1. Since there is nobody competing for east that is only one then east is also granted but we can see that there are 2 of them looking for west. Here is a place where switch arbitration happens.

So what we have learned here is we have couple of packets coming together and we are there in an NoC router and we find out through which input port they are coming and based upon the destination address we have to find out what is the desired output port. If there is only one packet that is competing for this particular output then that is been granted but if there are more packets looking for the same output port then we have to pick one among them and that is called switch arbitration.

In this case arbitration is done with the help of age based priority, so that packet which is having the higher age that is been given preference. Now we have a case where 2 packets are competing for the west output port. So one is packet P2 other one is packet P3, so if you carefully look you know that packet P2 has an age of 1 and packet P3 has an age of 3. So P3 is been given the west port and P2 is ask to buffer, so this is the status that we have.

(Refer Slide Time: 14:46)

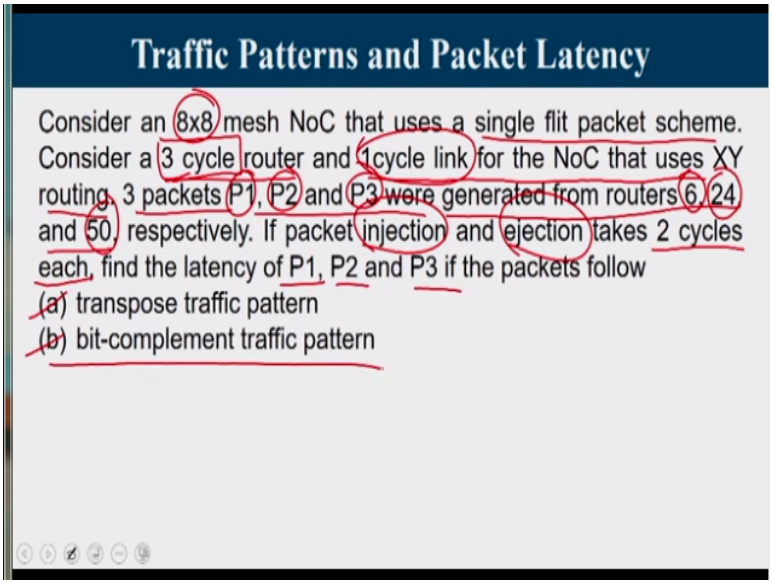


Now based on this observations there are certain questions that is been asked, P4 enters R through it is north input port. So the statement is P4 enters R through it is north input port, so P4 is entering through the west input port, so this statement is false. The second statement is both P2 and P3 wanted west output port at R, so P2 and P3 both they both wanted west output port at R that is true.

Now at the end of switch allocation phase P4 will remain in it is buffer, so this is the question but P4 actually got east output port. So the statement that it will remain in the buffer is false and then there exists an output port conflict between P1 and P4. So P1 this is output port conflict means they are both are requesting for the same, so P1 is requesting for south and P4 is requesting for east, so there is no conflict.

So the statement there exists an output port conflict between P1 and P4 is false and moving on to the last one. At the end of the switch allocation phase P1 gets south output port, so P1 it is getting the south output port, so that is true. So in this case based upon the routing that is been happening and the switch arbitration policy, few statements are given and we are checking whether they are true or false.

(Refer Slide Time: 16:05)



Now the next question is on traffic patterns and packet latency, consider an 8 by 8 mesh NoC, that use single flit packet scheme. Consider a 3 cycle router and a one cycle link for NoC that uses XY routing, 3 packets P1, P2, P3 were generated from router 6, 24 and 50 respectively. So we are talking about an 8 by 8 mesh NoC out of which 3 packets are generated, and here telling that in the router will take 3 cycle.

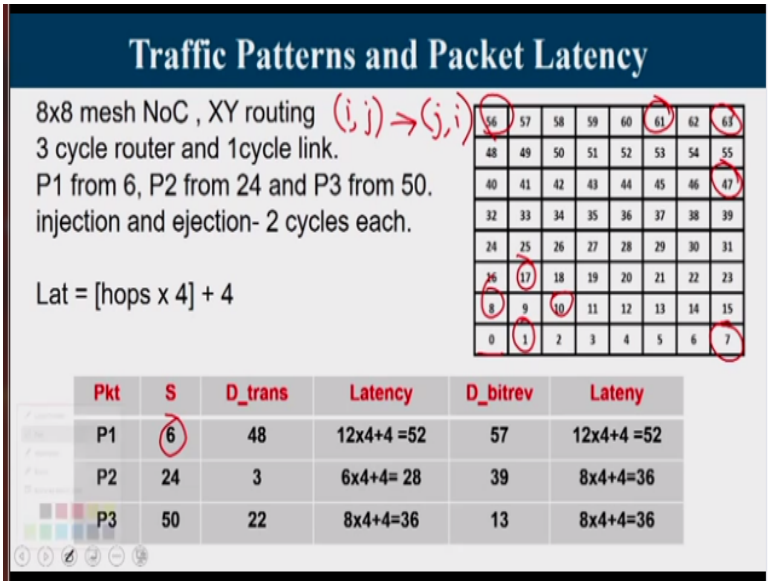
So we have learned different functionalities of the routers like buffer write, route computation virtual channel allocation, switch allocation and switch traversal together all these 5 operations

are being divided into 3 cycles and it take 1 cycle through the link. So moving from one router to another it will take 3 cycle inside the router, that is a processing tile inside the router and 1 cycle in the link, so total a hope will take 4 cycle, it is using XY routing.

Now we are talking about 3 packets say here P1, P2 and P3 are 3 packets which are starting from 6, 24 and 50 respectively, if packet injection and ejection take 2 cycles each. So the process by which a local tile create a new packet and inject into the local port, that is called injection and the process by which a packet is removed from a router to the tile that is known as ejection. So at the source the injection happened, at the destination the ejection happen.

So both injection and ejection process in this problem are taking 2 cycles each. Now we are ask to find out what is the latency of P1, P2 and P3, if the packets follow a transpose traffic pattern and a bit compliment traffic pattern.

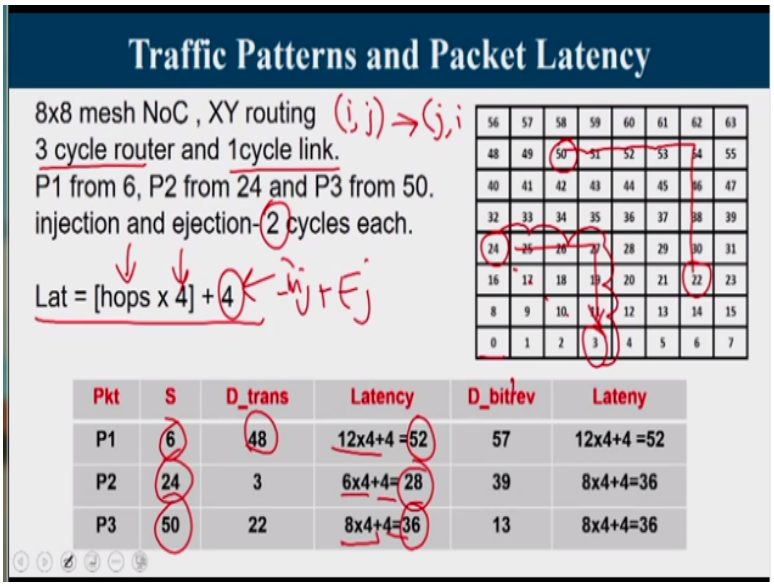
(Refer Slide Time: 17:45)



So this is the 8 by 8 mesh that we are talking about ranging from 0 all the way up to 63. Now this particular table you look at packet P1 it is source is 6. Now there are 2 traffic patterns that is been mentioned, the first a traffic pattern is called a transpose. So transpose means if a packet that is starting from row i and column j, in transpose pattern the packet is going to row j and column I, so interchanging the row number and column number.

So a packet from 1 will go to 8, now a packet 10 will go to 17, a packet 7 will go to 56, similarly a packet 61 will go to 47, so this is just to give a picture about how the transpose pattern works.

(Refer Slide Time: 18:43)



Now in this case we are been given 3 packets P1, P2 and P3 and your packet P1 is starting from source s from 6, packet P1 is starting from source 6. Now as per transpose pattern from 6 the packet should move to 48, so 48 is a transpose node. All packet generated from 6 are going to 48 that is why, so this kind of transpose is a synthetic traffic pattern we are artificially creating traffic, meaning we are mentioning where is a source and where is a destination.

So all packets from 6 are moving into destination 48, now what is the peculiarity, from 6 to 48 it take 1 hope to reach 5. So 2 hope 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 it will take 12 hops from 6 to reach 48, in each of the hope we are going to take 3 cycle in the router and 1 cycle in the link. So total 4 cycles are needed to complete one hop plus you have 2 cycles for injecting in the packet at 6 and 2 cycles for removing or ejecting the packet at 48, so you have another 4 more cycles.

So the equation is latency of a packet is defined us number of hops into 4 + 4, this second 4 will take care of injection as well as ejection. So 12 into 4, so 12 hops is the packet has to take 12 into 4, 48 + 4 52, so it takes 52 clock cycles for a packet starting from 6 which follows a transpose pattern.

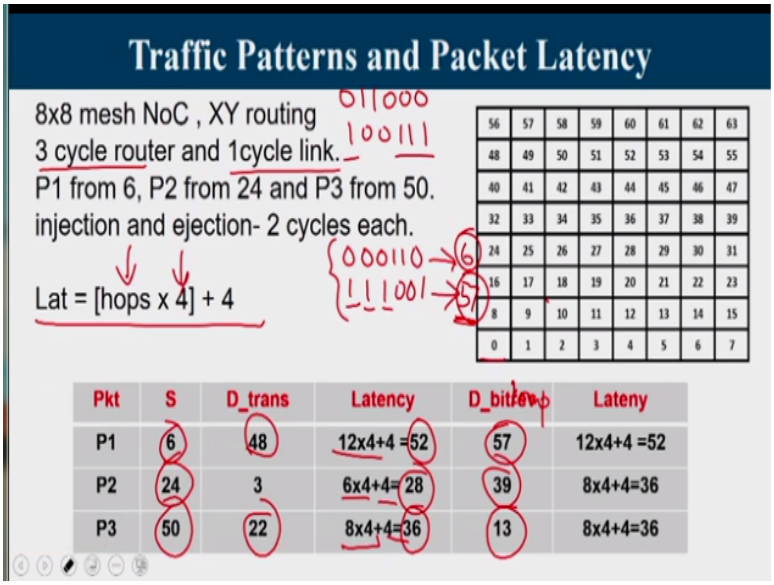
12 hops *4 =48 + 4more cycles =52 cycles

Similarly packet P2 the source is 24, now when the source is 24 the destination ask for transpose pattern is destination is 3, we interchange row and column.

So all packets on 24 are traveling through 3 and this is the path by XY routing it will take, so how many hops are there 1, 2, 3, 4, 5, 6 so we have 6 hops 6 into 4, 24 + 4, so it take 28 cycles for this packet. And similarly we have P3 that is starting from 50 and as per transpose pattern the destination is 22 and it takes 8 hops 1, 2, 3, 4, 5, 6, 7 and 8, so 8 into 4 + 4 total 36 cycles are needed for a packet to reach 22 from router number 50.

For the same question rather than transpose pattern, they are asking for a second traffic that is known as bitrev or bit compliment traffic. In the question it is mentioned we have to find 4 bit compliment traffic as well. Packets starting from 6 the destination is 57, now we have to find out how a bit compliment destination is there.

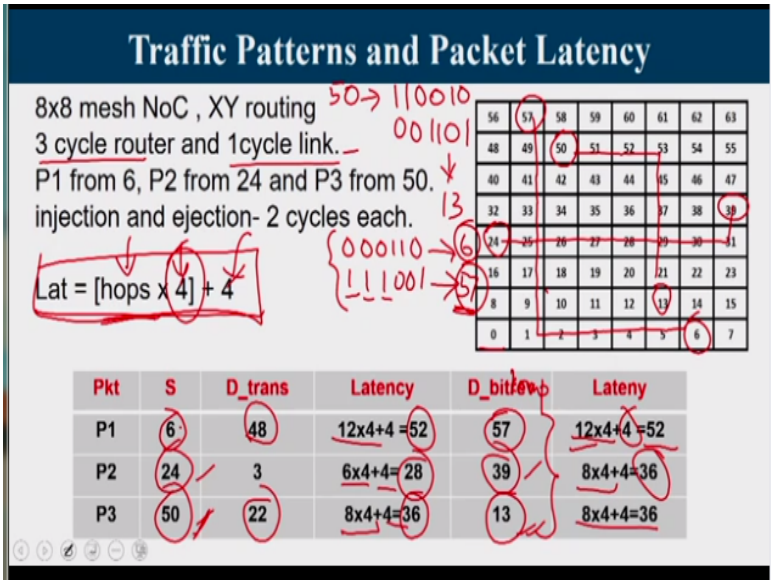
(Refer Slide Time: 21:52)



In the case of a transpose pattern we have to find the transpose node, so in the case of bit compliment our source here it is given as 6 packet P1, the source is 6. So when you write 6, this is the binary value of 6, we compliment this, so we get this as the compliment. And this correspond to, so packets from 6 are moving to 57 when you follow bit compliment traffic pattern, similarly packet 24.

So if you look at what is the compliment node for 24, 24 is represented by this is 24, now it is compliment is, so it is 32 + 7, 39. So as per bit compliment pattern packets generated from router 24, the destination is 39. Similarly if you write 22 you can find that the destination is going to be 13, so how are you going to assess it.

(Refer Slide Time: 23:24)



So packet P3 is starting from 22, so for the case of packet P3 the source is given us 50. Now source of 50 means, 50 is defined as 32 + 16 that is 48, this is 50, so if you compliment and this indicates 13. So a packet that is starting from 50 it destination is 13, now the problem is same like what we discussed previously. So the whole purpose of using transpose and bit compliment is to understand what is a destination, so in transpose it is interchanging of i and j.

In the case of bit compliment you write the binary value of the source node and then compliment it in binary and that is going to be the destination. So packet from 6 will go to 57, 24 will go to 39 and 50 will go to 13. Now from 6 to 57 there are 12 hops, you can see that 6 all the way to 57 and as per XY routing it is going like this 1, 2, 3,4, 5, 6, 7, 8, 9, 10, 11, 12, so 12 hops

$$12 * 4 + 4$$

that is for injection and ejection that gives you 52 cycles.

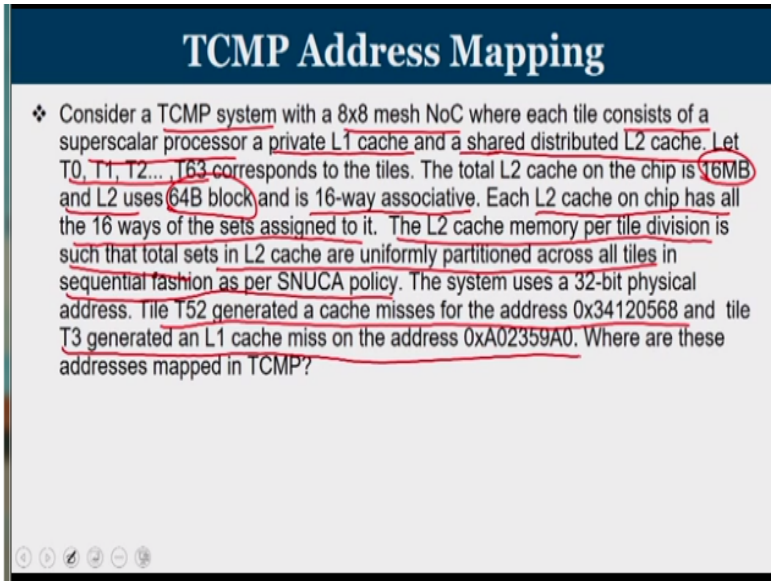
Now coming into the second one from 24 to reach 39, so this is 24 and I have to reach 39, so 1, 2, 3, 4, 5, 6, 7, 8 so it is

$$8 * 4 = 32 + 4 = 36 \text{ cycles}$$

And the last one is moving from 50 all the way to 13, so this is 50 and 13 is given here, so 1, 2, 3, 4, 5, 6, 7, 8, 8 * 4, 32 + 4, 36. So in this way we are able to find out how much a packet is taking to travel through the network. So based upon the source and destination and the number of hops it has to take, we have to apply this equation.

The latency of a packet is defined as it depends on number of hops and in each hop and It will take 4 cycles to complete and the initial overhead of 2 cycles for injecting and the final 2 cycles for rejecting this packet.

(Refer Slide Time: 25:38)



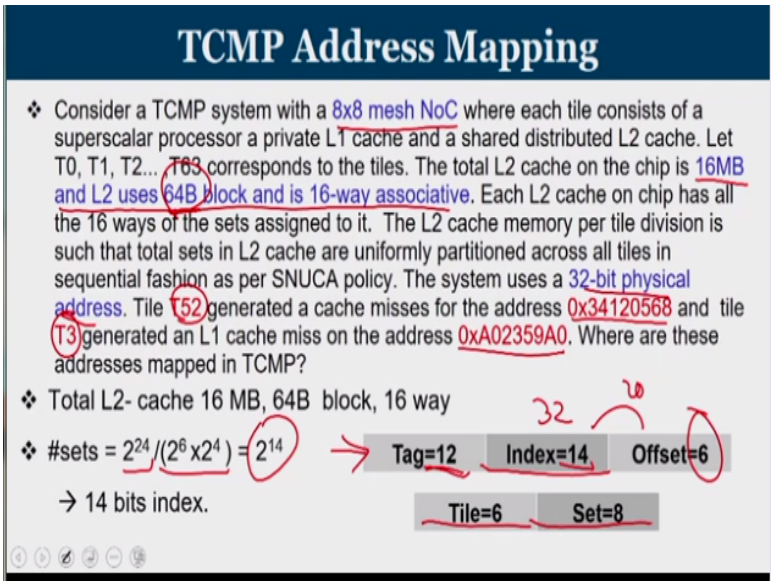
Now let us move into TCMP address mapping, tiled chip multi core processors, so this is a typical question which covers all the aspect of cache memory and NoC together. Consider a TCMP system in an 8 by 8 mesh NoC where each tile consists of a superscalar processor, a private L1 cache and a shared distributed L2 cache. We have seen it in the lecture each tile consists of a processor L1 and L2 cache. But a L2 cache is shared and distributed let T0 up to T63 correspond to the tiles, the total L2 cache on the chip is 16 MB.

So I have 16 MB of L2 cache that is scattered across 64 tiles, and each L2 uses 64 byte blocks and the L2 cache is 16 way associative. Each L2 cache on-chip has all the 16 ways of the set assigned to it. The L2 cache memory per tile division is such that total sets in L2 cache are

uniformly partitioned across all tiles in a sequential fashion as per SNUCA policy SNUCA means static non-uniform cache access policy.

So you have total of 64 tiles are there, now the 16 MB of L2 cache is been scattered across all the tiles, so each type will be having a uniform share and this policy is known as the SNUCA policy. The system use 32 bit physical address, now tile 52 generated a cache miss for an address that has been given and tiled T3 generated an L1 cache miss on address. So 2 L1 cache miss addresses are been given, now we have to find out from which tile you are able to find the corresponding L2 cache mapping, where are these addresses mapped in TCMP.

(Refer Slide Time: 27:29)



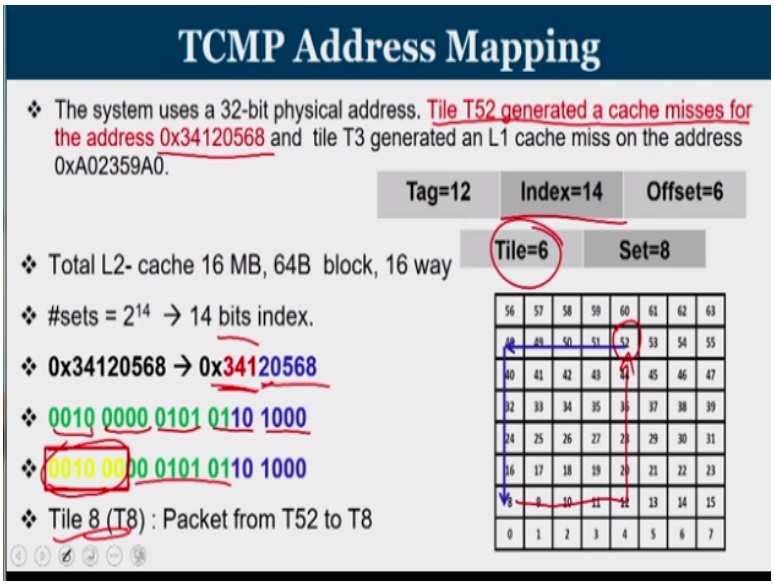
So let us rephrase **the** the most important aspects of the question I am talking about an 8 by 8 mesh NoC. And the property of L2 cache is it is shared distributed L2 cache 16 MB of L2 cache which is 16 way associative and 64 byte block. We are using 32 bit physical address and T52 is generating an address 0x34120568 and T3 is generating another address to 0xA02359A0 where are these addresses mapped in TCMP.

So first is we have to find out the total number of sets in L2 cache, it is 16 MB, so cache size 2 power 24 bytes divided by block size. We are talking about blocks of 64 bytes, so it is 2 power 6 and associativity is 16, so altogether I have 2 power 14 sets that is scattered across 64 tiles. So to represent a set number we require 14 bits index. Now if you look at the address I am talking

about the 32 bit physical address having a 14 bit index this is cache memory principle what we have learned.

So the 32 bit address is divided into 14 bits index and since the block size is 64 bytes it is already mentioned, the last 6 bit is used for offset. So that makes 14 + 6 = 20, 32 - 20 I have a 12 bit tag. Now I have 2 power 12, 2 power 14 sets are there which is scattered across 64 tiles, so the most significant 6 bit will tell you which is the tile number. And the last 8 bits of the set index will tell you within a tile I can host total of 256 sets, so what is a set number.

(Refer Slide Time: 29:17)

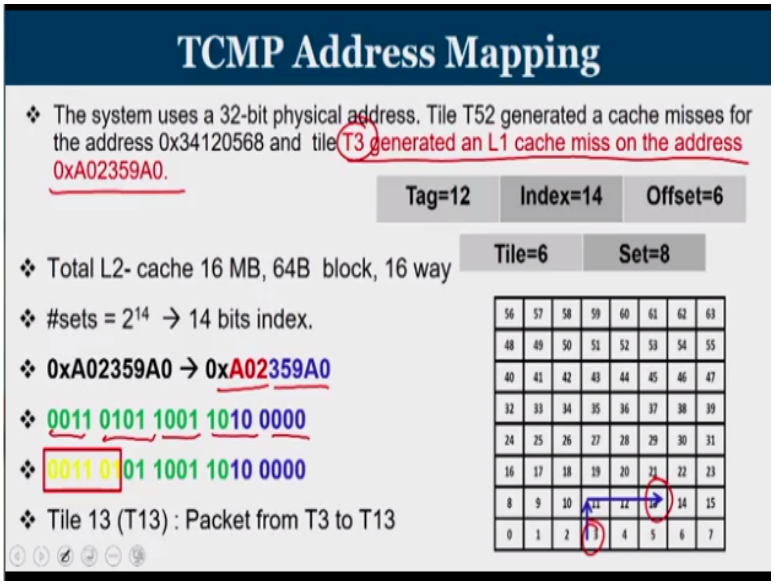


So this is the question, so in this case the system use 32 bit physical address tile T52 generated a cache miss for the address. So it is 0x34120568, so we have already found out the split up of the address. Now this is the address that I am talking about out of which the one that is been shown in the red 0x341 it is a hexadecimal value. So this 12 bits will represent my tag and the remaining 20 bits will represent my index and offset together.

Let me expand what is written in this blue color 20568, so hexadecimal 20010, this is 0, this is 5, this is 6 and this is 8. So the green portion indicates a set number, now in the set index number take the first 6 bit, so the first of 6 bit is 001000 and this is the set number within that tile. So what is seen in the yellow color within this rectangular box that tell you it is tile 8, that means this miss that is generated from tile 52.

So we have an address from 52 which upon looking in L1 cache encountered a miss, that means then you have to go and fetch it from L2 cache but L2 cache it is not a private cache for tile number 52. It is a shared distributed cache, so the request is going from 52 all the way to 8 and what we have found out is it is tile 8. We do not know where it is by looking at that address looking at these 6 bits will tell you where the request is going. And the cache block will be returned from 8 to 52 by XY routing following this path.

(Refer Slide Time: 31:02)



Now tile T3 generated an L1 cache miss on the address 0xA02359A0, the same process is been applied where the red portion indicate tag and the blue portion indicates index and offset bits together I am trying to expand the index and offset bits. So it is 3, this is 5, 9, A and 0, extract the first 6 bits and the first of 6 bits will give you tile 13. So this is the request that started from tile T3 so T3 gave a cache miss and L1 cache miss and this address is mapped into tile 13.

So this is the way how the packet is going, so this particular problem is a classical example of how generally a tiled chip multi core processor works. Generally we have many such processors which are there in these tiles and each of these tile houses a processor a private L1 cache from which the processor directly interact. So the fetching operation happens from this private L1 cache, once in a while you will miss in this cache.

Then we know that when you miss something in L1 cache you have to go to L2 cache and bring it, but L2 is not located in one place. In the case of TCMP architectures the L2 cache is shared and distributed. So from the address few bits in the address the physical address will tell you which tile this particular L2 cache is located or L2 address is mapped. And this policy is called SNUCA static non-uniform cache access.

So divide the address in all these problems divide the address into tag index and offset, the index bit will tell you what are the total number of bits reserved for the L2 cache set but L2 cache set is scattered. So the most significant bits will tell you where is the tile or which tile is been mapped and the least significant the address bits of the index will tell you within the tile what is a set number.

So once you extract this more significant bits of the set index it will tell you the mapping and from that particular tile a packet, an NoC packet is been generated. And in the previous question we have seen how much cycles it will take for the packet to move. So this is a classical combination of the processor the clause and NoC together, that is what we are trying to achieve in this course.

In the advanced computer architecture course how the concept of processor and we started with pipelining, we start fetching from a memory and this fetching can sometime result in a miss and when you miss you how to go to the next level and now the levels are scattered. So whole concept that we learned throughout this course is been used in this particular example. So these are very important, so kindly go and refresh this kind of questions more, with this we come to the end of this tutorial.

So all the tutorial sessions are over, so by this time our entire course which is spanning across 8 weeks, there were tutorial sessions, there were gem 5 practice sessions. So altogether this lecture videos which is complimented by the tutorial session will give you enough exposure in understanding the advanced computer architecture concepts. There are a few more courses in line with this related to multi core architectures, cache coherence and all.

So there will be more moves courses in this direction, so if you find this kind of topics interesting I urge you to continue studying in this fashion, thank you.