

Multi-core Computer Architecture - Storage and Interconnects.

Dr. John Jose

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

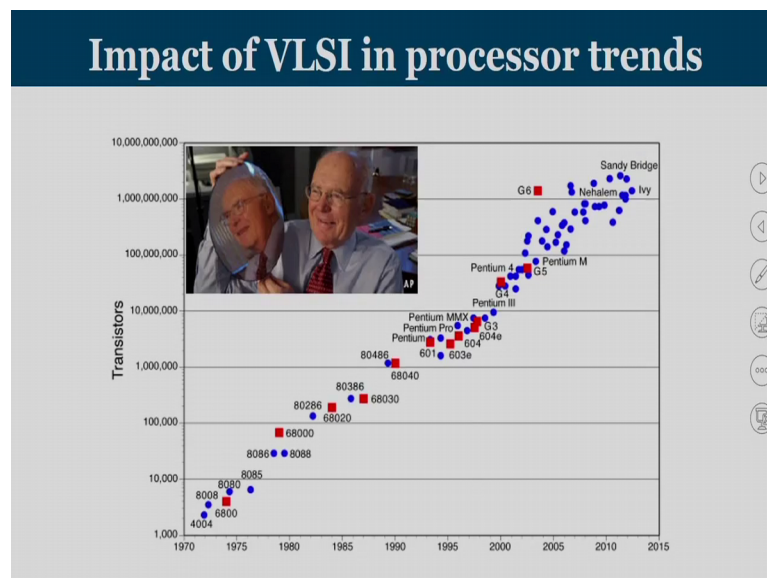
Lecture – 20

Introduction to Tiled Chip Multicore Processors

Welcome to the 12th lecture of the course. So, far we have had enough explanations and discursions on the way by which processors, caches, main memories and some address translation mechanism works. These where the techniques that were there over the last 2 decades; slowly we wanted to shift our attention and discussion into modern multicore processors.

So, today I am going to introduce to the recent multicore processors the recent advancement in processor design. And to start with I will be introducing the concept of multicore processors. What was the need to move from single core processors to multicore processors so, introduction to tiled chip multicore processors.

(Refer Slide Time: 01:31)



We can see that this graph shows the trend of transistors that is available inside a single chip. Gordon Moore predicted that the number of transistors in inside an IC is going to double in every 18 months.

As we can see from this graph over the years, lot of processors were developed and manufactured and you could see all these force are different different processors that came into the semiconductor market. And the number of transistors is plotted on the y axis. So, we can see that the transistors are steadily increasing every year.

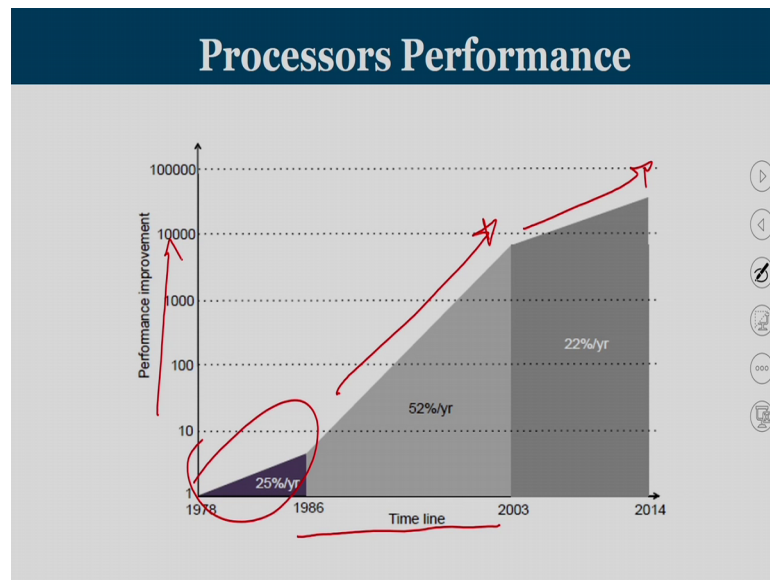
So, what is that why we are able to accommodate more number of transistors in the latest processors; thanks to the effort of VLSI design techniques such that we could squeeze in more transistors in a given space.

The package density of transistors has gone up to very high numbers. Let us assumed that we need 10 transistors to realize a circuit. Now, if there are 100 transistors available inside a circuit then we could realized 10 such units, let us say if it is an adder then we can have 10 adders. If you have still more for example, 1000 transistors available inside a circuit we could realize 100 adders.

So, more number of transistors that will give us the chance to realize or implement more number of functional units, more amount of storage, more amount of control logic or the advancements that we wanted to realize inside a processor is possible when we have more number of transistors available.

So, when we are having enough number of transistors that led to more advanced processors whatever we have seen advanced branch prediction mechanism, superscalar processors, advanced cache memory optimization techniques everything was possible because we could realize more of circuits inside a chip.

(Refer Slide Time: 03:40)

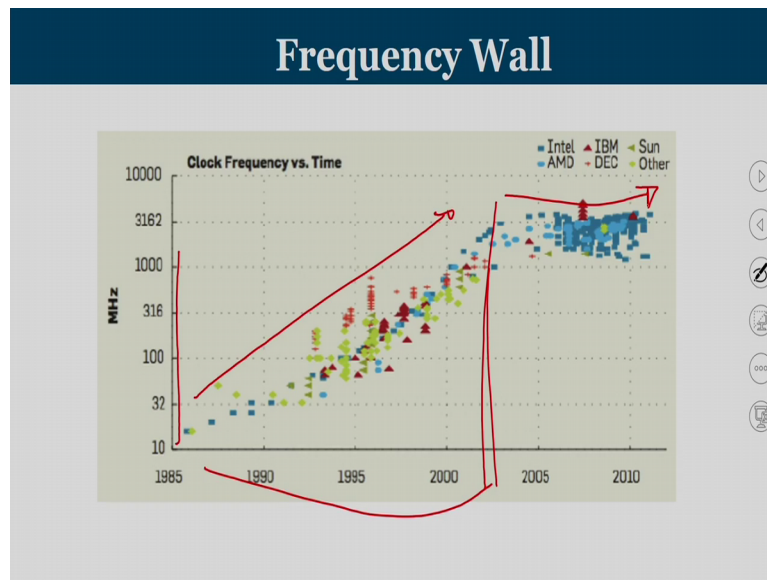


When we look at the performance of processors from late 7 ties up to the recent few years, this graph is going to give us an idea about how it was. In the initial few years if you look at, in the initial few years you can see that the performance of processors that were coming to market in the subsequent years were roughly 1.25 times more powerful than that of the previous generations.

And then we are in an era from 1986 to 2003 where 52 percent was the performance improvement across the previous year. So, that was considered to be the golden period in processor industry. It was happened because designers were trying to exploit instruction level parallelism can we come up with the better decoding techniques, multiple parallel units, super scalar processing all advanced features that we have discussed in the initial few couple of lectures of this course where all led to this performance improvement that you could see during this time. And close to around 2003 to 2005 architects found that almost all the available type of exploitation that we could do inside processors, where almost reaching to a saturation point.

That's why we can see that in the last one decade we were not able to improve much in terms of performance of processors. There are couple of reasons for it, we will try to find out what are the reasons in this context.

(Refer Slide Time: 05:26)



When we talk about any processor, the most important component that comes to a mind a feature that governs a performance of processor is the clock that drives the processor. We could see that in the last 2 to 3 decades the operating clock frequency of processors were linearly increasing from say 10 megahertz all the way up to 1 gigahertz.

This was the trend up to say 2000-2003, but after 2003 we could see that the trend is no longer continuing. Processors operating frequency was more or less around 3 gigahertz. Why is this like this?

The main problem associated with scaling further processor frequency is architects have realized that; increasing the frequency is no longer possible beyond the point. Because, when you increase the processors operating frequency that is going to impact the power dissipation that happens as power is proportional to the frequency of operation.

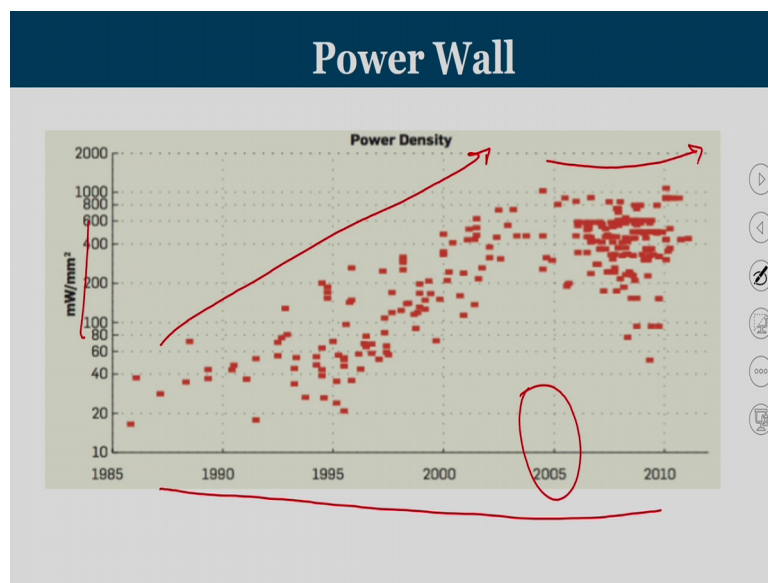
Moreover, we have learned from the pipeline in principle from instruction pipeline in principle that, every stage of pipeline requires approximately 1 clock cycle time. Modern processors are having deeper pipelines with more than 10 to 15 stages and each stage you supposed to complete its operation in 1 clock cycle.

When you increase the processors clock frequency; that means, that automatically the physical time available under 1 clock cycle is very limited, is very much small. So, the amount of work that the processor is supposed to do let us say it is instruction fetching or

instruction decoding, computation of effective address all the some operations that is happening inside an instruction pipeline cannot be completed in this available time.

So, increasing the frequency beyond a point was not possible because of these reasons. This is what we known as frequency wall; meaning we have reached a saturation point or a wall beyond which we cannot go further by exploiting frequency increase.

(Refer Slide Time: 08:02)



Let us look into the other important factor which is known as power wall, we can see it as this is much the y axis is milliwatts per millimeter square. The amount of heat that is getting dissipated over a millimeter square area of a processor chip and these are much is the power density of various processors, that we were coming to semi conductor market in the last 3 decades.

We could see that the trend was over the years the power dissipation of processors was strictly on an increasing line and somewhere around 2005 we could see that, then there was not much increase.

So, that was the time where architects were trying to incorporate many advanced architectural features inside processors like high performing pipelines, would branch prediction strategies, advanced decoding features, multiple parallel pipelines. All these were exploited from around late 90's to early 2000.

Since, more combinational logic or more intelligence is fed into this hardware units, they are going to work little bit more rigorously thereby dissipating more power. So, those processors where relatively consuming little bit of higher end of power that is what we can see.

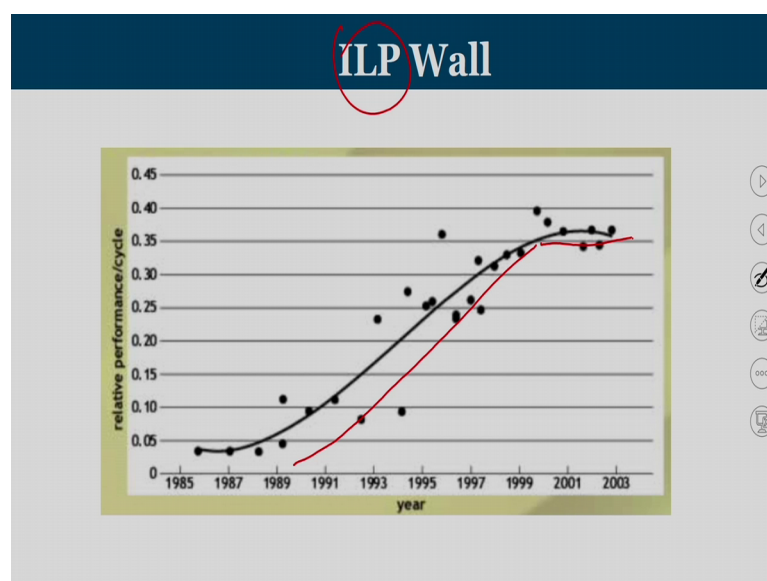
Architects realize that this trend cannot continue, because power dissipation is another important crucial parameter. We cannot afford a processor to dissipate that much heat then the cooling mechanism has to be appropriately competent enough to handle the scenario.

So, they were trying for low power design that is what you can see that somewhere the trend goes down where processors were designed in such a way that they were consuming lower amount of power.

So, the idea of having advanced intelligent units inside processor that will take care of the instruction execution in a more effective way that leads a wall and that is what is known as power wall.

So, frequency you cannot scale frequency wall, power you cannot increase. So, you cannot have more intelligent devices inside processors or intelligent circuits inside processors and then we are trying for the third wall which is known as the ILP wall.

(Refer Slide Time: 10:15)

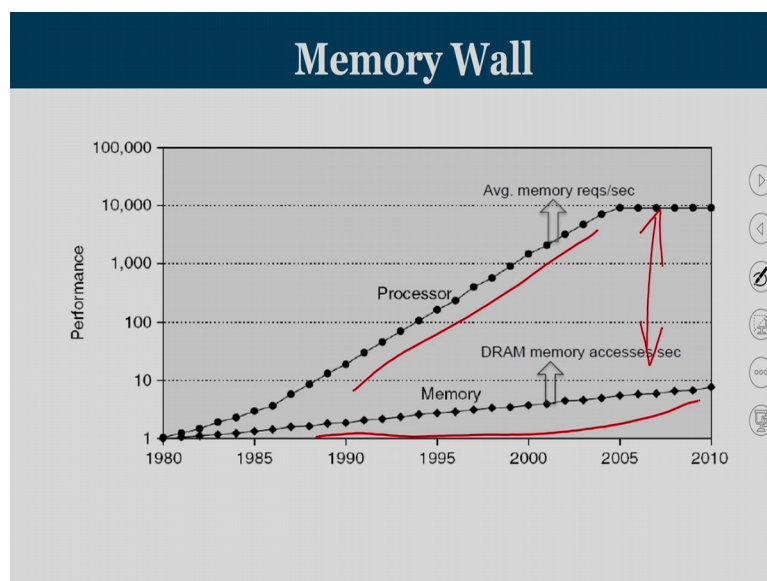


The basic principle by which execution of instructions were improved was through effective instruction pipeline. The throughput of the number of instructions that we say the throughput means the number of instructions that we complete in unit time was increased by using advanced pipelines.

In all these cases architects were trying to exploit instruction level parallelism that is called ILP Instruction Level Parallelism. So, there that also reach the wall everything possible to exploit between instructions can I take care of dependencies, operand forwarding, branch predictions, dealing with structural hazard, everything what we have seen in the case of an instruction that was already fully exploited.

There also we can see that even though we come up with good designs, then we were not able to improve performance of processors by merely exploiting instruction level parallelism also that led to ILP wall, instruction level parallelism wall.

(Refer Slide Time: 11:24)



We have seen that in a 5 stage risk pipeline instruction pipeline out of the 5 stages 2 stages touches memory it is a instruction fetch and the mem stage. When we talk about advanced processors which is operating at higher clock frequency, we have to understand that all the stages of the instruction pipeline also has to scale up accordingly. They have to complete their task in a lesser amount of time if you wanted to operate them at higher clock frequency.

For that we know that the instruction fetch operation and the memory access operation for accessing data, they cannot be done fully inside the processor because, it is trying to access a memory location, trying to access a cache memory.

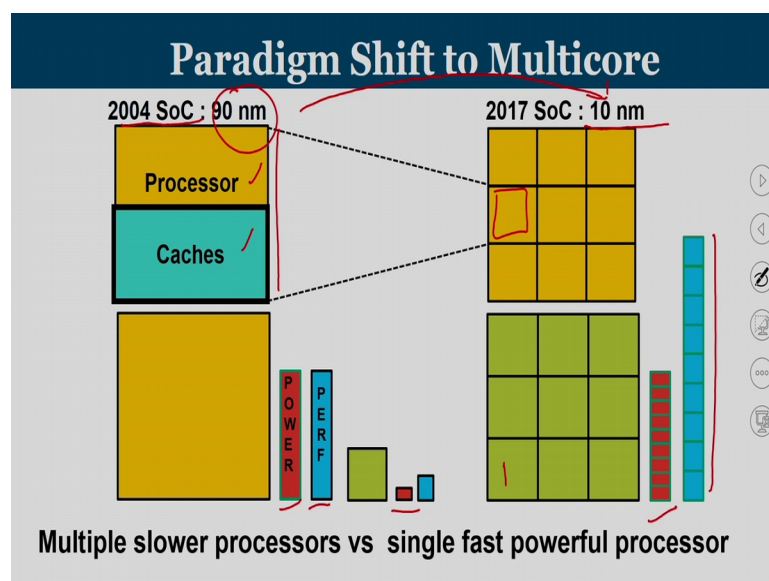
So, memory access technologies has to be equally fast then only processor can achieve performance at higher rate. But, it so happened that processor is built with one type of technology whereas; memory is built with another type of technology. The access speed the operating speed between processor and memory they were widening over the years, this is what we could see.

The trend of processors where developed and performance of processors were slightly at a higher rate. Even though memory technology was also improving the gap was widening.

So, these 4 where finding out some hindrances in the way how growth happened in processor industry and that was the reason that after 2000 onwards we were not able to generate more powerful high performing processors.

Due to limitations in frequency increase, limitations in power dissipation, limitations in further exploiting instruction level parallelism and limitations on memory wall. This led to something known as a paradigm shift to multicore processors what is given in this diagram.

(Refer Slide Time: 13:27)



Is that in 2004 consider a system on chip that is operating on 90 nanometer process technology, let us say this much was an IC which was taking care of processing and caches for a processor.

When I talk about 90 nanometer; that means, it is the process technology by which this chip is manufactured. It is basically a parameter which tells how closely 2 adjacent components are get inside an IC.

So, here 2 components like to transistors are kept as close as roughly 90 nanometer and it is known as a system on chip. Everything that is required to realize a system, let us say it is a processor or a coprocessor, a DSP unit, a timer, an interrupt controller, peripheral interface, serial interface, user all these are put together inside a single chip and that is what is known as system on chip.

Let us say if this much was the space that is needed to realize a processor and it is corresponding cache in system on chip. And 2004 at 90 nanometer technology with the advancement of VLSI technology whatever you could see the same functionality that you could see in 2004 we were able to accommodate in a lower space.

Still, the size of the processors remains same meaning we could accommodate 9 such system on chip devices or 9 such processors and their corresponding cache in the same available space. Roughly, we are moving from 90 nanometer to 10 nanometer. Even though 10 nanometer technology is now not that much ward this is just to give a scaling parameter.

So, a functionality that was available let us say 15 years before, the same functionality could be realized inside a chip which will take only 1 by 9th of the space that it was consuming 15 years before. Again process technologies shrinking, we could realize chips in such a way that adjacent components are kept as close as 10 nanometer.

What is you take away? Assume that let us say this is the size of a processor. Let us say it consists of some 100 million transistors and this was the power graph and the performance well it is a the power dissipated by a processor.

So, since you have a 100 million transistors since there are lot of transistors available, you could realize a extremely high end superscalar processor. But, architects realize that

whatever complex is the processor, whatever advance is the processor there is a limitation to each of these processor. It could handle only one task at a time. Even with superscalar features, there is a limit to which you can do parallelism.

So, to draw your attention to another exercise, do not take all this entire transistors that is available inside the chip. Take only 1 by 9th of it and still we could realize a simple processor using that it may not be of that much high end processor. Still a simple in order processor with a reasonably good features.

Since, I am using only 1 by 9th of a transistor the power consumption is roughly 1 by 9ths of that of the original big superscalar processor. But, the performance is not 1 by 9, it still lower than the advanced processor. But, still it be slightly lower than 1 by 9.

The takeaway is if you could realize a processor with 1 by 9th of the available number of transistors and space, in the same total space I we could realize 9 such simple processors. We have to understand that all these 9 simple processors are kept inside the same chip.

So, now, a chip is going to house 9 such processors. But, when you look at the power aspect it is going to be same because, I am going to use a same number of transistors. And these processors may be working at a slightly lower level. But a performance is going to be slightly on the higher end.

If you wanted to draw a simile or an example to understand this, let us say a farm has a 5 lakhs rupees. Should I employee a senior manager who can takes care of everything by paying him for 5 lakh rupees for 1 month or should I employee relatively junior level officers, let us say 5 of them and paying 1 lakh each?

So, rather than having one sophisticated employee with lot of capabilities versus 5 employees with mediocre capability, since you have 5 employees they could parallely do 5 task and if we can effectively integrate and coordinate among them the throughput is going to increase.

So, the take away that we are going to have from this multiples lower processors versus a single powerful fast processor.

(Refer Slide Time: 18:45)

Paradigm Shift to Multicore

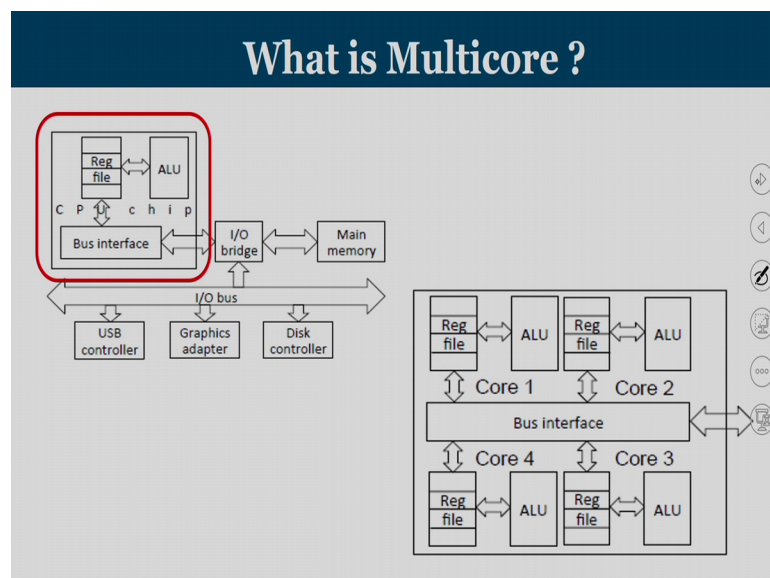


Multiple slower processors is better than single fast powerful processor

Something like this, this paradigm shift can be explained using this phenomenon. Let us say somebody who is joined, who is going for wrestling, can be compared with this is something like the uncore system. Is uncore system versus multicore system, multiple simple processors. So, easily multiple simple processors can beat a single uncore processor.

So, multiple slower processors is better than a single fast powerful processor, that is the take away from this.

(Refer Slide Time: 19:20)

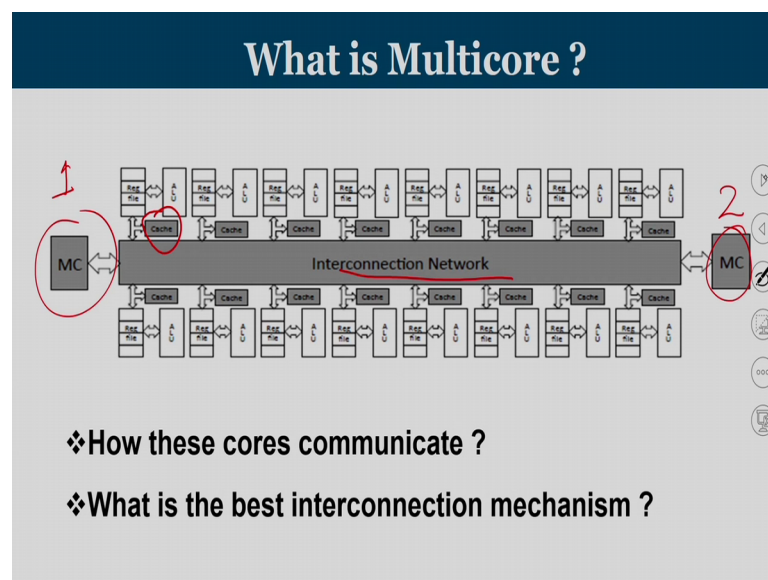


Now, what you mean by multicore? What is scenario of multicore? So, consider a general computer that you see in this diagram now whatever you see in this red square that is what is known as a core.

So, a core consists of the control unit, the registers and the functional units like ALU and 14 point units. Whereas, your main memory associated hard disk peripheral controllers all are going to be outside the core. So, when I talk about multicore whatever you see in the red that is going to be multiplied.

So, this is a design where you have 4 cores 1, 2, 3 and 4 and each of the core has its own set of registers, its own set of functional units, and its own set of control unit we will take care of this. What if I wanted to scale further?

(Refer Slide Time: 20:14)



This is a diagram where in you have 16 cores. These 16 cores are connected by some kind of an interconnection mechanism. And what do you see on the end that is known as memory controllers.

Memory controllers are the point through which the contents from an external memory like D ram is going to enter. We have learned about channels in memory. So, assume your D ram is having 4GB of memory, out of the 4GB; 2GB will be physically connected to one of the memory controller. And the second 2GB is connected to the other memory controller.

So, any data that is belonging to the first half of memory will enter through the first memory controller to the chip. And any data pertaining to the last half of the memory will enter the chip through the second memory controller.

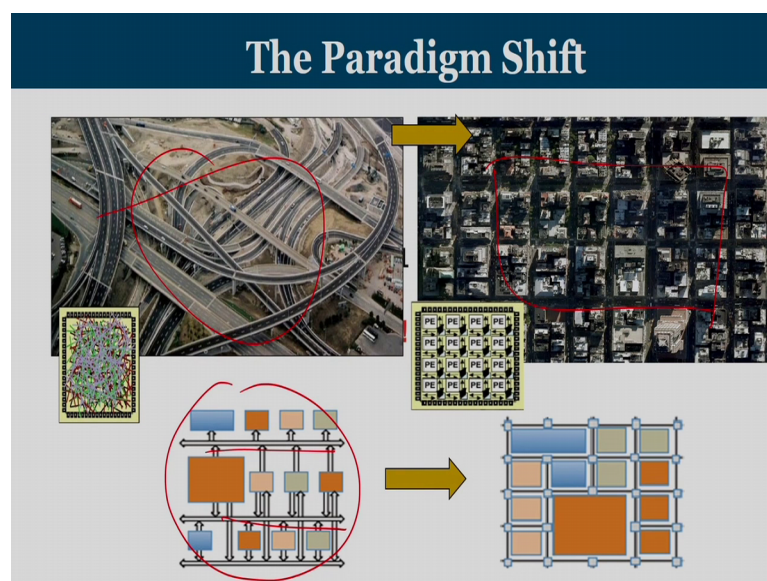
We have seen that in the previous diagram when you have 2 to 3 or 4 cores, cores can communicate each other using buses. That is what we can see it is bus interface. But, when we move into larger number of cores, we can see that buses wants k , we have to do some other interconnection mechanism. And each of the core has having k cache.

So, all 16 cores what is given in this diagram, they have their respective caches. And these caches will be having the frequently used or recently used program and data. And from this cache this processes will fetch and execute the task.

Whenever we encounter a cache miss then you have to go and get the data from the main memory. So, you come to the interconnection network, whatever be the mechanism, reach the memory controllers, go of the chip, get the data and come back, fill up the cache, resume the operation.

Now, the question that we are going to discuss today is how are these cores going to communicate. Since, the cores are not connected with the bus structure, scaling is a big problem in the case of traditional bus based communication and what is the best interconnection mechanism in this?

(Refer Slide Time: 22:28)

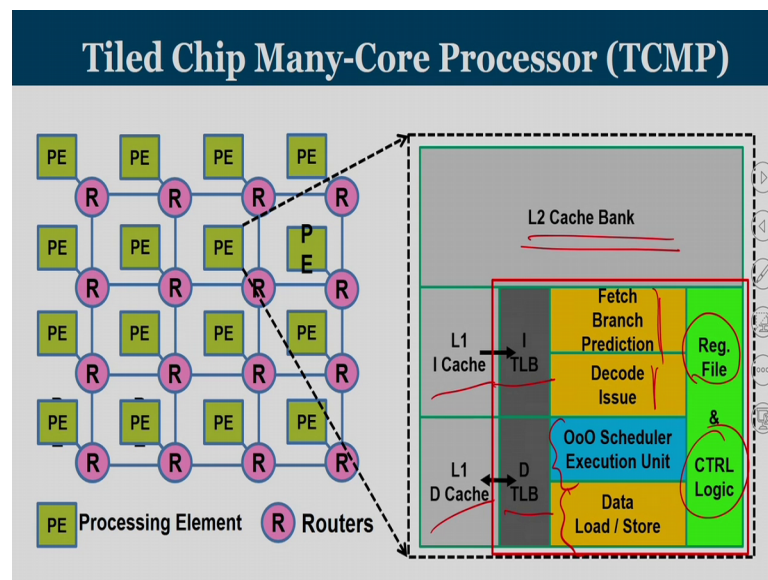


So, we are something like having a paradigm shift, rather than a bus based communication that you see here where various units inside the chip are connected by buses. That is going to be replaced with a well defined network.

So, various portions of a city rather than connecting them using flyovers. So, directly you can have perpendicular layout of roads horizontally and vertically that you can see there in the right side diagram. And then you have your apartments, buildings, office spaces everything in between them.

So, the paradigm shift is moving from a bus based interconnection mechanism in a multicore setup to a network base interconnection mechanism.

(Refer Slide Time: 23:16)



So, this led to the concept of tiled chip multicore processors. So, what you see here, you said we have 16 different tiles. So, the processing elements we call it as PE, Processing Elements we arrange processing elements inside a chip in a matrix format, in a tiled format, row wise and column wise.

And these processing elements will house the processors and its associated levels of cache memories. And these processing elements R connected to R, this R represent routers in this. Let us see what is inside this processing element.

The processing element has the register files. So, each processing element is essentially a processor which has a register file, the control logic, the fetch unit, the branch prediction

unit, decoding and issue out of order scheduling and the execution, memory access, load and data and we have seen what is the role of TLB.

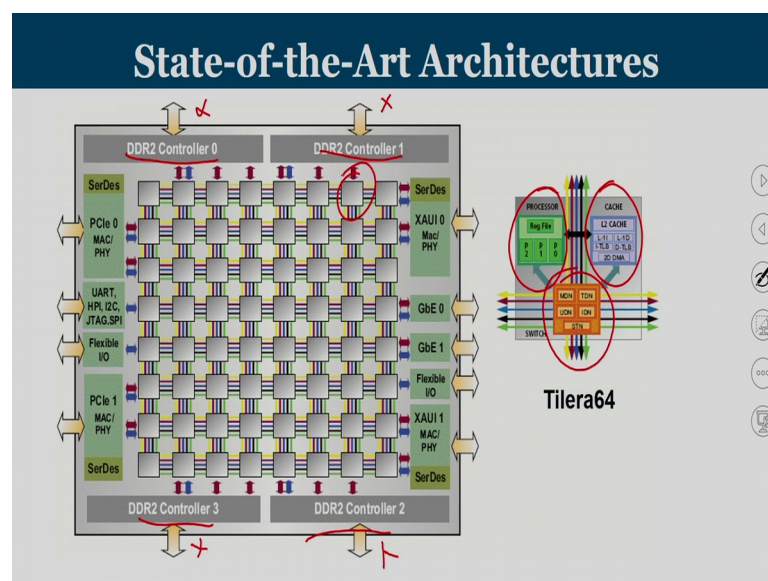
Whatever is a virtual address that is created by the processor, the virtual address has to be converted to physical address. So, instruction translation lookaside buffer, data translation lookaside buffer and this are going to take data from the, I cache and D cache.

We have to understand one more thing here we are using multi level caches. So, each of this tiled will house a processor plus I cache and D cache plus a share of L2. So, L2 for example, in this case let us say your entire L2 cache is divided into 16 segments and each of this tile will accommodate 1 by 16 of the entire cache memory.

So, one set segment if you divide the entire cache memory into 16 segments, 1 set segment is physically located inside a tile. So, in short, when you talk about tiled chip multicore processor. So, tiled chip many core processors, what happens there is we are having multiple such processors organized inside a chip. The fabrication is done in such a way that multiple such processors are house inside a chip.

And each of the tiled has what we see, we have a processing unit which has registers, control unit, fetching, decoding, the execution, the instruction pipeline, the address translation mechanisms like the TLB s plus the I cache and D cache and a slice of L2 cache. This is what you see the processors.

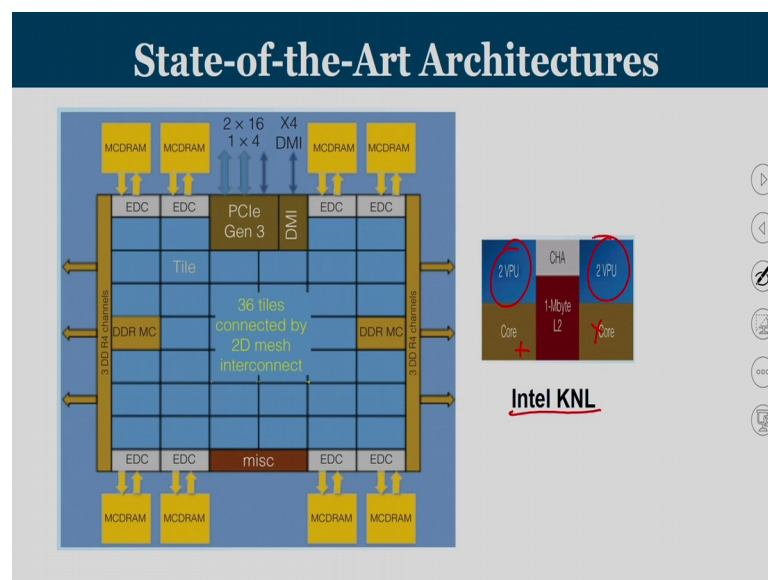
(Refer Slide Time: 25:52)



Now, do we have processors which have this many number of tiles? Yes, we do have. One of them is Tiler64 it is having 64 processors that you can see or rather 64 tiles are there. And each of the tile has a set of the registers that is a green portion will show the processing side, the blue portion will show the cache side and then you have a communication portion. We can see that the memory controllers here it has 4 channels a memory controllers are connected at the edges.

So, the data from the D ram is going to enter the chip through these 4 entry points which are nothing, but channels. And each of this tile is going to house and out of order superscalar processor, a private element cache and a slice of a shared L2 cache.

(Refer Slide Time: 26:52)



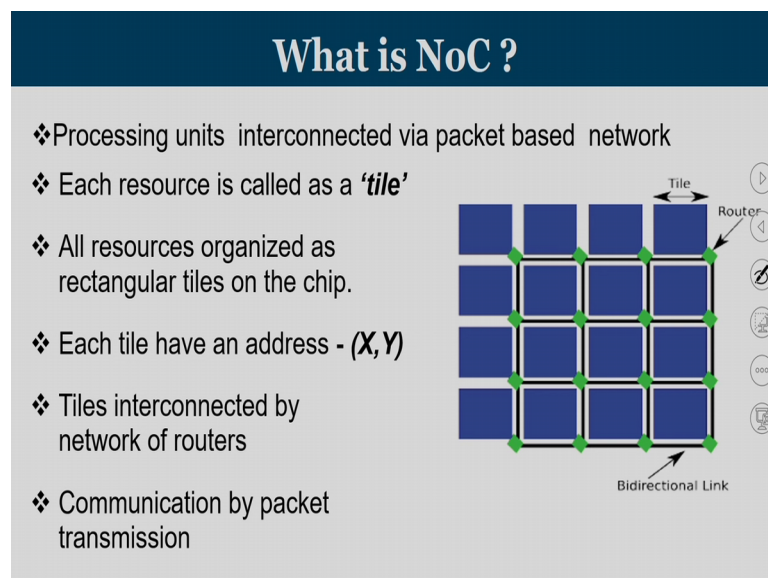
We have one more state of the architecture that is been recently discuss it is the Intel KNL processor, Intel knight landing processor, also known as Intel xeon phi. Where we have 36 tiles organized the 6 rows and 6 columns. Each of this tile can accommodate 2 cores with 2 virtual processing unit. So, tremendous amount of processing power is integrated into this.

So, Intel KNL processor, Tiler64 all these are examples for tiled chip multi or many core processors. Now we are going to understand how these processors work, this tiled chip processors work.

Since, it is already mentioned that all these processors are going to have its own instruction pipeline and cache memories which will take care of the execution of the instructions and accessing of data. At another aspect which we have not discussed so far is about the communication aspect. And that is going to be the discussion for the rest of our course..

The course itself multi core computer architecture storage that we have already seen, how cache memory and main memory works and now we are going to talk about the interconnection, storage and interconnects.

(Refer Slide Time: 28:11)



So, what is going to be the terminology called network on chip? So, this is what do you see it as a tiled chip multicore processor with 16 tiles, whatever you see in blue colour that is a tile. The green portion is the router and that routers are connected by bidirectional links. Essentially these routers are going to be the communication backbone for tiled chip multicore processors.

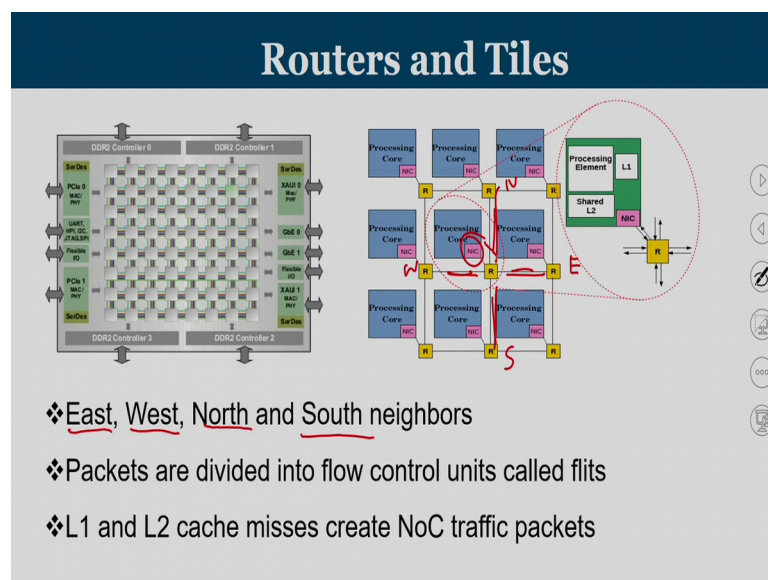
So, what happens in this kind of TCMP's? Processing units interconnected via packet based in network and each of this resource is known as a tile. And tiles are organized this rectangular fashion. And each tile have address x and y and tiles are interconnected by a network of routers. We can see that a set of well organized routers. Routers are connected by bidirectional links and communication is in terms of packets.

So, far for the processes we have seen in the earlier generation processors we have buses that connect processor to memory. And these are all couple of parallel wires which is known it as bus. You put a value in one end of the bus the value will propagate through the bus reach the other side and the data is been accessed.

The concept of bus is totally removed in the case of TCMP's rather than that we are going to have an underlying network that is you have a set of routers and bidirectional link that connect this routers which act as a network inside a chip and that is known as network on chip.

And packet communication happens in network on chip. It is not like bus based communication or signals and you know that what you mean by a packet based communication whatever data that you have and you are going to attach few more bits on top of this data, which is known as a header. The header may content from where the packet start the source address, then the destination address, the sequence numbers, a couple of control information all these are going to be integrated.

(Refer Slide Time: 30:15)



Now, let us see how the routers and tiles are going to interact each other. This is a diagram where you can see a tile consists of processing element L1 and L2 cache it is a 9 tile chip. And we can see that the routers are going to communicates each other. So, your router has 4 neighbors and it is connecting to the local processing core also.

So, we called is neighbors of the routers as East, West, North and South neighbors let us say if you to take this particular router. This is the East neighbor, this is the West neighbor, you have the North neighbor and you have the South neighbor and it is connected to the local also. So, that is the fifth one is the local.

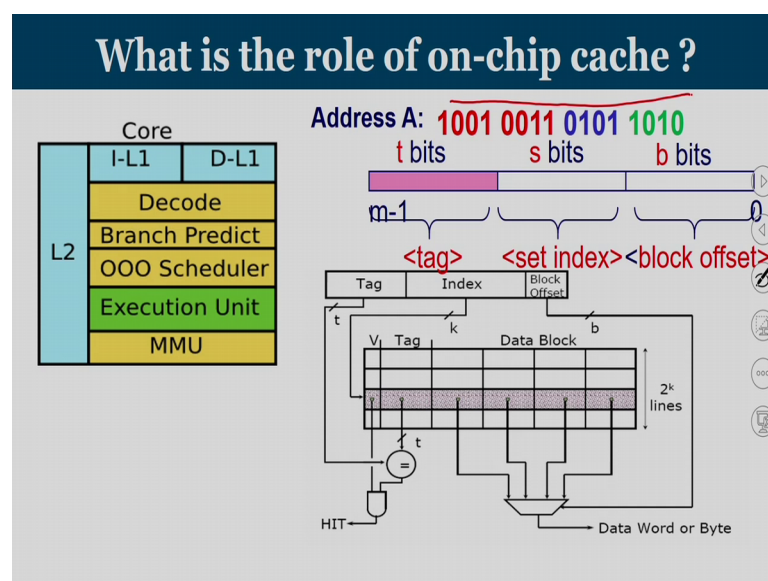
Now, a data that need to be send from one tile to another that is known as a packet. Let us say if the data is 512 bits, I may not be able to send all the 512 bits together. It depends upon what is a bandwidth between a pair of routers.

So, packets are further subdivided into flits and flit is the basic unit of flow control between a pair of routers. So, if there accesses 128 buyers in one direction between a pair of routers; that means, I could transfer 128 bits from one router to another.

So, if the data that is to be transmitted from one end of the tile to another end of the tile, for one from one tile to another tile is 512 bits and my in the router bandwidth is 128 bit. This 512 bit of data that is your packet has to be divided into smaller flow control units called flits.

So, essentially we required flow or flits to transfer this data. Whenever there are L1 and L2 cache misses that is going to create packets in the NoC. So, NoC traffic in a typical TCMP is generated whenever there are L1 and L2 cache misses.

(Refer Slide Time: 32:16)

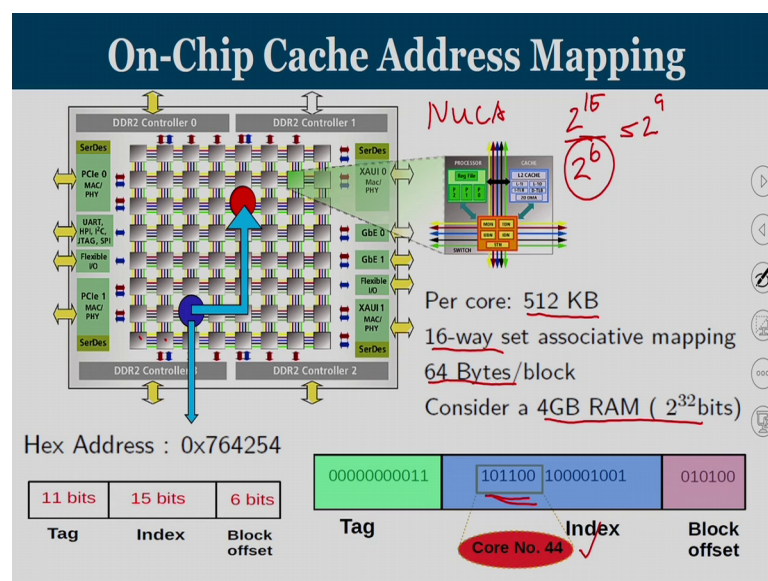


Now, what is the role of on chip cache in the case of a TCMP? We have seen that whenever processor is going to give an address, the address is going to be divided into tag, index and offset. Using the index with you go into the corresponding cache memory. Perform tag comparison and once if it is a hit you extract the corresponding word with the block of set that is been given.

Now we have to understand when you come to a tiled chip multicore processor cache memory is not located in one tile. Cache memory, the L2 cache is fragmented across all the tiles. So, when I talk about a set number N.

It may be present in tile number 1; it may be present in some other tile also tile number 2 or tile number 10. So, this is called address mapping where a set is located. So, I will introduce you to the concept of address mapping.

(Refer Slide Time: 33:17)



On chip cache address mapping. So, consider the case that you are talking about TCMP where per core there is 512KB of L2 cache. Let say the L2 cache is 16 way associative and 1 line cache line is 16 bytes. We have a total of 4GB ram; that means, total of 2 power 32 is the location or so, 32 bits is the address.

Consider the case that if you have 512 KB of L2 cache per core and which is a design with 64 cores. So, 512 KB into 64 that will give you 32 MB of cache 32 MB of L2 cache is available inside this chip and that 32 KB is divided across 64 tiles.

So, 1 tile is going to hold only 512 KB of data. Upon division you come to know that there are 11 bit set for tag, 15 bits for index and 6 bit is for offset because, we are going to use 64 bytes block so, it is 6 bit offset. So, this 15 bits so, you have total of 2^{15} sets. These 2^{15} sets are divided across 64 tile. So, 2^{15} divided by 64, 2^6 .

So, if you work it over 2^{15} sets are there and that is to be divided across 64 tiles that will give you 2^9 sets. So, each of the tile is going to have 512 sets each.

So, sets 0 to set 511 will be in tile 0, set 512 to set 1023 will be in tile 1, like that seen linearly this going to arrange and this is known as static NUCA, non uniform cache access. There is a static address mapping that is being that.

Now, consider your processor is going to encounter and L1 cache miss from this core which is marked as. Let say this is the address that processor is looking for hexadecimal 0X 764254; it is a 32 bit value. If it is not 32 you can add up zeros on the left side.

So, this 32 bit value we have to divide into tag, index and offset. Once you divide them into tag, index, there are 11 bits for tag, there are 15 bits for the index, that will shown in blue colour and there are 6 bits for the offset.

Now, the entire number of sets there are total of 2^{15} sets these sets are linearly arrange across the tiles. Take the most significant 6 bits why 6 bits? Because, you are having 64 sets so, the most significant 6 bits will tell where is this particular set number mapped. So, the mapping is 24 the by the decimal value of 101100 is 44; that means, whatever is marked in the red that is tile number 44, the address 0X764254 which was missed by this blue core is mapped into the L2 cache of this red core.

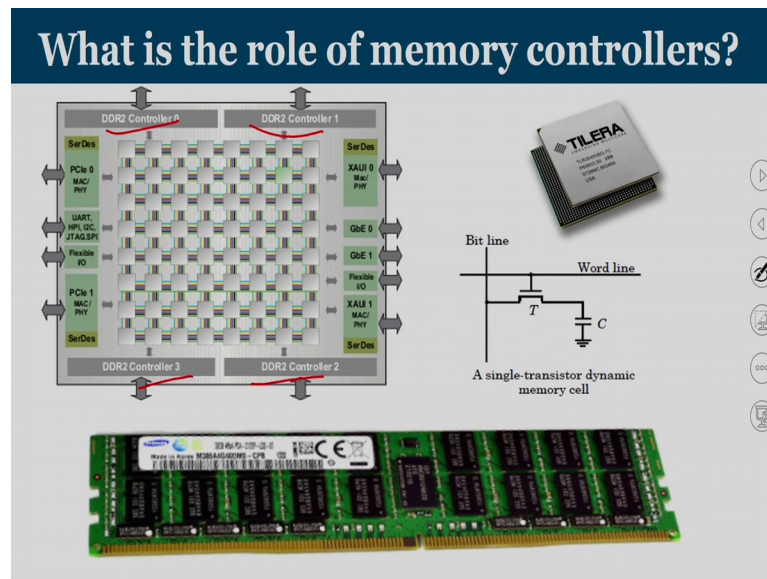
So, a packet has to be generated from blue to red and then it has to be brought back to the blue for the processor to work. Similarly, you have to understand that all the 64 core processors are going to generate addresses for instruction fetch and they will take it from the IL1 cache and smoothly work.

Whenever there are cache misses, using this address split up the cache controller of the tile will find out where this address is mapped and once the address mapping is there you have to generate packet into that tile. How will this packet reach that tile? You have to

create a header with source address, destination address and many other control fields are put and then this packet is going to travel through the network by hopping through these individual routers.

In this way a packet will start from one end of the router.

(Refer Slide Time: 37:37)



And it will reach the other end and what is the role of these memory controllers that you see in the TCMP. This memory controllers are gateway for the main memory data to enter into the chip.

(Refer Slide Time: 37:48)

Packets & Flits

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

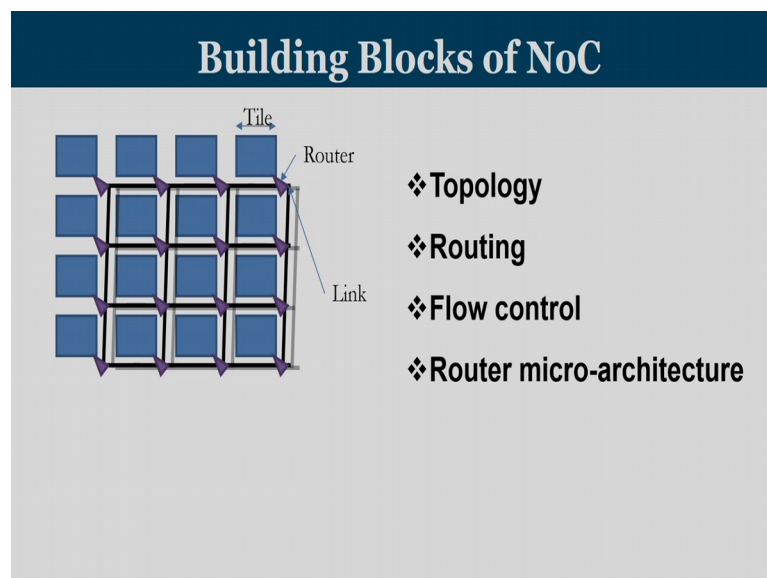
Packet

Head Flit Flits Tail Flit

So, what do you mean by packets and how does it differ? Packet is a basic unit of transfer in the network, where as flit is the basic unit of transfer between a pair of routers and flit is a basic unit of flow control within the network. So, data is moving from one router to another not in terms of packets, but they are in terms of flits. The packet is divided into smaller flow control units called flits.

So, we have head flit and then we have a set of body flits and the last body flit we typically call it as tail flit. In a case of a normal worm holed routing that we will going to see what kind of routings are there. All the flits of a packet will travel through the same route in a traditional network on chip. We will see about variations after a couple of lectures.

(Refer Slide Time: 38:41)



Now, we will see what are the basic building blocks of a network on chip. A traditional network on chip study involves study of the topology of the network, then routing, flow control and router micro architecture.

(Refer Slide Time: 39:02)

Building Blocks of NoC

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

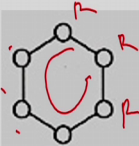
Now, what do you mean by topology? Topology specifies the way switches are wired, how you connect various switches or various routers. And routing algorithm tells how a packet move from one tile to another. And buffering in flow control basically deal with what is a basic unit of transfer and how handshake signals are been passed from one router to another.

To start with today will try to understand what are the basic topologies.

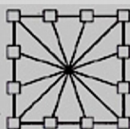
(Refer Slide Time: 39:34)

Topology

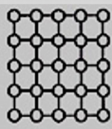
- ❖ Determines the physical layout and connection pattern between nodes and channels in the network.



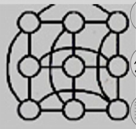
Ring



Spidragon



2D-mesh



2D-torus

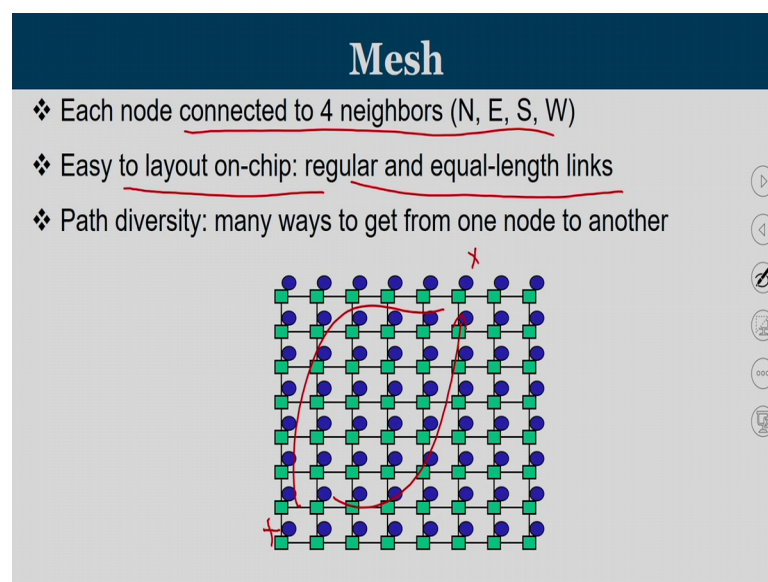
So, topology determines the physical layout and connection pattern between nodes and channels in a network. We can see that there are different topologies given in the slide. It is a ring topology that you see; in a spidergon topology it is a 2 dimensional mesh, 2 dimensional torus. Will have a quick overview of what are these topologies all about.

So, ring topology like what they it is diagram shows whatever you see in the circles these are all routers. So, all the routers are connected by a ring. So, the data flows from one router to another and each router has exactly 2 neighbors. And data flows from one of the neighbor reaches you and then you will forward the data to other.

Now, we can consider this is a graph problem. How the connection between the routers it is based upon the path that exist and the ring network has its own merits and demerits. The cost is very less because a number of channels required to connect a set of routers is very less. But it takes lot of time to reach from one end of the ring to another provided the number of routers is very large.

So, we will try to understand how all these things work.

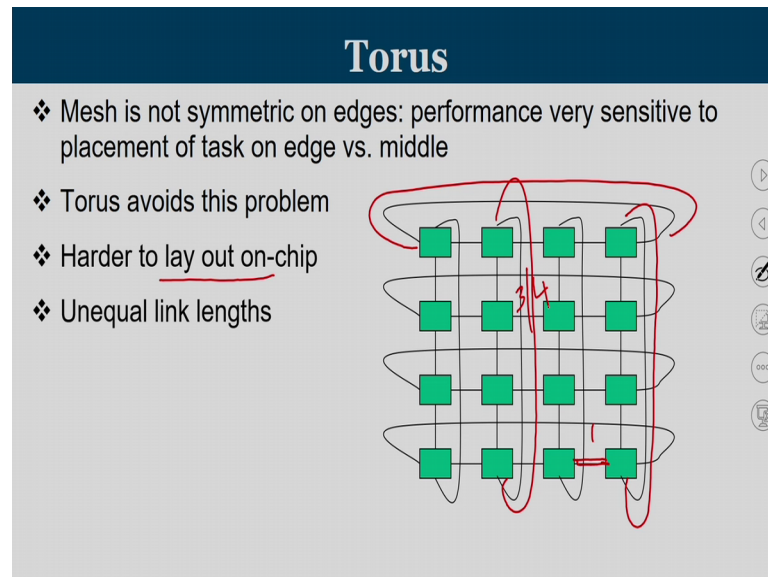
(Refer Slide Time: 40:57)



First topology is mesh. So, each of the router that you see in the diagram this blue circles indicate the tiles and the green squares indicate the routers, the green routers are connected to each other. So, each node is connected to 4 of the neighbors North, East, South and West. It is very easy to layout in this case because it is a planar structure.

So, we have links, regular equal length links are going to connect these routers and their accesses lot of paths from one router let us say from this router to another router there exists different paths are there. So, you can exploit path diversity.

(Refer Slide Time: 41:38)

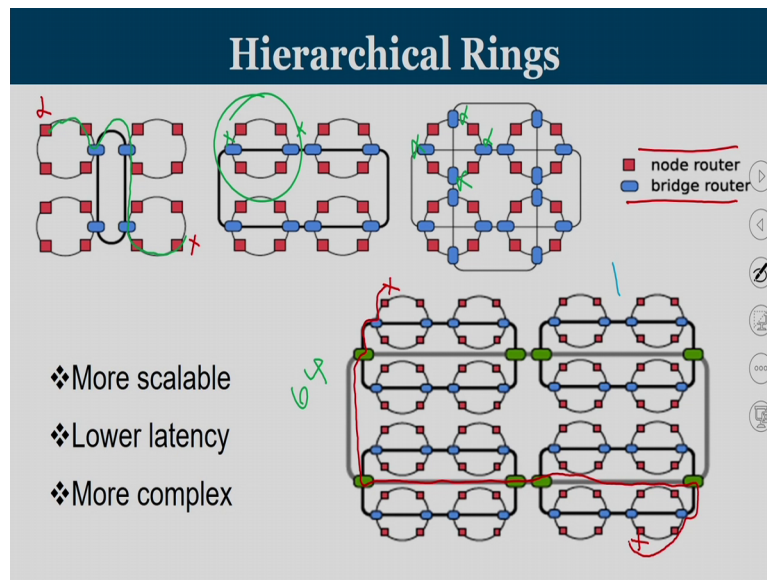


Another topology is known as torus because, mesh is not symmetric on edges because, in the case of mesh when you go to the edges they do not have 4 neighbors.

Here the symmetric concept is introduced every router has 4 neighboring routers and each of them will be connected to a local processing tile. So, but the layout is bit tricky here and some of the links are very long there exist a long wire connection from one end to the chip to another. So, we have short length links as well as long length links.

So, short length links may take one cycle whereas long length link may take 3 or 4 clock cycles. So, torus and mesh are basically same only thing is there exist an end-to-end connection from one end to the chip to another. Let us now we try to understand hierarchical rings.

(Refer Slide Time: 42:33)



So, what you see here is a conventional mesh topology and within that you have 16 routers what do you see with this red squares. Now, rather than connecting them with mesh fashion, connecting every router to all the 4 neighbors, here 4 routers inside a quadrant they are connected by a ring.

So, you have 4 rings inside the chip and the rings are connected by a second level of router and they are known as the blue colored one, they are known as the bridge router. So, you have node router which are connecting to your processing tiles and you have bridge router that connects between rings.

So, if there is a communication that is needed from one router to another their part of the same ring then they will move through the same ring. If the source and destination let say this is the source and this is a destination. If you are part of the 2 smaller rings you travel through the source ring, reach the bridge router. Then go through the hierarchical ring, reach the destination bridge router. So, the path is something like this I will try to illustrate this path moment.

So, when you have a case where there is a source router and destination router part of different links, let us say this is the source and this is going to be the destination, then the packet first travels through the ring, reach the bridge router, travels through the bridge router, reaches the destination ring and then finally, it reaches this.

But, the problem with this kind of hierarchical ring is there exist one bridge router per smaller ring. So, sometimes you have to travel all the way in the ring in order to reach the bridge router. You can realize the design in such a way that there exist 2 bridge routers per smaller ring or 4 bridge routers per small ring. In all these cases there are only 16 normal routers and there exist say 4 bridge router, 8 bridge router or 16 bridge routers. So, bridge routers are going to add complications in this.

Now, consider your scale into 64 tile, if you move to 64 tile then there you are trying to introduce one more level of hierarchy. So, this much portion is your normal 16 tile. Similarly, we have 16 into 4. So, there are going to be 4 such regions inside your tile. This is the third region and the last one is going to be your fourth region.

So, 4 regions are there, any communication within a region. So, it is region 1, it is a region 2, region 3 and region 4 any kind of communication that is happening within a region is using the first level of bridge router, the blue bridge router.

But, if you wanted to communicate from one region to another, first from a normal router you catch up the blue router, let us bridge router and then you go one more level up and you enter the green router that is called hierarchical ring.

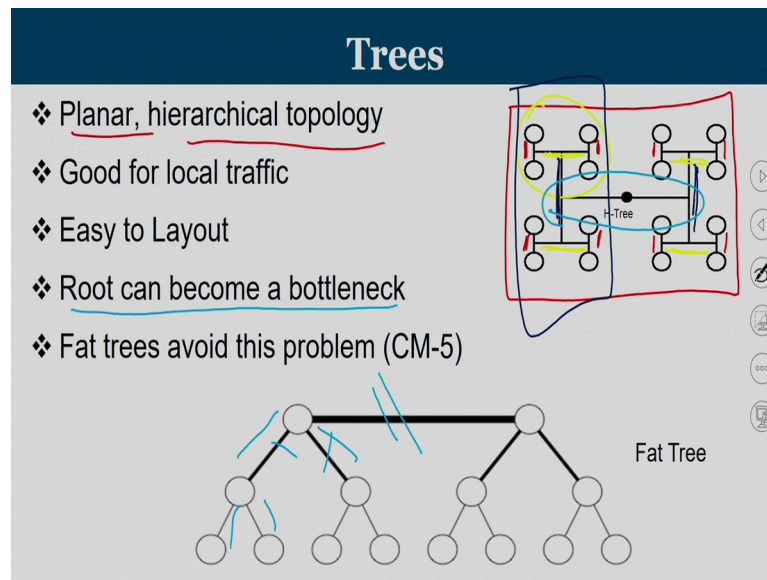
So, to understand what is the power that is being consumed by a hierarchical ring let us take a case of a simple example, let us say there is a packet there is going to travel from here all the way up to this router.

So, first is travel through this smaller ring, reach the bridge router. Now, you reach the first level and then you reach the green router. Travel through the green router through the second hierarchy, reaches the destination green router move through the blue and then at the end you reach.

So, some kind of communication we need only the basic ring, sometimes you travel through the next ring and in the worst case you travel to the third ring. But, fabrication of the third level hierarchical ring is going to be slightly tricky. Efforts are put by architects to realize this the thing in future multicore processors. It is all about hierarchical ring.

Yet, another structure is known as trees, trees are planar.

(Refer Slide Time: 47:13)



They have hierarchical topology as well and you can see that the entire 16 tiles can be organized like this. These are the 16 routers the circles indicate the routers.

So, rather than a complete mesh connection between every pair of neighbors there exist a very short connection between adjacent routers, this is what you see short interconnects. Now, you have a slightly longer type of an interconnect between you are next set of neighbors. So, there may take one or two clock cycles more.

So, if the communication is within this range then it will be of short distance link that is like communicating between the siblings of a same tree. Now, if the communication happens to be inside one section or region we may have to use slightly longer link that is going to connect. And if it is from the 2 ends then still we have to understand that there need to be a separate path or we have to move through the root of the tree.

But, root can create a single point bottleneck and this can be avoided by having fat tree, root this is like a slightly wider highway, you have multiple such parallel tracks, it is just like 2 cities are connected by a multi line highway. So, many parallel tracks are there, many vehicles can move parallelly.

Also you move into the smaller hierarchies the width of the channel is going to reduce and once you move into the children the last level hierarchy it is going to be your normal flit bandwidth.

So, today we have seen different kind of topologies like mesh where you have 4 neighbors and a local processing tile is going to be connected. And then we have the torus topology where the end routers are connected by a longer link. And we have seen about the rings, hierarchical rings, first level, second level and third level rings. And the last topology we studied was about trees and fat tree structures.

To sum up today's discussion we were trying to understand what led to the paradigm shift from uncore machine to multicore machines. We have seen that multiple slower processors are preferred over a single powerful processor. And traditional bus based communication which used to connect from one end of an IC to another is no longer scalable.

So, a network was introduced inside a chip with routers at regular intervals connected by short length links and that is what is known as network on chip. And we have seen that there is packet based communication rather than bus based signal link and packets are having headers, head flit packets or head flits, body flits and tile flits and they are going to be neatly connected.

So, there are basically 4 building blocks for an NoC, first one is routing, then you have topology, flow control and router micro architecture. Today we discussed about various topology takes possible. So, with that we conclude today's session. If there is any queries regarding this feel free to post it into the forum, will be happy to answer your queries.

Thank you.