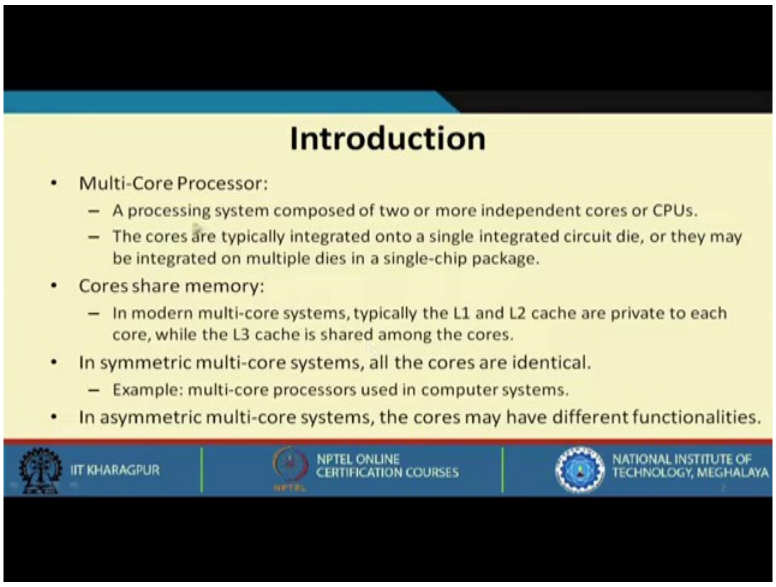
Computer Architecture and Organization
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 62
Multi – Core Processors

So far we have seen how we can make our processor faster by using concepts of pipelining, superscalar, VLIW concepts. These are become very standard concepts nowadays. Whenever you buy a computer system with an embedded processor, you will see invariably these technologies built into that as a standard feature. Now the question comes if you look at the modern processors that goes inside our desktops and laptops and also our mobile phones today, you have heard about multi-core processors.

So, what it is? Multi-core essentially means I have the capability of fabricating large systems on a VLSI chip; I am fabricating more than one processor inside the chip, which is very roughly the concept of multi-core. The question arises why we are actually going for multi-core, let us try to address these issues in this lecture.

(Refer Slide Time: 01:45)



Introduction

- Multi-Core Processor:
  - A processing system composed of two or more independent cores or CPUs.
  - The cores are typically integrated onto a single integrated circuit die, or they may be integrated on multiple dies in a single-chip package.
- Cores share memory:
  - In modern multi-core systems, typically the L1 and L2 cache are private to each core, while the L3 cache is shared among the cores.
- In symmetric multi-core systems, all the cores are identical.
  - Example: multi-core processors used in computer systems.
- In asymmetric multi-core systems, the cores may have different functionalities.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

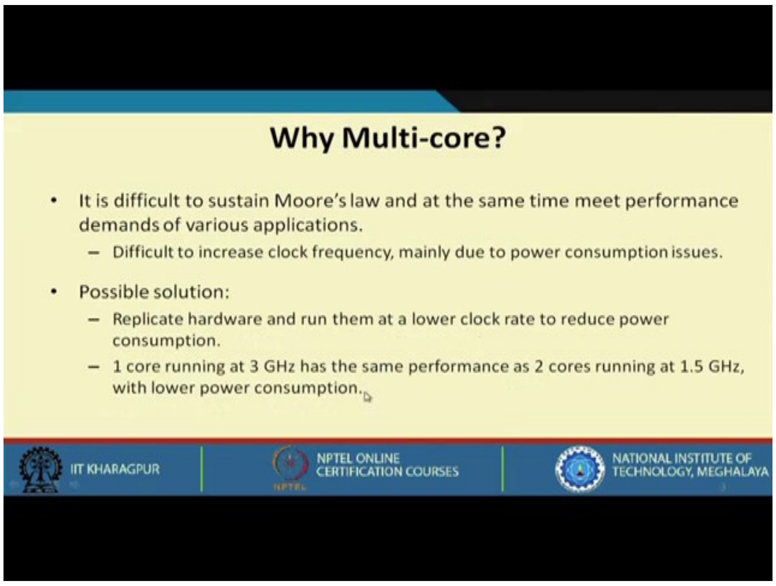
Our topic of discussion is multi-core processors. As I said a multi-core processor is nothing but a system, which consists of two or more CPUs. CPUs are typically independent, they are not strongly connected or they are not dependent on each other. They can independently execute different independent programs if required.

So, for a multi-core processing system we have two or more independent cores or CPUs. These cores come inside a single package, you can buy a core i3, i5, i7 processor from the market. Inside the same package there will be multiple cores, 2 or 4 or 6 whatever. Now these cores or processors are all fabricated inside the same die within the chip. There are two alternatives. You can either have a single die; die means a silicon wafer where the chips are all fabricated. Or you can have multiple dies that are connected or bonded together inside the same package.

There are multiple cores inside the same package, you buy a VLSI chip inside which there are multiple cores or processors. Now typically speaking if you look at the modern processors, the level 1 and level 2 caches are private to each of the cores; however, the level 3 cache is common and is shared by all the cores. These are symmetric systems where all the cores are identical, like the core i3, i5, i7 multi-core processors that we use in our processor systems. These are called symmetric – they are all identical. But in asymmetric multi-core systems, the cores may have different functionalities.

You think of the multi-core system that goes inside your mobile phones. You have heard of quad-core, octa-core processors, but there is no reason to believe that all those 4 or 8 cores are identical; they have some specific purposes. Some of them may be specific DSP cores, some of them may be specific processor cores like ARM, and some of them may be doing some specific functionality.

(Refer Slide Time: 04:57)



Why Multi-core?

- It is difficult to sustain Moore's law and at the same time meet performance demands of various applications.
  - Difficult to increase clock frequency, mainly due to power consumption issues.
- Possible solution:
  - Replicate hardware and run them at a lower clock rate to reduce power consumption.
  - 1 core running at 3 GHz has the same performance as 2 cores running at 1.5 GHz, with lower power consumption.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

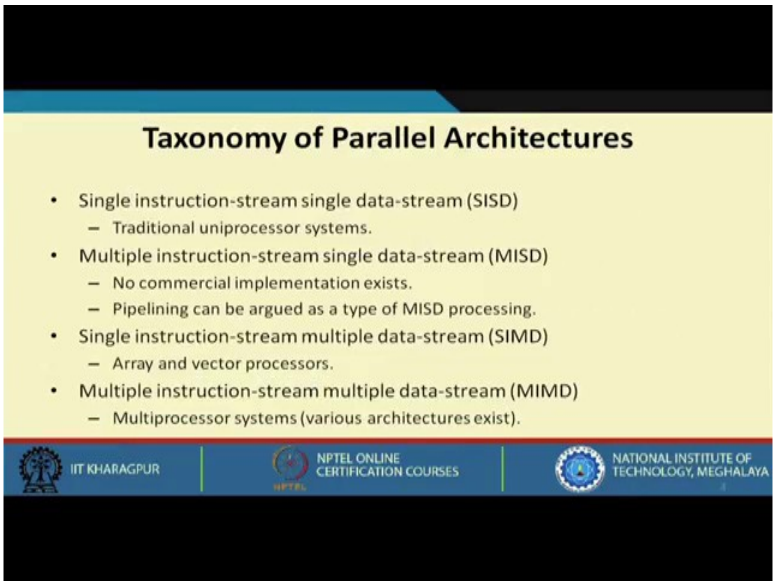
They may be running at different clock frequencies, different instruction set, and different capabilities; that is called asymmetric.

So, why we go for multi-core? You have earlier heard about Moore’s law. Moores law says that the number of transistors or gates that you can put inside a single chip will grow exponential with time; will double approximately every 18 months or so. Now, Moore’s law somehow has been sustained till now, but regarding the performance demands there have been some constraints. Like although you can potentially increase the frequency of the clock, but you are unable to do so, why? Because your chip power consumption directly depends on the clock frequency, and if you increase it your power consumption can become unmanageably high and your chip might be burnt out.

So, you will have to limit your clock frequency within a manageable level, simply to tackle the power consumption issue. There are two possible solutions you can think of. The first solution can be to replicate hardware. You used two computer systems that are running side by side, and run them at a lower clock frequency; like for example, just compare a core that is running a 3 GHz clock, and two cores running at 1.5 GHz clock. Both will be having the same performance, but the second one will be having much lower power consumption because power increases in a super linear fashion with increase in the clock frequency. That is why you will see that in the modern generation of processors the number of cores is increasing.

I mean over the last ten years or so, the clock frequency has not increased much, 1.5 to 2.5 GHz it still remains within that level.

(Refer Slide Time: 07:22)



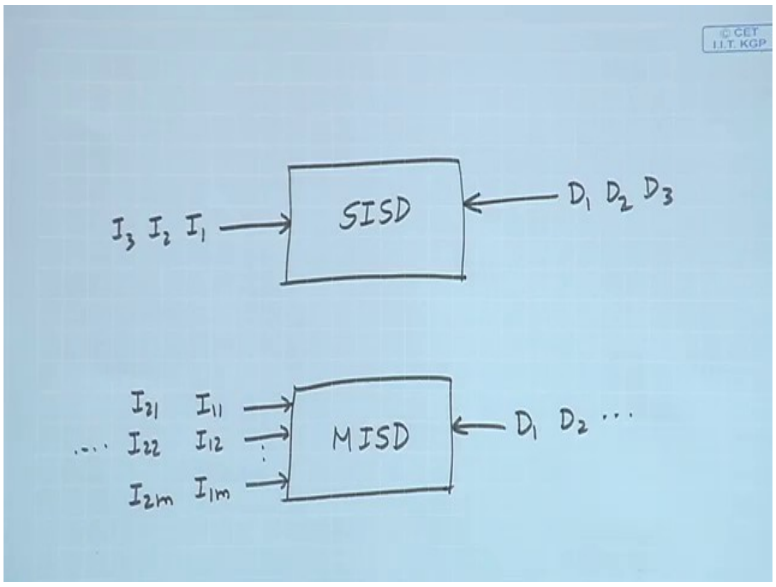
Taxonomy of Parallel Architectures

- Single instruction-stream single data-stream (SISD)
  - Traditional uniprocessor systems.
- Multiple instruction-stream single data-stream (MISD)
  - No commercial implementation exists.
  - Pipelining can be argued as a type of MISD processing.
- Single instruction-stream multiple data-stream (SIMD)
  - Array and vector processors.
- Multiple instruction-stream multiple data-stream (MIMD)
  - Multiprocessor systems (various architectures exist).

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Talking about parallel architectures, a very rough classification called Flynn's classification goes like this. We categorize depending on the instruction streams and data streams. First is representative of the traditional single processor or uniprocessor systems.

(Refer Slide Time: 07:59)



This is called SISD. This means I have a computing system where single instructions are coming one by one I1, I2, I3 like this. These instructions you are operating on data, data are also coming sequentially one by one D1, D2, D3.

Suppose I1 is working on D1, I2 is operating on D2, I3 is operating on D3 like that. This is called single instruction stream single data stream (SISD).

The second classification is MISD. It says that I am feeding several instructions let us say I11, I12, I1m together; then I feed I21, I22, I2m and so on.

But it says that they are working on single data streams. Like these m instructions are working on single data. Conceptually speaking it is a little difficult to visualize this kind of computation where we have a single data, but multiple instructions are working on it. Strictly speaking, no real systems are there that can fall under this MISD category. But some people argue that pipeline processing falls under this category. Let us say in the MIPS32 pipeline there are 5 stages. You can say that there are 5 instructions that are executing in different stages at the same time; as if there are multiple instructions. They are working on the same data set; it is not individual data they are calling a data set. For a program it take some input data it produces some output. That input data you can argue that it is a data set. Like this you can argue that pipeline processor can be considered as MISD.

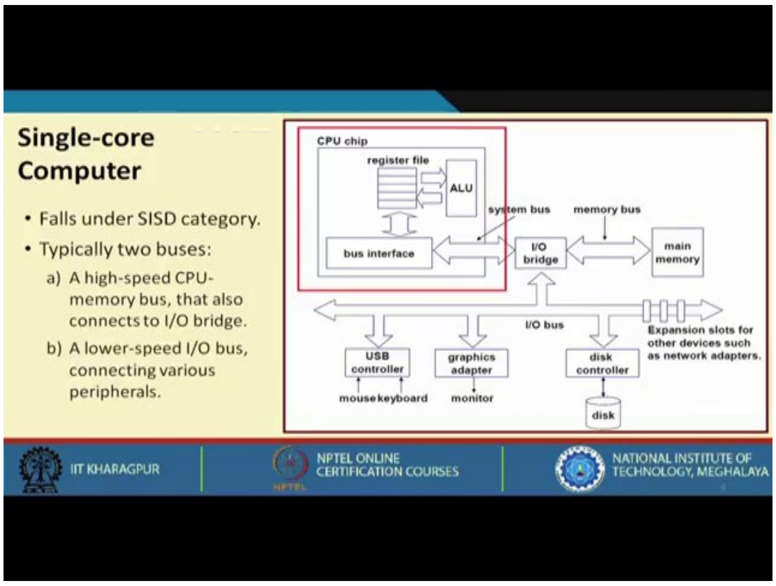
And single instruction stream multiple data stream (SIMD) is something that you can correlate with vector processors. There a single vector instruction operates on all the elements of a vector.

There is another kind of architecture that we do not see that much today, vector processor much more predominant. There were some computers called array processors, which worked on arrays instead of vectors of numbers. You could arrange the numbers in the form of an array, typically two-dimensional. But nowadays you do not see that kind of architecture anymore.

Lastly there are parallel processors, this is MIMD. The multi-cores fall under this category. There are four cores, four instructions can be executed at the same time, and they might be working on four different data.

So, broadly speaking this is one way of classifying computer systems depending on instruction or data stream parallelism.

(Refer Slide Time: 12:06)



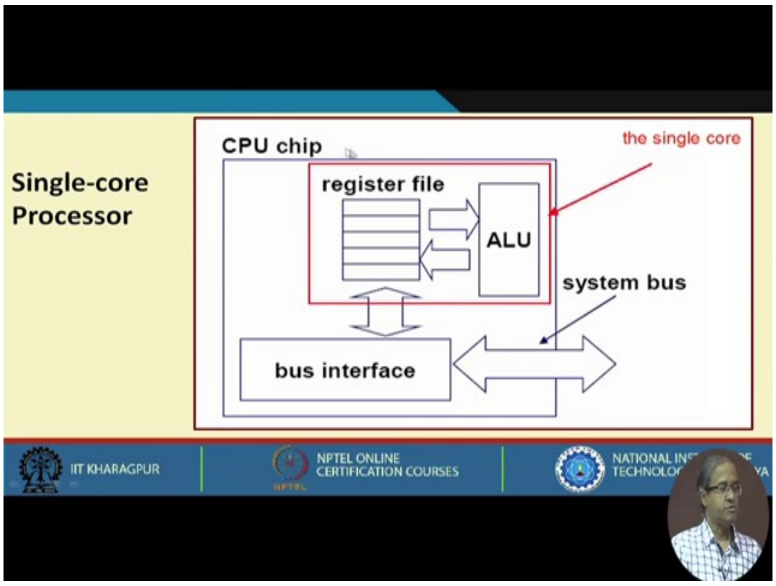
Single-core Computer

- Falls under SISD category.
- Typically two buses:
  a) A high-speed CPU-memory bus, that also connects to I/O bridge.
  b) A lower-speed I/O bus, connecting various peripherals.

Let us look at a single-core computer, the one that falls under the SISD category. The example that we have taken is similar to what we see inside our PCs or desktops, but not the desktops of today, rather desktops of earlier years where there were single cores. This is a typical architecture that you used to see. This is the CPU that consists of arithmetic logic unit, registers, some other circuits, bus interfaces and there was something called IO Bridge. This IO bridge had two purposes; it interfaces the CPU with the memory system with a very high speed bus, and there was a relatively lower speed IO so that the IO devices could transfer data to main memory and maybe the CPU can also access some of them.

The IO bus had connections to some USB controllers that provide some USB ports through which we can connect a variety of devices like mouse, keyboard, and many others. Through graphic adaptor you can connect your display monitor; various disk controllers, you can connect to a disk subsystem and there are expansion slots.

(Refer Slide Time: 13:48)

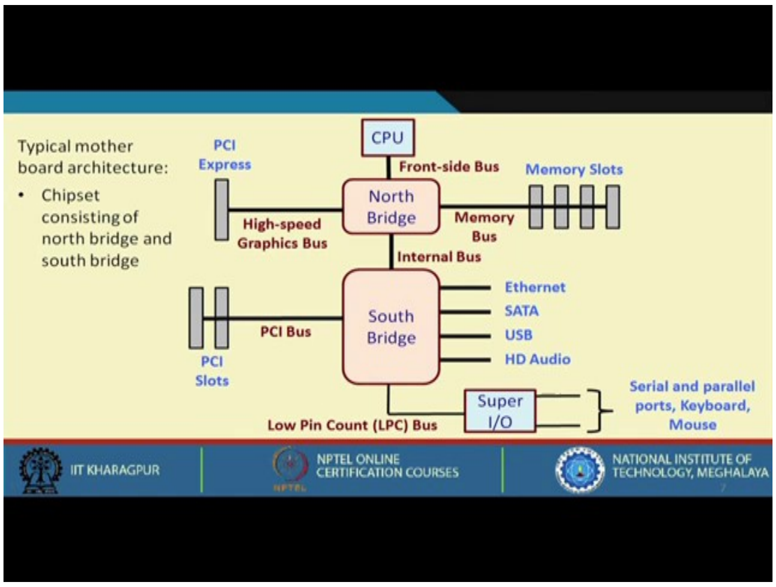

Single-core Processor

CPU chip
the single core
register file
ALU
system bus
bus interface
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

You could add some more devices like network adaptors and others. This was a very typical kind of architecture and this was your single core. This is a blown up picture of the same thing, this is technically the core and this is the interface to the external world.

When you talk off a core, every core essentially consists of a register file, ALU, and of course the control unit. The control unit is not shown here.

(Refer Slide Time: 14:20)

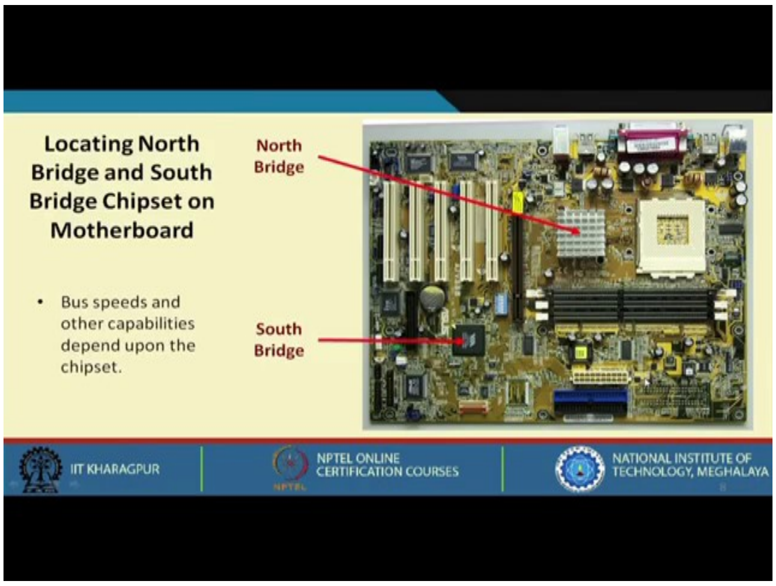

Typical mother board architecture:

- Chipset consisting of north bridge and south bridge

This is the typical motherboard architecture with the terminologies that are used in a desktop PC. The processor comes here. The processor is typically connected to a chip that is called the North Bridge, this is a very fast switch. This North Bridge interfaces the CPU with the memory bus; in the memory bus there can be memory slots. In the slots you can plug in your memory modules; there are memory modules of various capacities, you can plug those memory modules depending on how much memory you need.

And on the other side you have a PCI express bus that is typically used to connect high-speed graphics for your monitors and other things, which need very high speed data transfer. You have this PCI express bus. And on the other side it is connected to another switch called South Bridge, this is of relatively much slower speed. South Bridge interfaces with so-called PCI bus, which consists of several PCI slots through which you can connect network adaptors and different kind of circuit boards. There are also direct connections available in modern motherboards, which connects to network interface like Ethernet, disk interfaces like SATA, USB, audio interface. There is another bus that is meant for low-speed devices this is called LPC or low pin count bus. This is typically a serial bus working at a much lower speed.

(Refer Slide Time: 16:29)



Locating North Bridge and South Bridge Chipset on Motherboard

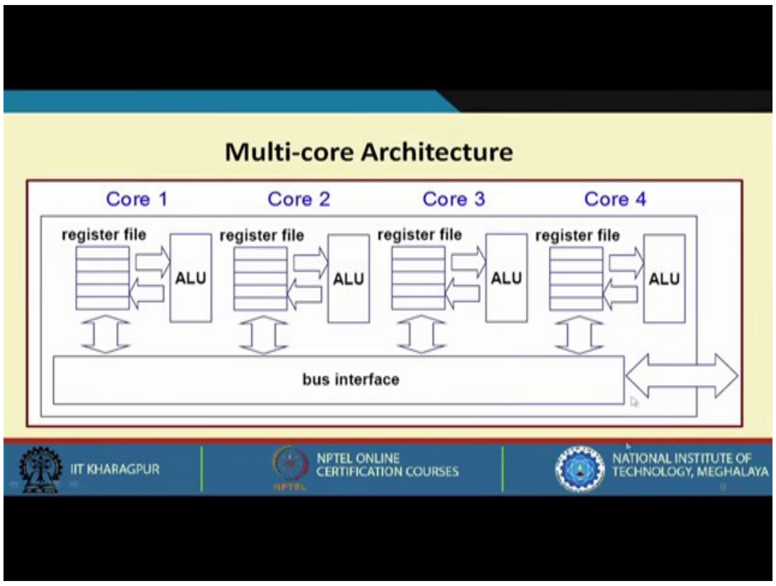
- Bus speeds and other capabilities depend upon the chipset.

North Bridge
South Bridge
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

This is the rough picture of the motherboard of a desktop. I am showing you pictorially one of the motherboards here, you can see this is your processor and this is your north bridge. The fins that you can see is actually heat sink, this is a very high speed chip, and dissipates lot of power. So, you need a heat sink to dissipate the heat. And this small chip out here this is the south bridge. These are the PCI slots, these are the memory slots and the other components. You see this is a motherboard that used to be several years back, but if you see the motherboard of one of the modern day PCs, you will see that many of the circuits have been collapsed into a single chip. There are very few chips and a very small motherboard.

That is why it is possible to squeeze the size of the computer system that much today. You can have a desktop system where the entire processing cabinet can be as small as a small rectangular box.

(Refer Slide Time: 17:49)

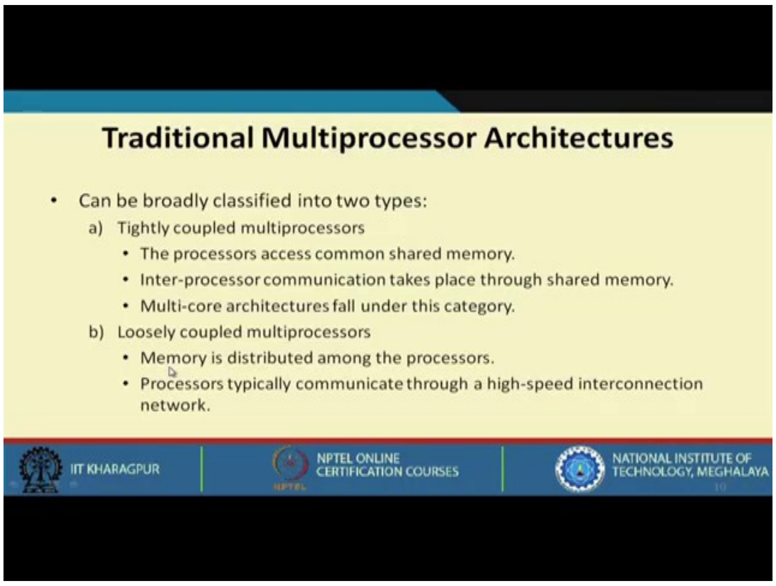


Multi-core Architecture

We looked at one core consisting of a register file, ALU, plus control. Now for multi core you have several such, and you have a some kind of a fast bus interface using which they can communicate among themselves, and there will be an interface with north bridge memory to the outside world.

This is what you have, but this is a very generic picture. Let us look into some more specifics.

(Refer Slide Time: 18:34)



Traditional Multiprocessor Architectures

- Can be broadly classified into two types:
  - a) Tightly coupled multiprocessors
    - The processors access common shared memory.
    - Inter-processor communication takes place through shared memory.
    - Multi-core architectures fall under this category.
  - b) Loosely coupled multiprocessors
    - Memory is distributed among the processors.
    - Processors typically communicate through a high-speed interconnection network.

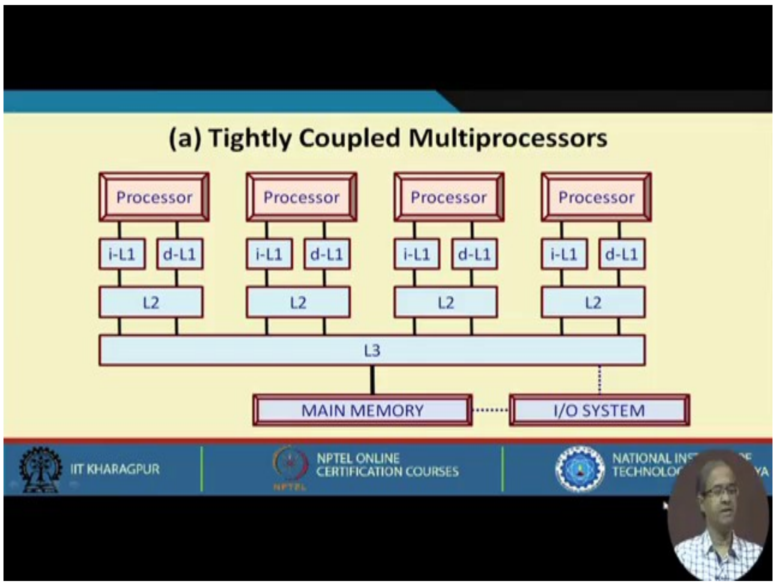
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Broadly speaking multiprocessors can be classified as either tightly coupled or loosely coupled. Tightly coupled means the processors or the cores have access to common shared memory, in other words, the processors are all inside the same box. Inside the same box there are the processors, there is also the common shared memory. When you buy a computer system, you buy the system along with all the processors and also memory.

Inter processor communication can take place through shared memory. Like one of the processor can write some data into the memory, the other processor can read the data from that memory. The standard multi core architectures usually fall under this category. But another kind of a multiprocessor architecture is also available, this is called loosely coupled multiprocessors. This is sometimes also called clusters or cluster computing. Here there is no shared memory. So, how is the memory organized? Memory is distributed among the processor; each processor has its own memory, and communication here is not via shared memory, but through some high-speed interconnection network.

There are processors, there will be a very high-speed interconnecting switch, the processors will be connecting via that switch. They will be sending and receiving messages from other processors. This is how loosely coupled processors work.

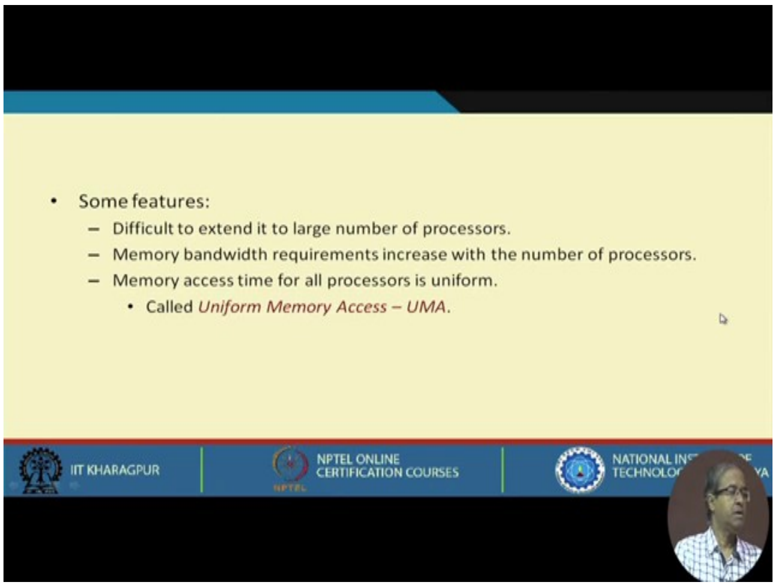
(Refer Slide Time: 20:33)



(a) Tightly Coupled Multiprocessors

Pictorially a tightly coupled multiprocessor looks like this. You can see the different cores, each of the cores is having its own instruction and data cache in L1 and also L2. These are all private to the processors, but typically the L3 cache is common across all the processors, and of course the main memory is also common.

(Refer Slide Time: 21:18)



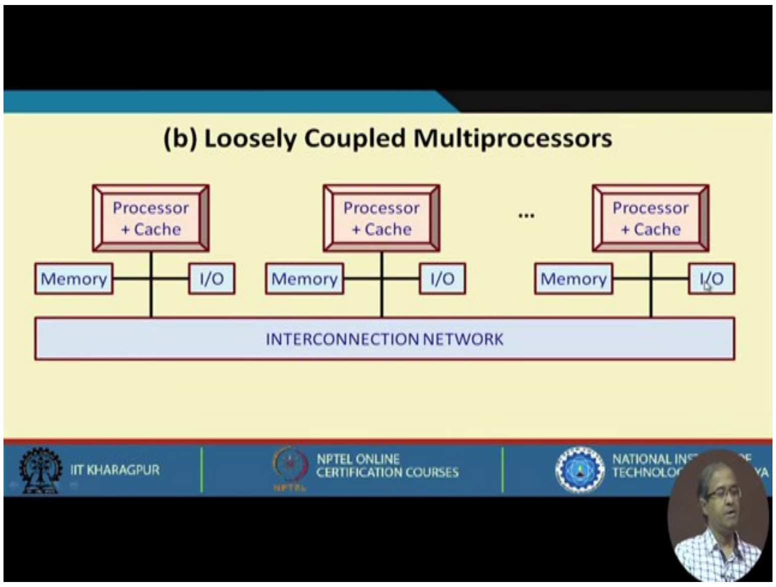
- Some features:
  - Difficult to extend it to large number of processors.
  - Memory bandwidth requirements increase with the number of processors.
  - Memory access time for all processors is uniform.
    - Called *Uniform Memory Access – UMA*.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Here you see that so many processors are trying to access L3 and the main memory. So, if you increase the number of processors to say 8, 16, or 32 they will lot of load on this L3 and main memory. So, there is a scalability issue. You cannot very easily extend to larger number of cores or processors, because the memory bandwidth requirement will also increase. In this kind of architecture there is one feature, the memory access time for all processors is uniform, this is sometimes called uniform memory access or UMA.

Why it is uniform? Because irrespective of which processor you are, the path to access memory is this same.

(Refer Slide Time: 22:18)

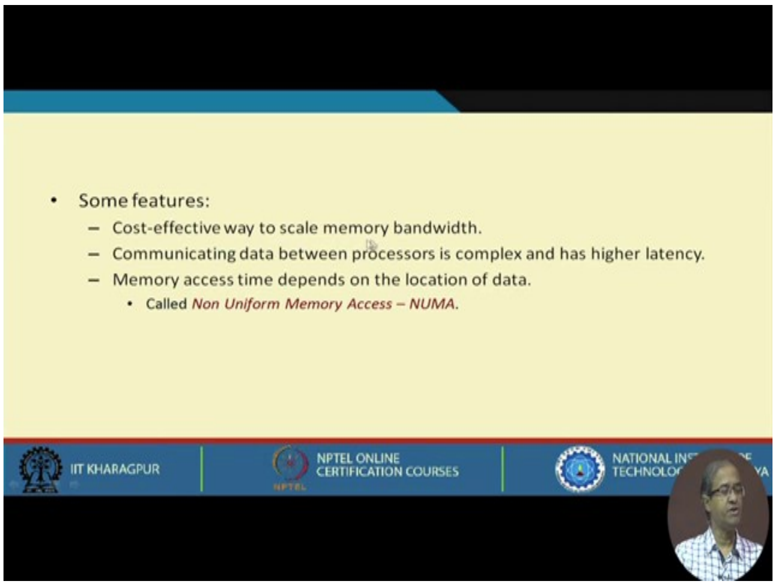


Since the delay is the same that is why it is called uniform memory access.

In contrast loosely coupled multiprocessor look like this. There will be multiple processors with their own private caches. So, all L1, L2, L3 can be inside this, they will be having the local memory. They will be connected via an interconnection network. A processor can access its local memory, and also via the interconnection network it can access the memory of the other processors.

So, you can see the access pattern will not be uniform, accessing local memory will be faster, but accessing the other memories will be slower. This is sometimes called non uniform memory access or NUMA.

(Refer Slide Time: 23:11)



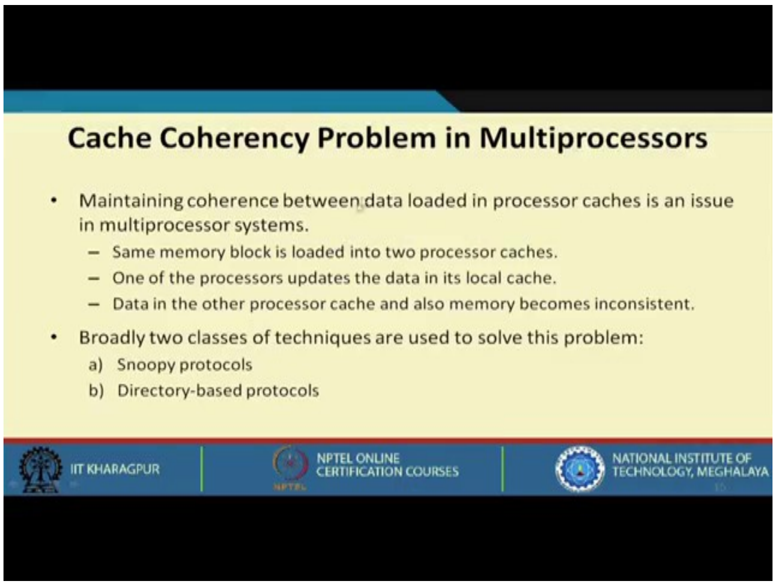
- Some features:
  - Cost-effective way to scale memory bandwidth.
  - Communicating data between processors is complex and has higher latency.
  - Memory access time depends on the location of data.
    - Called *Non Uniform Memory Access – NUMA.*

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY

This is also a cost-effective way to scale memory bandwidth, because you see most of the time when you run a program on a core, it will be accessing its local memory only. Very rarely it will be trying to access the memory of the other processors. So, in this kind of architecture, even if I increase the number of processors, scalability will not be that much of a challenge.

The only flip side is that the communication between the processors is complex using the switching network and has higher latency. So, when you are mapping some applications to run on them, you try to map in such a way that most of the time

(Refer Slide Time: 24:08)



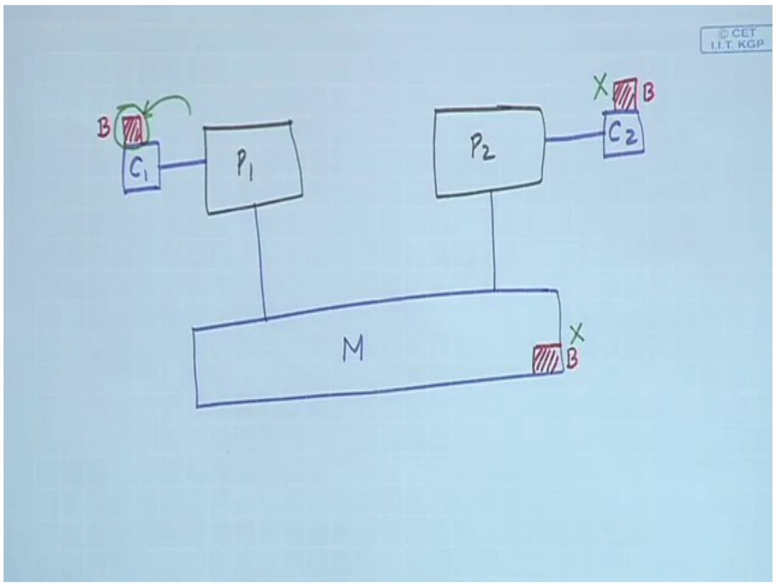
Cache Coherency Problem in Multiprocessors

- Maintaining coherence between data loaded in processor caches is an issue in multiprocessor systems.
  - Same memory block is loaded into two processor caches.
  - One of the processors updates the data in its local cache.
  - Data in the other processor cache and also memory becomes inconsistent.
- Broadly two classes of techniques are used to solve this problem:
  a) Snoopy protocols
  b) Directory-based protocols

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

they will be accessing the local memory and very rarely they will be accessing the other processors and their memories.

(Refer Slide Time: 24:26)



Cache coherence is a very important problem; I am trying to illustrate this. Suppose I have two processors P1 and P2, assume they have their own cache memories C1 and C2.

Let us also assume they have access to a common memory M. So, what may happen now? You see there can be one block in this memory, this is the block and the processor might be using this block. So, this block will be loaded in the cache. It is possible that the other processor is also trying to access this. So, this block will also be loaded in the cache. Let us say B is the original block; a copy is here and a copy is here. Now the processor P1 might write something into this block B that will make this block dirty. But as soon as it writes what will happen you can imagine, these two blocks will become invalid. Whatever is loaded in P2’s cache will be stale because this block was recently modified and the copy in memory will also be stale.

This is called the cache coherency problem, where there are multiple copies of the block that may be located in several places. One of them may be the good version, while others may be incoherent or wrong. If you can see here same memory block is loaded into two processor caches, one of the processor updates the data. So, data in the other processor cache and also memory becomes inconsistent.

There are many methods and techniques that have been proposed to solve this problem. One is call the snoopy protocol where in the memory bus itself you continuously check whether someone is doing a write into its local cache. If it does you immediately send the information to the other memory controls, they can check whether they are also having a copy of that same block in their cache. If so, they will immediately invalidate. The other is directory based protocol where you store the information about the blocks in some central place where the memories are stored. You store that a memory block is currently held by these processors.

In a processor update you know which are the other processors that will also get affected, but I am not going with the detail of this protocols here. I just tried to give a very broad bird’s eye view about what kind of parallel computing platforms are being used today and what are the different variations therein.

So with this I come to the end of this lecture. In this lecture we tried to give you a broad classification of computer systems with some examples, what kind of processing or computing systems fall under which category and some of the problems and issues. If you are interested to know more detail on this, there are advanced level courses on computer architecture available; you can get to know about these in those courses.

Thank you.