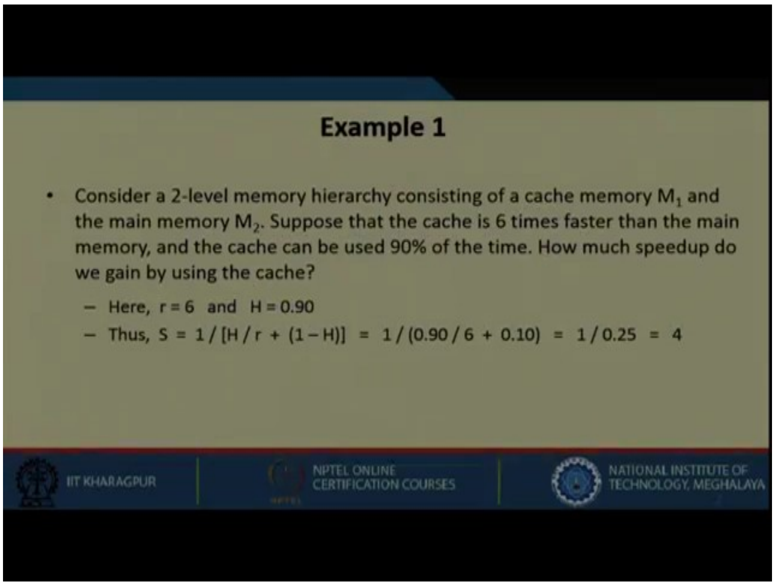
Computer Architecture and Organization

Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture - 29
Memory Hierarchy Design (PART II)

Welcome to next lecture 29. Here we will be discussing about memory hierarchy design in detail.

(Refer Slide Time: 00:28)



Example 1

- Consider a 2-level memory hierarchy consisting of a cache memory M_1 and the main memory M_2. Suppose that the cache is 6 times faster than the main memory, and the cache can be used 90% of the time. How much speedup do we gain by using the cache?
  - Here, $r = 6$ and $H = 0.90$
  - Thus, $S = 1 / [H / r + (1 - H)] = 1 / (0.90 / 6 + 0.10) = 1 / 0.25 = 4$

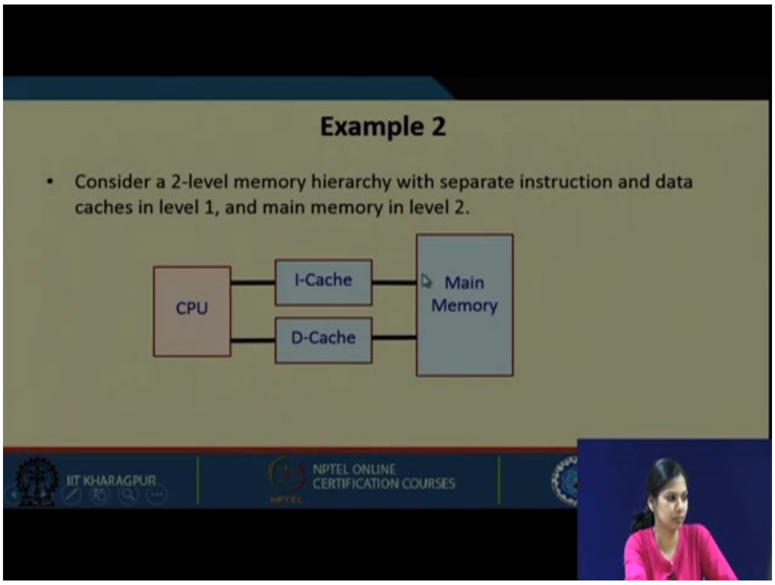
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Let us take this example here. Consider a two level memory hierarchy consisting of a cache memory M1 and main memory M2. Generally, the level which is closest to the processor we call it cache memory and then the next level of memory can be different levels of cache memory; or we can also have a main memory.

Suppose that cache is 6 times faster than main memory and the cache can be used 90% of the time. How much speed up do we gain by using cache?

So, here r will be 6 and H will be 0.90. So, we simply put in into this formula and we get 4. So, we are getting a speed up of 4.

(Refer Slide Time: 02:25)



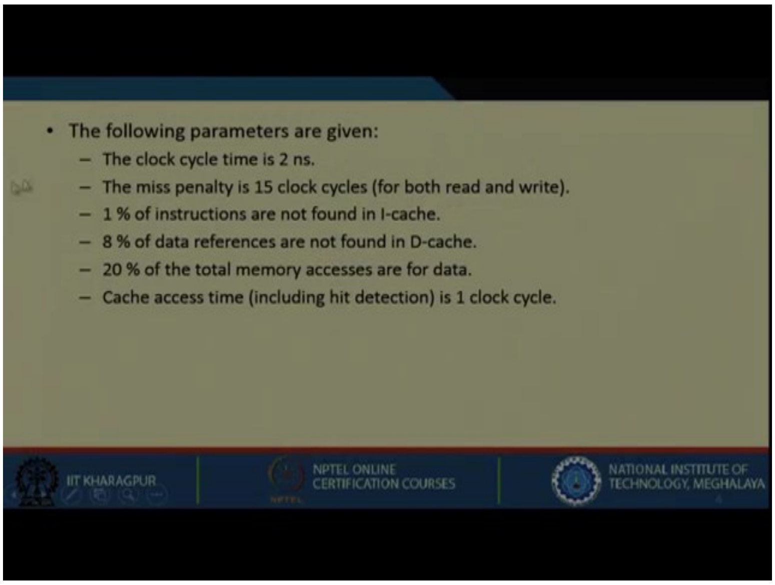
Example 2

- Consider a 2-level memory hierarchy with separate instruction and data caches in level 1, and main memory in level 2.

This is pretty similar to what we discussed way back in Amadahl’s law. Now I take another example where we consider two-level memory hierarchy with separate instruction and data caches in level 1, and main memory in level 2. So, here the scenario is we have I-cache, and we have D-cache. The instructions are brought and kept in instruction cache the data are brought and kept in data cache.

The advantage would be if you want to access data and instructions simultaneously, you can do that.

(Refer Slide Time: 03:13)



- The following parameters are given:
  - The clock cycle time is 2 ns.
  - The miss penalty is 15 clock cycles (for both read and write).
  - 1 % of instructions are not found in I-cache.
  - 8 % of data references are not found in D-cache.
  - 20 % of the total memory accesses are for data.
  - Cache access time (including hit detection) is 1 clock cycle.

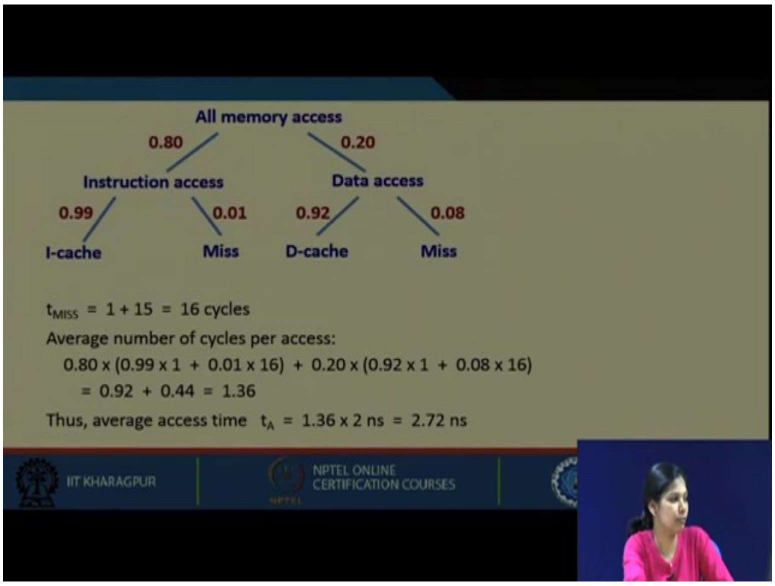
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

The following parameters are given.

For miss penalty total 15 clock cycles are required. Out of which it says that 1% of instructions are not found in I-cache; and 8% of data references are not found in D-cache. 20% of the total memory access are for data, and cache access time including hit detection is 1 clock cycle.

So, these are the information that are provided and we will be seeing how we can find the average access time.

(Refer Slide Time: 04:32)



t_{MISS} = 1 + 15 = 16 cycles

Average number of cycles per access:

$0.80 \times (0.99 \times 1 + 0.01 \times 16) + 0.20 \times (0.92 \times 1 + 0.08 \times 16)$

$= 0.92 + 0.44 = 1.36$

Thus, average access time $t_A = 1.36 \times 2$ ns $= 2.72$ ns

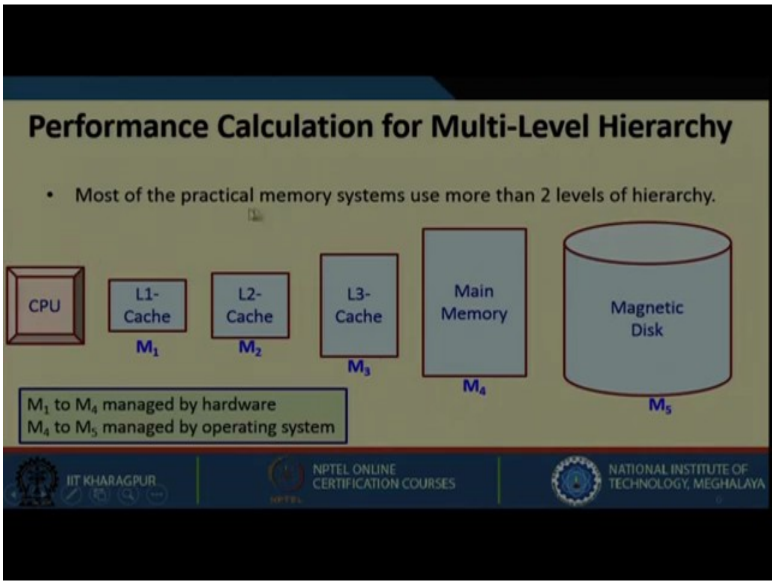
So, let us see, out of total memory access 80% are for instruction and 20% are for data. Out of this 80% of instruction, I-cache can be used 99%, and miss rate is 0.01 where the data is not found in I-cache.

Similarly, for data access 8% there is miss. So, 0.92 will be hit.

The average access time is calculated as shown.

So, in this example we have seen what happens when there is a miss, what happens when data is found in cache, and now the cache is divided into two different levels, one is I-cache and another is D-cache.

(Refer Slide Time: 07:55)



Performance Calculation for Multi-Level Hierarchy

- Most of the practical memory systems use more than 2 levels of hierarchy.

M_1 to M_4 managed by hardware
M_4 to M_5 managed by operating system

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

So, when you have two different caches, the calculation will be made in this fashion. Let us see performance calculation for multi level hierarchy.

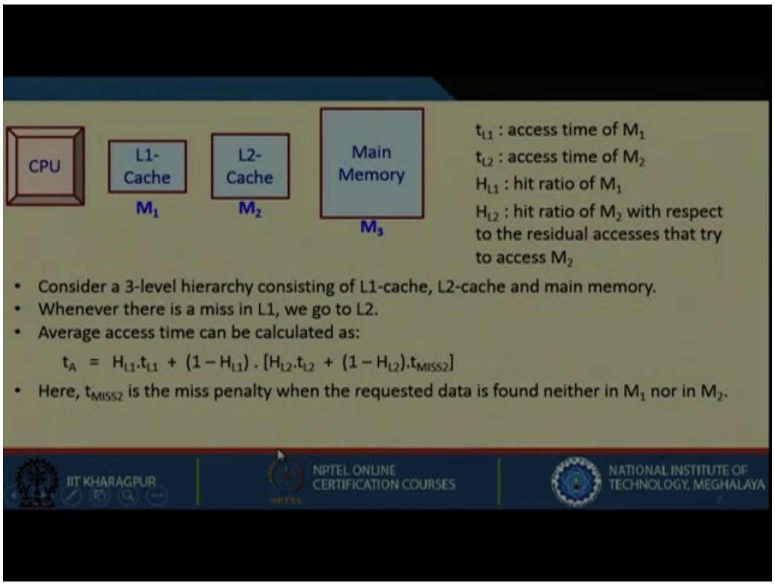
In general, we have multi-level hierarchy rather than two-level hierarchy. In two-level hierarchy, we have only two levels M1 and M2, but in reality we have multiple levels. For multiple levels whatever we have done for two levels can be extended to multi levels.

So, if you see this particular diagram which is considered as the most practical memory system that; obviously, we use more than two levels of hierarchy. The first level is M1 which is L1 cache, the next level is M2 which is using L2 cache, the next level is M3 which is L3 cache, then M4 is the main memory, and finally M5 is the magnetic disk.

So, how do we calculate? As we can see the access time of this will be much faster than this, then this, then this, and data from main memory is brought through L3 to L2 and then to L1, and when it is in L1 because of locality of reference, we will get the data whenever CPU is asking for it. But initially remember there will be a miss because there will be no data in the cache.

Whenever the data is loaded in main memory, whenever it is required it has to be first brought from main memory through different levels to L1 cache, and finally it goes to CPU. So, for the first time there will be miss always, but for consecutive times there will be hit because of locality of reference. So, M1 to M4 is managed by hardware; this is very important. But M4 to M5 is handled by operating system.

(Refer Slide Time: 10:31)



t_{L1} : access time of M_1

t_{L2} : access time of M_2

H_{L1} : hit ratio of M_1

H_{L2} : hit ratio of M_2 with respect to the residual accesses that try to access M_2

- Consider a 3-level hierarchy consisting of L1-cache, L2-cache and main memory.
- Whenever there is a miss in L1, we go to L2.
- Average access time can be calculated as:

$$t_A = H_{L1}.t_{L1} + (1 - H_{L1}) . [H_{L2}.t_{L2} + (1 - H_{L2}).t_{MISS2}]$$

- Here, t_{MISS2} is the miss penalty when the requested data is found neither in M_1 nor in M_2.

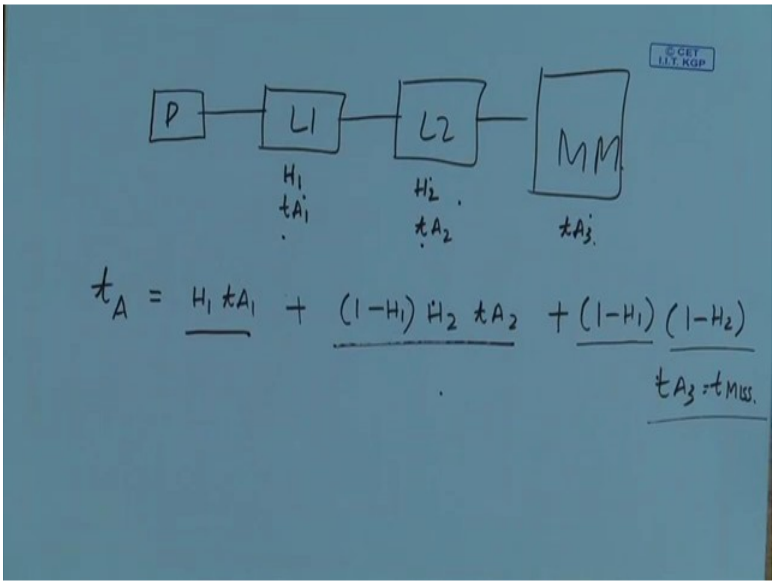
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Now, let us see this we have one more level; three levels of memory hierarchy, where tL1 is the access time of M1, tL2 is access time of M2, HL1 is the hit ratio of M1; that means, the percentage time the data or instruction is found in L1 cache. HL2 is the percentage time the data or instruction is found in L2 cache, which is not found in L1 cache.

So, hit ratio of M2 is with respect to the residual access that try to access M2. Consider a three level hierarchy that consists of L1 cache, L2 cache and main memory. Whenever there is a miss in L1 we go to L2. Obviously, if we do not find the data or instruction in L1 cache we move to L2 cache. What will be the average access time now? Earlier we have done average access time calculation taking into consideration that CPU first access L1 cache.

The access time tA is given by this expression.

(Refer Slide Time: 13:03)

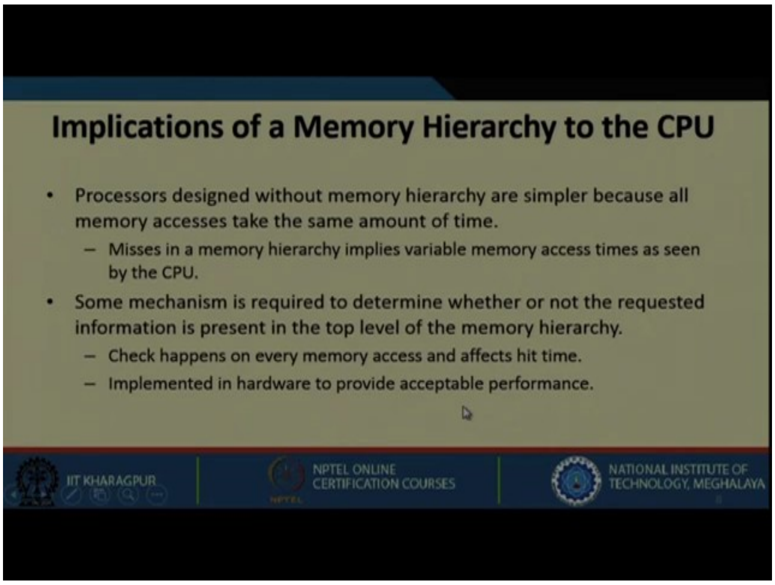


$$t_A = \underline{H_1 t_{A_1}} + \underline{(1-H_1) H_2 t_{A_2}} + \underline{(1-H_1)}\,\underline{(1-H_2)}\,\underline{t_{A_3} = t_{Miss}}$$

So, what will be the average access time? The expression is shown.

This is the equation for three levels of memory hierarchy.

(Refer Slide Time: 16:14)



Implications of a Memory Hierarchy to the CPU

- Processors designed without memory hierarchy are simpler because all memory accesses take the same amount of time.
  - Misses in a memory hierarchy implies variable memory access times as seen by the CPU.
- Some mechanism is required to determine whether or not the requested information is present in the top level of the memory hierarchy.
  - Check happens on every memory access and affects hit time.
  - Implemented in hardware to provide acceptable performance.

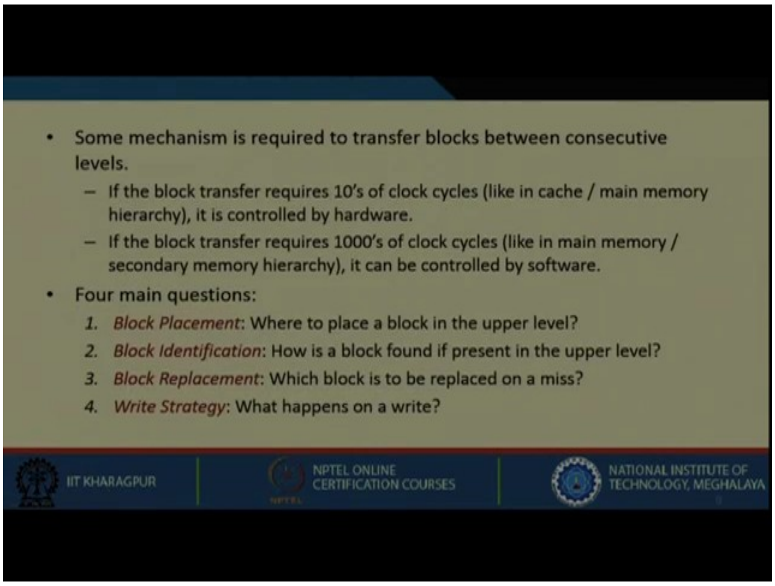
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

So, what is the implication of the memory hierarchy to the CPU? We can say that the processors designed without memory hierarchy are simple because all memory access takes same amount of time. Misses in a memory hierarchy implies variable memory access time as seen by the CPU, because access time of L1 cache is different from access time of L2 cache, and it is different from main memory.

Some mechanism is required to determine whether or not the requested information is present in top level of memory hierarchy. So, how can we check this? Check happens on every memory access and affects the hit time and implemented in hardware to provide acceptable performance. So, what we are doing.

We are first checking into the top level L1. Is there any mechanism that we in advance know that this particular word is there or not, or the checking can it be made little fast so that hit time can made little faster. So, there are some kinds of things that can be done, check happens on every memory access and that is why it affects the hit times. If we can have some kind of hardware implementation for this matching, it can be performed in reasonable time.

(Refer Slide Time: 18:33)



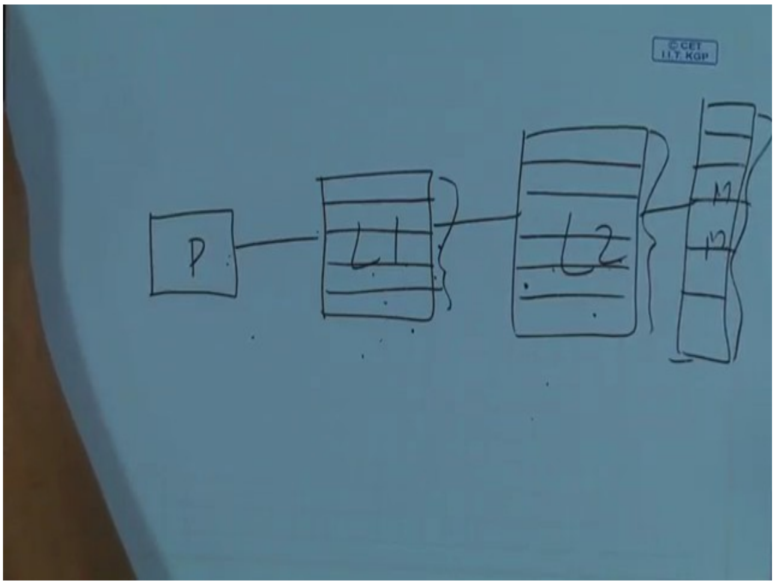
- Some mechanism is required to transfer blocks between consecutive levels.
  - If the block transfer requires 10's of clock cycles (like in cache / main memory hierarchy), it is controlled by hardware.
  - If the block transfer requires 1000's of clock cycles (like in main memory / secondary memory hierarchy), it can be controlled by software.
- Four main questions:
  1. *Block Placement*: Where to place a block in the upper level?
  2. *Block Identification*: How is a block found if present in the upper level?
  3. *Block Replacement*: Which block is to be replaced on a miss?
  4. *Write Strategy*: What happens on a write?

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

Some mechanism is required to transfer blocks between consecutive levels; as we know that if there is a miss in both L1 and L2, you have to bring a block from main memory to L2 level and then from L2 to L1 level. So, whenever there is a miss in some level you have to transfer a block. If the block transfer requires 10s of clock cycles like cache or main memory, it is controlled by hardware; but if the block transfer requires 1000s of clock cycles like in main memory, secondary memory etc., it can be controlled by software.

At this point of time few questions arise. One is block placement. Let us say we have different levels of memory.

(Refer Slide Time: 19:45)



So, again we have processor, we have L1 cache, we have L2 cache, and we have main memory. Now it has got some blocks.

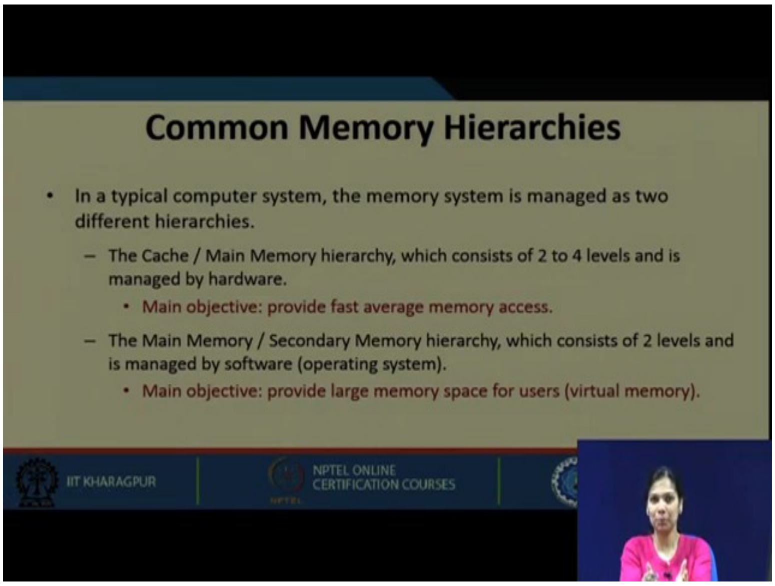
Now, the point is where we will make the placement of block. From here we will bring the block and place it here. It needs to be understood next is block identification. What is blocked identification? How is a block found if present in a upper level? So, we must be matching something when the CPU generates a logical address and it says that this is the logical address I am looking for this particular word, then something much must be matched to know if that block is present in the upper level or not. So, block identification is also important.

Similarly, block replacement … which block is to be replaced on a miss. What do you mean by that? Now you see in this particular diagram we have more number of blocks, here we have less number of blocks, and we are saying that every time the processor accesses the L1 cache it will be the fastest. So, all the programs or data are brought into L1 cache and then it is accessed.

Now, the size of this is small. So, at a time limited number of blocks can be present here. If you want to run your program that requires more number of blocks, in that case what you have to do? You have to replace some of the existing blocks from here and bring a new block. So, for that some block replacement strategies must be there; that means, some blocks can be removed from this particular level and then some blocks from this level can be brought in. So, this is called block replacement.

Write strategy means what happens on a write whenever we are performing a write operation. What strategies must be used for this write operation?

(Refer Slide Time: 22:56)



Common Memory Hierarchies

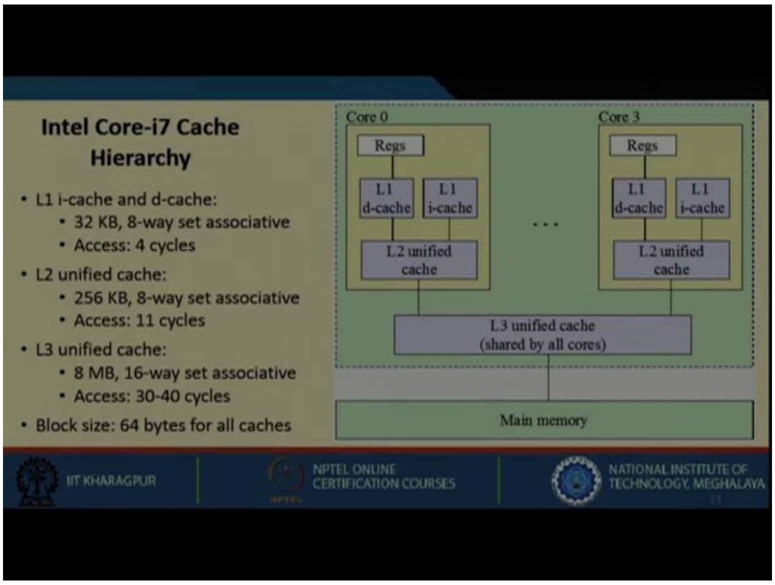
- In a typical computer system, the memory system is managed as two different hierarchies.
  - The Cache / Main Memory hierarchy, which consists of 2 to 4 levels and is managed by hardware.
    - Main objective: provide fast average memory access.
  - The Main Memory / Secondary Memory hierarchy, which consists of 2 levels and is managed by software (operating system).
    - Main objective: provide large memory space for users (virtual memory).

In a typical computer system the memory system is managed as two different hierarchies, cache memory / main memory hierarchy which consists of 2 to 4 levels and is managed by hardware. The main objective of this particular hierarchy is to provide fast average access time for instructions and data.

Then we have main memory / secondary memory hierarchy that consists of two levels and is managed by software, that is by the operating system which is the interface between the user and the hardware part of our computer. The main objective is to provide large memory space for the user, called virtual memory.

So, basically this virtual memory is a concept where it says that you have a large memory space at your disposal, but actually you are having the space as main memory only. Programs and data are brought from secondary storage to main memory as and when it is required. This is called virtual memory, which will not be taken up in this particular course in detail.

(Refer Slide Time: 24:51)



Intel Core-i7 Cache Hierarchy

- L1 i-cache and d-cache:
  - 32 KB, 8-way set associative
  - Access: 4 cycles
- L2 unified cache:
  - 256 KB, 8-way set associative
  - Access: 11 cycles
- L3 unified cache:
  - 8 MB, 16-way set associative
  - Access: 30-40 cycles
- Block size: 64 bytes for all caches

Core 0

Regs

L1 d-cache

L1 i-cache

L2 unified cache

. . .

Core 3

Regs

L1 d-cache

L1 i-cache

L2 unified cache

L3 unified cache (shared by all cores)

Main memory

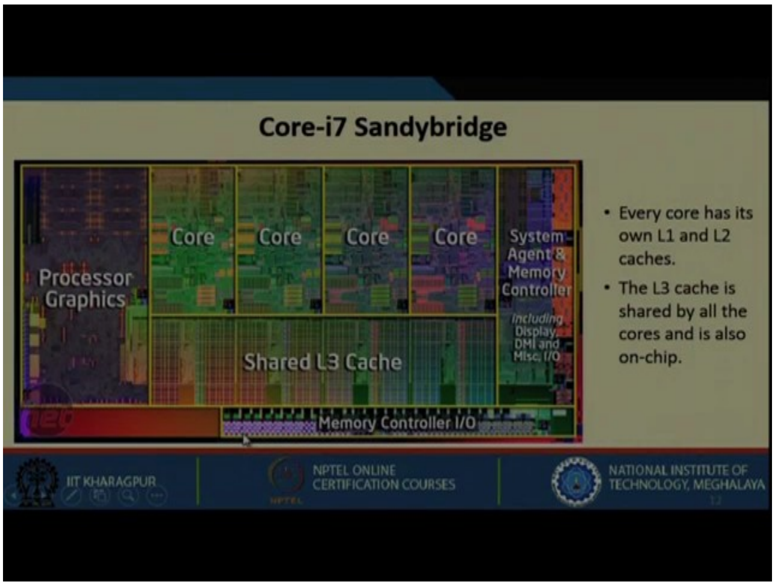
IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

This is the typical memory hierarchy of Intel core i. There are four cores; core 0, core 1, core 2, and core 3. Inside each of the cores you have registers, you have separate L1 cache for data and instruction, we have an unified L2 cache.

(Refer Slide Time: 27:39)



Core-i7 Sandybridge

- Every core has its own L1 and L2 caches.
- The L3 cache is shared by all the cores and is also on-chip.

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NATIONAL INSTITUTE OF TECHNOLOGY, MEGHALAYA

Next is core-i7 SandyBridge. Here you have 4 cores, every core has its own L1 and L2 cache, and L3 is shared by all the cores and is also on chip. This is the processor, graphics processor, the memory controller, etc.

So, we come to the end of lecture 29 where we discussed about memory hierarchy in details. We have seen that how memory hierarchy actually affects the overall performance.

Thank you.