Seminar 1

Memory Hierarchy

Let us remember the following two slides from Lecture 2-3

Cache Organization

Example:

- a cache of 64 Kbytes
- data transfer between cache and main memory is in *blocks* of 4 bytes; we say the cache is organized in *lines* of 4 bytes;
- a main memory of 16 Mbytes; each byte is addressable by a 24-bit address (2^{24}=16M)

  - the cache consists of 2^{14} (16K) lines
  - the main memory consists of 2^{22} (4M) blocks

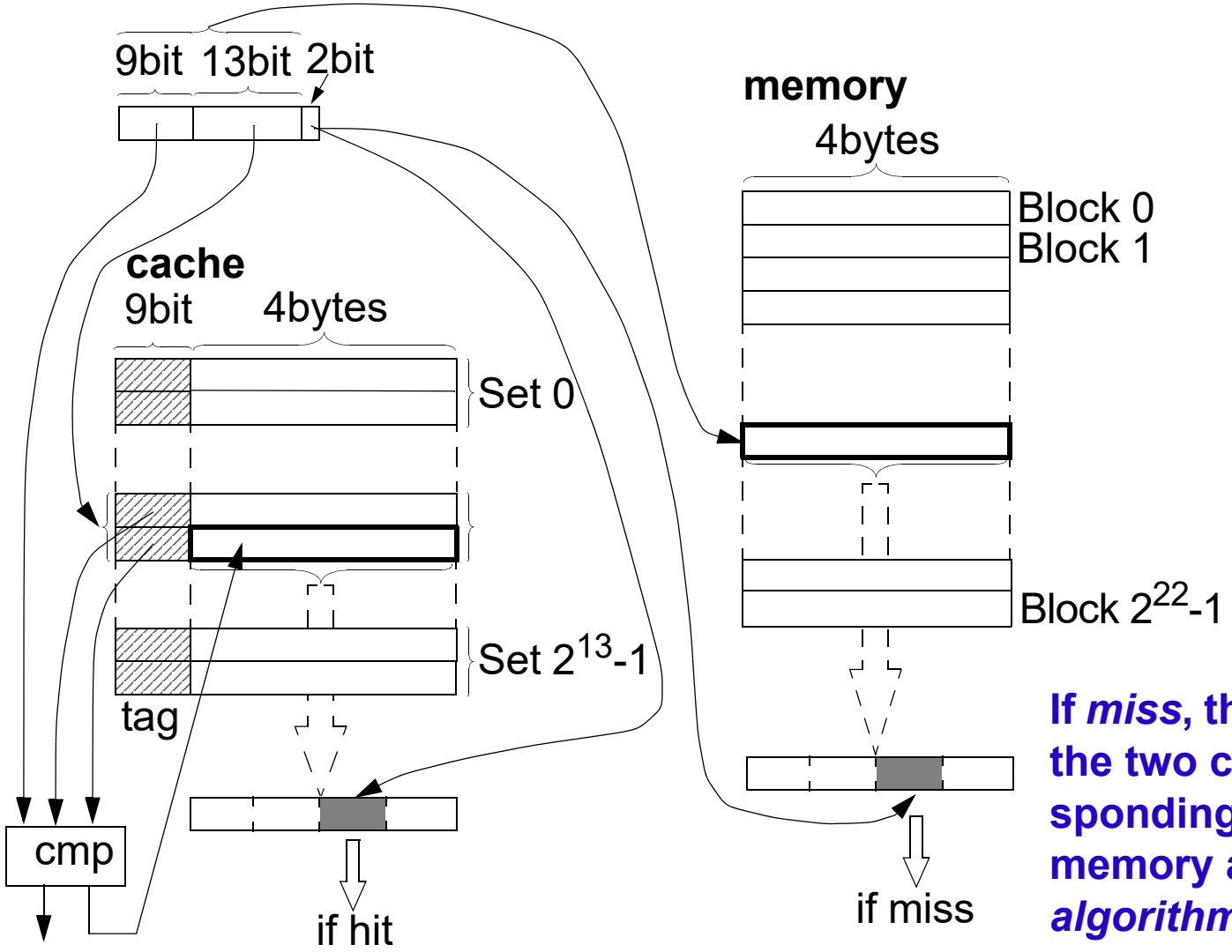
Questions:

- when we bring a block from main memory into the cache where (in which line) do we put it?
- when we look for the content of a certain memory address
  - in which cache line do we look for it?
  - how do we know if we have found the right information (*hit*) or not (*miss*)?

Set Associative Mapping

Two-way set associative cache

- 13 bits are needed to identify the cache set
- 22 bits are needed to address a block in main memory
- *tag* size is 22 - 13 = 9 bits



If *miss*, the block is placed in one of the two cache lines in the set corresponding to the 13 bits field in the memory address. The *replacement algorithm* decides which line to use.

Problem 1

A set-associative cache consists of 64 lines, divided into four-line sets. Main memory contains 4K blocks of 128 bytes each. Show the format of the main memory addresses.

Problem 1

A set-associative cache consists of 64 lines, divided into four-line sets. Main memory contains 4K blocks of 128 bytes each. Show the format of the main memory addresses.

Solution:

- **4K = 2^{12} blocks in main memory $\Rightarrow$ 12 bits are needed to address a block in main memory.**

Problem 1

A set-associative cache consists of 64 lines, divided into four-line sets. Main memory contains 4K blocks of 128 bytes each. Show the format of the main memory addresses.

Solution:

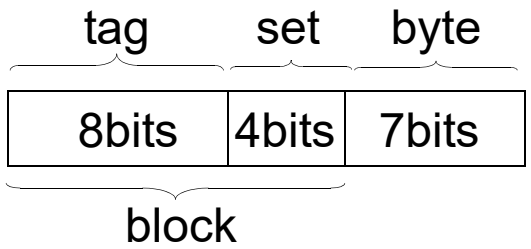
- **4K = 2^{12} blocks in main memory $\Rightarrow$ 12 bits are needed to address a block in main memory.**
- **We have 64/4 = 16 = 2^4 sets $\Rightarrow$ 4 bits are needed to identify the cache set.**
- ***tag* size is 12 - 4 = 8 bits.**

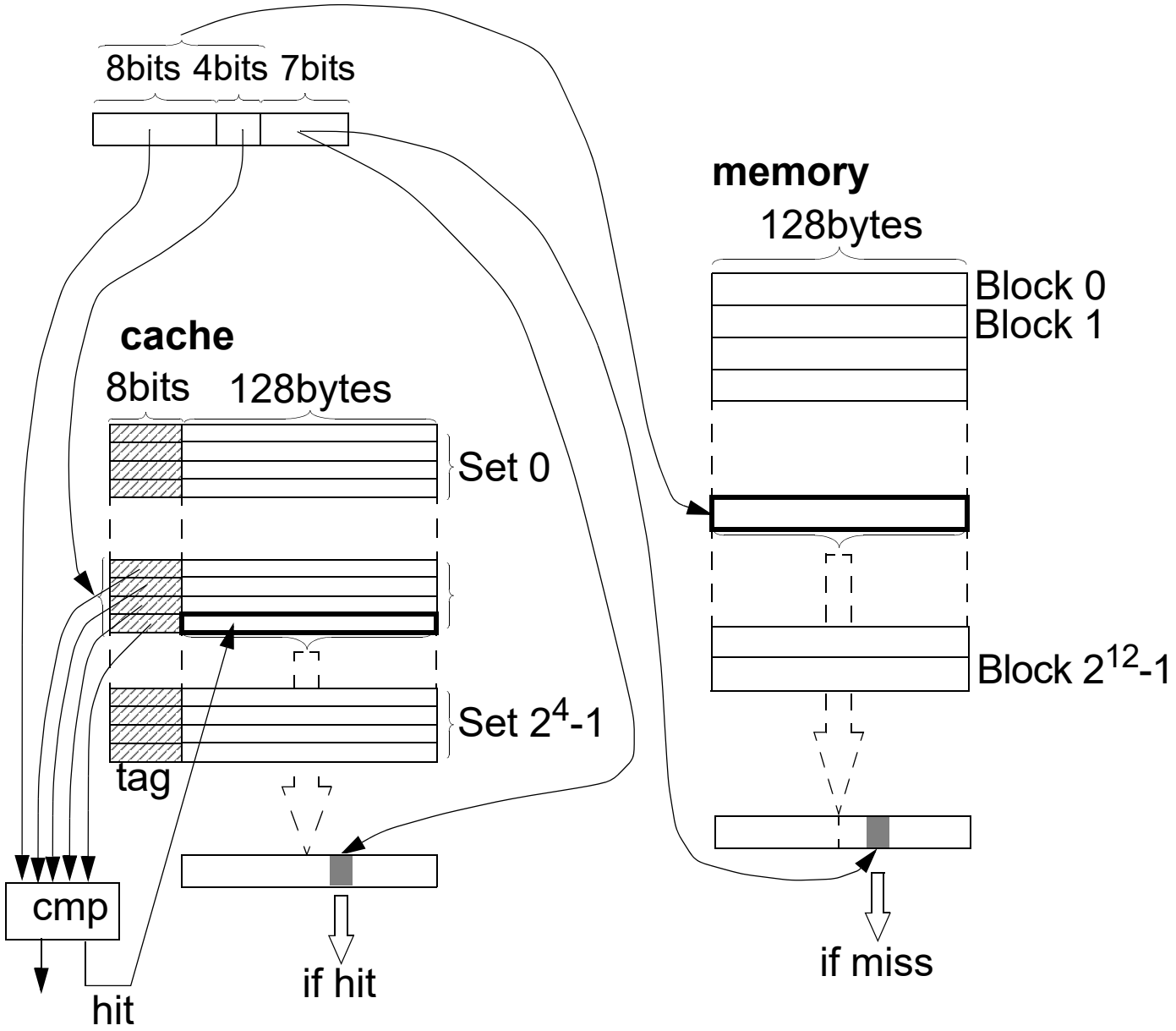
Problem 1

A set-associative cache consists of 64 lines, divided into four-line sets. Main memory contains 4K blocks of 128 bytes each. Show the format of the main memory addresses.

Solution:

- **4K = 2^{12} blocks in main memory $\Rightarrow$ 12 bits are needed to address a block in main memory.**
- **We have 64/4 = 16 = 2^4 sets $\Rightarrow$ 4 bits are needed to identify the cache set.**
- ***tag* size is 12 - 4 = 8 bits.**
- **Block/line length = 128 = 2^7 $\Rightarrow$ 7 bits are needed to select a byte out of a block/line.**




8bits 4bits 7bits
memory
128bytes
Block 0
Block 1
cache
8bits 128bytes
Set 0
Set 2^4-1
Block 2^{12}-1
tag
cmp
hit
if hit
if miss

Problem 2

A two-way set-associative cache has lines of 16 bytes and a total size of 8 Kbytes. The 64-Mbyte main memory is byte addressable. Show the format of main memory addresses.

Problem 2

A two-way set-associative cache has lines of 16 bytes and a total size of 8 Kbytes. The 64-Mbyte main memory is byte addressable. Show the format of main memory addresses.

Solution:

- **Main memory consists of $2^{26}/2^4 = 2^{22}$ blocks $\Rightarrow$ 22 bits are needed to address a block in main memory.**

Problem 2

A two-way set-associative cache has lines of 16 bytes and a total size of 8 Kbytes. The 64-Mbyte main memory is byte addressable. Show the format of main memory addresses.

Solution:

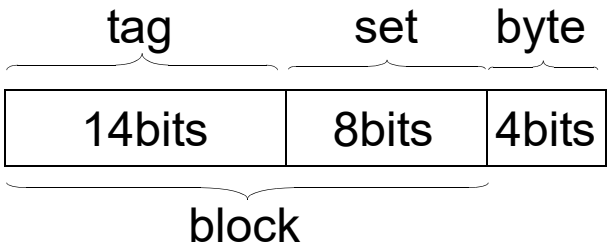
- Main memory consists of $2^{26}/2^4 = 2^{22}$ blocks $\Rightarrow$ 22 bits are needed to address a block in main memory.
- We have $2^{13}/2^4 = 2^9$ lines in the cache; we have $2^9/2 = 2^8$ sets $\Rightarrow$ 8 bits are needed to identify the cache set.
- *tag* size is 22 - 8 = 14 bits.

Problem 2

A two-way set-associative cache has lines of 16 bytes and a total size of 8 Kbytes. The 64-Mbyte main memory is byte addressable. Show the format of main memory addresses.

Solution:

- Main memory consists of $2^{26}/2^4 = 2^{22}$ blocks $\Rightarrow$ 22 bits are needed to address a block in main memory.
- We have $2^{13}/2^4 = 2^9$ lines in the cache; we have $2^9/2 = 2^8$ sets $\Rightarrow$ 8 bits are needed to identify the cache set.
- *tag* size is 22 - 8 = 14 bits.
- Block/line length = $16 = 2^4 \Rightarrow$ 4 bits are needed to select a byte out of a block/line.



Problem 3

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used.

a. How is a 16-bit memory address divided into tag, line number, and byte number?

b. Into what line would bytes with each of the following addresses be stored?
   0001 0001 0001 1011
   1100 0011 0011 0100
   1101 0000 0001 1101
   1010 1010 1010 1010

c. Suppose the byte with address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it?

d. How many total bytes can be stored in the cache?

e. Why is the tag also stored in the cache?

Problem 3

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used.

a.

- Main memory consists of $2^{16}/2^3 = 2^{13}$ blocks $\Rightarrow$ 13 bits are needed to address a block in main memory.

Problem 3

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used.

a.

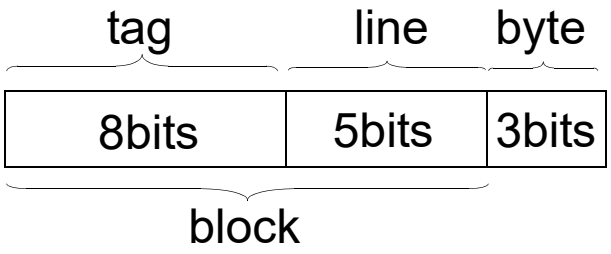
- Main memory consists of $2^{16}/2^3 = 2^{13}$ blocks $\Rightarrow$ 13 bits are needed to address a block in main memory.
- We have 2^5 lines in the cache $\Rightarrow$ 5 bits are needed to identify the line in the cache.
- *tag* size is 13 - 5 = 8 bits.

Problem 3

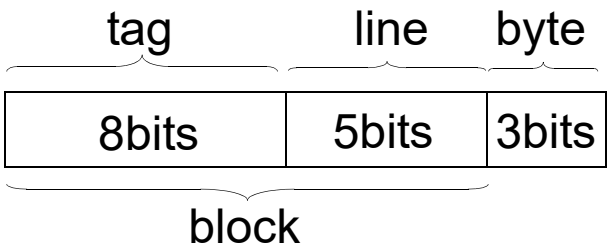
Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used.

a.

- **Main memory consists of $2^{16}/2^3 = 2^{13}$ blocks $\Rightarrow$ 13 bits are needed to address a block in main memory.**
- **We have 2^5 lines in the cache $\Rightarrow$ 5 bits are needed to identify the line in the cache.**
- ***tag* size is 13 - 5 = 8 bits.**
- **Block/line length = 2^3 $\Rightarrow$ 3 bits are needed to select a byte out of a block/line.**



Problem 3



b.

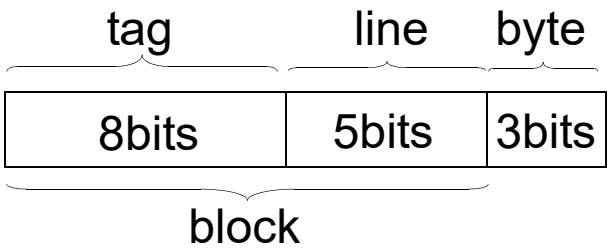
0001 0001 0001 1011

1100 0011 0011 0100

1101 0000 0001 1101

1010 1010 1010 1010

Problem 3



b.

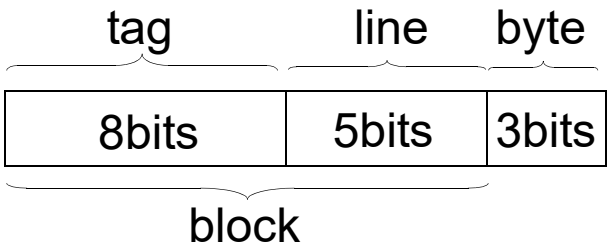
0001 0001 0001 1011 ⇒ cache line 3

1100 0011 0011 0100 ⇒ cache line 6

1101 0000 0001 1101 ⇒ cache line 3

1010 1010 1010 1010 ⇒ cache line 21

Problem 3



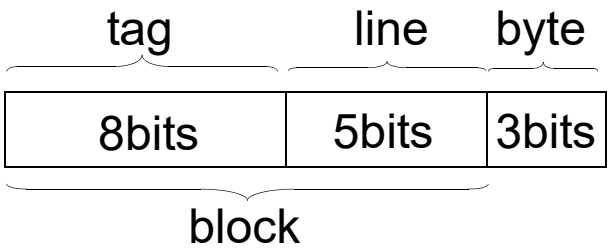
c.

0001 1010 0001 1010 is stored in line 3

When this byte is loaded into the cache, the whole block of 8 bytes is loaded; the main memory addresses of those 8 bytes are:

0001 1010 0001 1000
0001 1010 0001 1001
0001 1010 0001 1010
0001 1010 0001 1011
0001 1010 0001 1100
0001 1010 0001 1101
0001 1010 0001 1110
0001 1010 0001 1111

Problem 3



d.

Cache size is (2^5 lines) x (2^3 bytes per line) = 2^8 = 256 bytes.

e.

In order to distinguish between the 2^8 different blocks that can be stored in the same cache line (see lecture 2).

Problem 4

Consider the following code:

```
for (i=0; i < 20; i++)
    for (j = 0; j < 10; j++)
            a[i] = a[i]*(j+1);
```

a. Give one example of the spacial locality in the code.
b. Give one example of the temporal locality in the code.

Problem 4

Consider the following code:

```
for (i=0; i < 20; i++)
    for (j = 0; j < 10; j++)
        a[i] = a[i]*(j+1);
```

a. Give one example of the spacial locality in the code.
b. Give one example of the temporal locality in the code.

Solution:

a.

- When a[i] is used, in the next iteration of the outer loop a[i+1] will be used.
- After an instruction is executed, immediately the following one is executed.

b.

- Inside the inner loop, the same a[i] is accessed ten times in a short interval.

Problem 5

Consider an L1 cache with an access time of 1 ns and a hit ratio of H = 0.95. Suppose that we can change the cache design (size of the cache, cache organization) such that we increase H to 0.97, but increase the cache access time to 1.5 ns. What conditions must be met for this change to result in improved performance?

Problem 5

Consider an L1 cache with an access time of 1 ns and a hit ratio of H = 0.95. Suppose that we can change the cache design (size of the cache, cache organization) such that we increase H to 0.97, but increase the cache access time to 1.5 ns. What conditions must be met for this change to result in improved performance?

Remember from Lecture 2:

Average access time T_s :

$$T_s = H \times T_1 + (1 - H) \times (T_1 + T_2) = T_1 + (1 - H) \times T_2$$

- T_1 : Access time to cache
- T_2 : Access time to main memory
- H : Hit ratio

Problem 5

Consider an L1 cache with an access time of 1 ns and a hit ratio of H = 0.95. Suppose that we can change the cache design (size of the cache, cache organization) such that we increase H to 0.97, but increase the cache access time to 1.5 ns. What conditions must be met for this change to result in improved performance?

$$T_s = H \times T_1 + (1 - H) \times (T_1 + T_2) = T_1 + (1 - H) \times T_2$$

Solution

$$T_s^1 = 1 + (1 - 0{,}95) \times T_2 = 1 + 0{,}05\,T_2$$

Problem 5

Consider an L1 cache with an access time of 1 ns and a hit ratio of H = 0.95. Suppose that we can change the cache design (size of the cache, cache organization) such that we increase H to 0.97, but increase the cache access time to 1.5 ns. What conditions must be met for this change to result in improved performance?

$$T_s = H \times T_1 + (1 - H) \times (T_1 + T_2) = T_1 + (1 - H) \times T_2$$

Solution

$$T_s^1 = 1 + (1 - 0{,}95) \times T_2 = 1 + 0{,}05\,T_2$$

$$T_s^2 = 1{,}5 + (1 - 0{,}97) \times T_2 = 1{,}5 + 0{,}03\,T_2$$

Problem 5

Consider an L1 cache with an access time of 1 ns and a hit ratio of H = 0.95. Suppose that we can change the cache design (size of the cache, cache organization) such that we increase H to 0.97, but increase the cache access time to 1.5 ns. What conditions must be met for this change to result in improved performance?

$$T_s = H \times T_1 + (1 - H) \times (T_1 + T_2) = T_1 + (1 - H) \times T_2$$

Solution

$$T_s^2 = 1{,}5 + (1 - 0{,}97) \times T_2 = 1{,}5 + 0{,}03\,T_2$$

$$T_s^1 = 1 + (1 - 0{,}95) \times T_2 = 1 + 0{,}05\,T_2$$

$$1 + 0{,}05\,T_2 > 1{,}5 + 0{,}03\,T_2 \implies T_2 > 25 \text{ ns}$$

Increasing hit ratio to 0.97 at the cost of increasing cache access time to 1.5 will improve average memory access time in the case that main memory access time is larger than 25 ns.

Problem 6

A Computer has a cache, main memory, and a disk used for virtual memory. If a referenced word is in the cache, 20ns are required to access it. If it is in main memory but not in the cache, 60 ns are needed to load it into the cache, and then the reference is started again. If the word is not in main memory, 12 ms are required to fetch the word from disk, followed by 60 ns to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.6. What is the average time in nanoseconds required to access a referenced word on this system?

Problem 6

A Computer has a cache, main memory, and a disk used for virtual memory. If a referenced word is in the cache, 20ns are required to access it. If it is in main memory but not in the cache, 60 ns are needed to load it into the cache, and then the reference is started again. If the word is not in main memory, 12 ms are required to fetch the word from disk, followed by 60 ns to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.6. What is the average time in nanoseconds required to access a referenced word on this system?

Solution

Let's generalise the formula from Lecture 2 (see Problem 5) to a three level memory including cache, main memory, and disk:

$$T_s = H_1 \times T_1 + (1 - H_1) \times H_2 \times (T_1 + T_2) + (1 - H_1) \times (1 - H_2) \times (T_1 + T_2 + T_3)$$

- T_1, T_2, T_3 : access times to cache, main memory, disk
- H_1, H_2: Hit ration to cache, main memory
- H_1: Probability to hit the cache
- $1 - H_1$: Probability to go to main memory
- $(1 - H_1) \times H_2$: Probability to fetch from main memory
- $(1 - H_1) \times (1 - H_2)$: Probability to fetch from disk

Problem 6

A Computer has a cache, main memory, and a disk used for virtual memory. If a referenced word is in the cache, 20ns are required to access it. If it is in main memory but not in the cache, 60 ns are needed to load it into the cache, and then the reference is started again. If the word is not in main memory, 12 ms are required to fetch the word from disk, followed by 60 ns to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.6. What is the average time in nanoseconds required to access a referenced word on this system?

Solution

Let's generalise the formula from Lecture 2 (see Problem 5) to a three level memory including cache, main memory, and disk:

$$T_S = H_1 \times T_1 + (1 - H_1) \times H_2 \times (T_1 + T_2) + (1 - H_1) \times (1 - H_2) \times (T_1 + T_2 + T_3)$$

- **T_1, T_2, T_3 : access times to cache, main memory, disk**
- **H_1, H_2: Hit ration to cache, main memory**
- **H_1: Probability to hit the cache**
- **$1 - H_1$: Probability to go to main memory**
- **$(1 - H_1) \times H_2$: Probability to fetch from main memory**
- **$(1 - H_1) \times (1 - H_2)$: Probability to fetch from disk**

$$T_S = 0{,}9 \times 20 + 0{,}1 \times 0{,}6 \times 80 + 0{,}1 \times 0{,}4 \times 12000080 = 480026 ns$$