27.05.2020

Istanbul Technical University
Department of Computer Engineering

BLG 322E – Computer Architecture
Recitation 3

Note: If you have questions about the recitation, you may contact the research assistants of the course (ozcelikfu@itu.edu.tr).

QUESTION 1:

A computer system has a **64-KB** (B: Byte) main memory and a **1-KB** cache. Data transfer between main memory and cache is in blocks of **8 bytes**. The cache control unit uses set-associative mapping where each set contains **two frames** (*2-way set associative*). When necessary, **FIFO** is used as the replacement technique.

For read operations **Read Through** is used. For write operations, **Write Through** (WT) with **Write Allocate** (WA) is used.

Cache memory is used only for data, not for instructions.

The cache access time is tc =10 ns, the main memory access time is tm = 50 ns, and the block transfer time between cache and main memory is t_B = 100 ns.

Assume that the cache memory is empty in the beginning.

The CPU performs the arithmetic operation C = A + B, where each variable is one byte.

a) **i)** Assign proper main memory addresses to the three variables in this operation (A, B, and C) so that accesses to them are as fast as possible (best case); give exemplary addresses.

ii) What is the total time spent in accessing these variables in the best case? Write the equation.

b) **i)** Assign main memory addresses to the three variables, A, B, and C, so that accesses to them are as slow as possible (worst case); give exemplary addresses.

ii) What is the total memory access time in the worst case? Write the equation.

SOLUTION 1:

a)

i) In the best case, all variables are in the same block of main memory and in the same cache frame. Same set number and same tag value.

For example:
A: 0000 0000 0000 0000 = $0000
B: 0000 0000 0000 0001 = $0001
C: 0000 0000 0000 0010 = $0002
OR
A: 0000 0000 0000 1000 = $0008
B: 0000 0000 0000 1001 = $0009
C: 0000 0000 0000 1010 = $000A

ii) Operations:
Generate the address of A: miss. Transfer a block from main memory to cache, and at the same time, get A.
Read B: hit.
Write C: hit, write to cache and main memory (because of WT).

ta = tB + tc + tm = 100 + 10 + 50 = 160 ns

b)

i) In the worst case, all variables try to share the same set, but they are in different blocks of main memory. Set numbers are the same, but their tags are different.

16 bits

Tag	set num.	word num.
7 bits	6 bits	3 bits

For example:
A: 0000 0000 0000 0000 = $0000
B: 0000 0010 0000 0000 = $0200
C: 0000 0100 0000 0000 = $0400

ii) Operations:
Generate address of A: miss. Transfer the block from main memory to cache; at the same time, get A (set:0, frame=0).
Generate address of B: miss. Transfer the block from main memory to cache; at the same time, get B (set:0, frame=1).
Generate address of C: miss. Transfer block from main memory to cache (because of WA) (set:0, frame=0); write to cache and main memory (because of WT).

$t_a = t_B + t_B + t_B + t_m = 100 + 100 + 100 + 50 = 350$ ns

QUESTION 2:

a) **Note:** in RAID 4, disks operate independently (not synchronized). Large strips (blocks) are used. Draw a RAID 4 system with a total of 4 disks (data + parity), and distribute 6 blocks (block 0 – block 5) over the disks. Assume that the access time for each disk is **ta.**

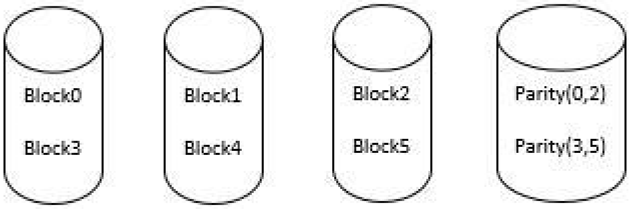
i) How long does it take to read words from two blocks (for example, block 0 and block 4) in two different disks? (One word from block 0, one word from block 4)

ii) How long does it take to update (write) words of two blocks (for example, block 0 and block 4) in two different disks? Explain. (One word in block 0 and one word in block 4)

b) Answer the questions in **Part (a)** (**i** and **ii**) for the RAID 5 system.

SOLUTION 2:

a)



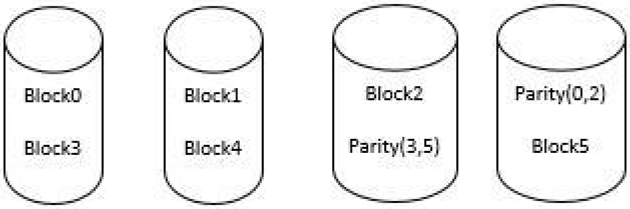
i) Two different blocks from two different disks can be read in **ta.**

ii) Two read and two write operations (**2ta**) should be performed for an update operation in RAID 4 (see the lecture notes). Since parity update operations cannot be performed independently (in parallel) (there is only one parity disk), it takes **4ta** to update words of two blocks in two different disks.

Updating Block0: Read Block0 and Parity(0,2): **ta**
Update Block0 and Parity(0,2): **ta**
Total: **2ta**
Updating Block4: Read Block4 and Parity(3,5): **ta**
Update Block4 and Parity(3,5): **ta**
Total: **2ta**

Total: **4ta**

b)



i) Same as in RAID 4: **ta**.

ii) For each data update, two read and two write operations are necessary. Different from RAID4, now parity update operations can be performed in parallel, because parity strips are distributed to different disks: **2ta**.

Updating Block0: Read Block0 and Parity(0,2): **ta**
Update Block0 and Parity(0,2): **ta**
Total: **2ta**
Updating Block4: Read Block4 and Parity(3,5): **ta**
Update Block4 and Parity(3,5): **ta**
Total: **2ta**

For two update operations, different disks are accessed, so these operations can be performed in parallel.
Total: **2ta**