# Ve370 Introduction to Computer Organization

Homework 6

杨毅文 519370910053

# Assigned: November 11, 2021

**Due: 2:00pm on November 18, 2021 Submit a PDF file on Canvas**

1. (10 points) The following code is written in C, where elements within the same row are stored contiguously. Assume each word is a 32-bit integer.

for (I=0; I<8; I++)

for (J=0; J<8000; J++) A[I][J]=B[I][0]+A[J][I];

* 1. Which variable references exhibit temporal locality? (5 points)

I, J, B[I][0]

* 1. Which variable references exhibit spatial locality? (5 points)

A[I][J]

1. (40 points) Below is a list of 32-bit memory address references, given as word addresses: 0x03, 0xB4, 0x2B, 0x02, 0xBF, 0x58, 0xBE, 0x0E, 0xB5, 0x2C, 0xBA, 0xFD
   1. For each of these references, identify the tag and the cache index given a direct-mapped cache with 8 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty. (10 points)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0x03 | 0xB4 | 0x2B | 0x02 | 0xBF | 0x58 | 0xBE | 0x0E | 0xB5 | 0x2C | 0xBA | 0xFD |
| tag | 0 | 10110 | 101 | 0 | 10111 | 10110 | 10111 | 1 | 10110 | 101 | 10111 | 11111 |
| index | 11 | 100 | 11 | 10 | 111 | 0 | 110 | 110 | 101 | 100 | 10 | 101 |
| h/m | miss | miss | miss | miss | miss | miss | miss | miss | miss | miss | miss | miss |

1. For each of these references, identify the tag and the cache index given a direct-mapped cache with two-word blocks and a total size of 4 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty. (10 pints)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0x03 | 0xB4 | 0x2B | 0x02 | 0xBF | 0x58 | 0xBE | 0x0E | 0xB5 | 0x2C | 0xBA | 0xFD |
| tag | 0 | 10110 | 101 | 0 | 10111 | 10110 | 10111 | 1 | 10110 | 101 | 10111 | 11111 |
| index | 01 | 10 | 01 | 01 | 11 | 00 | 11 | 11 | 10 | 10 | 01 | 10 |
| h/m | miss | miss | miss | miss | miss | miss | hit | miss | hit | miss | miss | miss |

1. You are asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with a total of 8 words of data: C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 35 cycles, and C1 has an access time of 2 cycles, C2 takes 3 cycles, and C3 takes 5 cycles, which is the best cache design? (20 points)
2. For C1, miss rate = 100%, and for C2, miss rate = 10/12 = 83.3%.

For C3, miss rate = 11/12 = 91.7%

Then C2 is the best.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0x03 | 0xB4 | 0x2B | 0x02 | 0xBF | 0x58 | 0xBE | 0x0E | 0xB5 | 0x2C | 0xBA | 0xFD |
| tag | 0 | 10110 | 101 | 0 | 10111 | 10110 | 10111 | 1 | 10110 | 101 | 10111 | 11111 |
| index | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| h/m | miss | miss | miss | miss | miss | miss | hit | miss | miss | miss | miss | miss |

2) C1: 35 12 + 2 12 = 444 cycles

C2: 35 10 + 3 12 = 386 cycles

C3: 35 11 + 5 12 = 445 cycles

Then C2 is the best.

1. (50 points) For a direct-mapped cache design with a 32-bit byte address, the following bits of the address are used to access the cache.

|  |  |  |
| --- | --- | --- |
| Tag | Index | Offset |
| 31–10 | 9–5 | 4–0 |

* 1. What is the cache block size (in words)? (5 points)

8

* 1. How many blocks does the cache have? (5 points)

32

* 1. What is the ratio between total bits required for such a cache implementation over the data storage bits? (5 points)

(32\*8\*32+32\*22+32\*1)/(32\*8\*32)=1.09

Beginning from power on, the following byte addresses for cache references are recorded.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | | | | | | | | | | | |
| 0x00 | 0x04 | 0x10 | 0x84 | 0xE8 | 0xA0 | 0x400 | 0x1E | 0x8C | 0xC1C | 0xB4 | 0x884 |

* 1. (20 points) For each reference, list
     1. its tag, index, and offset
     2. whether it is a hit or a miss, and
     3. How many blocks were replaced (if any)?

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0x00 | 0x04 | 0x10 | 0x84 | 0xE8 | 0xA0 | 0x400 | 0x1E | 0x8C | 0xC1C | 0xB4 | 0x884 |
| tag | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 11 | 0 | 10 |
| index | 0 | 0 | 0 | 100 | 111 | 101 | 0 | 0 | 100 | 0 | 101 | 100 |
| offset | 0 | 1 | 100 | 1 | 10 | 0 | 0 | 111 | 11 | 111 | 101 | 1 |
| h/m | miss | hit | hit | miss | miss | miss | miss | miss | hit | miss | hit | miss |
| replace |  |  |  |  |  |  | yes | yes |  | yes |  | yes |

* 1. What is the hit ratio? (5 points)

4/12 = 33.3%

* 1. Show the final state of the cache, with each valid line represented as <index, tag, data>. (10 points)

<000, 11, Mem[1100000xxxxx]>

<100, 10, Mem[1000100xxxxx]>

<111, 0, Mem[0000111xxxxx]>

<101, 0, Mem[0000101xxxxx]>