

VE370 Homework 6 Wang Lan 519370910084

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)

(2) Which variable references exhibit spatial locality? (5 points)

1) I, J, B[1][0]

2) A[I][J]

2. (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)

(2) 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 points)

(3) 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)

		tag	index	H/M
1) 0x03	= 0000 0011	00000	011	M
0xB4	= 1011 0100	10110	100	M
0x2B	= 0010 1011	00101	011	M
0x02	= 0000 0010	00000	010	M
0xBF	= 1011 1111	10111	111	M
0x58	= 0101 0000	01011	000	M
0xBE	= 1011 1110	10111	110	M
0x0E	= 0000 1110	00001	110	M
0xB5	= 1011 0101	10110	101	M
0x2C	= 0010 1100	00101	100	M
0xBA	= 1011 1010	10111	010	M
0xFD	= 1111 1101	11111	101	M

		tag	index	M/H
C2	0x03 = 0000 0011	00000	01	M
	0xB4 = 1011 0100	10110	10	M
	0x2B = 0010 1011	00101	01	M
	0x02 = 0000 0010	00000	01	M
	0xBF = 1011 1111	10111	11	M
	0x58 = 0101 1000	01011	00	M
	0xBE = 1011 1110	10111	11	H
	0x0E = 0000 1110	00001	11	M
	0xB5 = 1011 0101	10110	10	M
	0x2C = 0010 1100	00101	10	M
	0xBA = 1011 1010	10111	01	M
	0xF0 = 1111 1101	11111	10	M

(3)		tag	index	M/H
C3	0x03 = 0000 0011	00000	0	M
	0xB4 = 1011 0100	10110	1	M
	0x2B = 0010 1011	00101	0	M
	0x02 = 0000 0010	00000	0	M
	0xBF = 1011 1111	10111	1	M
	0x58 = 0101 1000	01011	0	M
	0xBE = 1011 1110	10111	1	H
	0x0E = 0000 1110	00001	1	M
	0xB5 = 1011 0101	10110	1	M
	0x2C = 0010 1100	00101	1	M
	0xBA = 1011 1010	10111	0	M
	0xF0 = 1111 1101	11111	1	M

Miss Rate : C1 : 100%

$$C2: \frac{10}{12} = 83.3\% \Rightarrow C2$$

$$C3: \frac{11}{12} = 91.7\%$$

Time : C1 :  $2 \times 12 + 35 \times 12 = 444$  cycles

C2 :  $3 \times 12 + 10 \times 35 = 386$  cycles  $\Rightarrow C2$

C3 :  $5 \times 12 + 11 \times 35 = 445$  cycles

3. (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)
- (2) How many blocks does the cache have? (5 points)
- (3) What is the ratio between total bits required for such a cache implementation over the data storage bits? (5 points)

$$\begin{aligned}
 (1) \quad & 2^5 = 32 \text{ bytes} = 8 \text{ words} \\
 (2) \quad & 2^5 = 32 \\
 (3) \quad & \frac{32 \times (1 + 22 + 32 \times 8)}{32 \times (32 \times 8)} = 1.09
 \end{aligned}$$

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

- (4) (20 points) For each reference, list
  - a) its tag, index, and offset
  - b) whether it is a hit or a miss, and
  - c) How many blocks were replaced (if any)?
- (5) What is the hit ratio? (5 points)
- (6) Show the final state of the cache, with each valid line represented as <index, tag, data>.

(10 points)

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

tag	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x0	0x0	0x3	0x0	0x2
index	00000	00000	00000	00100	00111	00101	00000	00000	00100	00000	00101	00100
offset	00000	00100	10000	00100	01000	00000	00000	11110	01100	11100	10010	00100

H/M      M    H    H    M    M    M    M    H    M    H    M

4 blocks    (For tag, higher bits are omitted)

$$(5) \frac{4}{12} = 33.3\%$$

- (6)
  - 00000    0x3    Mem[0x100] - Mem[0xC1F]
  - 00100    0x2    Mem[0x880] - Mem[0x89F]
  - 00101    0x0    Mem[0xA00] - Mem[0xBF]
  - 00111    0x0    Mem[0xD00] - Mem[0xFF]