Lebanese American University
School of Arts and Sciences
Department of Computer Science and Mathematics

CSC 320 - Computer Organization

Problem Set 7: Large and Fast: Exploiting Memory Hierarchy

## Exercise 1

In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously.

| a. | for $(\mathrm{I}=0 ; \mathrm{I}<8 ; \mathrm{I}++)$ <br> for $(\mathrm{J}=0 ; \mathrm{J}<8000 ; \mathrm{J}++)$ <br> $\mathrm{A}[\mathrm{I}][\mathrm{J}]=\mathrm{B}[\mathrm{I}][0]+\mathrm{A}[\mathrm{J}][\mathrm{I}] ;$ |
| :---: | :--- |
| b. | for $(\mathrm{J}=0 ; \mathrm{J}<8000 ; \mathrm{J}++)$ <br> for $(\mathrm{I}=0 ; \mathrm{J}<8 ; \mathrm{I}++)$ <br> $\mathrm{A}[\mathrm{I}][\mathrm{J}]=\mathrm{B}[\mathrm{I}][0]+\mathrm{A}[\mathrm{J}][\mathrm{I}] ;$ |

1.1 How many 32-bit integers can be stored in a 16-byte cache line?

Solution:
4
1.2 References to which variables exhibit temporal locality?

Solution:
a. I, J
b. $\mathrm{B}[I][0]$
1.3 References to which variables exhibit spatial locality?

Solution:

| a. | $\mathrm{A}[\mathrm{I}][\mathrm{J}]$ |
| :---: | :--- |
| b. | $\mathrm{A}[\mathrm{J}][\mathrm{I}]$ |

Locality is affected by both the reference order and data layout. The same computation can also be written below in Matlab, which differs from C by contiguously storing matrix elements within the same column.
$\square$
1.4 How many 16-byte cache lines are needed to store all 32-bit matrix elements being referenced?

Solution:

| a. | $8 \times 8000 / 4 \times 2-8 \times 8 / 4+8000 / 4=33984$ |
| :---: | :--- |
| b. | $8 \times 8000 / 4 \times 2-8 \times 8 / 4+8 / 4=31986$ |

1.5 References to which variables exhibit temporal locality?

Solution:
a. I, J
b. $\quad$ I, J, B(I, 0)
1.6 References to which variables exhibit spatial locality?

Solution:

| a. | A(J, I) |
| :---: | :--- |
| b. | A(I, J), A(J, I), B(I, 0) |
|  |  |

## Exercise 2

For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

|  | Tag | Index | Offeset |
| :---: | :---: | :---: | :---: |
| a. | $31-10$ | $9-5$ | $4-0$ |
| b. | $31-12$ | $11-6$ | $5-0$ |

2.1 What is the cache line size (in words)?

Solution:

| a. | 8 |
| :---: | :--- |
| b. | 16 |

2.2 How many entries does the cache have?

Solution:

| a. | 32 |
| :---: | :--- |
| b. | 64 |

2.3 What is the ratio between total bits required for such a cache implementation over the data storage bits?

Starting from power on, the following byte-addressed cache references are recorded

| Address | 0 | 4 | 16 | 132 | 232 | 160 | 1024 | 30 | 140 | 3100 | 180 | 2180 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

Solution:
a. $1+(22 / 8 / 32)=1.086$

| b. | $1+(20 / 8 / 64)=1.039$ |
| :--- | :--- |

2.4 How many blocks are replaced?

In the following parts, consider a direct mapped cache with 64 blocks and 16 bytes/block

| Adress | 0 | 4 | 16 | 132 | 232 | 160 | 1024 | 30 | 140 | 3100 | 180 | 2180 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| Line ID | 0 | 0 | 1 | 8 | 14 | 10 | 0 | 1 | 8 | 1 | 11 | 8 |
| Hit/miss | M | H | M | M | M | M | M | H | H | M | M | M |
| Replace | N | N | N | N | N | N | Y | N | N | Y | N | Y |

2.5 What is the hit ratio?

Solution: 0.25
2.6 List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

Solution:

```
<0000012, 00012, mem[1024]>
<0000012, 00112, mem[16]>
<0010112, 00002, mem[176]>
<0010002, 00102, mem[2176]>
<0011102,00002, mem[224]>
<0010102,00002, mem[160]>
```

