CSCI 4591: Computer Architecture
HW Assignment # 8
Due Date: May 2, 2021 @ 11:55 PM

Problem 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. Assume each word is a 64-bit integer.

```
for (I = 0; I < 8; I++)
   for (J = 0; J < 8000; J++)
      A[I][J] = B[I][0] + A[J][I];
```

A) How many 64-bit integers can be stored in a 16-byte cache block?

64/8 = 8. 16/8 = 2 integers

B) Which variable references exhibit temporal locality?

I and J and `B[I][0]`

C) Which variable references exhibit spatial locality?

A[I][J] and `B[I][0]`

Problem 2) By convention, a cache is named according to the amount of data it contains (i.e., a 4 KB cache can hold 4 KB of data); however, caches also require SRAM to store metadata such as tags and valid bits. For this exercise, you will examine how a cache's configuration affects the total amount of SRAM needed to implement it as well as the performance of the cache. For all parts, assume that the caches are byte addressable, and that addresses and words are 64 bits.

A) Calculate the total number of bits required to implement a 32 KiB cache with two-word blocks.

cache data size = 32 Kib = 32*1024 = 2^{15}
block size = 2 words = 64bits *2 = 2^4
total number of blocks = cache/block = $2^{15}/2^4 = 2^{11} = 2048$
index = 11

offset = 4

data = 64

tag = 64- (11+4) = 49

valid bits = 1

valid bit + tag + offset(word bits * 64)

1 + 49 + (2*64) = 178

right number of total bits = 178 * 2048 = 364544

or to do it the wrong way but think that this is valid to use elsewhere
($2^{\text{index bits}}$) * (valid bits + tag bits + (data bits * $2^{\text{offset bits}}$) = 2199552 bits

B) Calculate the total number of bits required to implement a 64 KiB cache with 16-word blocks. How much bigger is this cache than the 32 KiB cache described in Exercise A? (Notice that, by changing the block size, we doubled the amount of data without doubling the total size of the cache.)

Word = 8 bytes
block = 16 words * 64 = 1024 → 2^{10}
cache data size = 64 KiB = 64 * 1024 = 2^{16}
total number of blocks = cache/block = $2/2^4 = 2 = 2048$
index = 16

offset = 10

data = 64

tag = 64- (16+4) = 44

valid bits = 1

valid bits + tag + offset
1 + 44 + 10(64) = 685
total bits = 685 * 2048 = 1 402 880

Problem 3) For a direct-mapped cache design with a 64-bit address, the following bits of the address are used to access the cache.

Tag(54)	**Index (5)**	**Offset**
63 – 10	9 – 5	4 – 0

A) What is the cache block size (in words)?
$2^5 = 32 \rightarrow 5$ from number of offset bits
32 bytes / 8 bytes = 4 words

B) How many blocks does the cache have?
$2^5 = 32$ blocks - 5 from number of index bits

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

Total bits = 32 entries *(1 valid bits + 54 tag bits + (64*4 data bits) = 32* 311

#32 entries from the range of index implying that it is a direct mapped cache

Data bits = 32 entries * (64*4 data bits) = 32 * 256
Ratio = (311)/(256) = 1.21

(Tag bits + valid bit + data bits)/data bits