CprE 381 – Computer Organization and Assembly Level Programming

HW-10

[Note from Joe: This is the first of two assignments on memory hierarchy. This assignment focuses on cache architecture and performance issues, while the second will focus more on other programmer's perspective issues.]

Reading: Patterson & Hennessy, Sections 5.1-5.3

1) Principle of Locality

(a) Describe the general characteristics of a program that would exhibit very little temporal and spatial locality with regard to *data* accesses. Provide an example program (pseudocode is fine).

(b) Describe the general characteristics of a program that would exhibit very high amounts of temporal but very little spatial locality with regard to *data* accesses. Provide an example program (pseudocode is fine).

(c) Describe the general characteristics of a program that would exhibit very little temporal but very high amounts of spatial locality with regard to *data* accesses. Provide an example program (pseudocode is fine).

(d) Describe the general characteristics of a program that would exhibit very little temporal and spatial locality with regard to *instruction* accesses. Provide an example program (pseudocode is fine).

(e) Describe the general characteristics of a program that would exhibit very high amounts of temporal but very little spatial locality with regard to *instruction* accesses. Provide an example program (pseudocode is fine).

(f) Describe the general characteristics of a program that would exhibit very little temporal but very high amounts of spatial locality with regard to *instruction* accesses. Provide an example program (pseudocode is fine).

2) Cache Configuration and Performance

(a) Here is a series of address references given as word addresses: 2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6, and 11. Assuming a direct-mapped cache with 16 one-word blocks that is initially empty, label each reference in the list as a hit or a miss and show the final contents of the cache.

(b) Cache C1 is direct mapped with 16 one-word blocks. Cache C2 is direct mapped with 4 four-word blocks. Assume that the miss penalty for C1 is 8 memory bus clock cycles and

the miss penalty for C2 is 11 memory bus clock cycles. Assuming that the caches are initially empty, find a reference string for which C2 has a lower miss rate but spends more memory bus clock cycles on cache misses than C1. Use word addresses.

(c) Consider three processors with different cache configurations:

- *Cache 1*: Direct mapped with one-word blocks
- *Cache 2*: Direct mapped with four-word blocks
- *Cache 3*: Two-way set associative with four-word blocks

The following miss rate measurements have been made:

- *Cache 1*: Instruction miss rate is 4%; data miss rate is 6%.
- *Cache 2*: Instruction miss rate is 2%; data miss rate is 4%.
- *Cache 3*: Instruction miss rate is 2%; data miss rate is 3%.

For these processors, one-half of the instructions contain a data reference. Assume that the cache miss penalty is 6 + Block size in words. The CPI for this workload was measured on a processor with cache 1 and was found to be 2.0. Determine which processor spends the most cycles on cache misses.

3) Breaking Locality

Complete the following C function that scales each element of a 2D array by a scalar. First, complete it in row-major ordering (https://en.wikipedia.org/wiki/Row-_and_column-major_order). Second, complete it in column-major ordering. Which is faster on your computer (report the time each takes to execute 1000 calls to `scale` and what processor model you have)? Why is it faster? Please use the C template provided with this homework, and submit your modified code as `prob3_rowmajor_<NetID>.c` and `prob3_colmajor_<NetID>.c`.

```
void scale(int n, int m, int array[n][m], int scale);
```