Fall ’18 CIS 314 Assignment 7 – 100/100 points – Due Monday, 11/26, 11:59 PM

Please submit individual source files for coding exercises (see naming conventions below) and a single solution document for non-coding exercises (.txt or .pdf only). Your code and answers need to be documented to the point that the graders can understand your thought process. Full credit will not be awarded if sufficient work is not shown.

1. [40] Assume that you’re given the task of optimizing graphics software that operates in the CMYK (Cyan, Magenta, Yellow, Black) color space. Specifically, you need to determine the efficiency of the following algorithm on a machine with a 4096B cache with 64B blocks and 64 sets (i.e., direct mapped). You are given the following definitions:

```
struct ColorPoint {
    long c;
    long m;
    long y;
    long k;
};

struct ColorPoint points[16][16];
```

Assume the following:

- sizeof(long) == 8.
- points begins at memory address 0.
- The cache is initially empty.
- The only memory accesses are to the entries of the array points. Variables *i*, *j*, and *sum* are stored in registers.

Determine the cache performance of the following code:

```
long sum = 0;
for (int i = 0; i < 16; i++) {
    for (int j = 0; j < 16; j++) {
        sum += square[i][j].c;
        sum += square[i][j].m;
        sum += square[i][j].y;
        sum += square[i][j].k;
    }
}
```

A. (15) What is the total number of memory reads? Why? It may help to think in terms of movq instructions.

B. (15) What is the total number of memory reads that miss in the cache? Why?

C. (10) What is the miss rate?

Write your answers in your solutions document.

2 [60] Assume that we're going to simulate a 256B cache with 16B blocks and 16 sets (i.e., direct mapped) for a 32-bit architecture. Write a C program with the following functions, using only bitwise ops for the arithmetic (no division or modulo operators):

- (15) *unsigned int getOffset(unsigned int address)* – returns the byte offset of the address within its cache block.
- (15) *unsigned int getSet(unsigned int address)* – returns the cache set for the address.
- (15) *unsigned int getTag(unsigned int address)* – returns the cache tag for the address.
- (15) A main() function to test your functions.

Here are some test runs:

```
0x12345678: offset: 8 - tag: 123456 - set: 7
0x87654321: offset: 1 - tag: 876543 - set: 2
```

Name your source file 7-2.c.

Zip the source files and solution document (if applicable), name the .zip file <Your Full Name>Assignment7.zip (e.g., EricWillsAssignment7.zip), and upload the .zip file to Canvas (see Assignments section for submission link).