CMPSCI 201 – Fall 2005

Final Exam Part #2 – December 14, 2005

Professor William T. Verts

Open Book, Open Notes!

This part of the exam is taken-home and is due at the time of the final exam on Tuesday, December 20 at 1:30pm, when you will take part 1 of this exam. Over the weekend you may discuss the questions with the other students or work alone, as you choose, but *your answers must be unique and in your own words*.

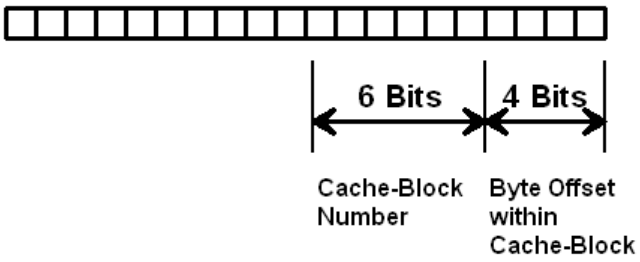
You are allowed and encouraged to type in your answers into a word processor such as Microsoft Word, but if you do please follow these guidelines:

1. Put your name in the header/footer region so it appears on every page,
2. Put automatic page numbers in the header/footer as well,
3. Write your answers in the same order as the questions in this document,
4. Write all text in 12-point Times New Roman, except:
5. Use the `Courier New` typeface for any ARM code,
6. Staple your printout to this exam before you turn it in.

NAME	

PROBLEM	SCORE	POSSIBLE
10		25
TOTAL		25

<10> 25 Points – In a computer with a 20-bit address space (addressing up to 2^{20} bytes), there is a ***direct-mapped*** cache system, arranged as follows:



A. 1 Point – How many blocks in the cache are there?

B. 1 Point – How many bytes are in each block?

C. 1 Point – How many blocks of primary memory map onto each block in the cache?

Currently in primary memory is a small program that starts at location `$00000` and takes up exactly 2 kilobytes of memory just for the executable code. It references a 64-kilobyte two-dimensional array of bytes, called `MyData`, that starts at fixed location `$A0000`. The array contains 256 rows and 256 columns, both indexed by values from 1 to 256, and the cells in the array are stored in row-major order. In Pascal, this would be allocated with the following declaration:

```
Var MyData : Array [1..256,1..256] Of Byte Absolute $A0000 ;
```

Here are two approaches to storing zeroes in every element of the array:

```
For R := 1 To 256 Do
     For C := 1 To 256 Do
          MyData[R,C] := 0 ;
```

```
For C := 1 To 256 Do
     For R := 1 To 256 Do
          MyData[R,C] := 0 ;
```

D. 1 Point – What is the mapping function for the array, which maps row `R` and column `C` onto a linear byte offset (0…65535) relative to the start of the array in physical memory?

E. 1 Point – How many blocks of primary memory does the array occupy?

F. 1 Point – How many blocks of the array map onto each corresponding cache block?

G. 1 Point – Which block numbers in the cache will be used by the array?

H. 1 Point – Which block numbers in the cache will be used by the program?

I. 7 Points – Essay: Of the two approaches to clearing the array shown above, which one will result in the least amount of "cache thrashing"? Why?

J. 10 Points – Essay: How would changing the cache mechanism to ***two-way set associative*** improve overall performance in this particular example?