CS 351 Exam 1, Section 1 Wed. 10/5/2016

Name: ______________________________

Rules and Hints

- You may use one handwritten 8.5 × 11″ cheat sheet (front and back). This is the only additional resource you may consult during this exam. *No calculators.*
- You may write your answers in the form *[mathematical expression][units]*. There is no need to actually do the arithmetic.
- Write your answers on scratch paper. Clearly label your answer to each question.

Grade

	Your Score	**Max Score**
Problem 1: Short answer		22
Problem 2: Processor performance and power		24
Problem 3: Assembly arrays		22
Problem 4: Assembly functions		32
Total		**100**

Problem 1: Short answer (22 points)

Part A (2 points)

+1 extra credit point if entire class gets it right.
How many bits are in a byte?

Part B (6 points)

You are designing the next version of the LEGv8 ISA. Your colleague proposes decreasing the number of bits in the opcode for the CB format so that you can have more room for the CondBrAddr field. Give one advantage and one disadvantage of this approach.

Part C (8 points)

Translate these two LEGv8 assembly instructions to 32-bit binary machine code:

```
    CBZ X19, exit
    ADD X10, X19, X20
exit:
```

Part D (6 points)

Write a snippet of LEGv8 code to set the second bit from the right (the second-least significant bit) of register X19 to 1, while preserving the other bits. You can use the temporary registers if you need them, but you should leave the other register values unchanged.

Problem 2: Processor performance and power (24 points)

You have a workload with 1 trillion dynamic instructions. 40% are integer instructions, 30% are floating-point instructions, 10% are branches, and 20% are memory instructions. You want to run it on one core of a processor with the following characteristics:

- Frequency of 2 GHz
- Average CPI of 1 for integer instructions; 2 for branches; 4 for floating-point; 5 for memory
- Static power consumption of 25 W and dynamic power consumption of 20 W/core

Make the overly simplistic assumptions that inactive cores consume 0 dynamic power and that your program is the only one running.

Part A: Execution time (6 points)

What is the execution time of your workload if it runs single-threaded on a single core of this processor?

Part B: Optimization (8 points)

You have two options for optimizing your program. Calculate the new execution time for each option (independently):

- Overclock the processor to 2.5 GHz
- Activate 3 additional cores, splitting the memory and floating-point instructions evenly across all 4 active cores.

Part C: Power and Energy (10 points)

How much power and energy will your processor use in each of the configurations from Part B? Be sure to give units in your answers.

Problem 3: Assembly arrays (22 points)

Translate the following C code to LEGv8 assembly. Assume that *s* is a `char *` in `X19`. At the end of your snippet of code, all of these registers should have their correct values according to the C semantics. Use temporary registers for all intermediate values.

```
while (*s != 0) {
    if (*s >= 97 && *s <= 122)
        *s -= 32;
    s++;
}
```

Problem 4: Assembly functions (32 points)

Translate the following C functions to LEGv8, using only the instructions on your reference sheet. You should translate each function independently and not attempt to combine them. Obey all LEGv8 conventions about functions, registers, and stack usage.

```
unsigned long long f(unsigned long long x, unsigned long long y) {
    return (x + y) / 2;
}

unsigned long long g(unsigned long long x, unsigned long long y) {
    return f(x, 10) + f(y, 10);
}

unsigned long long h (unsigned long long a[], unsigned long long N) {
    unsigned long long sum = 0;
    for (unsigned long long i = 0; i < N; i+= 2) {
        sum += f(a[i], a[i+1]);
    }
    return sum;
}
```