CS 351 Exam 1 Wed. 2/19/2014

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.*
- Include step-by-step explanations and comments in your answers, and show as much of your work as possible, in order to maximize your partial credit.
- You may use the backs of these pages if you need more space, but make it clear where to find your answer to each question.
- You may write your answers in the form *[mathematical expression][units]*. There is no need to actually do the arithmetic.

Grade

	Your Score	**Max Score**
Problem 1: Warmup		10
Problem 2: Speedup and energy		10
Problem 3: Processor performance and power		30
Problem 4: MIPS functions		25
Problem 5: MIPS arrays		25
Total		**100**

Problem 1: Warmup (10 points)

Part A (2 points)

+1 extra credit point if entire class gets it right.

How many bits are in a byte?

Part B (4 points)

Put the following in order from smallest to largest:

- Gigabyte (GB)
- Kilobyte (KB)
- Megabyte (MB)
- Terabyte (TB)

Part C (4 points)

The x86 ISA has an instruction

mov %reg, %reg

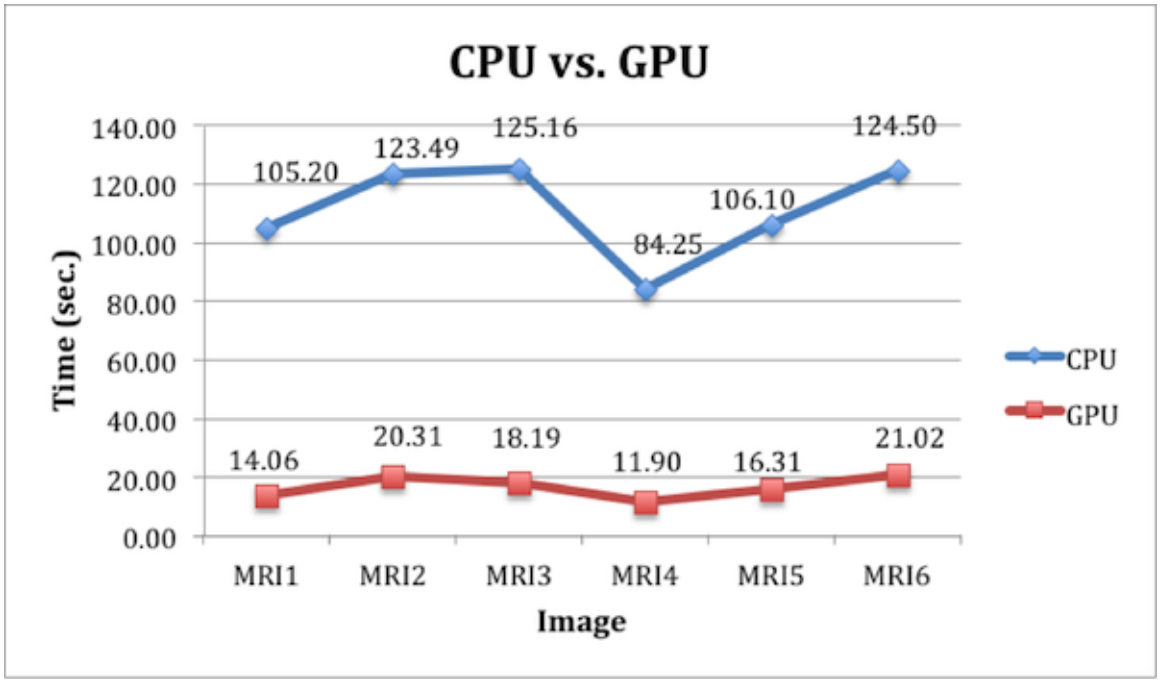
which copies a value from one register to another. Answer the following questions:

- If you were to add this instruction to the MIPS ISA, what encoding format would you use, and how would you assign values to each field?

- Why do you think MIPS doesn't have this instruction?

Problem 2: Speedup and energy (10 points)

Consider the following graph that I stole from the Internet:



Part A: Speedup (5 points)

To process the image MRI1, which is faster: the CPU or the GPU? What is its speedup over the slower one?

Part B: Energy (5 points)

If the CPU consumes an average power of 50 W, and the GPU consumes 150 W, which one uses more energy to process MRI1? Explain/show work.

Problem 3: Processor performance and power (30 points)

You have a workload with the following characteristics:

- 1 billion dynamic instructions
- 30% integer instructions
- 30% floating-point instructions
- 20% branches
- 20% loads and stores
- The integer and floating-point instructions are perfectly parallelizable. In other words, if you have N cores, you can have all cores executing $\frac{1}{N}$ of the original instructions at the same time. However, the branches and load/stores are serial.

You have a processor with the following characteristics:

- 4 cores
- Clock speed of 2.8 GHz
- Operating voltage of 1.5 V
- Static power of 20 W
- Dynamic power of 60 W

Part A: Execution time (10 points)

What is the execution time of your workload on this processor?

Parts B and C: Optimization

The processor manufacturer is considering two new versions of this processor:

Option A An 8-core version with a dynamic power of 90W and the same static power and clock speed.

Option B A 3.2 GHz version with an operating voltage of 1.75 V and the same static power, capacitive load, and number of cores.

Part B: Optimized performance

What are the speedups of Option A over the original processor and Option B over the original processor?

Part C: New power consumption

What is the *total power consumption* of Option A? What about Option B?

Problem 4: MIPS functions (25 points)

Translate the following C code to MIPS. Obey all MIPS conventions about functions, registers, and stack usage.

```
int f(int a) {
    if (a < 2) return a;

    return a % 2 + f(a / 2);  // note the recursive call
}
```

Bonus (3 points)

Provide a concise description in English of what this function computes.

Problem 5: MIPS arrays (25 points)

Translate the following C code to MIPS. Obey all MIPS conventions about functions, registers, and stack usage. Assume that the array `a` is an array of **characters** and is located at the top of the stack.

```
for (int i=0; i < 100; i++) {
    if (a[i] == 'e' || a[i] == 'E')
        a[i] = '3';  // so l33t
}
```