The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Comp 411 Computer Organization
Spring 2011

Lab #5: Basic MIPS Assembly
Issued Fri. 2/11/11; Due Fri. 2/18/11 (beginning of lab)

Note: Please submit your answers to Exercise 1, and your code and screenshot of the output for Exercise 2, electronically via Blackboard. These could be separate files, or pasted together into a Word or PDF document.

Exercise 1. "Name that Loop" (20 points)

a) Describe in your own words the function performed by the following code fragment when it reaches the label `end`. Consider the following hints. The code fragment operates on the contents of register a0, and the loop will always complete. Assume a machine with 32-bit registers.

```
          addi  $t0,$0,32
loop:     andi  $t1,$a0,1
          sub   $t0,$t0,$t1
          srl   $a0,$a0,1
          bne   $a0,$0,loop
end:      add   $a0,$t0,$0
```

Answer:

b) Describe how the operation of the above code fragment changes if the `srl` instruction is replaced with an `sra` instruction with the same arguments.

Answer:

c) Describe the function of the following code fragment in your own words:

```
          la    $t1,a
loop:     lw    $t0,4($t1)
          sw    $t0,($t1)
          addi  $t1,$t1,4
          bne   $t0,$0,loop
```

Answer:

d) Describe the function of the following code fragment in your own words:

```
          la    $t1,a
loop:     lw    $t0,($t1)
          addi  $t1,$t1,4
          bne   $t0,$0,loop
          la    $t0,a
          sub   $t0,$t1,$t0
          sra   $t0,$t0,2
```

Answer:

Exercise 2. "A Loop of Your Own" (80 points)

Download and install MARS, the MIPS instruction set simulator (see the class website for download link). Then, write the assembly program described below.

You are given an array called `pixels`, each element of which is a 32-bit word representing a color value. The lower significant 8 bits of each color value denote an unsigned integer (from 0 to 255) representing the color's "blue" value, the next 8 bits are the "green" value, the next 8 bits are the "red" value, and the most significant 8 bits are all zeroes. For example, the pixel with value 0x0001ff22 has color components:

red = 1 (or 0x01), green = 255 (or 0xff), blue = 34 or (0x22)

For this programming assignment, you will read through this array of pixels, and for each pixel, convert the color pixel into a grayscale pixel using a simple formula:

gray value = (red + green + blue) / 3

For the above example, the gray value would be (1+255+34) / 3 or 96.

After calculating the gray value for a pixel, print it out to the console (only one element per line). The expected output is given in the comments in the starter file on the course website.

Hint: Study and understand the divide instruction thoroughly before attempting to use it!

Submission: A starter assembly file is provided for you on the class website. Fill in your name and onyen in the comments at the top. Add your code where indicated; do not change anything else in the file! When you have completed your assignment, submit the code and a picture the program's output, along with your answers to Exercise 1, electronically via Blackboard.

Grading: Your program will be graded as correct (full-credit), partially correct (half-credit), or incorrect (no credit). Get help early if necessary!