

CDA 3101 Computer Organization

Assignment

Assignment 1

(5 points) This assignment exercises concepts in assembly languages and numeric representation. Although you are encouraged to study with others and to cooperate in understanding the textbook material, you are to work on these exercises individually. You will not receive credit for answers unless you show all corresponding work and calculations.

Read the questions carefully, and be sure to **answer the question**, including all parts of the question.

This assignment is due at the beginning of recitation on Tuesday, September 20, 2005.

Name: _____ Section: _____

1. Below is a MIPS assembly function called "tweak".

```
tweak:    slt   $s0, $t0, $t1
          beq   $s0, $0, twink
          addu  $t0, $t0, $t1
          j     done
twink:    subi  $t0, $t1, 1
done:     jr    $ra
```

- a. Given starting values of $\$t0 == 2$ and $\$t1 == 3$, what is $\$t0$ after the function is executed?
 - b. As it turns out, "beq" is a pseudo-instruction. That is, "beq" is not part of the MIPS instruction set, but most MIPS assemblers will convert it into actual MIPS instructions. Rewrite the above "dawdle" function, replacing the "beq" command with actual MIPS instructions that accomplish the same function. Do not use more than four total instructions to replace the one "beq". (Hint: "bne" is an actual MIPS instruction.)
2. Suppose you wish to load the byte at the very end of the 32-bit memory size. That is, the address you wish to access has all 32 bits set to '1' (equivalent to $2^{32} - 1$ in decimal). Eventually you wish to execute the instruction

```
lb $t0, $t1    # load byte from memory at address $t1
```

where the register $\$t1$ contains the "all 1" address. Assume that you cannot obtain the address from memory storage.

- a. Show the minimal MIPS instructions necessary to set the value of $\$t1$ before you execute the above load. Hint: the binary number consisting of 16 bits all set to '1' is equivalent to 65535 in decimal.

- b. What is it about the fields in the instruction format of MIPS arithmetic instructions that requires you to use the instructions that you did in setting \$t1?
3. Convert the following base-10 decimal numerals to the equivalent base-2 binary numerals. Use two's complement notation, and you may restrict your answer to 16 bits.
- $(49)_{10}$
 - $(-101)_{10}$
 - the smallest negative number which can be represented in 16-bit two's complement notation.
4. Convert the following 32-bit base-2 binary numerals (in two's complement notation) to the equivalent base-10 decimal numerals.
- $(0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010\ 1010)_2$
 - $(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 1011)_2$
 - $(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1001\ 1000)_2$
 - $(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111)_2$
5. Convert the following unsigned base-2 binary numeral
- $$1110\ 0101\ 0010\ 0111\ 1000\ 1010\ 0110\ 1101$$
- to its equivalent base-8 octal numeral,
 - to its equivalent base-16 hexadecimal numeral.
6. Convert the following unsigned base-16 hexadecimal numeral
- $$3af72d04$$
- to its equivalent base-2 binary numeral
 - to its equivalent base-8 octal numeral
7. Add the following pair of two's complement 16-bit binary numerals.
- $$\begin{array}{r} 0111\ 1111\ 1010\ 0110 \\ +0000\ 0000\ 0111\ 1001 \\ \hline \end{array}$$
- Give the 16-bit two's complement result.
 - Did an overflow occur? Explain what indicates that?
8. Subtract the following pair of two's complement 16-bit binary numerals.

$$\begin{array}{r} 0000\ 0000\ 1010\ 0110 \\ -0000\ 0000\ 0001\ 0001 \\ \hline \end{array}$$

Give the 16-bit two's complement result. (Hint: since the numerals are in two's complement notation you may wish to convert one of the numerals).

9. Convert the base-10 decimal numeral $(-10.5)_{10}$ to single-precision IEEE Floating Point Standard format.
 - a. What is the S, F, and E value?
 - b. Give your final result in a 32-bit, base-2 binary numeral
 - c. Give your final result in a base-16 hexadecimal numeral
10. Convert the following IEEE Floating Point Standard format value into the equivalent base-10 decimal numeral.

$$(1FA00000)_{16}$$