COMP2611: Computer Organization and Design Spring 2015

Problem Set 1

Prepared by: Prof. Brahim BENSAOU CSE, HKUST

Warning:
• This is not meant to be a model of the exam paper. It is just a set of exercises for you to practice "thinking" the solutions in assembly language.

Contents:
• Number representation, MIPS coding.

Exercise 1:

The following program tries to copy a series of words starting at the one whose address is in register $a0 to another series of memory words starting at the address stored in register $a1, counting the number of words copied in register $v0, and stopping the copy when it finds a word equal to 0.

```
Loop:lw   $v1, 0($a0)
     addi $v0, $v0, 1
     sw   $v1, 0($a1)
     addi $a0, $a0, 1
     addi $a1, $a1, 1
     bne $v0, $zero, loop
```

The program has bugs. Fix the bugs and explain the reason of your choice. Test your version on Mars to make sure it is correct.

Exercise 2:

Consider the following two MIPS procedures used to clear an array of integers (i.e., set its elements to 0).

The first version called clearA uses the C array approach while the second called clearP uses the C pointers version. Give two sequences of MIPS code that implement the two procedures as faithfully as possible to their C code versions. Assume the array base address and its size are available in $a0 and $a1 respectively.

```
clearA(int array[], int size)
{
        int i;
        for (i = 0; i < size; i+= 1) array[i] = 0;
}

clearP(int *array, int size)
{
        int *p;
        for (p = &array[0]; p < &array[size]; p = p + 1) *p = 0;
}
```

Exercise 3:

a) Convert the following decimal numbers into their 32-bit two's complement binary representation
$+4096_{ten}$
-2047_{ten}
$-2{,}000{,}000_{ten}$

b) What decimal number do these two's complement binary number represents?
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 1111_{two}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 1111_{two}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 0000\ 0110_{two}$

c) What is the value of the IEEE754 floating point number: 0x40C00000

d) Assume this number is stored in memory in a word labelled "ieee:" in the data segment. Write a short sequence of MIPS instructions to multiply this number by 2^i, where i is an integer stored in register $t1 with $1 < i < 5$.

Exercise 4:

The following instruction does not exist in MIPS. It takes a register (e.g., $t2) and returns its absolute value in the same resister
abs $t2
Give the minimal sequence of real MIPS instructions to implement this operation. Hint: it can be done in 3 instructions

Exercise 5:

MIPS does not cause an exception if there is an overflow with unsigned arithmetic operations. For example if addu $t0, $t1, $t2 is executed and the result of the addition is greater than $2^{32} - 1$, the program still continues execution.

a) From the above, using only boolean operations express the condition on which the addition of two unsigned numbers in registers $t1 and $t2 runs into an overflow.

b) Provide a MIPS sequence of instructions to detect if an overflow took place after addu is executed.

```
addu $t0, $t1, $t2
...
```

Exercise 6:

The following code fragment processes an array and produce two important values in registers $v0 and $v1 . Assume that array consists of 5000 words indexed through 4999. and its base address is stored in $a0

and its size (5000) in $a1. Describe in one sentence what this code does. Specifically, what will be returned in $v0 and $v1

```
        add  $a1, $a1, $a1
        add  $a1, $a1, $a1
        add  $v0, $zero, $zero
        add  $t0, $zero, $zero
outer:  add  $t4, $a0,  $t0
        lw   $t4, 0($t4)
        add  $t5, $zero, $zero
        add  $t1, $zero, $zero
inner:  add  $t3, $a0, $t1
        lw   $t3, 0($t3)
        bne  $t3, $t4, skip
        addi $t5, $t5, 1
skip:   addi $t1, $t1, 4
        bne  $t1, $a1, inner
        slt  $t2, $t5, $v0
        bne  $t2, $zero, next
        add  $v0, $t5, $zero
        add  $v1, $t4, $zero
next:   addi $t0, $t0, 4
        bne  $t0, $a1, outer
```

Exercise 7:

You should do all the take-home problems provided at the end of each tutorial.