**CPRE 381- Intro to Computer Organization & Implementation**

**HW4**

**Due Date: February 27, 2017**

Name: Ningyuan Zhang

Section: A

1. Consider the factorial *fact* program given in Appendix A.6 on Pages A-27-A29. Make it more efficient by decreasing its instruction count. What address is this program stored in the memory (Check in Spim)? Is it the starting address of the text segment given in Figure A.5.1 on Page A-21? **[10 points]**

Addresses are different between different computers. There are a lot of methods to make the program more efficient by elimination the frame pointer or using loop.

1. Consider the program:

.data

myArray: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

.text

la $s0, myArray

li $s1, 0

loop: sll $t0, $s1, 2

add $t0, $t0, $s0

lw $s2, 0($t0)

lw $s3, 4($t0)

add $s2, $s2, $s3

sw $s2, 0($t0)

addi $s1, $s1, 1

slti $t1, $s1, 9

bne $t1, $zero, loop

.end

Explain what does this program do? How is the data bound from the .data segment to the base address register $s0? What address does Spim use for 0th element of array in $s0? **[10 points]**

**User sata segment [10000000] . . [10040000]**

[10000000] . . [1000ffff] 00000000

[10010000] 00000001 00000002 00000003 00000004 . . . . . . . .

[10010010] 00000005 00000000 00000000 00000000 . . . . . . . .

[10010020] . . [1003ffff] 00000000

The program that given us Is a for loop, it increases elements of an array by 1. And the details about the addresses were explained through the machine code that I typed above.

1. Someone suggested that unsigned overflow on addition of A+B can be checked by checking if (((A+B) < A) AND ((A+B)<B)). Is it true? Prove it. Keep in mind that on overflow you lose the 33rd bit losing 232 value. Specifically, on unsigned overflow your result is only A+B-232. Write a MIPS assembly program to check for unsigned overflow using this rule. **[10 points]**

The statement is true. Let C = A + B. Assume that (C<A) AND (C<B). Because this is unsigned addition, C must lose the 33rd bit to let itself have less value than both A and B.

MIPS code:

Assum: $t1 = A, and $t2 = B

addu $t0, $t1, $t2 # $t0 = A+B

slt $t3, $t0, $t1 # $t3 =1 if A+B <A

slt $t4, $t0, $t2 # $t4 = 1 if A+B <B

and $t5, $t3, $t4 # $t5 = 1 if A+B <A and A+B <B

beg $t5, $zero, NOF # $t5 =0 , jump to No Over flow branch

j OF