

Name: KEY

Id#

## ICS 233, Term 063

### Computer Architecture & Assembly Language

#### Quiz# 2

Date: Monday, July 30, 2007

**Q1.** Assume that you have a two-dimensional array of integers, **ITArray**, and that you are required to write a procedure, **TArraySum**, to compute the sum of all the integers in the array and return the result in \$v0. Assume that the address of the array, its number of rows and its number of columns will be passed in registers \$a0, \$a1, and \$a2, respectively. Implement the following algorithm for **TArraySum**:

```
Sum=0
For (i=0; i<#rows; i++){
    Sum = Sum + RowSum(i)
}
```

**RowSum** is to be implemented as another procedure that receives the address of the array, the row number to be added and its number of columns in registers \$a0, \$a1, and \$a2, respectively, and will return the sum of the integers in a row in \$v0. Write a program to use the procedure **TArraySum** to compute the sum of the array given below and then display it:

```
ITArray: .word 1, 2, 3, 4, 5
         .word 6, 7, 8, 9, 10
         .word 11, 12, 13, 14, 15
```

Note that the sum displayed in this case should be 120.

```
##### Data segment #####
.data
ITArray: .word 1, 2, 3, 4, 5
         .word 6, 7, 8, 9, 10
         .word 11, 12, 13, 14, 15

##### Code segment #####
.text
.globl main
main: # main program entry

# Computing the array sum
    la $a0, ITArray
```

```
li $a1, 3
li $a2, 5
jal TArraySum
```

```
# Displaying the sum
move $a0, $v0
li $v0, 1
syscall
```

```
li $v0, 10    # Exit program
syscall
```

# RowSum is a procedure that receives the address of the array, the row number to be added  
# and its number of columns in registers \$a0, \$a1, and \$a2, respectively, and will  
# return the sum of the integers in a row in \$v0.

RowSum:

```
move $t2, $a2
xor $v0, $v0, $v0    # sum=0
mul $t0, $a1, $a2    # compute starting address of row
sll $t0, $t0, 2
add $t0, $t0, $a0
```

NextC:

```
lw $t1, ($t0)
add $v0, $v0, $t1
addi $t0, $t0, 4      # increment address to next column
addi $t2, $t2, -1
bnez $t2, NextC
jr $ra
```

# TArraySum is a procedure that receives the address of the array, its number of rows  
# and its number of columns in registers \$a0, \$a1, and \$a2, respectively, and will  
# return the sum of the integers in the array in \$v0.

TArraySum:

```
addi $sp, $sp, -4    # pushing return address on stack
sw    $ra, ($sp)
move $s2, $a1
xor $s1, $s1, $s1    # sum=0
xor $s0, $s0, $s0    # loop counter
```

ForLoop:

```
move $a1, $s0
jal RowSum
add $s1, $s1, $v0
addi $s0, $s0, 1
blt $s0, $s2, ForLoop
move $v0, $s1
lw $ra, ($sp)
addi $sp, $sp, 4    # popping return address from stack
jr $ra
```