**Lab Assignment 7: SPIM**

1. **Write MIPS assembly code to add two positive integers.**

#add two numbers

.text #text section

.globl main #call main by SPIM

main: la $t0, value #load address 'value' into $t0

 lw $t1, 0($t0) #load word 0(value) into $t1

 lw $t2, 4($t0) #load word 4(value) into $t2

 add $t3, $t1, $t2 #add two registers $t1 and $t2 temp store $t3

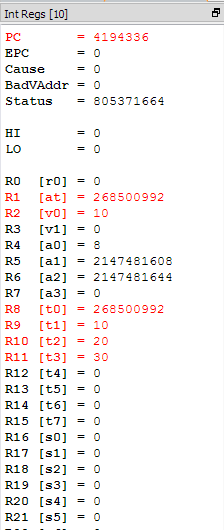
 sw $t3, 8($t0) #store word $t3 offset 8($t0)

 jr $ra #exit

.data #data section

value: .word 10, 20, 0 #data for addition

**Register t1 and t2 are added into t3 using the add instruction.**



1. **Write MIPS assembly code for swapping the contents of two registers.**

#swap two numbers

.text #text section

.globl main #call main

main: la $t0, value       #load address from 'value'

 lw $t1, 0($t0) #load first word into t1

 lw $t2, 4($t0) #load second word into t2

 add $t3, $0, $t1 #temp store t1 into t3

 add $t1, $0, $t2 #store t2 into t1

 add $t2, $0, $t3 #store t3 into t2

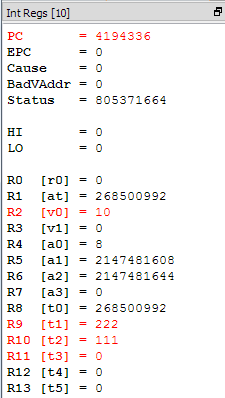
 add $t3, $0, $0 #clear t3

 jr $ra #exit

.data #data section

value: .word 111, 222 #data to be swapped

**Initially, register t1 was 111 and t2 was 222. With temporary store in t3, t1 and t2 are swapped.**



1. **Write MIPS assembly code to reverse the bits in a register. Assume the register of interest is $t3.**

#reverse the bits

.text #text section

.globl main #call main

main:

la $t0, value #load address from 'value'

li $t1, 0 #counter

li $t2, 32 #termination

lw $t3, 0($t0) #load word from address

li $t4, 1 #1

li $t5, 0 #00000000 00000000 00000000 00000000 (eventually final)

add $t7, $0, $t3 #copy of t3

loop:

beq $t1, $t2, end #jump to end if equal

and $t6, $t4, $t3    #bitwise and 1 with word

sll $t5, $t5, 1 #shift final left once

bne $0, $t6, one #if 1 is found branch

sra $t3, $t3, 1 #word shift right 1

addi $t1, $t1, 1 #add 1 to counter

j loop #jump to loop

one:

add $t5, $t5, 1 #add one to shifted value

sra $t3, $t3, 1 #shift right old value

addi $t1, $t1, 1 #add 1 to counter

j loop #back to loop

end:

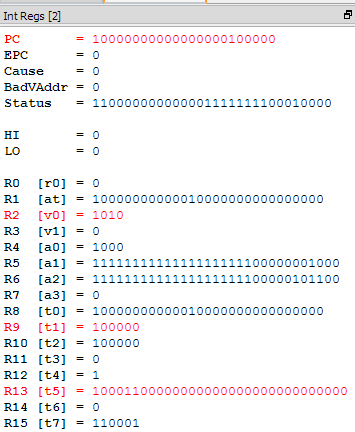
sw $t5, 0($t0) #store word

jr $ra #exit

.data #data section

value: .word 49 #data to be reversed

**Original value (110001) is temporary shown for display at t7. Our final value is stored in t5. If main loop with shift the final value store left. From the original value, if a one is found, it will branch to add a one, else it will continue with the loop. Once it has looped 32 times it will end the loop, store the word and exit.**



1. **Write MIPS assembly code to print your name in the following format on the screen/console.**

#reverse the bits

.text #text section

.globl main #call main

main:

la $a0, name #load name

li $v0, 4 #load print\_string

syscall #print\_string

la $a0, last #load last name

syscall

la $a0, first #load first name

syscall

la $a0, last #print last name

syscall

jr $ra #exit

.data #data section

name: .asciiz "My name is "

first: .asciiz "Jomar "

last: .asciiz "Pueyo "

**Using loading immediate into $v0 is the function to print out string values. Syscall will output that value into the console. Load address is needed to load the different variables to output in order.**

