Cover Page

Computer Architecture

CPE 315 - Section 05

Lab Section 06

3-30-16

Chad Benson

Nghia Nguyen

CHad Benson

Nghia Nguyen

Introduction ( & purpose of project ):

The first lab introduces the student to the basic MIPS instruction set- how to add registers together, load values into registers, implement control structures, how to print values to console, how to halt the program, and so forth. It also introduces the student to basic pointer arithmetic and how to access values using a base + offset syntax, such as

4($t1). Here, the address would be the following address relative to register $t1.

Functional Requirements ( What is to be accomplished ):

For this lab, two simple adding programs must be developed, one adds a bunch of 32 bit values together and the other adds a bunch of 8 bit values. The sums are then displayed to console. We must also learn how to load and store values, as well as use pointers to access values at various addresses.

Approach used ( algorithms & methods ):

First, a series of values of a specific bit size ( word or byte ) are stored using a label.

To add a series of values together, we load the address of the first value to be added. From that address, we can advance the pointer to the next address and then add that value with the current sum. A looped is used to continue to sum all the values together until the final sum is computed. A syscall is used to print the values to console.

Source Code:

# Program 1

# add a bunch of word sized values together….

.data

number: .word 4155543, 3112 , -2, 1054, -33543, 1233, -433433, 10101, 16384

.text

main:

la $t4, number #load the address of number in to t4

li $t3, 9 #set counter to 10

li $t1, 0 #total

loop:

lw $t0, 0($t4) #load the number into t0

add $t1, $t0, $t1 #t1 = t1 + t0

addi $t4, $t4, 4 #get to the next address of number - 4 bytes with word

addi $t3, $t3, -1 #decrease the counter

bne $t3, $zero, loop #jump back to loop if t3 is not 0

move $a0, $t1 # move result into printing register

li $v0, 1 # signal command to print

syscall # print in

li $v0, 10 # stop program

syscall

.end

# Program 2

# add a bunch of byte size values together...

.data

number: .byte 40, 33, -127, 122, 4, 0, 16, 24, 32, -5, 123

.text

main:

la $t4, number #load the address of number in to t4

li $t3, 10 #set counter to 10

li $t1, 0 #total

loop:

lb $t0, 0($t4) #load the number into t0

add $t1, $t0, $t1 #t1 = t1 + t0

addi $t4, $t4, 1 #get to the next address of number - 1 byte with byte

addi $t3, $t3, -1 #decrease the counter

bne $t3, $zero, loop #jump back to loop if t3 is not 0

move $a0, $t1 #move t1 into a0 for printing

li $v0, 1 #print int

syscall

li $v0, 10 #stop program

syscall

.end

Discussion ( difficulties and or concerns with reliability or security ) :

The development went rather smoothly, especially since Prof. Retz gave us most the code; however, we did run into an issue with declaring a bunch of word sized values- we learned that the compiler sometimes requires a blank space to separate each value, otherwise it would not compile.

Summary:

We learned how to use loops, load and store values, add values togethers, and use pointer arithmetic.

Console results:

Word sum: 3720449

Byte sum: 139