COMP 2030R Lab 2 Keys

NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. MIPS has two groups of data transfer instructions: ‘load’ to store data in a register and ‘store’ to save register data in a memory location.

‘load’ has four variations:

li $t0, 3 # load immediate: store a constant in a register

la $t0, A # load address: store an address (label) in a register

lw $t0, A($t9) # load word: fetch a word (4 bytes) from memory to store in a register

lb $t0, A($t9) # load word: fetch a single byte from memory to store in a register

Given the three integers in the data segment, consider the program below. In the comment, write the updated value of the destination register when the instruction completes its execution. Assume the data segment starts at address 0x1100.

.data

array: .word 0x37, 0x55, 0xF

.text

la $s0, array # $s0 = 0x1100

la $s1, array+4 # $s1 = 0x1104

la $s2, 8($s0) # $s2 = 0xF

li $a0, 4 # $a0 = 4

lw $t0, 0($s0) # $t0 = 0x37

  lw $t3, array($a0) # $t3 = 0x55

lw $t1, 8($s0) # $t1 = 0xF

lb $t2, ($s0) # $t2 = 0x37

1. Suppose you have a 10-element array ‘arr’ with integers (words), declared with initial values of 0:

.data

arr: .word 0:10

* 1. Write MIPS instructions to store $t0 and $t1 to the first and second elements of arr.

sw $t0, arr

sw $t1, arr+4

* 1. Write MIPS instructions to add the 2nd and 3rd elements of arr and save the sum in the 4th element.

lw $t0, arr+8

lw $t1, arr+12

add $t1, $t0, $t1

sw $t1, arr+16

1. Suppose that all numbers are limited to 6-bit binary quantities. In parentheses, write decimal values of the given binary numbers.
   1. x=001011 ( 11 )

Left shift x by one bit, and write the resulting binary bit sequence and its decimal equivalent in the parenthesis.

y=010110 ( 22 )

* 1. Let’s say that the binary sequence in part (a) is called ‘y.’

Left shift y by one bit, and write the resulting binary bit sequence and its decimal equivalent in the parenthesis.

101100 ( 44 )

* 1. z=011010 ( 26 )

Right shift z by one bit, and write the resulting binary bit sequence and its decimal equivalent in the parenthesis.

001101 ( 13 )

* 1. The left shift of x by one bit is writtein in C as

y = x << 1;

What arithmetic operation is equivalent to a left shift by one bit? Rewrite the statement y=x<<1 using only arithmetic operations (+, -, \*, or /):

y = x \* 2;

multiplication by 2