Patrick Austin

CS 219 Homework 2

2/16/2016

1. add $t0, $s1, $s2 # add w and x

sub $t1, $t0, $s3 # subtract y

add $t1, $t1, $s4 # add z

2. lw $t0, 40($s1) # load A[10] to t0

lw $t1, 24($s1) # load A[6] to t1

sub $t2, $s0, $t0 # subtract A[10] value from g, save result to t2

add $t2, $t2, $t1 # add A[6] value to t2

sw $t2, 64($s1) # save result value to A[16]

3. Instruction memory is accessed 5 times. Data memory is accessed 3 times.

4. 1. h = 4 \* j;

f = h + i;

f = f + g;

2. B[g] = A[f+1] + A[f];

f = A[f];

5. li $t0, 1 # start i at 1

li $t1, 100 # will loop until i hits this value, 100

start:

lw $t2, 0($a1) # load A[i-1]

lw $t3, 4($a2) # load B[i]

add $t4, $t2, $t3 # add A[i-1] and B[i], save in t4

sw $t4, 4($a1) # save t4 to A[i]

addi $a1, 4 # increment i in array A

addi $a2, 4 # increment i in array B

addi $t0, 1 # i++

bne $t0, $t1, start # compare i with t1, loop until equal

Extra Credit. A bge pseudo-instruction of the form “bge $rs, $rt, L1” where L1 is a label can be implemented using these lines of code:

slt $t0, $rs, $rt # if rs < rt, sets $t0 to 1

beq $t0, $zero, L1 # if t0 is 0, jump to L1

# In other words: if rs < rt, $t0 will be set to 1 and the jump in the

# beq line will not fire. Otherwise, $t0 will be 0 and the jump will

# occur. This equates to a ‘greater than or equal to’ condition for

# jumping.