Name: Ningyuan Zhang

Section: K

Instructor: Joseph Zambreno

Date: 9/19/2016

Home Work #3 Solutions

1)

(a)

-(2^15) to( 2^15) - 1

(b)

MIPS has 3 instruction formats, all 32 bits in length

R-Format

op rs rt rd shamt funct

6 5 5 5 5 6

I-Format

op rs rt imm

6 5 5 16

J-Format

op target

6 26

(c)

Register addressing: jr

Base/displacement addressiong: lw/sw

Immediate addressing: addi and other instructions with constants

PC-relative addression: beq/bne

Pseudo-direct addressing: j

2)

(a)

D because we see that the code first judge if A[i] is greater or equal to 10, and if yes it will directly jump to Loop, and if not, it will judge again if A[i] is smaller than 0, so it is a or (||) condition.

(b)

# assume a = $a0, b = $a1, i = $t0, c = $s0

# C code: for (i=0; i<=100; i=i+1) { a[i] = b[i] + c; }

add $t0, $t0, $zero #t0:i = 0

Loop: slti $t4, $t0, 101 #t4:i < 101

beq $t4, $zero, Exit #if t4 == 0, goto Exit

sll $t1, $t0, 2

add $t2, $t1, $a0

add $t3, $t1, $a1

lw $t2, 0($t2) #t2: a[i]

lw $t3, 0($t3) #t3: b[i]

add $t2, $t3, $s0

addi $t0, $t0, 1

Exit:

(c)

# C code :int f(int a, int b, int c, int d) {

# return func(func(a,b), c+d);

# }

# a: $a0, b: $a1, c: $a2, d: $a3

f:

addiu $sp, $sp, -8

sw $ra, 0($sp)

addiu $t0, $a2, $a3

sw $t0, 4($sp)

jal func

move $a0, $v0

lw $a1, 4($sp)

jal func

lw ra, 0($sp)

addiu $sp, $sp, 8

jr $ra

3)

(a)

i. sub $t3, $t1, $t2

bltz $t3, TARGET

ii. sub $t3, $t1, $t2

bgtz $t3, TARGET

iii. sub $t3, $t1, $t2

blez $t3, TARGET

iv. sub $t3, $t1, $t2

bgez $t3, TARGET

v. addi $t1, $zero, 0xABCDE

vi. add $t1, $zero, $t2

(b)