Colt Darien
829 722 055

ECE 452: Computer Organization and Design

Spring 2015
Homework 2: Instruction Set Architectures

Assigned: 3 Feb 2015
Due: 12 Feb 2015

Instructions:

- Please submit your assignment solutions via Canvas in a word or pdf file.
- Some questions might not have a clearly correct or wrong answer. In such cases, grading is based on your arguments and reasoning for arriving at a solution.

Q1 (25 points) Translate the following MIPS code snippets into C code

a. **(10 points)** For the MIPS assembly instructions below, what is the corresponding C statement? Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

```
sll  $t0, $s0, 2     # $t0 = f * 4
add  $t0, $s6, $t0   # $t0 = &A[f]
sll  $t1, $s1, 2     # $t1 = g * 4
add  $t1, $s7, $t1   # $t1 = &B[g]
lw   $s0, 0($t0)     # f = A[f]
addi $t2, $t0, 4
lw   $t0, 0($t2)
add  $t0, $t0, $s0
sw   $t0, 0($t1)
```

1.a)
```
int q1( int f, g, h, i, j){
        B[g]=A[f]+A[f+1];
}
```

b. **(15 points)** Translate the following MIPS code to C. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

```
addi $t0, $s6, 4
add  $t1, $s6, $0
sw   $t1, 0($t0)
lw   $t0, 0($t0)
add  $s0, $t1, $t0
```

1.b)
```
int q1b( int f, g, h, i, j){
        A[1]=&A[0];
        f=&A[0]+&A[0];
}
```

Q2 (15 points) Find the shortest sequence of MIPS instructions that extracts bits 16 down to 11 from register $t0 and uses the value of this field to replace bits 31 down to 26 in register $t1 without changing the other 26 bits of register $t1.

2.)
```
andi $t0, $t0, 0x0000FC00    #and t1 with only bits 16---11 set to 1(meaning that the rest are 0)
sll $t0, $t0, 15             # shift bits 16–11 to the right 15 spaces (now in bits 31–26)
andi $t1, $t1, 0x03FFFFFF    #zeroes bits 31–26 and leaves the rest unharmed
or $t1, $t1, $t0          #transfers all of the bits from 31–26 in $t0 to the same position in $t1
```

Q3 (20 points) Translate the following C code to MIPS assembly code. Use a minimum number of instructions. Assume that the values of a, b, i, and j are in registers $s0, $s1, $t0, and $t1, respectively. Also, assume that register $s2 holds the base address of the array D.

```
for(i=0; i<a; i++)
    for(j=0; j<b; j++)
        D[4*j] = i + j;
```

3.)

```
iLoop:  addi $t0, $t0, 1 #i=i+1
        bne $t0, $s0, Exit      #branch to exit if i >=a
jLoop   addi $t1, $t1, 1 #j=j+1
        bne $t1, $s1, jexit     #branch to iexit if j>= b
        sll $t2, $t1, 4         #$t2=4*4*$t1 (=4*j from code and
                                #4*more for the word offset )
        add $t2, $t2, $s2       #$t2=&D[4*j]
        add $t2, $t0, $t1       #D[4*j]=i+j
        j jLoop
jexit   j iLoop
Exit
```

Q4 (30 points) Answer the following

a. **(10 points)** Provide the type and assembly language instruction for the following binary value: 0000 0010 0001 0000 1000 0000 0010 0000_{two}

b. **(10 points)** Provide the type and hexadecimal representation of following instruction: sw $t1, 32($t2)

c. **(10 points)** Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields: op=0, rs=3, rt=2, rd=3, shamt=0, funct=34

4.)

a. 00-0000 1-0000 1-0000 1-0000 0-0000 10-0000=> R type sll $s0 $s0 0

b. I type, 2B 9 A 20=> [1010 11][01 001][0 1010][0000 0000 0100 0000]=>AD 2A 00 40

c. R type=>[000000][00011][00010][00011][00000][110100] where [] seperate parts of the instruction

Op rs rt rd shamt funct

Q5 (10 points) Assume $t0 holds the value 0b00101000. What is the value of $t2 after the following instructions?

```
        slt  $t2, $0,  $t0
        bne  $t2, $0,  ELSE
        j    DONE
ELSE:   addi $t2, $t2, 2
DONE:
```

5.)

#set $t2 if $t0 is positive or 0

#go to else if true? (if $t2 is not 0)

0b00101000 is positive so $t2 will be 1 then jump to else where 2 is added

$t2=3