CSE 2312: Computer Organization & Assembly Language Programming Fall 2017

Homework #2

Student Name: Andrew Duong

Student ID: 1000867697

Directions: Answer the questions on the following pages. Show all applicable steps for any problems requiring the use of formulas or calculations. Submit your completed assignment electronically as a single PDF document with this completed coversheet as the first page and your name written at the top of all additional pages. You may also submit the document in person before the deadline, in which case this coversheet must be completed and stapled to your solution pages.

NOTE: Refer to the MIPS Reference Data Card for all questions on this assignment.

1. Consider the following C statement:

```
f = (g - 7) + (h + 79);
```

a) Assuming *g* and *h* are stored in registers $t0 and $t1, and *f* is to be stored in register $s0, assemble the statement into a sequence of MIPS instructions.

b) Give the machine code representation (binary) of your answer to part a.

2. Assume the following register contents (hex value):

```
$t0 = 0x00FF00FF, $t1 = 0xABCD1234
```

a) What is the value of $t2 after the following statements are executed?

```
sll $t2, $t0, 7
or  $t2, $t2, $t1
```

b) What is the value of $t2 after the following statements are executed?

```
sll  $t2, $t0, 3
andi $t2, $t2, -9
```

c) What is the value of $t2 after the following statements are executed?

```
srl  $t2, $t0, 6
andi $t2, $t2, 0x0000EEAD
```

3. Translate the following MIPS code into C. Let integers x, y, and z be stored in $a0, $a1, and $a2, respectively.

x y z

```
fun:  add $t0, $a0, $a1   # g = x + y
      sub $t0, $t0, $a2   # g = g - z
      add $v0, $t0, $zero # f = g
      jr  $ra             # return
```

4. Consider the following MIPS code:

```
LOOP:   slt  $t2, $zero, $t1   # h = (g > 0) ? 1 : 0
        beq  $t2, $zero, DONE  # while (h != 0)
        subi $t1, $t1, 1       #     g--;
        addi $s2, $s2, 3       #     f += 3;
        j    LOOP
DONE:
```

$t1 = g $t2 = h $s2 = f

a) Assume that register $t1 is initialized to the value 8 (in base 10), and $s2 is initialized to zero. What is the value of $s2 after executing the code segment above?

b) Convert the MIPS instructions above to C code. Assume that registers $s2 and $t1 contain the variables *B* and *i*, respectively.

c) How many MIPS instructions would be executed if the register $t1 was initialized to the value *N*?

5. Translate the following C code to MIPS assembly. Assume that the values of *a*, *b*, *i*, and j are in registers $s0, $s1, $t0, and $t1, respectively. Also assume that $s2 holds the base address of the array D.

```
for(i = 0; i < a; i++)
    for(j = 0; j < b; j++)
        D[2 * j] = i + j;
```

6. Write a series of MIPS assembly instructions to implement the following: "*If the 8th bit of register $s0 is 1, then add 2 to register $s1; else set the 8th bit of register $s0 to 1.*"

1.

```
f = (g - 7) + (h + 79);
```

a)

```
sub  t2, t0, 7
add  t3, t1, 79
add  s0, t2, t3
```

b)

(2.)

$t0 = 0x00FF00FF, $t1 = 0xABCD1234

a)

sll $t2, $t0, 7

$t0 = 0b 0000 0000 1111 1111 0000 0000 1111 1111

$t2 = 0b 0000 0000 0111 1111 1000 0000 0111 1111 1000 0000

or $t2, $t2, $t1

$t2 = 0b 0000 0000 0111 1111 1000 0000 0111 1111 1000 0000

$t1 = 0b 0000 0000 1010 1011 1100 1101 0001 0010 0011 0100

$t2 = 0b 1111 1111 1100 1101 0111 1111 1011 0100

$t2 = 0xFFCD7FB4

b)

(2.)

$t0 = 0x00FF00FF, $t1 = 0xABCD1234

b)

sll $t2, $t0, 3

$t0 = 0b 0000 0000 1111 1111 0000 0000 1111 1111

$t2 = 0b 0000 0111 1111 1000 0000 0111 1111 1000

andi $t2, $t2, -9

$t2 = 0b 0000 0111 1111 1000 0000 0111 1111 1000

9 = 0b 0000 0000 0000 0000 0000 0000 0000 1001

= 0b 1111 1111 1111 1111 1111 1111 1111 0110 1's complement

-9 = 0b 1111 1111 1111 1111 1111 1111 1111 0111 2's complement

$t2 = 0b 0000 0111 1111 1000 0000 0111 1111 0000

$t2 = 0x07F807F0

(2.)

$t0 = 0x00FF00FF, $t1 = 0xABCD1234

c)

srl $t2, $t0, 6

$t0 = 0b 0000 0000 1111 1111 0000 0000 1111 1111

$t2 = 0b 0000 0000 0000 0011 1111 1100 0000 0011

andi $t2, $t2, 0x0000FEAD

$t2 = 0b 0000 0000 0000 0011 1111 1100 0000 0011

0b 0000 0000 0000 0000 1111 1110 1010 1101

$t2 = 0b 0000 0000 0000 0000 1110 1100 0000 0011

$t2 = 0x0000EC03

(3.)

```
fun: add $t0, $a0, $a1  # g = x + y
     sub $t0, $t0, $a2  # g = g - z
     add $v0, $t0, $zero # f = g
     jr  $ra             # return
```

```
int fun(x, y, z)
{
    g = x + y;
    g -= z;
    return g;
}
```

④

g	h	r
$t1	$t2	$s2
8	1	0
7	1	0
7	1	3
7	1	3
6	1	3
6	1	6
6	1	6
5	1	6
5	1	9
5	1	9
4	1	9
4	1	12
4	1	12
3	1	12
3	1	15
3	1	15
2	1	15
2	1	18
2	1	18
1	1	18
1	1	21
1	1	21
0	1	21
0	1	24
0	0	24

a) $\boxed{\$s2 = 24 \text{ (base 10)}}$

④ b)

```
while (i > 0)
{
    i--;
    B += 3;
}
```

c) $N = 5 * 8 - 3 =$ **37 number of times**