CS447 Lab #4 (Week 5) - Solutions

Part a: Conversion between binary and hexadecimal formats

For all problems in this section, assume unsigned representations (no two's complement).

Solutions are presented with bit widths that are a multiple of 1 byte. This is done to provide consistent formatting, but it is not critical for representing the number accurately.

As a warm up, try the following simple conversions from decimal format:

1. **Decimal value 37 as binary?**
   0010 0101_2
2. **Decimal value 37 as hexadecimal?**
   0x25

The following problems involve conversions from binary to hexadecimal (or vice versa):

3. **Binary value 0110 1011_2 as hexadecimal?**
   0x6b
4. **Hexadecimal value 0x7f as binary?**
   0111 1111_2
5. **Binary value 0000 0001 0010 1010 0100 0000 0010 0000_2 as hexadecimal?**
   0x012a4020
   Side note: This is the numerical representation of the MIPS instruction add $t0, $t1, $t2

6. **Hexadecimal value 0x3230000f as binary?**
   0011 0010 0011 0000 0000 0000 0000 1111_2
   Side note: This is the numerical representation of the MIPS instruction andi $s0, $s1, 15

Part b: Operations for extracting bitfields

The following problems ask you to predict the results of logical and shifting operations.

7. **Assume that some nonzero value is stored in register $t0. The instruction**

```
sll $t0, $t0, 3
```

   is equivalent to the multiplication of the contents of $t0 by what number?

   $2^3 = 8$

8. **For the following problems, assume that *x* = 0100 1000_2 and *y* = 0111 0100_2.**
   a. **What is the binary result of *and(x,y)*? (This can also be represented as *x*&*y*).**
      0100 0000_2

b. What is the binary result of *or(x,y)*? (This can also be represented as *x*∨*y*).

$0111\ 1100_2$

c. What is the binary result of *not*(*x*)? (This can also be represented as ~*x*).

$1011\ 0111_2$

9. Assume that x = 15, and y = 65 (both in decimal format). What is the result (in a base of your choice) of *and(x,y)*?

1 (in any base)

10. Assume that $t0 contains the decimal value 48, and $t1 contains the decimal value 16. The following MIPS instructions are run:

```
and $t2, $t0, $t1
srl $t3, $t2, 2
```

In other words, we perform $t2 = *and*($t0, $t1), shift two places to the right the contents of $t2, and store the final result in $t3. What is the value (in any base) stored in $t3?

4 (in hex or decimal)

11. Assume that we have decided to store the day of the year (for example, "September 30th") as a binary representation in the 32-bit MIPS register $s0. We will not make use of the two most significant bytes of the register. In the least significant bytes (bit positions 15……0), we will store the month and day in binary, according to the following format:

position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
contents	binary representation of month								binary representation of day							

For example, if we wished to store the date "September 30th" (9/30), the binary representation of positions 15…0 in $s0 would be:

$0000\ 1001\ 0001\ 1110_2$

Assume we wanted to extract the day (e.g., the 30th) from $s0 and store the result in register $s1. Write one or more MIPS instructions that can accomplish this goal. Multiple solutions could work.

One solution:

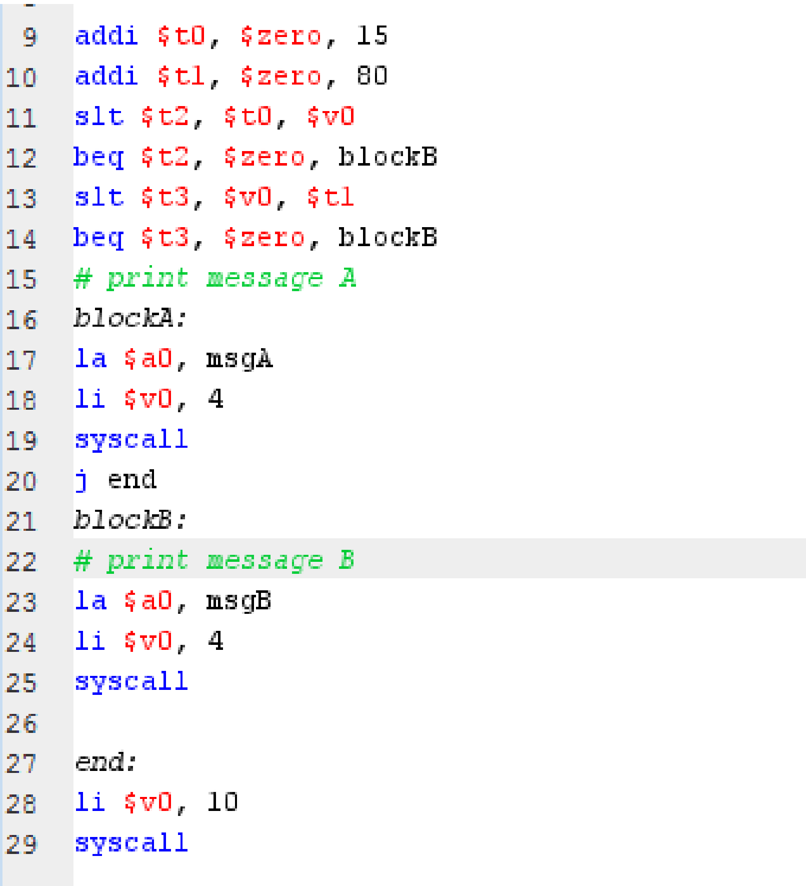
```
andi $s1, $s0, 255
```

This instruction would place the contents of the least significant byte of $s0 into $s1. If one assumed a calendar system in which the day of the month could not exceed 31, then an immediate value of 31 (00011111 in binary) would also work.

Different solutions that do not use *andi* are also possible.

Part c: Branching

12. Consider the following MIPS instructions. Assume that $v0 holds the value of some positive integer in the range of 1-99.



```
 9  addi $t0, $zero, 15
10  addi $t1, $zero, 80
11  slt $t2, $t0, $v0
12  beq $t2, $zero, blockB
13  slt $t3, $v0, $t1
14  beq $t3, $zero, blockB
15  # print message A
16  blockA:
17  la $a0, msgA
18  li $v0, 4
19  syscall
20  j end
21  blockB:
22  # print message B
23  la $a0, msgB
24  li $v0, 4
25  syscall
26
27  end:
28  li $v0, 10
29  syscall
```

In another language (for example, C/C++, Java, Python), lines 9-25 could be represented with a single if…else statement, with a Boolean operator (for example, && or ||) in the condition, and print statements in the *if* and *else* blocks. Try writing this if….else statement in the language of your choice. You can use whatever variable name you prefer to represent $v0.

In Java (assume *v* represents the value of $v0):

```
if (v > 15 && v < 80) {
   System.out.println(msgA);
}
else {
   System.out.println(msgB);
}
```

The program “lab4question12.asm” (also in the Github repository) demonstrates the instructions used for this question.