CSCI-UA.0201

Computer System Organization

Homework Assignment 2

(Total: 40 points)

Note: For the problems that require writing code assume the following (unless stated otherwise):

- **The inputs given to your functions will be correct. So do not worry about error handling.**
- **Do not write a full-fledged program, just the needed function only.**
- **Do not call other functions or libraries. Your function must be self-content.**

1. [5 points] Write a C function, called *compare()*, (no need to write a full-fledged program, just the function) that takes two integers as arguments and returns 1 if the two numbers are equal and 0 if they are not. However, you must not use any comparison operators.

2. [5 points] Write a C function, called *set_bits()*, (no need to write a full-fledged program, just the function) that takes three arguments: an unsigned int x, and unsigned int l, and an unsigned int r. The function sets the bits from l to r (inclusive) in x to 1. Remember that an integer is 32 bits, starting from the bit 0 (in the right) till bit 31 (in the left).

3. [2 point] What does the following function do (i.e. what does the value of x represents at the end)?

```
unsigned strange(unsigned int k)
{
    unsigned int  x = 0;
    while(k)
    {
       k = k & (k-1);
       x++;
     }

     return x;
}
```

4. [3 points] Suppose you have 3 N-bit unsigned integers: a, b, and c. What is the *minimum* number of bits required to present: a*b+c ?

5. [9 points] Suppose we have a 5-bit floating point number where 1 bit is used for the sign, 3 bits for exponent, and 1 bit for fraction. The same rules of normalized, denormalized, and special values numbers in IEEE 754 standard are applied.

a) What will be the bias? How did you get it?
b) What is the smallest non-zero positive number that can be presented (in decimal)? Show all the steps.
c) What is the largest positive number than can be presented (in decimal)? Show all the steps.

6. [6 points] Given the unsigned int x, using only shifting left and right, add, and subtract, write the statements to calculate the following (each problem below is independent from the others). Try to optimize as much as possible.

a) y = 7*x;
b) y = 27*x;
c) y = 67*x;

7. [5 points (-0.5 for each mistake)]

Assume we are running code on a 6-bit machine using two's complement arithmetic for signed integers. A "short" integer is encoded using 3 bits. Fill in the empty boxes in the table below. The following definitions are used in the table:

```
short sy = -3;
int y = sy;
int x = -17;
unsigned ux = x;
```

Note: You need not fill in entries marked with "–".

Expression	Decimal Representation	Binary Representation
Zero	0	
–	-6	
–		01 0010
ux		
y		
TMax		
$-$TMin		

-TMin and TMax are the minimum and maximum two's complement numbers. Remember that the whole program assumes 6-bit number. So, for this problem, an in is 6 bits and a short is 3 bits.

8. [A: 1 point, B: 2 points, C: 2 points]

A. On a x86 32-bit machine, Alice intends to use the expression `if ((x & mask) !=0))` to test if the 5th bit of `x` from the right is one or not. (The rightmost bit of x is considered as the 0-th bit). The value of `mask` should be ________ (decimal).

B. Which of the following expressions generate the desired `mask` value in Question (A)? Select all that apply.

(a) `1 << 6`
(b) `1 << 5`
(c) `~(1<<6)`
(d) `~(1<<5)`
(e) `1 >> 26`
(f) `1 >> 27`

C. Please give the expression which sets the 5th bit of `x` to be one and leave the rest of the bits of `x` unchanged. Your expression should only use the `mask` value in Question (A) and no other constants.