Scientific Computing

Assigment 2

Andres Alam Sanchez Torres

November 2, 2023

Problem 1

Let `x:i32`, when is `(x & (x - 1)) == 0` true and why?

The expression `x & (x - 1) = 0`, holds when every pair of corresponding bits in x and $x - 1$ are either different *or* both equal to zero.

More formally, let's denote $x_i, 0 \leq i < n$ as the i-th digit (from least to more significant) of a base-2 number x with n digits. Parting from the initial statement, let $x, y = x - 1$ be integers, then $x \ \& \ y = 0$ holds if and only if

$$x_i = y_i = 0 \lor x_i \neq y_i \qquad \text{for every i} \tag{1}$$

Note if y has less digits that x, the missing corresponding digits are considered 0.

Now consider there are only two cases for the value of x:

1) If the rightmost digit of x is 1, i.e. $x_0 = 1$, then the digits of y are the same of x except for the rightmost digit, then

$$\begin{aligned} x_i &= y_i & \forall i \neq 0 \\ x_i &\neq y_i & i = 0 \end{aligned}$$

Therefore, all digits, other than the rightmost, of x must be zero to hold (1), i.e. $x = 1$.

2) If the rightmost digit of x is 0, after substracting 1, the least significant bit is set to 0, and all the bits on its right are set to 1, i.e. let k be the position of the least significant bit of x, then

$$\begin{aligned} x_i &= y_i & \forall i > k \\ y_i = 0, x_i &\neq y_i & i = k \\ y_i = 1, x_i &\neq y_i & i < k \end{aligned}$$

Therefore, for (1) to hold, x_i must be 0 for every i on the left of the least significant bit. In other words, x must have only a single bit set to 1, i.e. **be a power of 2**.

Therefore, the expression is true for **all powers of 2 greater than 0**. However, in the context of Rust, and programming languages in general, negative numbers are also represented in binary as 2's complement. Since all negative numbers have a leftmost digit of 1, the only case representing a power of 2 is the lower bound of `i32`, so `x -1` is out of bounds. There's also the case of $\mathbf{x = 0}$ (all bits set to 0), and since $x - 1$ is represented as $2^{32} - 1$ (all bits set to 1), the expression also holds.

Problem 2

The following are the binary values of some single precision (32 bit) IEEE754 floating point values:

```
402D F854
7F80 0000
8000 0000
3DCC CCCD
3FB5 04F3
```

Which numbers do they represent exactly in decimal and which real number are they supposed to represent?

1. `402D F854`
   exact decimal: 2.71828174591064453125
   real: 2.7182817459106445
2. `7F80 0000`
   Infinity
3. `8000 0000`
   exact decimal: -0
   real: 0
4. `3DCC CCCD`
   exact decimal: 0.100000001490116119384765625
   real: 0.10000000149011612
5. `3FB5 04F3`
   exact decimal: 1.41421353816986083984375
   real: 1.4142135381698608