

Lab Homework 1 Report

이번 랩 시간에 랩 과제 수행을 위한 환경 설치에 대한 설명과 수업시간에 배운 내용을 복습하였다. 1바이트는 8비트로 표현되고, 비트 연산은 다른 비트에 영향을 주지 않는 bitwise의 방식으로 계산된다. 비트 연산자 &, ~, |, ^와 shift operator >>와 <<에 대해 배웠다. 그리고 이를 사용하여 해결하는 5개의 문제가 나왔고, 해결 과정은 다음과 같다.

1. Problem1 (bitNor)

- 본 과제의 Problem 1은 '~'와 '&'만을 사용하여 $\sim(x|y)$ 를 구현하는 문제이다.
- 이 연산자는 0과 0을 연산했을 경우에만 1이 나오고 나머지 경우에는 0이 나온다.
- 이 연산자를 구현하기 위해 x, y에 '~'을 붙인 후 and 연산을 수행하여 구현하였다. 함수에서 return하는 값은 다음과 같다.

$\sim x \& \sim y$

2. Problem2 (isZero)

- 본 과제의 Problem2는 입력 값이 0인지 확인하여 0이면 1, 0이 아니면 0을 return하는 함수를 구현하는 문제이다.
- 이 문제를 구현하기 위해 논리연산자 '!'를 사용하였다. 0이 아닌 수에 대해 True가, 0에 대해 False를 return하므로 '!'를 사용하여 해당 기능을 구현하였다. 함수에서 return하는 값은 다음과 같다.

!x

3. Problem3 (addOK)

- 본 과제의 Problem3는 $x+y$ 를 계산할 때 overflow가 발생하는지 판단하는 함수를 구현하는 문제이다.
- w-bit로 표현된 x의 각 자리를 LSB부터 MSB까지 각각 1~w번째라고 하고, w번째와 w-1번째에서 발생하는 carry를 각각 c1, c2라 하자. 이때 c1과 c2가 같은 경우 overflow가 발생하지 않고 c1과 c2가 다를 경우 overflow가 발생한다.

- c1과 c2를 각각 비트연산자로 나타내면 다음과 같다.

$$c2 = ((x \mid (1 \ll 31)) + (y \mid (1 \ll 31))) \gg 31,$$

$$c1 = (((x \gg 31) \& (y \gg 31)) \mid ((x \gg 31 \mid y \gg 31) \& c2))$$

- c1과 c2가 다른 경우에만 overflow가 발생하므로 XOR 연산자(^)를 사용한 후 논리연산자 '!'를 사용하였다. 함수에서 return하는 값은 다음과 같다.

$$!(c1 \wedge c2)$$

4. Problem4 (absVal)

- 본 과제의 Problem4는 입력을 받아 절대값을 return하는 함수를 구현하는 문제이다.
- 입력 값 x가 양수일 경우 입력 값을 그대로, 음수일 경우 $\sim x + 1$ 을 return해야 한다.
- 이를 비트 연산으로 구현하기 위해 'or' 연산을 사용하였다.
- ' $\gg 31$ '을 사용하여 양수일 경우 x를 그대로 return하고 음수일 경우 ' $\sim x + 1$ '을 return하도록 하였다. 함수에서 return하는 값은 다음과 같다.

$$((\sim x \gg 31) \& x) \mid ((x \gg 31) \& (\sim x + 1))$$

5. Problem5 (logicalShift)

- 본 과제의 Problem5는 정수 x와 n을 입력 받아 x를 n만큼 logical shift를 사용하여 옮겨 이를 return하는 함수를 구현하는 문제이다.
- 입력 받는 양수 x가 양수일 경우 arithmetic shift와 동일하고, 음수일 경우 앞의 n개의 bit를 1에서 0으로 바꾸어야 한다.
- 이를 구현하기 위해 x에 n만큼 arithmetic shift를 한 값과 앞의 n개의 비트는 1로, 나머지 비트는 0으로 이루어진 수와 XOR 연산을 하여 앞의 n개의 비트가 1일 경우 이를 0으로 바꿀 수 있도록 했다. 함수에서 return하는 값은 다음과 같다.

$$(x \gg n) \wedge (((x \gg 31) \ll 31) \gg n \ll 1)$$