CSED211: Lab 2

손량(20220323)

Last compiled on: Monday 25th September, 2023, 22:22

1 개요

몇 가지 간단한 정수 관련 함수와 IEEE 754 표준에 따른 floating point number 관련 함수를 구현해 본다.

2 코드 설명

2.1 `negate` – Negate Signed Integer

Two's complement 방식의 정의대로 계산하면 된다. 주어진 숫자의 bitwise not에 1을 더하였다.

2.2 `isLess` – Compare Signed Integers

`x`가 `y`보다 작은 경우는 `x`가 음수, `y`가 0 이상이거나 둘의 부호가 같고 `x - y`가 음수인 경우임에 주목하여 코드를 작성하였다.

2.3 `float_abs` – Absolute Value of Float

우선 MSB만 0이고 나머지 비트는 1인 32비트 상수를 `sign_mask`에 저장한 다음, 이를 `uf`와 bitwise and하여 결과값 `res`를 얻었다. `uf`가 `NaN`일 경우 그대로 반환해야 하기 때문에, `NaN` 판정을 위해 `nan_mask` 변수를 사용하였다. 이 변수에는 `Infinity`에 해당하는 값이 저장되어 있는데, `nan_mask`와 `uf`의 bitwise and 결괏값이 `nan_mask`와 같은 경우, `uf`는 `Infinity`, `-Infinity`, `NaN` 셋 중 하나일 것이다. `res`가 `Infinity`가 아닌 경우, `uf`는 `NaN`이므로 `uf` 그대로를 반환한다. 아닌 경우에는 `res`에 담긴 결괏값이 정확함을 알 수 있고, `res`를 반환한다.

2.4 `float_twice` – Double a Float

우선 IEEE 754 floating point의 exponent 부분에 해당하는 비트만 1인 상수를 `exp_mask`에 저장하고, `uf`에 저장해 exponent 부분을 가져온다. 만약 exponent에 해당하는 비트들이 모두 1이라면 이미 `Infinity`나 `NaN`이므로, `uf`를 그대로 반환한다. 그렇지 않은 경우에는 exponent에 해당하는 값을 1 증가시킨 후 `uf`의 exponent 부분을 덮어씌운 숫자를 반환한다.

2.5 `float_i2f` – Signed Integer to Float

주어진 정수 `x`의 크기에 따라 동작이 달라진다. `x`의 leading zero를 제외한 부분의 길이를 `bitlen`에 저장하고, `bitlen`이 24보다 작은 경우에는 mantissa에 적절히 `x`의 절댓값을 left shift하여 저장하고, `bitlen`이 24인 경우에는 절댓값 그대로를 mantissa에 저장한다. 만약 `bitlen`이 24보다 큰 경우에는 round to even 규칙에 맞추어 rounding을 수행하여 mantissa를 설정한다. 최종 결과는 `x`의 부호에 맞게 sign bit를 설정하고, exponent와 mantissa를 표준에 맞게 합쳐 얻는다.

2.6 `float_f2i` – Float to Signed Integer

`uf`의 sign bit를 right shift로 얻어 음수인지 판정하여 `is_negative`에 저장한다. `uf`의 exponent 부분과 mantissa 부분을 각각 `exp`, `mant`에 저장한 뒤, `exp`에 따라 변환을 다르게 수행한다. `exp`가 0 미만인 경우, `uf`는 절댓값이 1보다 작은 수를 나타내므로 결괏값은 0이 된다. `exp`가 31이고 `uf`가 음수이며 mantissa가 0인 경우에는 결괏값은 `INT_MIN`이다. 그 이외에 `exp`가 31 이상인 경우에는 범위 밖으로 나간 것으로 간주하였다. `exp`가 이외의 값을 가지는 경우에는 `mant`를 적당한 값만큼 shift하여 결괏값을 얻었다.