

# fn x\_mul\_2k

Michael Shoemate

This proof resides in “**contrib**” because it has not completed the vetting process.

Proves soundness of the implementation of `x_mul_2k` in `mod.rs` at commit `f5bb719` (outdated<sup>1</sup>).

## 1 Hoare Triple

### Precondition

#### Compiler-Verified

None

#### User-Verified

$k \neq \text{i32.MIN}$

### Pseudocode

```
1 def x_mul_2k(x: RBig, k: i32) -> RBig:
2   num, den = x.into_parts() #
3   if k < 0:
4     den <=<= cast(-k, usize)
5   else:
6     num <=<= cast(k, usize)
7
8   return RBig.from_parts(num, den)
```

### Postcondition

**Theorem 1.1.** Return  $x \cdot 2^k$ .

*Proof.* Since  $x$  is a fraction, line 2 splits  $x$  into its numerator and denominator.

Consider two cases:

- If  $k < 0$ , then multiplying the denominator by  $2^{-k}$  is equivalent to multiplying by  $2^k$ . Since  $k$  is negative, then  $-k$  is positive, meaning the cast to `usize` is valid. Shifting  $-k$  zeros to the left is equivalent to multiplying the denominator by  $2^k$ . Negation of  $k$  is well-defined for all values of `i32`, except for `i32.MIN`, which is not allowed by the precondition. Therefore, the result is  $x \cdot 2^k$ .
- If  $k \geq 0$ , then we multiply the numerator by  $2^k$ . Since  $k$  is positive, the cast to `usize` is valid. Shifting  $k$  zeros to the left is equivalent to multiplying the numerator by  $2^k$ . Therefore, the result is  $x \cdot 2^k$ .

In both cases, the result is  $x \cdot 2^k$ . □

---

<sup>1</sup>See new changes with `git diff f5bb719..f0b74b0 rust/src/measurements/noise/nature/float/utilities/mod.rs`