

Integer operations

Complementary arithmetic method

Using a' to represent the two's complement of a .

Assume that the bit width of an integer is N .

We have $a' = a$ if a is positive and $a' = 2^N + a$ if a is negative.

Addition:

$$(a + b)' = a' + b'$$

Proof:

a and b are both positive: $(a + b)' = a + b = a' + b'$

a and b are both negative: $(a + b)' = 2^N + (a + b) = 2^{N+1} + a + b = (2^N + a) + (2^N + b) = a' + b'$

(Since the maximum bit width is N , 2^N has the same representation with 2^{N+1} .)

a is positive but b is negative: $(a + b)' = 2^N + (a + b) = a + (2^N + b) = a' + b'$

Subtraction:

$$(a - b)' = a' + \sim(b') + 1$$

Proof:

$$a - b = a + -b$$

$$-b = \sim(b') + 1$$

Multiplication:

$$(a * b)' = a' * b'$$

Proof:

$$\begin{aligned}
(a * b)' \bmod 2^N &= ((a + a_{N-1}2^N) * (b + b_{N-1}2^N)) \bmod 2^N \\
&= (a * b + (a_{N-1}b + ab_{N-1})2^N + x_{N-1}y_{N-1}2^{2N}) \bmod 2^N \\
&= (x * y) \bmod 2^N
\end{aligned}$$

Division:

$(a / b)' = (|a'| / |b'|)$, when a and b have the same sign

$(a / b)' = (|a'| / |b'|)'$, when a and b have different signs

Since $|a'| = |a|$.

Module:

$(a \% b)' = (|a'| \% |b'|)'$, when a is negative

$(a \% b)' = (|a'| \% |b'|)$, when a is positive

Since $|a'| = |a|$.

Contrast about true form, complement, and offset binary

True form:

The value of an integer is visible.

Convenient to make addition on two positive integers.

Inconvenient to process subtraction.

Complement:

Convenient to make addition on arbitrary two integers with no effect of their signs.

The value of a negative integer is not so clear, but it's not important.

offset binary:

Has advantages in float representation.

Convenient to compare two integers.

Inconvenient to know the real value of an integer.

Bit expansion

True form:

Only keep the highest bit.

Complement:

If the highest bit is 0, then all expansion bits are 0.

If the highest bit is 1, then all expansion bits are 1.

Offset binary:

All expansion bits are set to be 0.

Comparison

True form:

First the sign bit, and then the absolute value.

Complement:

First the sign bit.

Two positive integers: larger the complement is, larger the integer is.

Two negative integers: larger the complement is, smaller the integer is.

A positive integer and a negative integer: obviously the positive one is larger.

Offset binary:

Compare directly.

Overflow

True form:

When two integers with the same sign bit make addition and obtain a integer with a different sign, an overflow occurred.

Complement:

When two integers with the same sign bit make addition and obtain a integer with a different sign, an overflow occurred.

Offset binary:

If two positive integers make addition and the result is smaller than any of them, an overflow occurred.

If two negative integers make addition and the result is larger than any of them, an overflow occurred.