Two's Complement

Randy Shepherd
NYU
Definition

- **Property** Two's complement representation allows the use of binary arithmetic operations on signed integers, yielding the correct 2's complement results.

- **Positive Numbers** Positive 2's complement numbers are represented as the simple binary.

- **Negative Numbers** Negative 2's complement numbers are represented as the binary number that when added to a positive number of the same magnitude equals zero.
**Signed Numbers**

- The most significant (leftmost) bit indicates the sign of the integer; therefore it is sometimes called the **sign bit**.
- If the sign bit is zero, then the number is greater than or equal to zero, or positive.
- If the sign bit is one, then the number is less than zero, or negative.
Calculation of Two’s Compliment

- To calculate the 2's complement of an integer, invert the binary equivalent of the number by changing all of the ones to zeroes and all of the zeroes to ones, and then add one.
Continued

- These examples show conversion of a decimal number to 8-bit twos complement.
- The bit size is always important with twos complement, since you must be able to tell where the sign bit is. The steps are simple.
- First, you convert the magnitude of the number to binary, and pad to the word size (8 bits). If the original number was positive, you are done. Otherwise, you must negate the binary number by inverting the bits and adding 1.
Two's Compliment Addition

- Two's complement addition follows the same rules as binary addition.
- For example,

\[ 5 + (-3) = 2 \]
\[ 0000 \ 0101 = +5 \]
\[ + 1111 \ 1101 = -3 \]
\[ \underline{0000 \ 0010} = +2 \]
Two's Complement Subtraction

- Two's complement subtraction is the binary addition of the minuend to the 2's complement of the subtrahend (adding a negative number is the same as subtracting a positive one).

- For example,

\[
\begin{align*}
7 - 12 &= (-5) \\
0000 \ 0111 &= +7 \\
+ \ 1111 \ 0100 &= -12 \\
\hline
1111 \ 1011 &= -5
\end{align*}
\]
Two's Complement Overflow Rules

The rules for detecting overflow in a two's complement sum are simple:

→ If the sum of two positive numbers yields a negative result, the sum has overflowed.
→ If the sum of two negative numbers yields a positive result, the sum has overflowed.
→ Otherwise, the sum has not overflowed.
The reason for the rules is that overflow in two's complement occurs, not when a bit is carried out of the left column, but when one is carried into it. That is, when there is a carry into the sign. The rules detect this error by examining the sign of the result. A negative and positive added together cannot overflow, because the sum is between the addends. Since both of the addends fit within the allowable range of numbers, and their sum is between them, it must fit as well.
Examples

-39 + 92 = 53:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}
\]
Carryout without overflow. Sum is correct.

104 + 45 = 149:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0
\end{array}
\]
Overflow, no carryout. Sum is not correct.

10 + -3 = 7:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1
\end{array}
\]
Carryout without overflow. Sum is correct.

-19 + -7 = -26:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}
\]
Carryout without overflow. Sum is correct.

-75 + 59 = -16:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1
\end{array}
\]
No overflow nor carryout.

127 + 1 = 128:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}
\]
Overflow, no carryout. Sum is not correct.

44 + 45 = 89:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0
\end{array}
\]
No overflow nor carryout.

-103 + -69 = -172:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}
\]
Overflow, with incidental carryout. Sum is not correct.

-1 + 1 = 0:
\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}
\]
Carryout without overflow. Sum is correct.