1 Binary operations

Addition

$$
\begin{aligned}
0 + 0 &= 0\\
0 + 1 &= 1\\
1 + 0 &= 1\\
1 + 1 &= 10
\end{aligned}
$$

$1\ +\ 1\ = 2$ but since we are working with binary here, we write $(2)_{10}$ as $(10)_2$.

Signed binary numbers

To represent a negative number in binary, we apply 2's complement.

Consider the decimal number 2.

The 32-bit representation of $(2)_{10}$ is:

$$(00000000\ 00000000\ 00000000\ 00000010)_2$$

To find the 2's complement, we **first invert all the bits of the binary representation of** 2. And this what we get:

$$(11111111\ 11111111\ 11111111\ 11111101)_2$$

This is 1's complement of 2.

Now, for 2's complement, we **add a** 1 **to this** 1**'s complement**.

$$
\begin{array}{r}
11111111111111111111111111111101\\
+1\\
\hline
11111111111111111111111111111110
\end{array}
$$

So, $(11111111111111111111111111111110)_2$ is the binary representation of -2.

Now, if we add -2 and 2, we end up with 0.

This condition is satisfied by their binary representations as well.

$$
\begin{array}{r}
00000000000000000000000000000010\\
+11111111111111111111111111111110\\
\hline
00000000000000000000000000000000
\end{array}
$$

Now, notice there that we still have a 1 as a carry. In assembly languages, this 1 goes into a special register known as **carry register**.