Binary Arithmetic

\[ 0 + 0 = 0 \]
\[ 0 + 1 = 1 \]
\[ 1 + 1 = 0 \text{ carry 1 to next higher bit} \]

**Ex1:**
\[
\begin{align*}
(110)_2 + (111)_2 &= (1101)_2 \\
(6)_{10} + (7)_{10} &= (13)_{10}
\end{align*}
\]

\[
(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0
\]

\[ = 1 + 0 + 4 + 8 = (13)_{10} \]

Complement of a Binary Number

Twos Complement

\[ N^*_2 = 2^n - N \]

\[ n - \text{number of bits in binary integer} \]

**Ex2:** \[ N = (1101)_2 \quad n = 4 \]

\[ N^*_2 = 2^4 - (1101)_2 \]

\[ = (10000)_2 - (1101)_2 = ? \]
This can be calculated by hand, but it's easy to make a mistake. There is an easier way.

**Ones Complement**

\[ N_1^* = N_2^* - 1 \]

The ones complement of a binary number is found by switching all the ones and zeros to zeros and ones, respectively.

**Ex:** \( N = (1101)_2 \)

\[ N_1^* = (0010)_2 \]

The two complement is given by

\[ N_2^* = N_1^* + 1 = (0010)_2 + (0001)_2 \]

\[ = (0011)_2 \]

Subtraction is carried out in the computer by using the twos complement.
\[ M - N = M + N_2^* \]

where \( M \) and \( N \) are binary numbers

Ex 3: Suppose \( M = (01101)_2 \) and \( N = (01010)_2 \) and the computer word length is 5 bits

Find the ones complement of \( N \)

\[ N_1^* = (10101)_2 \]

Find the twos complement of \( N \)

\[ N_2^* = N_1^* + 1 = (10101)_2 + (00001)_2 \]

\[ = (10110)_2 \]

Calculate the difference

\[ M - N = M + N_2^* = (01101)_2 + (10110)_2 \]

\[ = (100011)_2 \]

since computer can handle only 5 bits drop the leftmost bit
\[ M - N = (00011)_2 \]

Verify result with decimal calculation

\[ M = (01101)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \]

\[ = (13)_{10} \]

\[ N = (01010)_2 = 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \]

\[ = (10)_{10} \]

\[ M - N = (13)_{10} - (10)_{10} = (3)_{10} \]

From the binary calculation

\[ M - N = M + N^* = (00011)_2 \]

\[ = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \]

\[ = 1 + 2 = (3)_{10} \]
<table>
<thead>
<tr>
<th>Bits</th>
<th>Straight</th>
<th>Offset</th>
<th>Twos Complement</th>
<th>Ones Complement</th>
<th>AVPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>-7</td>
<td>+0</td>
<td>+0</td>
<td>-0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>-6</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>-5</td>
<td>+2</td>
<td>+2</td>
<td>-2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
<td>-4</td>
<td>+3</td>
<td>+3</td>
<td>-3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>-3</td>
<td>+4</td>
<td>+4</td>
<td>-4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
<td>-2</td>
<td>+5</td>
<td>+5</td>
<td>-5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
<td>-1</td>
<td>+6</td>
<td>+6</td>
<td>-6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
<td>0</td>
<td>+7</td>
<td>+7</td>
<td>-7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>0</td>
<td>-8</td>
<td>-7</td>
<td>+0</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
<td>1</td>
<td>-7</td>
<td>-6</td>
<td>+1</td>
</tr>
<tr>
<td>1010</td>
<td>10</td>
<td>2</td>
<td>-6</td>
<td>-5</td>
<td>+2</td>
</tr>
<tr>
<td>1011</td>
<td>11</td>
<td>3</td>
<td>-5</td>
<td>-4</td>
<td>+3</td>
</tr>
<tr>
<td>1100</td>
<td>12</td>
<td>4</td>
<td>-4</td>
<td>-3</td>
<td>+4</td>
</tr>
<tr>
<td>1101</td>
<td>13</td>
<td>5</td>
<td>-3</td>
<td>-2</td>
<td>+5</td>
</tr>
<tr>
<td>1110</td>
<td>14</td>
<td>6</td>
<td>-2</td>
<td>-1</td>
<td>+6</td>
</tr>
<tr>
<td>1111</td>
<td>15</td>
<td>7</td>
<td>-1</td>
<td>0</td>
<td>+7</td>
</tr>
</tbody>
</table>