Computer Whole Number Representation

Andy Chong Sam

December 24, 2023

1 Introduction

Math is most commonly performed using a decimal system. We call this a decimal system because ten symbols are used: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Any number can be decomposed using a member from the set of symbols and a power of ten, take the number 1421 for example, which can be written as:

$$1421 = (1)(10)^3 + (4)(10)^2 + (2)(10)^1 + (1)(10)^0$$

Computers use a binary system, where the set of usable symbols has been reduced to just two: $\{0, 1\}$. This is referred to as a binary system. This document explains the algorithms to convert between decimal and binary systems. It will also explain how negative and fractional amounts are represented in binary.

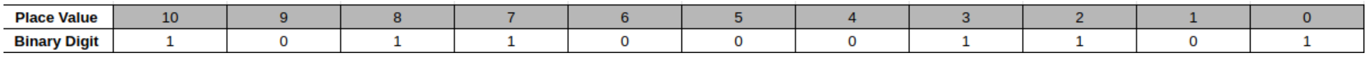
2 Decimal to Binary

The algorithm to translate decimal to binary involves repeated divisions by 2, and recording the remainders. The stopping point is when the dividend becomes 1. The list of remainders is then reversed. We will use the same number as before, 1421 to illustrate this.

$$
\begin{array}{r} 710 \\ 2\overline{)1421} \\ \underline{14} \\ 02 \\ \underline{2} \\ 01 \end{array}
\quad
\begin{array}{r} 355 \\ 2\overline{)710} \\ \underline{6} \\ 11 \\ \underline{10} \\ 10 \\ \underline{10} \\ 0 \end{array}
\quad
\begin{array}{r} 177 \\ 2\overline{)355} \\ \underline{2} \\ 15 \\ \underline{14} \\ 15 \\ \underline{14} \\ 1 \end{array}
\quad
\begin{array}{r} 88 \\ 2\overline{)177} \\ \underline{16} \\ 17 \\ \underline{16} \\ 1 \end{array}
\quad
\begin{array}{r} 44 \\ 2\overline{)88} \\ \underline{8} \\ 08 \\ \underline{8} \\ 0 \end{array}
\quad
\begin{array}{r} 22 \\ 2\overline{)44} \\ \underline{4} \\ 04 \\ \underline{4} \\ 0 \end{array}
\quad
\begin{array}{r} 11 \\ 2\overline{)22} \\ \underline{2} \\ 02 \\ \underline{2} \\ 0 \end{array}
\quad
\begin{array}{r} 5 \\ 2\overline{)11} \\ \underline{10} \\ 1 \end{array}
\quad
\begin{array}{r} 2 \\ 2\overline{)5} \\ \underline{4} \\ 1 \end{array}
\quad
\begin{array}{r} 1 \\ 2\overline{)2} \\ \underline{2} \\ 0 \end{array}
\quad
\begin{array}{r} 0 \\ 2\overline{)1} \end{array}
$$

The recorded list of remainders is 10110001101. Reversing the list gets us the binary representation of 1421: **10110001101**.

Let's try the reverse process now. Suppose that I have 10110001101 and I want to transform that into a decimal number. Similar to the example in the introduction, we can assign place values to all the binary digits:



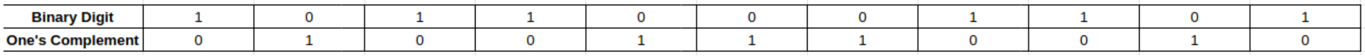
Place Value	10	9	8	7	6	5	4	3	2	1	0
Binary Digit	1	0	1	1	0	0	0	1	1	0	1

We then proceed to multiply each binary digit by two raised to the place value. All these terms will then be added together:

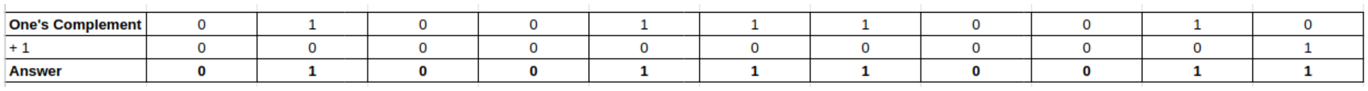
$$2^{10} + 2^8 + 2^7 + 2^3 + 2^2 + 1 = 1421$$

3 Negative Integers

The binary equivalent of a negative number is formulated using the two's complement technique. First, we take the one's complement which involves flipping each zero to a one, and each one to a zero. Finally we add a binary 1. Suppose we wanted to represent -1421:



Binary Digit	1	0	1	1	0	0	0	1	1	0	1
One's Complement	0	1	0	0	1	1	1	0	0	1	0



One's Complement	0	1	0	0	1	1	1	0	0	1	0
+ 1	0	0	0	0	0	0	0	0	0	0	1
Answer	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**

We are left with the result **01001110011**. This result occupies 10 positions (or 11 bits). In an actual computer, the most significant bit is known as the **sign bit**, and for negative numbers it will be set to 1. The left most bit is the most significant bit.

If we above result was processed through a 12 bit computer, -1421 would be stored as **101001110011**.