

University CEU San Pablo

Department of Information Technology

Gabriel Caffarena

Academic year 2019/2020



Fast binary conversion – Part I

1. Introduction

Binary is the basis of digital systems and it is important to be fluent when it comes to obtain the decimal value of a binary number. In this post we will start dealing with fast binary to decimal conversion for numbers up to 16 bits. This will be handy if you work in the domain of embedded systems and digital design.

The binary system is a base-2 positional numeral system. A base- B positional numeral system represents numbers with digits that can have values from 0 to $B-1$ (from 0 to 1 for binary; from 0 to 9 for decimal, etc.), and they are written one after the other. A digit is in a position i which in turn determines its weight. Assuming that the digit located to the right of the number has position 0 and that the digit located to the left of the number has position $N-1$ (because there are N digits), the weight of each digit is B^i . Given that an N -digit number with values b_i ($b_i \in [0 \dots B - 1]$ and $i \in [0 \dots N - 1]$), its value can be computed as:

$$value = \sum_{i=0}^{N-1} b_i \cdot 2^i = b_{N-1}2^{N-1} + b_{N-2}2^{N-2} + \dots + b_12^1 + b_02^0.$$

For instance, the binary number 10011_2 has a decimal value of $1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 2 + 1 = 19_{10}$.

Learning by heart a few powers of two is useful in many ways. Firstly, it is necessary to do any mental conversion from binary to decimal. So do your homework and learn the powers of two from 2^0 to 2^{16} ; that is: **1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768** and **65536**. Once you know these powers of two, it is easy to find out the minimum number of bits that we need to represent an integer number. For instance, if we know the upper closest power of two of a number x , this will lead us to the number of bits required to represent that number.

Let's say that we want to convert to binary number **240**. The upper closest power of two is **256**, which is 2^8 , which tells us that **8** bits are required. Note, that if the number is exactly a power of two (i.e. $x=256$) then the number of bits has to be increased by 1. And knowing these powers of two will be also essential in performing fast binary to decimal conversions.

So now, let's start with some "**shortcuts**".

2. Consecutive ones

Let's talk now about something quite trivial. If we add **1** to number **999**, we obtain **1000**. So $999=1000-1$, that is $999 = 10^3 - 1$, so, the exponent of the power of ten matches the number of consecutive nines. We can generalize this so that a number in base B composed of a sequence of N consecutive digits is equal to $(B^N - 1)_B$, and particularizing for binary ($B=2$) a sequence of N consecutive ones is equal to $(2^N - 1)_2$. So now, assuming that we master the powers of two, we can easily calculate the decimal value of numbers such as:

- $11111_2 = 2^5 - 1 = 32 - 1 = 31$
- $1111111111_2 = 2^{10} - 1 = 1024 - 1 = 1023$
- Etc.

3. Zeros on the right

Adding a zero to the right of a number is equivalent to multiplying the number by the base. For instance, $560_{10} = 56 \times 10$, $110_2 = 11_2 \times 2$, etc. Thus, adding N zeros to the right of a binary number is equivalent to multiplying the number by 2^N . This could be useful sometimes:

$$10100_2 = 101_2 \cdot 2^2 = 5 \cdot 4 = 20 \quad (\text{also, } 16+4=20)$$

$$1111111111000_2 = 1111111111_2 \cdot 2^3 = (2^{10} - 1) \cdot 8 = 1023 \cdot 8 = 8184$$

Etc.

4. More ones than zeros

If a binary number has just a few zeros you might as well consider the number as a sequence of consecutive ones that has a few ones missing. Thus, number 111111110111_2 can be seen as $111111111111_2 - 1000_2 = (2^{12} - 1) - 2^3 = 4095_{10} - 8_{10} = 4087_{10}$. Let's see some examples:

$$10111101_2 = (2^8 - 1) - 2^7 - 2^1 = 255 - 66 = 189$$

$$111111111100_2 = (2^{12} - 1) - 3 = 4092 \quad (\text{or, } 1023 \cdot 4 = 4092)$$

5. Putting all together

Let's try to find the best way to perform fast conversion to several numbers. I am assuming here that you master powers of two and that you easily recognize short binary numbers.

$$10100000101 = 1024 + 256 + 5 = 1285$$

$$10100000101 = 5 \cdot 2^8 + 5 = 5 \cdot 257 = 1285$$

$$11110000 = 15 \cdot 16 = 240$$

$$11110111000 = (255 - 8) \cdot 8 = 247 \cdot 8 = 1976$$

$$11110111000 = 2047 - 64 - 7 = 2040 - 71 = 1976$$

$$1000000000001 = 4096 + 1 = 4097$$

$$1000000001110 = 4096 + (7 \cdot 2) = 4096 + 14 = 4110$$

$$111101000 = 8 \cdot (63 - 2) = 8 \cdot 61 = 488$$

Etc.

Well, it is not rocket science, but surely it is useful. You will find many occasions when these tricks cannot be applied, and you will have to add all the weights corresponding to the ones to obtain the decimal values. Also, you can use a calculator...

In this article we have covered unsigned numbers. Part II we'll be about fast conversion of signed binary numbers: **two's complement numbers**. Work hard on your binary and soon you will master the binary language of moisture vaporators.