Numbering Systems and Base Encoding

Abyan Majid

July 5, 2023

1 Numbering systems typically found in computer systems

There are 4 different numbering sytems that are typically found in computer systems:

- Decimal (base 10): Decimal numbers may ONLY be represented by the values [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- Binary (base 2): Binary numbers may ONLY be represented by the values [0 and 1].
- Octal (base 8): Octal numbers may ONLY be represented by the values [0, 1, 2, 3, 4, 5, 6, 7]
- Hexadecimal (base 16): Hexadecimal numbers may ONLY be represented by the values [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F]

Decimals are what we humans currently use to count. Binaries are what computers use. Octal and hexadecimals are useful abbreviations for binaries because they're often lengthy.

For example, the decimal number 25 can also be represented as:

- "11001", on base 2 (binary). Or, we write "0b11001" to clearly indicate that it is a binary.
- "31", on base 8 (octal). Or, we write "0o31" to clearly indicate that it is an octal.
- "19", on base 16 (hexadecimal). Or, we write "0x19" to clearly indicate that it is a hexadecimal.

2 Base encoding

In this section, we will explore ways to represent a number, with the example of 25, in base 10, 2, 8, and 16.

2.1 Translating base 10 to base 2 (decimal to binary)

You can convert from base 10 to base 2 (ie decimal to binary) through successive division.

$$\frac{25}{2} = 12 \times 2 + \boxed{1} \text{ (least significant bit)}$$

$$\frac{12}{2} = 6 \times 2 + \boxed{0}$$

$$\frac{6}{2} = 3 \times 2 + \boxed{0}$$

$$\frac{3}{2} = 1 \times 2 + \boxed{1}$$

$$\frac{1}{2} = 0 \times 2 + \boxed{1} \text{ (most significant bit)}$$

What happened here is we kept on dividing by 2 until we no longer can. However, each time we divide by 2, we do not write literal decimals but rather in quotient and remainder format.

IMPORTANT: The REVERSE order of the remainders is the binary or base 2 representation of the decimal. So, in this case, the binary or base 2 representation of the decimal "25" is $\boxed{11001}$, otherwise written as "0b11001".

2.2 Translating base 2 to 10 (binary to decimal)

To convert from base 2 to base 10 (ie binary to decimal) you multiply each bit by 2^n, where n is the position of the bit in the binary - starting from the last, rightmost bit, in which case $n = 0$.

So, basically, you perform $\boxed{b \times 2^n}$ starting from the rightmost bit until the leftmost. n should start at 0 and increments by 1 with every next bit.

$$\begin{tikzcd}
1 \arrow[d] & 1 \arrow[d] & 0 \arrow[d] & 0 \arrow[d] & 1 \arrow[d] \\
1 \times 2^4 \arrow[d] & 1 \times 2^3 \arrow[d] & 0 \times 2^2 \arrow[d] & 0 \times 2^1 \arrow[d] & 1 \times 2^0 \arrow[d] \\
16 & 8 & 0 & 0 & 1
\end{tikzcd}$$

The sum gives the decimal or base 10 representation.

$$16 + 8 + 0 + 0 + 1 = \boxed{25}$$

2.3 Translating base 2 to base 8 (binary to octal)

When converting base 2 to base 8 (ie binary to octal), we group the binary into sets of 3, starting from the right. We do this because 2^3 (which is 8) is the common factor between the binary and octal number system.

$$
\begin{array}{c} 11001 \\ \downarrow \\ 011\ \ 001 \end{array}
$$

If a set has fewer than 3 bits, we add a prefix of leading zeros (e.g. 11 $\rightarrow$ 011).

Next, we get the corresponding octal value for each set of 3 bits.

3-bit Binary	**Octal**
000	0
$\boxed{001}$	$\boxed{1}$
010	2
$\boxed{011}$	$\boxed{3}$
100	4
101	5
110	6
111	7

So, the octal representation of the binary 11001 is $\boxed{31}$, otherwise written as "0o31".

2.4 Translating base 2 to 16 (binary to hexadecimal)

To convert base 2 to base 16 (ie binary to hexadecimal), you do the same as you would when converting base 2 to base 8. Except, in this case, you group the binary into sets of 4 bits instead because 2^4 (which is 16) is the common factor between the binary and hexadecimal systems.

$$
\begin{array}{c} 11001 \\ \downarrow \\ 0001\ \ 1001 \end{array}
$$

Don't forget to add leading zeros as a prefix to sets with less than 4 bits (eg 1 $\rightarrow$ 0001)

Now, we get the corresponding hexadecimal digits.

4-bit Binary	**Hexadecimal**
0000	0
$\boxed{0001}$	$\boxed{1}$
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
$\boxed{1001}$	$\boxed{9}$
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

So, the base 16 or hexadecimal representation of the binary 11001 is $\boxed{19}$, otherwise written as ”0x19”

2.5 Translating base 8 and 16 to base 2 (octal and hexadecimal to binary)

This should be self-explanatory, all you have to do is find the corresponding set of bits for each digit in the octal or hexadecimal form.

- Octal number ”0o31” to binary
  - ”3” is ”011” in binary
  - ”1” is ”001” in binary

  Therefore, the binary of ”0o31” is $\boxed{11001}$, with the leading extra zeros ”$\underline{0}$11001” removed.

- Hexadecimal number ”19” to hexadecimal
  - ”1” is ”0001” in binary
  - ”9” is ”1001” in binary

  Therefore, the binary of ”0x19” is $\boxed{11001}$, with the leading extra zeros ”$\underline{00}$011001” removed.

Refer to the binary to the octal/hexadecimal tables above for reference.