

Numerical systems

Preface

The computers are representing physically numbers in a binary form, it is important to understand how binary numbers does work in order to do anything more serious math coding on the c64 other then basic additions/substracitons. The hexadecimal system is not completely unavoidable, however it is so widely used that it is not possible to live without it.

Bits and bytes

A bit is a binary digit, taking a value of either 0 or 1. For example, the number 10010111 is 8 bits long, a 8 bit long number is called a byte.

Binary numbers

(from wikipedia)

One can think about binary by comparing it with our usual numbers. We use a base ten system. This means that the value of each position in a numerical value can be represented by one of ten possible symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9. We are all familiar with these and how the decimal system works using these ten symbols. When we begin counting values, we should start with the symbol 0, and proceed to 9 when counting. We call this the “ones”, or “units” place.

The “ones” place, with those digits, might be thought of as a multiplication problem. 5 can be thought of as 5×10^0 (10 to the zeroeth power, which equals 5×1 , since any number to the zero power is one). As we move to the left of the ones place, we increase the power of 10 by one. Thus, to represent 50 in this same manner, it can be thought of as 5×10^1 , or 5×10 .

```
500 = (5 \times 10^2) + (0 \times 10^1) + (0 \times 10^0)
5834 = (5 \times 10^3) + (8 \times 10^2) + (3 \times 10^1) + (4 \times 10^0)
```

When we run out of symbols in the decimal numeral system, we “move to the left” one place and use a “1” to represent the “tens” place. Then we reset the symbol in the “ones” place back to the first symbol, zero.

Binary is a base two system which works just like our decimal system, however with only two symbols which can be used to represent numerical values: 0 and 1. We begin in the “ones” place with 0, then go up to 1. Now we are out of symbols, so to represent a higher value, we must place a “1” in the “twos” place, since we don't have a symbol we can use in the binary system for 2, like we do in the decimal system.

In the binary numeral system, the value represented as 10 is $(1 \times 2^1) + (0 \times 2^0)$. Thus, it equals “2” in our decimal system.

Binary-to-decimal equivalence:

$$\begin{aligned}1 &= 1 * 2^0 = 1 * 1 \Rightarrow 1 \\10 &= (1 * 2^1) + (0 * 2^0) = 2 + 0 \Rightarrow 2 \\101 &= (1 * 2^2) + (0 * 2^1) + (1 * 2^0) = 4 + 0 + 1 \Rightarrow 5\end{aligned}$$

Here is another way of thinking about it: When you run out of symbols, for example 11111, add a “1” on the left end and reset all the numerals on the right to “0”, producing 100000. This also works for symbols in the middle. Say the number is 100111. If you add one to it, you move the leftmost repeating “1” one space to the left (from the “fours” place to the “eights” place) and reset all the numerals on the right to “0”, producing 101000.

Hexadecimal Numbers

hexadecimal, base-16, or simply hex, is a numeral system with a radix, or base, of 16, usually written using the symbols 0-9 and A-F, or a-f. For example, the decimal numeral 79, whose binary representation is 01001111, is 4F in hexadecimal (4 = 0100, F = 1111).

Hexadecimal is primarily used in computing to represent a byte, whose 256 possible values can be represented with only two digits in hexadecimal notation.

Binary may be converted to and from hexadecimal easily. This is due to the fact that the radix of the hexadecimal system (16) is a power of the radix of the binary system (2). More specifically, $16 = 2^4$, so it takes four digits of binary to represent one digit of hexadecimal.

The following table shows each hexadecimal digit along with the equivalent decimal value and four-digit binary sequence:

Hex	Dec	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

To convert a hexadecimal number into its binary equivalent, simply substitute the corresponding binary digits:

```
3A16 = 0011 10102  
E716 = 1110 01112
```

To convert a binary number into its hexadecimal equivalent, divide it into groups of four bits. If the number of bits isn't a multiple of four, simply insert extra 0 bits at the left (called padding). For example:

```
10100102 = 0101 0010 grouped with padding = 5216  
110111012 = 1101 1101 grouped = DD16
```

To convert a hexadecimal number into its decimal equivalent, multiply the decimal equivalent of each hexadecimal digit by the corresponding power of 16 and add the resulting values:

```
C0E716 = (12 × 163) + (0 × 162) + (14 × 161) + (7 × 160) = (12 × 4096) +  
(0 × 256) + (14 × 16) + (7 × 1) = 49,38310
```

From:

<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:

https://codebase64.org/doku.php?id=base:numerical_systems

Last update: **2015-04-17 04:33**

