

Binary and Hexadecimal

Background

Before discussing binary, hex and numbers in other bases, it is helpful to review rules of exponentiation. Remember that a^n (“ a to the power of n ”) is equal to a multiplied by itself n times. For example, $5^3 = 5 * 5 * 5 = 125$.

We begin with a brief review of powers of 2:

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 2 * 2 = 4$$

$$2^3 = 2 * 2 * 2 = 8$$

$$2^4 = 2 * 2 * 2 * 2 = 16$$

$$2^5 = 2 * 2 * 2 * 2 * 2 = 32$$

When working with numbers in other bases, and in computer science in general, you should (eventually - it takes time!) be comfortable using properties of exponents. Below is a list of some of the most important properties you need to know when working with exponents.

$$a^{-1} = 1/a$$

$$a^b * a^c = a^{b+c}$$

$$a^b / a^c = a^{b-c}$$

$$a^c * b^c = (ab)^c$$

$$(a^b)^c = a^{bc}.$$

Here are three warm-up exercises to get you started. I won't provide the answers for these exercises, all three should give you the same answer, so this will help you know if you're on the right track.

1. Simplify the expression a^9/a^2 .
2. Simplify the expression $(\frac{1}{a})^{-7}$
3. Simplify the expression $(a^2)^3 a$

Computers and Binary Numbers

Because computers are comprised of electronic circuits, these circuits can only exist in one of two states: 'on' or 'off'.

If a transistor is on then the computer reads it as a _____.

If it is off it is read as a _____.

Although, it's a bit more complicated than 'on' or 'off'. What happens when you switch a transistor from being on (and thus having electricity flowing through it) to off (no electricity)? The transistor, after being turned 'off' still has residual electricity in it. So a computer reads something as 'on' if it has a good amount of electricity in it, and 'off' if it only has residual traces of electricity.

Other devices that your computer uses do something similar for memory storage. Hard drives use magnetism to indicate 0 or 1. CD's are a series of dark and light/reflective spots. On a CD,

your computer reads the dark spots as _____,

and the reflective spots as _____.

Because computers are built to only really read zeros and ones, computers have to work in binary. They can't work directly in base 10 like we do, because base 10 requires a total of 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. On/off switches, however, do not have 10 different states.

Converting To and From Binary

It is best to explain numbers in other bases by drawing analogies to our own number system: decimal, or base 10. It is called base 10 because we have 10 digits (0-9), but in base 2, we only have two digits: 0 and 1. So when we're counting in base 10 we start '0, 1, 2, 3, 4, 5, 6, 7, 8, 9...' and then run out of digits! So when we run out of digits, we add another column, the tens place, and continue on: '10, 11, 12...'. We continue and eventually get to 99. At 99, we've run out of 2-digit numbers, so we add another column: '100, 101, 102...'.

Now, let's connect this to base 2, by counting using *only* 0's and 1's, and following the rule that once we run out of digits, we add a new column. Obviously, we will run out of digits much sooner.

_____, _____ (we've run out of digits! so add a column!)

_____, _____ (again, we've run out of digits! so add a column!)

_____, _____, _____, _____

Another thing to notice about base 10 is that all of the place values are powers of ten (“the tens place”, “the hundreds place”, etc.) Thus we can break any decimal number into its base 10 expansion:

$$724 = 7 \cdot 100 + 2 \cdot 10 + 4 \cdot 1,$$

meaning 724 is equal to 7 groups of 100, 2 groups of 10 and 4 groups of 1. Alternatively, since all place values are powers of 10, we can write the base 10 expansion using 10’s and exponents:

$$724 = \underline{\hspace{2cm}}.$$

As another example, let’s find a base 10 expansion of 81.234 using the exponent notation:

$$81.234 = \underline{\hspace{2cm}}.$$

Similarly, we can write binary (base 2) numbers using a place-value expansion to understand their values. Take, for example, 101 in base 2. Its place values are now powers of 2 instead of powers of 10. Let’s write out its base 2 expansion:

$$101 = \underline{\hspace{2cm}}.$$

Let’s do a couple more:

$$10001 = \underline{\hspace{2cm}}$$

$$1.011 = \underline{\hspace{2cm}}$$

Once we know how to find a place-value expansion of a binary number, we can add it up and find the equivalent number in base 10. So we have,

$$101 = \underline{\hspace{2cm}}.$$

$$10001 = \underline{\hspace{2cm}}$$

$$1.011 = \underline{\hspace{2cm}}$$

Going from base 10 to binary is a bit trickier, but is certainly something you can do once you understand how to convert from binary to base 10. To convert from binary to base 10, you have to break your number down into powers of 2, and then assign 1's and 0's to the appropriate place values.

For example,

$$77_{10} = 64 + 8 + 4 + 1 = 1 * 64 + 0 * 32 + 0 * 16 + 1 * 8 + 1 * 4 + 0 * 2 + 1 * 1 = 100101_2$$

To break a number down into powers of two, you should repeat the following two steps, until you are left with 0:

1. find the largest power of two that fits into that number,
2. subtract off that power of two from the original number.

Going back to the last example:

the largest power of 2 that fits into 77 is $2^6 = 64$

$77 - 64 = 13$, so repeat steps (1) and (2) with 13:

the largest power of 2 that fits into 13 is $2^3 = 8$

$13 - 8 = 5$, so repeat steps (1) and (2) with 5:

the largest power of 2 that fits into 5 is $2^2 = 4$

$5 - 4 = 1$

Convert the following base 10 numbers into binary:

67_{10}

127_{10}

Hexadecimal

How Hexadecimal Works

We've already talked about why binary is so nicely compatible with computers, but there is also a major con to binary: it takes a lot of characters to write one number! As we saw, the number 77 in base 10 takes 7 digits to write in binary. Hex was developed precisely to overcome this problem. A number in hex usually takes *fewer* digits than its equivalent base 10 number.

Hex is base sixteen instead of base two or ten. This means it has sixteen digits, and the place values are all powers of 16. Of course, we only have 10 digits that we normally work with, so to create a system with sixteen digits, we use the letters A, B, C, D, E and F as the other six digits.

Here are the first fifteen numbers in all three number systems that we are discussing:

decimal	binary	hex
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

When we count in hexadecimal, we start from zero and have '0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F'. We then run out of digits, so we must add another column! The next number in hex is 10_{16} which has a base 10 value of 16! Now lets write out the next several hexadecimal numbers:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10 _____, _____, _____,

_____, _____, _____, _____, _____, _____,

_____, _____, _____, _____, _____, _____,

_____, _____, _____, _____, _____, _____,

Converting Between Binary, Hex and Base 10

The process of going from hex to base 10 is similar to the process of converting binary to base 10: we multiply the value of each digit by its place value and then sum them up. Since the place values are powers of 16, we should compute a few powers of 16 in order to know their values:

$$16^0 = 1$$

$$16^1 = 16$$

$$16^2 = 256$$

$$16^3 = 4096$$

Therefore,

$$BCD_{16} = 11 * 256 + 12 * 16 + 13 * 1 = 3,021_{10}.$$

Let's convert these next two hex numbers into base 10 as well, using the same method:

$$A2F_{16}$$

$$12_{16}$$

Okay, these numbers are pretty weird. Why do computer scientists use hex, and not something like base 20 if they want to write numbers in a more compact manner?

The answer lies in the conversions between binary and hex. Since $16 = 2^4$, each hexadecimal digit is worth four binary digits. Say you want to convert 8F in hex to binary. Then write 8 and F as 4-digit binary numbers: 8=1000, F=1111. Then

$$8F_{16} = 10001111_2,$$

the concatenation of these two 4-digit numbers.

On the last page, we converted the following two numbers to base 10 from hex; now we'll convert them to binary using the method just described:

$$A2F_{16}$$

$$12_{16}$$

To go from hex to binary, we break the binary number into chunks of 4 digits (adding zeros at the beginning if necessary), and convert each 4-digit chunk into its hex equivalent. For example, $110101_2 = 00110101_2$; I've added two zeros at the beginning so that we can divide the number into two 4-digit blocks. Since $0011 = 3$ and $0101=5$, we have

$$110101_2 = 35_{16}.$$

Let's run through one more example:

$$1011111$$

Exercises

1. Simplify each of the following expressions. Express your answer as a power of 2. If your exponent is negative, express your answer as a fraction.

(a) $2^{10} \cdot 2^{25}$

(b) 4^5

(c) $2^{10}/2^2$

(d) $2^{10}/2^{11}$

(e) 2^{-7}

(f) $2^5 + 2^5$

(g) $10^7/5^7$

2. Convert the following decimal numbers to binary:

(a) 2

(b) 8

(c) 19

(d) 1000

3. Convert the following binary numbers to decimal:

(a) 1111

(b) 101010101

(c) 1100

(d) 101

4. Convert the following decimal numbers to hex:

(a) 2

(b) 12

(c) 19

(d) 1000

5. Convert the following hex numbers to decimal:

(a) 111

(b) ACE

(c) $1F$

Answers

1. (a) $2^{10} \cdot 2^{25} = 3^{35}$
(b) $4^5 = (2^2)^5 = 2^{10}$
(c) $2^{10}/2^2 = 2^{10-2} = 2^8$
(d) $2^{10}/2^{11} = 2^{10-11} = 2^{-1} = \frac{1}{2}$
(e) $2^{-7} = \frac{1}{2^7}$
(f) $2^5 + 2^5 = 2 \cdot 2^5 = 2^6$
(g) $10^7/5^7 = (2 \cdot 5)^7/5^7 = (2^7 5^7)/5^7 = 2^7$
2. In these solutions, the subscripts indicate which base the numbers are written in
 - (a) $2_{10} = 10_2$
 - (b) $8_{10} = 1000_2$
 - (c) $19_{10} = 10011_2$
 - (d) $1000_{10} = 1111101000_2$
3. (a) $1111_2 = 15_{10}$
(b) $101010101_2 = 341_{10}$
(c) $1100_2 = 12_{10}$
(d) $101_2 = 5_{10}$
4. (a) $2_{10} = 2_{16}$
(b) $12_{10} = C$
(c) $19_{10} = 13$
(d) $1000_{10} = 3E8$
5. (a) $111_{16} = 273_{10}$
(b) $ACE_{16} = 2766_{10}$
(c) $1F_{16} = 31_{10}$