Binary, Decimal, and Hex TACT

A **Numeral System** is a writing system for expressing numbers.
There are four common numeral systems in computer science:

1. Decimal
2. Binary
3. Octal
4. Hexadecimal

Decimal number	Binary representation	Octal representation	Hexadecimal representation
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

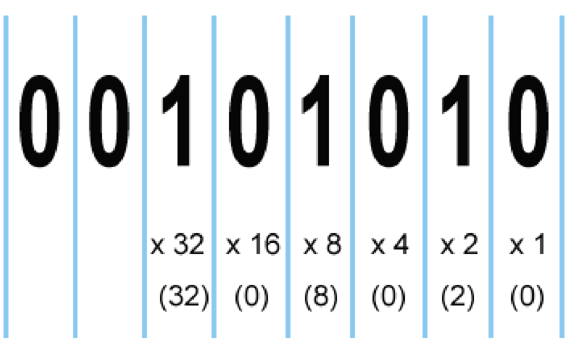
Number Bases

Decimal

The decimal number system uses digits 0-9 to represent any numeric value. We are familiar with this system so this will be used to compare and understand the other number systems.

Binary

The binary number system uses digits 0 and 1 to represent any numeric value. Just like in the decimal system, the smallest valued digit is on the right-most side, known as the Least Significant Bit (LSB). As the decimal system has a 1's, 10's, and 100's place, binary has a similar structure. Binary has a 1's, 2's, 4's, 8's, 16's place and so on as shown below.



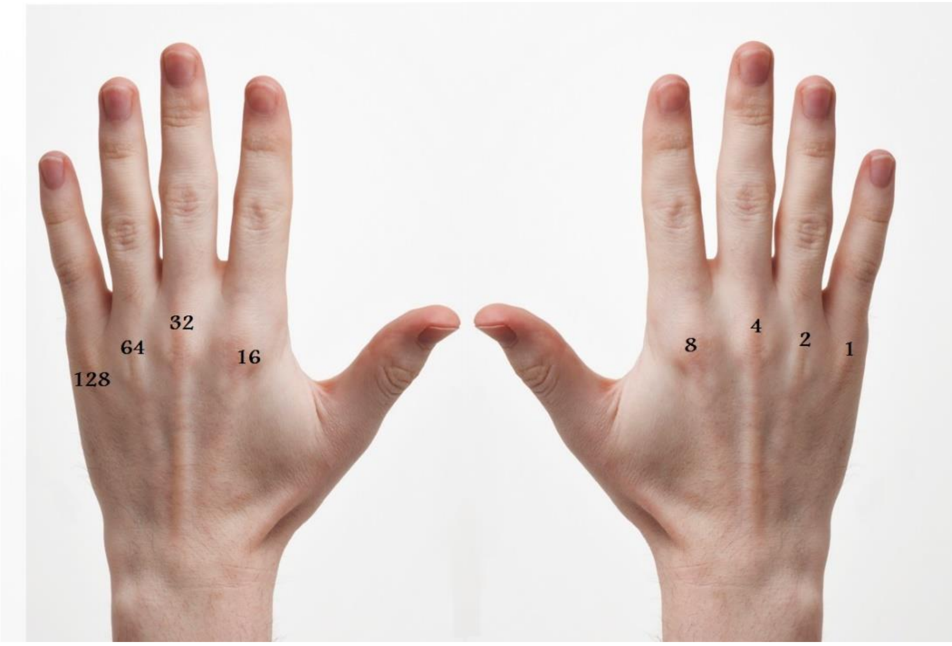
If there is a 1, then we count the numeric representation of that value. The 6th digit from the right represents 32, so we count 32 with all other values with a 1, which includes 8 and 2. So the numeric representation of the figure is

$$0 + 0 + 32 + 0 + 8 + 0 + 2 + 0 = 42$$

As you may notice the represented numbers double, this will continue to double, unless if there is a clear break such as a space to represent two different binary numbers

Binary Exercise

On your knuckles for both hands, write 128, 64, 32, 16, 8, 4, 2, 1 (skip the thumbs for now). So your hands will look like this:



Open/up fingers have values as shown. Closed/down fingers have a value of zero.

Can you represent the following on your hands?

1. 0001 1001 (25)
2. 129
3. 94
4. 511

It is best to start with the biggest numbers. Check to see if 128 needs to be open, if so, raise the finger and subtract your number by 128 and continue.

Here is a link to an excellent video that does this same counting method to count to 1023 (including thumbs):

https://www.khanacademy.org/math/math-for-fun-and-glory/grant-sanderson/inventing-math-tutorial/v/how-to-count-to-1000-on-two-hands

This works because binary works on base 2 (using two digits, 0 and 1):

one hundred twenty eights place	sixty four's place	thirty second's place	sixteen's place	eight's place	four's place	two's place	one's place
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	1	1	1

Octal

The octal number system uses digits 0-7 to represent a numeric value.

This number system may feel more familiar, with the exception that we do not use digits 8 or 9 at all. Below is a chart you can review to see the pattern of how octal counts.

Octal Table																			
Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22

Octal is sometimes used in computing instead of hexadecimal, perhaps most often in modern times in conjunction with file permissions under Unix systems. It has the advantage of not requiring any extra symbols as digits (the hexadecimal system is base-16 and therefore needs six additional symbols beyond 0–9). It is also used for digital displays.

Hexadecimal

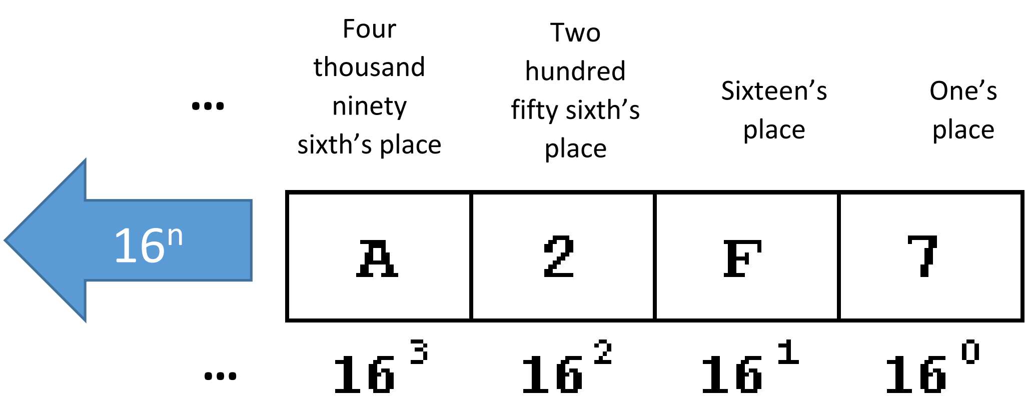
The hexadecimal number system uses digits 0-9 and A-F (Total of 16 symbols) to represent a numeric value. As you count up to 9, instead of 10 being the next value, in hexadecimal the value would be A. This continues until 15 which is F, as you can see on the comparison chart.

Hexadecimal Table																
Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Once you reach F, you will increment the 16's place and continue. Notice the pattern is similar to how the decimal system increments the 10's place.

Hexadecimal Table (cont.)																
Value	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Hex	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F

So 32 (decimal) in hex would be 20 (hex). Hexadecimal works on base 16:



When given a hexadecimal like the one above (A2F7) we consider their placement and multiply it by their value:

A = 10 * 4096 = 40960 Note: $16^3 = 4096$

2 = 2 * 256 = 512 $16^2 = 256$

F = 15 * 16 = 240 $16^1 = 16$

7 = 7 * 1 = 7 $16^0 = 1$

Then you add them all up:

A2F7 (Hex) = 40960 + 512 + 240 + 7 = 41719 (decimal)