Hexadecimals as an Effective Shorthand for Binary

Shaan Fulton

July 1, 2025

We discuss how a base-2, binary system can be used to represent any form of data in `representing-data`. Hexadecimals provide a convenient shorthand for representing larger strings of binary data. Hexadecimal is base-16, meaning there are 16 combinations per digit. There are also 16 combinations in a string of four binary digits. Thus we can create a perfect bijection (every input maps to a unique output and every output to a unique input) between one hexadecimal digit and four binary digits. This makes writing binary much cleaner:

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

Note we precede hexadecimals with `0x` and binary base-2 with `0b`. This is purely a shorthand. Hexadecimals have no real fundamental grounding in computer architecture. When we say `0xFBC` we're really just saying `0b111110111100` in a legible way.

Converting between positional number system is rather intuitive. A clear mathematical method is outlined in `translating-number-systems`.