Number Representation

Real world data is **analog** and must be converted to a **digital** format to transmit via **sampling**. Data may also be born digital.

Bits

N bits can represent at most $\mathbf{2^N}$ things.

Binary, Decimal, Hex

- Decimal: base10 (great for humans, especially when doing arithmitic)
- Binary: base2
  - 4 bit = 1 nibble = 1 hex digit = 2^4 *things*
  - 8 bits = 1 byte = 2 hex digits = 2^8 *things*
- Hex: base16 (a simpler way of looking at a string of bit)

Binary	Decimal	Hex
0b0000	0x0	0
0b0001	0x1	1
0b0010	0x2	2
0b0011	0x3	3
0b0100	0x4	4
0b0101	0x5	5
0b0110	0x6	6
0b0111	0x7	7
0b1000	0x8	8
0b1001	0x9	9
0b1010	0xA	10
0b1011	0xB	11
0b1100	0xC	12
0b1101	0xD	13
0b1110	0xE	14
0b1111	0xF	15

Representing Numbers

Unsigned Numbers

Unsigned integers lack a sign and range from [0, 2^N-1].

Sign and Magnitude

If we let the leftmost bit represent the sign, we can now represent negative integers. However, this has several shortcomings:

1. Complicated arithmetic circuit: the binary odometer does *not* wrap around.
2. 2 zeros: both `10...0` and `00...0` represent 0.

Two's Complement

We define the left most bit as $\mathbf{-2^{N-1}}$. For example, 1101_2 in two's complement is converted as follows:
`0b1101` $= 1*-2^3 + 1*2^2 + 0*2^1 + 1*2^0$
$= -8 + 4 + 0 + 1$
$= -3_{10}$

Bias Encoding

A number can be represented as an unsigned integer added to a bias term. Generally, a bias is chosen as $\mathbf{-2^{N-1}-1}$. For example,

Number Conversions

credit: JennoMai

Binary to Hexadecimal

Given a base-2 number *x*, we can convert to base-16 as follows:

1. Group *x* into 4-bit numbers, padding the left side with 0s as needed.
2. Evaluate each group to a (base-10) number between 0 and 15.
3. Substitute each group with the base-16 equivalent.

Hexadecimal to Binary

Similarly to the reverse, we can simply replace each base-16 digit with its 4-bit equivalent and remove leading 0s.