Unsigned Integer Representation

Integers are represented as n-bit vectors (series of 1's and 0's)

$$B = b_{n-1} \ldots b_1 b_0$$

The unsigned value represented is:

$$V(B) = b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

The *b's* are the coefficients in a powers of 2 polynomial

The number of bits (n) determines the range that can be covered

$$0 \text{ to } 2^n - 1$$

Unsigned overflow exists if an operation yields a value out of range

Signed Integer Representations

The total number of patterns = 2^n where n is the number of bits

This is called the modulus of the system

Incrementing the largest pattern (all 1's) rolls over to 0

Signed systems use some of the patterns for negative values

- Sign-and-magnitude
- One's complement
- Two's complement
- Biased (or excess)

Two's complement is by far the most common

Biased is used for integer exponents in floating point numbers

Signed Integer Representations

The first three systems are compared below (using 4 bits):

$b_3 b_2 b_1 b_0$	Sign and magnitude	1's complement	2's complement
0 1 1 1	+7	+7	+7
0 1 1 0	+6	+6	+6
0 1 0 1	+5	+5	+5
0 1 0 0	+4	+4	+4
0 0 1 1	+3	+3	+3
0 0 1 0	+2	+2	+2
0 0 0 1	+1	+1	+1
0 0 0 0	+0	+0	+0
1 0 0 0	−0	−7	−8
1 0 0 1	−1	−6	−7
1 0 1 0	−2	−5	−6
1 0 1 1	−3	−4	−5
1 1 0 0	−4	−3	−4
1 1 0 1	−5	−2	−3
1 1 1 0	−6	−1	−2
1 1 1 1	−7	−0	−1

Signed Integer Representations

$b_3b_2b_1b_0$	Sign and magnitude	1's complement	2's complement
0 1 1 1	+7	+7	+7
0 1 1 0	+6	+6	+6
0 1 0 1	+5	+5	+5
0 1 0 0	+4	+4	+4
0 0 1 1	+3	+3	+3
0 0 1 0	+2	+2	+2
0 0 0 1	+1	+1	+1
0 0 0 0	+0	+0	+0
1 0 0 0	−0	−7	−8
1 0 0 1	−1	−6	−7
1 0 1 0	−2	−5	−6
1 0 1 1	−3	−4	−5
1 1 0 0	−4	−3	−4
1 1 0 1	−5	−2	−3
1 1 1 0	−6	−1	−2
1 1 1 1	−7	−0	−1

In each system, the MSB is 0 for positive and 1 for negative values

Positive values have identical representations for each system

- Same as for the unsigned value

They differ in how negative values are represented

- Only negative values require complementing

Signed Integer Representations

Sign-and-magnitude:

MSB only indicates sign (0 for +, 1 for -)

Remaining bits give the magnitude

One's complement

Invert each bit to get the negative

Same as adding negative value to modulus-1

e.g. -5 is represented as -5 + (16-1) = 10

Two's complement

Invert each bit and add 1 to get negative

Same as adding negative value to modulus

e.g. -5 represented as -5 + 16 = 11

$b_3 b_2 b_1 b_0$	Sign and magnitude	1's complement	2's complement
0 1 1 1	+7	+7	+7
0 1 1 0	+6	+6	+6
0 1 0 1	+5	+5	+5
0 1 0 0	+4	+4	+4
0 0 1 1	+3	+3	+3
0 0 1 0	+2	+2	+2
0 0 0 1	+1	+1	+1
0 0 0 0	+0	+0	+0
1 0 0 0	−0	−7	−8
1 0 0 1	−1	−6	−7
1 0 1 0	−2	−5	−6
1 0 1 1	−3	−4	−5
1 1 0 0	−4	−3	−4
1 1 0 1	−5	−2	−3
1 1 1 0	−6	−1	−2
1 1 1 1	−7	−0	−1

Signed Integer Representations

Sign-and-magnitude properties:

two distinct representations for 0

+0 → 0 sign bit and all 0's for magnitude

-0 → 1 sign bit and all 0's for magnitude

Range = $-(2^{n-1}-1)$ to $+2^{n-1}-1$

One's complement:

two distinct representations for 0

+0 → bits are all 0's

-0 → bits all 1's

Range = $-(2^{n-1}-1)$ to $+2^{n-1}-1$

Two's complement:

Unique representation of 0 (all 0 bits)

Range = -2^{n-1} to $+2^{n-1}-1$

Extra negative pattern (-2^{n-1}) has MSB=1, all other bits = 0