Binary Sign Representations

Sign-magnitude: The left bit is the sign (0 for + numbers and 1 for – numbers).

All bits to right are the number magnitude

Left bit is the sign bit

Advantages to sign-magnitude:

- Simple to implement.
- Useful for floating point representation.

Disadvantage of sign-magnitude:

If m = 00000000, then m = 11111111; there are two zeros in this method as well.

In an n-bit representation, there are no extra bits! If adding 2 n-bit numbers results in n+1 bits, the left most bit is discarded! Thus: Let n = 0000 0000. Then m = n + 1, or m = 1111 1111 + 1 = (1) 0000 0000 = 0000 0000. The 1 is discarded, since in a computer, there are no extra columns. There are only 8-bits, so the (9th-column) 1 is “thrown away.”

Therefore, the 2’s complement of 0 is 0.

Finding Two's Complements: Examples

To convert a negative decimal number to 2's complement binary: Convert the decimal number to a positive binary number.

Take the 1's complement of that binary number and add 1.

- Converting negative numbers (still using a single 8-bit byte length):

– 50: 50 = 0011 0010; 1's C. = 1100 1101; 2's C. = 1100 1110.

– 127: 127 = 0111 1111; 1's C. = 1000 0000; 2's C. = 1000 0001.

– 1: 1 = 0000 0001; 1’s C. = 1111 1110; 2’s C. =1111 1111.

But: Positive decimal numbers are converted simply to positive binary numbers as before (no 2’s complement).

Example: +67 (using method of successive div.) → 0100 0011.

Two's Complement Binary to Decimal

Converting the 2's complement to decimal is also simple. Simply do the following: Check the sign bit (left-most bit).

If the sign bit is 0 (positive number), simply convert the number directly to a positive decimal number as we learned previously.

If the sign bit is 1, the number is a 2’s complement negative number. To convert this number to decimal: Take the 2’s complement of the negative binary number. Convert the resulting + number to decimal and add a negative sign.

Two's Complement Binary to Decimal (2)

Binary 2's complement-to-decimal examples, negative numbers:

1111 1111 → 0000 0000+1 = 0000 0001 = 1 → – 1.

1010 0011 → 0101 1100+1 = 0101 1101 = 93→ – 93.

1000 1111 → 0111 0000+1 = 0111 0001 = 113 → – 113.

1000 0010 → 0111 1101+1 = 0111 1110 = 126 → – 126.

But for a positive binary number:

0000 0001 → Not a negative number → 1.

0000 1111 → Not a negative number → 15.

0110 1100 → Not a negative number → 108.

0111 1111 → Not a negative number → 127.

To subtract binary number b from a, simply take the 2's complement of b, and add to a. That is:

$a - b = a + (\text{2's comp. of } b) = a + (b + 1) = a + b + 1.$

To add a positive and negative number, simply perform the addition and the minus sign (i.e., the left-most bit in the number) will take care of itself (assuming the result is within the range of representation).

Two's Complement Math

Subtract 0111 0101 from 0111 1100:

The 2's complement of 0111 0101 is 1000 1010 +1 = 1000 1011.

Adding:

```
    0111 1100   Check:   124
  +1000 1011            –117
(1)0000 0111 007         007
```

Add 1100 0001 + 0110 1110:

Note that the 2's complement of 1100 0001 is 0011 1111, so the first number is equivalent to – 63 decimal.

Adding:

```
    1100 0001    Check:  – 63
   +0110 1110            +110
 (1)0010 1111             +47
```

Subtract 1101 1101 from 0101 1100 (note we are subtracting a negative number):

Adding:	0101 1100	Check:	92
	+ 0010 0011		-(-35)
	0111 1111		+127

Add 1000 0001 + 0111 0010

	0111 0010	check:	114
+	1000 0001		+(-127)
	1111 0011		-13

Check: 2's C of 1111 0011 = 0000 1101 = 13, so the number = -13.