

SIGNED BINARY AND 2'S COMPLEMENT

Robert P. Webber, Longwood University

Humans use **signed decimal** notation for integers. We indicate whether a number is negative by preceding it with a minus sign. For instance, 837 is positive. (We could write +837, but a leading positive sign is usually omitted.) To make a number negative, just precede it with a minus sign: -837, for instance.

Integers could be written in **signed binary** notation. Just write the absolute value of the number in base two, and affix a negative sign if necessary. For example, 28 and -73 in signed binary are

$$\begin{aligned}28 &= 16 + 8 + 4 = 11100_2, \\ -73 &= -(64 + 8 + 1) = -1001001_2.\end{aligned}$$

This has a lot of advantages, including convenience and clarity. However, it has some disadvantages, one of which is that subtraction becomes very messy. It requires borrowing, which is tricky and error prone. Just recall all the problems you had in grade school learning to borrow in subtraction!

There are other ways to represent negative integers, one of which is **complement notation**. This notation has the advantage of greatly simplifying subtraction.

Let's see how it works in base 10. The **10's complement** of an integer is the difference between the number and a higher power of the base. For example, various complements of 837 are

$$\begin{aligned}1000 - 837 &= 163 \\ 10,000 - 837 &= 9,163 \\ 100,000 - 837 &= 99,163 \\ 1,000,000 - 837 &= 999,163\end{aligned}$$

Obtaining a 10's complement by traditional subtraction is not pleasant. It requires borrowing in every position! Fortunately, there is an easier way.

1. Fix the number of places, including at least one additional place on the left. Supply zeros in the extra places on the left.
2. Replace each digit in the original number by 9 - digit .
3. Add 1 to the result.

For example, find the six digit 10's complement of 837.

1. 000837
2. 999162
3. $999162 + 1 = 999163$

Notice this is the same result we obtained above.

Here's why complement notation is nice. Consider the problem of subtracting 837 from 2146, say. Lots of borrowing is necessary.

$$\begin{array}{r} \overset{1}{2} \overset{11}{1} \overset{3}{4} \overset{16}{6} \\ - 837 \\ \hline 1309 \end{array}$$

Now try adding the complement of 837 instead of subtracting. Use six digits for convenience in all numbers, and throw away any carry out of the left most digit.

$$\begin{array}{r} 2146 \\ - 837 \\ \hline \end{array} \Rightarrow \begin{array}{r} \overset{1}{0} \overset{1}{0} \overset{1}{2} 146 \\ + 999163 \\ \hline 001309 \end{array}$$

which is the correct result. We can subtract by adding!

This scheme works for negative results. Consider $268 - 837$.

$$\begin{array}{r} 268 \\ - 837 \\ \hline \end{array} \Rightarrow \begin{array}{r} \overset{1}{0} \overset{1}{0} 0268 \\ + 999163 \\ \hline 999431 \end{array}$$

The leading 9's are a signal that the number is negative. Convert it to signed notation by reversing the complementation (subtract 1 and replace each digit by 9 - digit) or by complementing the complement and affixing a negative sign.

$$999431 \Rightarrow \begin{array}{r} \overset{1}{0} \overset{1}{0} 00568 \\ + \quad \quad 1 \\ \hline -569 \end{array}$$

Sure enough, $268 - 837 = -569$.

So far, we have seen the following equivalences.

<i>Signed decimal</i>	<i>10's complement (six digits)</i>
-837	000163

-569

999431

There isn't much use for 10's complement notation, except as a curiosity. However, suppose we use base 2 and 2's complement.

To convert a negative signed binary integer to 2's complement,

1. Fix the number of bits. Write zeros in the extra places to the left.
2. Replace each bit by 1 - bit. Notice this amounts to reversing the 0's and 1's.
3. Add 1, ignoring any carry out of the left most bit.

For example, find the 2's complement binary representation of -45 using eight bits.

To find the representation, first convert 45 to binary.

$$45_{10} = 101101$$

Now pad with zeros on the left to make eight bits.

00101101

Switch 0's and 1's and add 1.

$$\begin{array}{r}
 11010010 \\
 + \quad \quad 1 \\
 \hline
 11010011
 \end{array}$$

To sum up,

<i>Signed decimal</i>	<i>Signed binary</i>	<i>2's complement binary (8 digits)</i>
-45	-101101	11010011

The 2's complement of a positive number is the same as the ordinary binary, with leading zeros affixed for emphasis.

<i>Signed decimal</i>	<i>Signed binary</i>	<i>2's complement binary (8 digits)</i>
-45	-101101	11010011
45	101101	00101101

Computers subtract by adding the 2's complement. Suppose we ask the computer to evaluate 85 - 38. In binary, using eight bits for convenience,

$$85_{10} = 01010101 \quad \text{and} \quad 38_{10} = 00100110,$$

so

$$-38_{10} = 11011001 + 1 = 11011010$$

Finally,

$$\begin{array}{r} 01010101 \\ + 11011010 \\ \hline 00101111 \end{array}$$

In base 10, $00101111_2 = 32 + 8 + 4 + 2 + 1 = 47$, which is the correct answer.

Computers use 2's complement notation for negative integers instead of signed binary. It has several advantages.

- It is easy to determine the sign of a 2's complement number: If it begins with 0, the number is positive; if it begins with 1, the number is negative.
- The programmer doesn't have to teach the computer to subtract by borrowing. Instead, it can complement and add.
- It is much simpler and less expensive to build an electronic circuit that can complement than it is to build one that can subtract by borrowing.

Exercises

1. Complete the table. The first two rows were done above.

<i>Signed decimal</i>	<i>Signed binary</i>	<i>2's complement (8 bits)</i>
-45	-101101	11010011
45	101101	00101101
100		
	-1100001	
		00011001
	1100	
		11000011
-66		
0		
	1	
-1		

2. Complete the table.

<i>Signed decimal</i>	<i>Signed binary</i>	<i>2's complement (8 bits)</i>
	-1101100	
17		
		01101100
	100001	
-58		
		10001001

In problems 3 through 10, do the decimal subtraction by using 10's complements, *not* by borrowing. Give the answers in signed decimal notation, however.

3. $37 - 12$

4. $37 - 18$

5. $37 - 37$

6. $37 - 62$

7. $37 - 78$

8. $4285 - 1277$

9. $462 - 498$

10. $805 - 1486$

In problems 11 through 16, do the binary subtraction by using 2's complements, *not* by borrowing.

11. $101 - 11$

12. $1011 - 10$

13. $1110 - 111$

14. $1001 - 1100$

15. $101 - 1101$

16. $11011011 - 1101101$

Will subtraction by adding the complement work if the first term is negative? In problems 17 through 20, do the arithmetic using 10's complements for all negative numbers, and check your answers using ordinary arithmetic.

17. $-44 - 21$

18. $-108 - 97$

19. $-16 - 95$

20. $-16 - 209$

21. What is the largest positive base 10 number that can be stored in 2's complement using 8 bits? What is the smallest negative number? *Hint:* Think about the bit patterns in base 2 and translate back to decimal.

22. What is the largest positive base 10 number that can be stored in 2's complement using 16 bits? What is the smallest negative number?