1.3 Signed binary numbers: Two's complement

Two's complement

Unsigned numbers involve only non-negative numbers, like 0 and 3. ***Signed numbers*** involve both positive and negative numbers, like 3 and -3.

In binary, a ***signed-magnitude representation*** uses the left bit for the sign: 0 means positive, 1 means negative. Ex: For 4-bit numbers, 0011 is 3, and 1011 is -3. Signed-magnitude representation is rarely used, because calculations involving negative numbers, such as 5 - 3, would require special circuits beyond an adder.

A more clever negative number representation exists that can use an adder for both positive and negative numbers. A ***complement*** of an N-digit number is another number that yields a sum of 100...00 (with N 0's), and can be used to represent the negative of that number.

PARTICIPATION ACTIVITY | 1.3.1: Two's complement signed number representation.

Start 2x speed

Base 10:

Replace by:

$$\begin{array}{r} 5 \\ -\ 3 \\ \hline 2 \end{array} \qquad \begin{array}{r} 5 \\ +\ 7 \\ \hline \cancel{1}\,2 \end{array} \text{ ignore carry}$$

Why?

7 + 3 = 10; 7 is the complement of 3.

Thus 5 + 7 is 10 too much, so the carry can be ignored.

Base 2:

$$\begin{array}{rl} 0101 & (5) \\ -\ 0011 & (3) \\ \hline 0010 & (2) \end{array} \qquad \begin{array}{rl} 0101 & (5) \\ +\ 1101 & (-3) \\ \hline \cancel{1}\,0010 & (2) \end{array} \text{ ignore carry}$$

Complement: invert bits, add 1.

0011 (3) has complement: (0011)' + 1

1100 + 1

So -3 is: 1101

Captions

1. Intuition in base 10.
2. In base two, a complement is obtained by inverting each bit and adding 1.
3. Subtraction is performed by adding the complement.

Feedback?

The above is called the ***two's complement representation***, which inverts every bit and adds 1. One's complement also exists, but is rarely used, and so is not discussed further. This material uses "complement" to mean two's complement.

The left bit indicates the sign. 0011 is +3. 1101 is a negative; complementing yields the positive version: 0010 + 1 = 0011, which is 3. So 1101 is -3.

Given a negative number like 1110, the value can be obtained by complementing, so 0001 + 1 = 0010, and negating, so -0010. Thus 1110 is -2.

PARTICIPATION ACTIVITY 1.3.2: Two's complement signed number representation.

1) In base ten , what is the complement of 33 (two digits)?

67

Check Show answer

Correct

67

33 + 67 = 100
Note: Base ten is introduced here only for intuition. Digital circuits use base two.

2) In base two , what is the complement of 0010 (four bits)?

1110

Check Show answer

Correct

1110

Invert the bits: 1101
Then add 1: 1110
Check result: 0010 + 1110 = 10000

3) What is -2 in four-bit two's complement representation?

1110

Check Show answer

Correct

1110

+2 is 0010. Complement to obtain the negative representation:
Invert the bits: 1101
Then add 1: 1110
So 1110 represents -2.

4) What is -7 in four-bit two's complement representation?

1001

Correct

1001

Check Show answer

7 is 0111. Complement to obtain the negative representation:
Invert the bits: 1000
Then add 1: 1001

5) Assuming four-bit two's complement representation, is 1011 positive or negative?

negative

Check Show answer

Correct

Negative

Left bit is 1, so negative.

6) Assuming two's complement representation, what base ten number does 1111 represent?

-1

Check Show answer

Correct

-1

Left bit is 1, so negative.
Complement to find positive.
0000 + 1 = 0001
Magnitude is 1. Negate to yield -1.

7) Assuming two's complement representation, what base ten number does 1001 represent?

-7

Check Show answer

Correct

-7

Left bit is 1, so negative.
Complement to find positive.
0110 + 1 = 0111
Magnitude is 7. Negate to yield -7.

8) In base two, for four bits, what is the complement of 0000?

0000

Check Show answer

Correct

0000

Invert the bits: 1111
Then add 1: 10000, which in four bits is 0000.
0000 is somewhat of an exception.

9) What is -3 in eight-bit two's complement representation?

11111101

Check Show answer

Correct

11111101

+3 is 00000011 in 8 bits.
Complement to find negative:

(00000011)' + 1 = 11111100 + 1 = 11111101

Feedback?

Note: This section uses 4-bit numbers for ease of example; wider numbers like 8 or 32 bits are more typical.

Subtracting by adding

Two's complement representation has the benefit of allowing an adder to be used even when dealing with negative numbers. Ex: 5 + -3 is just 0101(5) + 1101(-3) = 10010, or 0010(2) after ignoring the carry. No extensive special circuitry for negative numbers is needed.

PARTICIPATION ACTIVITY 1.3.3: Two's complement arithmetic.

Assume four-bit two's complement representation.

1) 6 + 2 is 0110 + ?

0010

Check Show answer

Correct

0010

2 is 0010 in binary. The leftmost 0 means positive.

2) 6 + -2 is 0110 + ?

1110

Check Show answer

Correct

1110

2 is 0010. The negative is the complement, so 1101 + 1 or 1110.

3) 3 + -4 is 0011 + 1100 = ?

1111

Check Show answer

Correct

1111

Once represented as two-complement, numbers can be added, even if one is negative.

4) 2 - 3 is 0010 + ?

1101

Check Show answer

Correct

1101

2 - 3 is 2 + -3.
-3 is 0011 complemented, so 1100 + 1 or 1101. (The result is 1101 + 0010 = 1111, which is -1).

5) -3 + 2 is ? + 0010

1101

Check Show answer

Correct

1101

Either or both numbers can be negative. The first number is -3, represented as the complement of 3 (0011), so 1100 + 1 = 1101. (The result is 1101 + 0010 = 1111, or -1).

Feedback?

Overflow

The largest positive four-bit two's complement number is 0111, or 7. The smallest negative is 1000, or -8 (0111 + 1 = 1000, so magnitude is 8). Adding two positives, or adding two negatives, may yield a value that can't be represented in the given number of bits, a situation known as ***overflow***. Ex: 0101 (5) + 0011 (3) incorrectly yields 1000, which is -8 in two's complement.

PARTICIPATION ACTIVITY 1.3.4: Overflow.

1 2 3 4 2x speed

Adding two positives
May overflow

0010 + 0011 = 0101 (ok)
2 + 3 = 5

0111 + 0001 = 1000 (overflow)
7 + 1 ≠ -8

Adding two negatives
May overflow

1111 + 1111 = (1)1110 (ok)
-1 + -1 = -2

1000 + 1111 = (1)0111 (overflow)
-8 + -1 ≠ 7

Adding positive and negativ
Cannot overflow

1000 + 0111 = 1111 (ok)
-8 + 7 = -1

0010 + 1000 = 1010 (ok)
2 + -8 = -6

Overflow occurs if numbers' sign bits were 0's but sum's is 1, OR numbers' sign bits were 1's but sum's is 0.

Captions

1. Adding two positives may overflow.
2. Adding two negatives may overflow. Note: Ignore carry-out bit.
3. Adding a positive and negative cannot overflow.
4. Overflow occurs if numbers' sign bits were 0's but sum's is 1, OR numbers' sign bits were 1's but sum's is 0.

Feedback?

As seen above, overflow occurs if the numbers being added have the same sign bit but the sum's sign bit differs. In other words, overflow occurs if two positives sum to a negative (clearly wrong), or two negatives sum to a positive (clearly wrong).

Adding a positive number and negative number (or vice-versa) cannot result in overflow. The sum always has a smaller magnitude than one or both of the numbers, so clearly can fit in the same number of bits. Ex: 7 + -2 = 5, and 5's magnitude is smaller than 7.

PARTICIPATION ACTIVITY 1.3.5: Overflow.

All numbers are in four-bit two's complement representation.

1) 0011 + 0010 results in overflow.

- ○ True
- ◉ False

Correct

0011 (3) + 0010 (2) = 0101 (5). The result can be represented in four-bit two's complement notation.

2) 0111 + 0110 results in overflow.

- ◉ True
- ○ False

Correct

0111 (7) + 0110 (6) ≠ 1101 (-3)

3) 0001 + 1111 results in overflow.

- ○ True
- ◉ False

Correct

0001 (1) + 1111 (-1) = 0000 (0). Adding a negative number and a positive number cannot result in overflow.

4) 1011 + 1110 results in overflow.

- ○ True
- ◉ False

Correct

1011 (-5) + 1110 (-2) = 1001 (-7)

5) Number A's sign bit is 0. Number B's sign bit is 0. The sum's sign bit is 1. The addition resulted in overflow.

- ◉ True
- ○ False

Correct

Overflow occurs when the numbers' sign bits match, but yield a sum with a different sign bit.

6) Number A's sign bit is 1. Number B's sign bit is 0.

Correct

The sum's sign bit is 0. The addition resulted in overflow.

- True
- False (selected)

Adding a negative number and a positive number cannot result in overflow. The numbers' sign bits differ, meaning a negative and positive are being added.

Feedback?

CHALLENGE ACTIVITY 1.3.1: Two's complement.

547404.4091098.qx3zqy7

Jump to level 1

1 2 3 4 5 6

Note: Answer with 4 bits.

Binary operation:

$$\begin{array}{r} 0110 \\ +\ 0111 \\ \hline \end{array}$$

1101

1 2 3 4 5 **6**

Check Next

Done. Click any level to practice more. Completion is preserved.

✓ Expected: 1101

Sum causes overflow:

$$\begin{array}{rl} 0110 & (6) \\ +\ 0111 & (7) \\ \hline 1101 & (-3) \end{array}$$

Feedback?

How was this section?

Provide section feedback