Binary Arithmetic -- Negative numbers and Subtraction

Binary Mathematics

Addition: this is performed using the same rules as decimal except all numbers are limited to combinations of zeros (0) and ones (1).

Using 8-bit numbers

1 1 1

$0000\ 1110_2$ which is 14_{10}

$+\ 0000\ 0110_2$ which is $+\ 6_{10}$

-------------------- -----------

$0001\ 0100_2$ 20_{10}

$0001\ 0100_2$ => we can ignore the leading zeros, since they have no impact on our conversion

~~000~~1 $0100_2 = 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 0x2^0$

$= 16 + 0 + 4 + 0 + 0$

$= 20_{10}$

What about negative numbers? Clearly not all numbers are positive.

What if we wanted to do: $14_{10} + (-6_{10}) = 8_{10}$

Recall that adding a negative number is the equivalent to subtracting a positive number.

So, : $14_{10} + (-6_{10}) = 8_{10}$ ➔ $14_{10} - (+6_{10}) = 8_{10}$

Subtraction: Use standard subtraction rules

$0000\ 1110_2$ which is 14_{10}

$-\ 0000\ 0110_2$ which is $-\ 6_{10}$

-------------------- -----------

$0000\ 1000_2$ 8_{10}

~~0000~~ $1000_2 = 1x2^3 + 0x2^2 + 0x2^1 + 0x2^0 = 8_{10}$

This is rather straightforward.

Try this example:

1

0 10 ~~10~~ 10

000~~1~~ ~~1~~001_2 → 25_{10}

$-\ 0000\ 1110_2$ → $-\ 14_{10}$

-------------------- -----------

$0000\ 1011_2$ 11_{10}

THIS WAS PAINFUL!

Because there is so much borrowing, this is highly error prone for people.

For a computer to solve this math problem using standard subtraction rules, not only is circuitry needed to deal with the subtraction, but additional circuitry would be needed to deal with all the complex borrowing.

Early Computer Scientists and Computer Engineers looked at this and determined that building this complex circuitry would be expensive and the calculations would be slow, even in computer time, because of all the borrowing.

An alternative technique was developed that enabled computer hardware to work with signed numbers using slightly modified addition hardware. This technique is called two's complement arithmetic.

Two's Complement

Two's complement is a way of representing negative numbers so that only addition hardware is required by the computer.

Step 1: Decide how many bits you are going to use for all your operations.

For example: if we use 4-bits, this leaves only 3 bits to hold the number.
This limits our numbers to only very small ones.

A 4-bit number would look like X X X X the left-most bit is considered the **sign** bit
|
This is the sign bit

If we were using 8-bits the left-most bit will contain the sign. This would leave 7 bits to hold the number.

X X X X X X X X
|
This is the sign bit

This sign bit is **reserved** and is no longer one of the digits that make up the binary number. If the sign bit is zero (0) the binary number following it is **positive**. If the sign bit is one (1) the binary number following it is **negative**.

In the binary number

$0000\ 1011_2$ the sign bit is 0 so the number is positive
The binary number is only 7-digits long,

$\sout{0000}\ 1011_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0$
$8 + 0 + 2 + 1 = 11_{10}$

Note: An easier notation for converting binary to decimal

$\sout{0000}\ 1\ 0\ 1\ 1_2$
(8) (2) (1) = $8 + 2 + 1 = 11_{10}$

This is exactly the same as writing out the powers of two, but we only need to remember the values (which are all multiples of two) and can get more quickly to the end result.

In the binary number

$1000\ 1100_2$ the sign bit is 1 so the number is negative
The binary number is only 7-digits long,

$1\text{~~000~~}\ 1\ 1\ 0\ 0_2$
(8) (4) $= 8 + 4 = 12_{10}$

Remember, Step 1, was determine the number of bits being used.
For our example we will use 8 bits.

Example:

$14_{10} - 6_{10} = 14_{10} + (-6_{10}) = 8_{10}$

$0000\ 1110_2 + 1000\ 0110_2 = ?_2$

left-most digit is 0, number is positive — left-most digit is 1, number is negative

Step 2: Find the two's complement of the negative number.

Let's use the number -6. How do we find the two's complement of -6?

Write down the number with the sign bit set to zero(0) -------	$0000\ 0110_2$
a) Flip all the digits The 1 → 0, the 0 → 1	$1111\ 1001_2$
b) Add 1 to this number	+ 1
c) This is now - 6 in the two's complement format	$1111\ 1010_2$

Step 3: **Add** the two's complement in place of the negative number.

So, $14_{10} + (-6_{10}) = 8_{10}$

$$
\begin{array}{r}
0000\ 1110_2 \\
+1111\ 1010_2 \\
\hline
1\ 0000\ 1000
\end{array}
$$

in two's complement format

1 — Overflow bit IGNORE

0000 1000 — this is the positive number 8 in binary

IT Worked!

Example I:

$12_{10} - 9_{10} = 3_{10}$

Or

$12_{10} + (-9_{10}) = 3_{10}$

$0000\ 1100_2 + 1000\ 1001_2 = ?_2$

Positive 12 Negative 9

Step 1: Determine the number of bits we are using. Choose 8 bits

$12_{10} = 0000\ 1100_2$

$-9_{10} = 1000\ 1001_2$

Step 2: Determine the 2's complement of the negative number, or the number to be subtracted.

Find the 2's complement of 9_{10}

Write down the number with the sign bit set to zero(0)	$0000\ 1001_2$	
A) Flip all the digits 0→ 1, 1→ 0	$1111\ 0110_2$	this is the 1's complement
B) Add One(1)	+ 1	

C) This is Two's Complement	$1111\ 0111_2$	Note: left-most bit is 1, negative number

Step 3: Add the numbers together. In this case, add 12_{10} in binary ($0000\ 1100_2$) and the two's complement of **-** 9_{10} in binary (1111 0111). Ignore any overflow.

	1 1 1 1 1
12_{10}	$0000\ 1100_2$
$+ (-9_{10})$	$+\ 1111\ 0111_2$
--------------	-------------------------
3_{10}	$1\ 0000\ 0011_2$

1 — IGNORE Overflow

$0000\ 0011_2$ — this is the positive number 3 in binary

IT Worked!

Example II:

$25_{10} - 14_{10} = 11_{10}$

Or

$25_{10} + (-14_{10}) = 11_{10}$

$0001\ 1001_2 + 1000\ 1110_2 = 0000\ 1011$

Positive 25 Negative 14 Positive 11

Step 1: Determine the number of bits we are using. Choose 8 bits

$25_{10} = 0001\ 1001_2$

Positive 14_{10} = $0000\ 1110_2$

Step 2: Determine the 2's complement of the negative number, or the number to be subtracted.

Find the 2's complement of 9_{10}

Write down the number With the sign bit set to zero(0)	$0000\ 1110_2$	
A) Flip all the digits 0→ 1, 1→ 0	$1111\ 0001_2$	this is the 1's complement
B) Add One(1)	+ 1 -------------------------	
C) This is the Two's Complement form of – 9	$1111\ 0010_2$	Note: left-most bit is 1, negative number

Step 3: Add the numbers together. In this case, add 12_{10} in binary ($0000\ 1100_2$) and the two's complement of 9_{10} in binary (1111 0111). Ignore any overflow.

```
                 1 11
  25₁₀          0001 1001₂
+ (-14₁₀)     + 1111 0010₂
-------------- ------------------------
  11₁₀        1 0000 1011₂
```

1 | IGNORE Overflow

0000 1011: this is the positive number 3 in binary

IT Worked!

In all of the above examples, the resulting number, the answer was positive.

cse@buffalo UNIVERSITY AT BUFFALO STATE UNIVERSITY OF NEW YORK COMPUTER SCIENCE & ENGINEERING

What happens if we are working with numbers where the result is negative?

The basic computational process is the same. Negative numbers are converted to their two's complement equivalent. But since the resulting number is negative, rather than positive, twos complement is used again to convert the number back to standard binary.

Example III:

$9_{10} - 14_{10} = 9_{10} + (-14_{10}) = $ **- 5**

Decimal	**Standard Binary**	
9_{10}	$0000\ 1001_2$	left-most digit says positive
$+ (-14_{10})$	$+\ 1000\ 1110_2$	left-most digit says negative
-------------	--------------------------------	
-5_{10}		

Step 1: Decide on the number of binary digits -- Choose 8
Step 2: Convert the negative number to it's twos complement equivalent.

Write down the negative number with sign bit set to 0	$0000\ 1110_2$
A) Flip the digits	$1111\ 0001_2$
B) Add one (1)	$+\ 1$

C) This is two's complement	$1111\ 0010_2$

Step 3: Add the two numbers together.

9_{10}	$0000\ 1001_2$	
$+ (-14_{10})$	$+\ 1111\ 0010_2$	in two's complement format
-------------	--------------------------------	
	$1111\ 1011_2$	***no overflow, this says result Is still in two's complement form***

Step 4: Convert from two's complement by doing the two's complement process again.

Write down the number	$1111\ 1011_2$	
A) Flip the digits	$0000\ 0100_2$	
B) Add one (1)	$+\ 1$	

C) This is two's complement	$0000\ 0101_2$	positive 5, in standard binary
D) We know the result Is negative, change sign bit	$1000\ 0101_2$	negative 5, in standard binary

Example IV:

- 25 + 18 = - 7

Proceed the same way, converting the negative number to it's two's complement equivalent.

Decimal	Standard Binary	
$- 25_{10}$	$1001\ 1001_2$	left-most digit says negative
$+ 18_{10}$	$+ 0001\ 0010_2$	left-most digit says positive
$- 7_{10}$		

Step 1: Decide on the number of binary digits -- Choose 8
Step 2: Convert the negative number to it's twos complement equivalent.

Write down the negative number with sign bit 0	$0001\ 1001_2$
A) Flip the digits	$1110\ 0110_2$
B) Add one (1)	+ 1
C) This is two's complement	$1110\ 0111_2$

Step 3: Add the two numbers together.

$- 25_{10}$	$1110\ 0111_2$	in two's complement format
$+ 18_{10}$	$+ 0001\ 0010_2$	
	$1111\ 1001_2$	***no overflow, this says result Is still in two's complement form***

Step 4: Convert from two's complement by doing the two's complement process again.

Write down the number	$1111\ 1001_2$	
A) Flip the digits	$0000\ 0110_2$	
B) Add one (1)	+ 1	
C) This is two's complement	$0000\ 0111_2$	positive 7, in standard binary
D) We know the result Is negative, change sign bit	$1000\ 1110_2$	negative 7, in standard binary

Try these problems:

1)	$9_{10} - 7_{10} = 2_{10}$	0000 1001 - 0000 0111 =
2)	$17_{10} - 12_{10} = 5_{10}$	0001 0001 - 0000 1100 =
3)	$18_{10} - 22_{10} = -4_{10}$	0001 0010 - 0001 0110 =
4)	$7_{10} - 19_{10} = -12_{10}$	0000 0111 - 0001 0011 =
5)	$-26_{10} + 10_{10} = -16_{10}$	1001 1010 + 0000 1010 =
6)	$-19_{10} + 6_{10} = -13_{10}$	1001 0011 + 0000 0110 =

Solutions:

1) $9_{10} - 7_{10} = 2_{10}$ 0000 1001 - 0000 0111 =

This is the same as

$9_{10} + (-7_{10}) = 2_{10}$ 0000 1001 + 1000 0111 =

Convert the negative number to it's 2's complement equivalent

```
1000 0111  →   start with sign bit 0    0000  0111
               Flip the digits          1111  1000
                                            +    1
                                        -----------------------
               This is 2's comp         1111  1001
```

Now add the numbers

```
    0000  1001
  + 1111  1001
-------------------------------
  1 0000  0010
  = 2₁₀
```

$= 2_{10}$

2) $17_{10} - 12_{10} = 5_{10}$ 0001 0001 - 0000 1100 =

This is the same as

$17_{10} + (-12_{10}) = 5_{10}$ 0001 0001 + 1000 1100 =

Convert the negative number to it's 2's complement equivalent

```
1000 1100  →   start with sign bit 0    0000  1100
               Flip the digits          1111  0011
                                            +    1
                                        -----------------------
               This is 2's comp         1111  0100
```

Now add the numbers

```
    0001  0001
  + 1111  0100
-------------------------------
  1 0000  0101
```

$= 5_{10}$

3) 18_{10} **-** 22_{10} = - 4_{10} 0001 0010 - 0001 0110 =
This is the same as
18_{10} **+ (**-22_{10}) = - 4_{10} 0001 0010 + 1001 0110 =

Convert the negative number to it's 2's complement equivalent

1001 0110 → start with sign bit 0 0001 0110
Flip the digits 1110 1001
Add one(1) + 1

This is 2's comp 1110 1010

Now add the numbers

```
   0001  0010
+  1110  1010
-------------------------------
   1111  1100
```

No Overflow
This is still in 2's complement
Apply 2's complement to number again

1110 1010 → Start with sign bit 0 0111 1100
Flip the digits 1000 0011
Add one(1) + 1

1000 0100

- 4_{10}

4) 7_{10} **-** 19_{10} = - 12_{10} 0000 0111 - 0001 0011 =
This is the same as
7_{10} **+ (**-19_{10}) = - 12_{10} 0000 0111 + 1001 0011 =

Convert the negative number to it's 2's complement equivalent

cse@buffalo UNIVERSITY AT BUFFALO STATE UNIVERSITY OF NEW YORK COMPUTER SCIENCE & ENGINEERING

1001 0011 →	start with sign bit 0	0001 0011
	Flip the digits	1110 1100
	Add one(1)	+ 1

	This is 2's comp	1110 1101

Now add the numbers

```
  0000  0111
+ 1110  1101
-------------------------------
  1111  0100
```

No Overflow
This is still in 2's complement
Apply 2's complement to number again

1111 0100 →	Start with sign bit 0	0111 0100
	Flip the digits	1000 1011
	Add one(1)	+ 1

		1000 1100
		-12_{10}

5) $-26_{10} + 10_{10} = -16_{10}$ 1001 1010 + 0000 1010 =

Convert the negative number to it's 2's complement equivalent

1001 1010 →	start with sign bit 0	0001 1010
	Flip the digits	1110 0101
	Add one(1)	+ 1

	This is 2's comp	1110 0110

Now add the numbers

```
  1110  0110
+ 0000  1010
-------------------------------
  1111  0000
```

No Overflow
This is still in 2's complement

Apply 2's complement to number again

1111 0000 →	Start with sign bit 0	0111 0000
	Flip the digits	1000 1111
	Add one(1)	+ 1

		1001 0000 = -16_{10}

6) $-19_{10} + 6_{10} = -13_{10}$ 1001 0011 + 0000 0110 =

Convert the negative number to it's 2's complement equivalent

1001 0011 →	start with sign bit 0	0001 0011
	Flip the digits	1110 1100
	Add one(1)	+ 1

	This is 2's comp	1110 1101

Now add the numbers

```
    1110  1101
+   0000  0110
-------------------------------
    1111  0011
```

No Overflow
This is still in 2's complement
Apply 2's complement to number again

1111 0011 →	Start with sign bit 0	0111 0011
	Flip the digits	1000 1100
	Add one(1)	+ 1

		1000 1101 = -13_{10}