Binary Addition

What is Binary Addition

The binary addition operation works similarly to the base 10 decimal system, except that it is a base 2 system. The binary system consists of only two digits, 1 and 0. Most of the functionalities of the computer system use the binary number system. The binary code uses the digits 1's and 0's to make certain processes turn off or on. The process of the addition operation is very familiar to the decimal system by adjusting to the base 2.

Rules of Binary Addition

Binary addition is much easier than the decimal addition when you remember the following tricks or rules. Using these rules, any binary number can be easily added. The four rules of binary addition are:

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 =10

How To Do Binary Addition?

Now, look at the example of the binary addition:101 + 101

Procedure for Binary Addition of Numbers:

101

(+) 101

- **Step 1:** First consider the 1's column, and add the one's column,(1+1) and it gives the result 10 as per the condition of binary addition.
- **Step 2:** Now, leave the 0 in the one's column and carry the value 1 to the 10's column.

1

101

(+) 101

_____

0

- **Step 3: Now add 10's place, 1+(0 + 0) = 1.** So, nothing carries to the 100's place and leave the value 1 in the 10's place

1

101

(+) 101

_____-

10

- **Step 4:** Now add the 100's place (1 + 1) = 10. Leave the value 0 in the 100's place and carries 1 to the 1000's place.

1

101

(+) 101

_____-

1010

So, the resultant of the addition operation is 1010.

When you cross-check the binary value with the decimal value, the resultant value should be the same.

The binary value 101 is equal to the decimal value 5

So, 5 + 5 = 10

The decimal number 10 is equal to the binary number 1010.

Binary Addition Table

The table of adding two binary numbers 0 and 1 is given below:

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	0 (where 1 is carried over)

You can see from the above table, x and y are the two binary numbers. So when we give the input for x = 0 and y = 0, then the output is equal to 0. When x = 0 or 1 and y = 1 or 0, then x+y = 1. But when both x and y are equal to 1, then their addition equals to 0, but the carryover number will equal to 1, which means basically 1 + 1 = 10 in binary addition, where 1 is carry forwarded to the next digit.

Examples of Binary Addition

A few examples of binary additions are as follows:

Example 1: 10001 + 11101

Solution:

```
      1
  1 0 0 0 1
(+) 1 1 1 0 1
_____________
  1 0 1 1 1 0
```

Example 2: 10111 + 110001

Solution:

```
      1 1 1
    1 0 1 1 1
(+) 1 1 0 0 0 1
_______________
  1 0 0 1 0 0 0
```

Binary Substraction

What is Binary Subtraction?

Can you subtract binary numbers? The answer is yes. Subtraction of binary numbers is an arithmetic operation similar to the subtraction of decimal numbers or base 10 numbers. For example, 1 + 1 + 1 = 3 in base 10 and 1 + 1 + 1 = 11 in binary number system. When you add and subtract binary numbers, you will need to be careful when borrowing as these will take place more often.

When you subtract several columns of binary digits, you must take into account the borrowing. When 1 is to be subtracted from 0, the result is 1 where 1 is borrowed from the next highest order bit or digit.

Binary Subtraction Table

The subtraction of binary numbers is given by:

Binary Number	Subtraction Value
0 – 0	0
1 – 0	1
0 – 1	1 (Borrow 1 from next high order digit)
1 – 1	0

Note:

The addition of two binary number 1 and 1 is 10, where we consider 0 and carry forward 1 to next high order. But in the case of subtraction of 1 and 1, the answer is equal to 0, and nothing is carried forward.

In case of decimal subtraction, when 1 is subtracted from 0, then we borrow 1 from next preceding number and make it 10, and after subtraction, it results in 9, i.e. 10 – 1 = 9. But for binary subtraction, it results in 1 only.

Binary Subtraction Rules

Rules and tricks: Binary subtraction is much easier than the decimal subtraction when you remember the following rules:

- 0 – 0 = 0
- 0 – 1 = 1 (with a borrow of 1)
- 1 – 0 = 1
- 1 – 1 = 0

Now, look at the example of the binary subtraction: 101 from 1010

How to Subtract Binary Numbers?

Learn how to do binary subtraction using the example: 1010 – 101

Procedure to do Binary Subtraction:

1010

(-) 101

- **Step 1:** First consider the 1's column, and subtract the one's column,(0 – 1) and it gives the result 1 as per the condition of binary subtraction with a borrow of 1 from the 10's place.
- **Step 2:** After borrowed 1 from the 10's column, the value 1 in the 10's column is changed into the value 0

1 Borrow

1 0 1 0

(-) 1 0 1

_____

1

- **Step 3:** So, subtract the value in the 10's place, (0 – 0) = 0.

1 Borrow

1 0 1 0

(-) 1 0 1

_____

0 1

- **Step 4:** Now subtract the values in 100's place. Borrow 1 from the 1000's place (0 – 1) = 1.

1 1 Borrow

1 0 1 0

(-) 1 0 1

_____

0 1 0 1

So, the resultant of the subtraction operation is 0101.

When you cross-check the binary subtraction resultant value with the decimal value, the resultant value should be the same.

The binary value 1010 is equal to the decimal value 10, and 101 is equivalent to 5

So, 10 – 5 = 5

Therefore, the decimal number 5 is equal to the binary number 0101.

Binary Subtraction Examples

Consider other examples of binary subtractions are as follows:

Example 1: 0011010 – 001100

Solution:

1 1 Borrow

0 0 1 1 0 1 0

(-) 0 0 1 1 0 0

_____

0 0 0 1 1 1 0

Decimal Equivalent :

0 0 1 1 0 1 0 = 26

0 0 1 1 0 0 = 12

Therefore, 26 – 12 = 14

The binary resultant 0 0 0 1 1 1 0 is equivalent to the 14

Example 2: 0100010 – 0001010

Solution:

1 1 Borrow

$0\ 1\ 0\ 0\ 0\ 1\ 0 = 34_{10}$

$(-)\ 0\ 0\ 0\ 1\ 0\ 1\ 0 = 10_{10}$

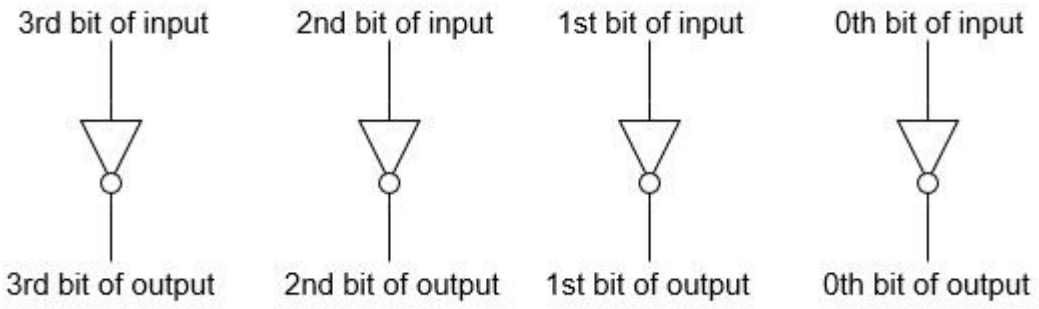
_____

$0\ 0\ 1\ 1\ 0\ 0\ 0 = 24_{10}$

1’s Complement

1’s Complement of a Binary Number:

There is a simple algorithm to convert a binary number into 1’s complement. To get 1’s complement of a binary number, simply invert the given number. You can simply implement logic circuit using only NOT gate for each bit of Binary number input. Implementation of logic circuit of 4-bit 1’s complement is given as following below.



Example-1: Find 1’s complement of binary number 10101110.

Simply invert each bit of given binary number, so 1’s complement of given number will be 01010001.

Example-2: Find 1’s complement of binary number 10001.001.

Simply invert each bit of given binary number, so 1’s complement of given number will be 01110.110.

Example-3: Find 1’s complement of each 3 bit binary number.

Simply invert each bit of given binary number, so 1’s complement of each 3 bit binary number will be,

Binary number	1’s complement
000	111
001	110
010	101
011	100
100	011
101	010
110	001
111	000

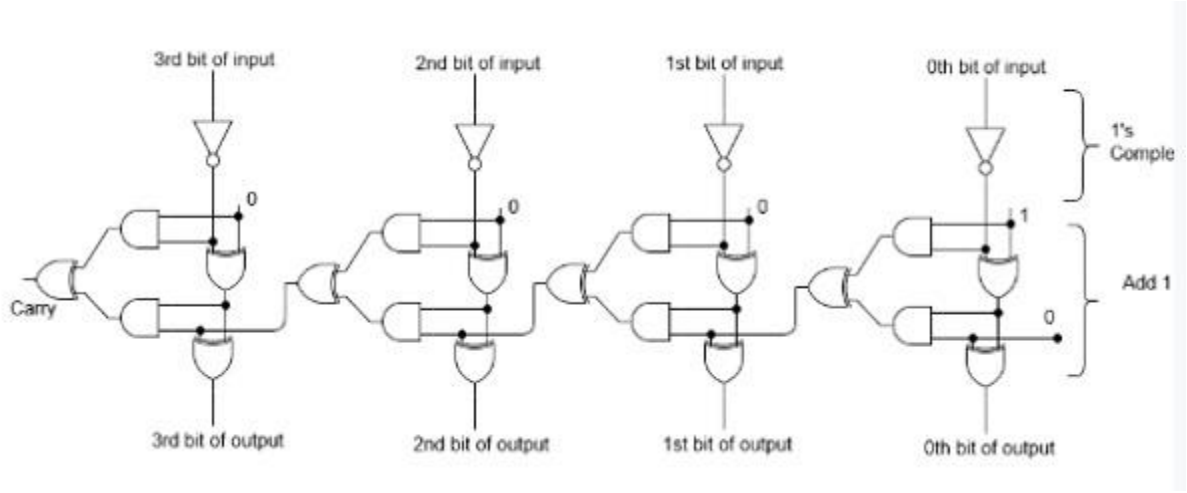
Uses of 1’s Complement Binary Numbers:

There are various uses of 1’s complement of Binary numbers, mainly in signed Binary number representation and various arithmetic operations for Binary numbers, e.g., additions, subtractions, etc.

2’s Complement

2’s Complement of a Binary Number

There is a simple algorithm to convert a binary number into 2’s complement. To get 2’s complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given result. Implementation of 4-bit 2’s complementation number is given as following below.



Example-1 − Find 2’s complement of binary number 10101110.

Simply invert each bit of given binary number, which will be 01010001. Then add 1 to the LSB of this result, i.e., 01010001+1=01010010 which is answer.

Example-2 − Find 2’s complement of binary number 10001.001.

Simply invert each bit of given binary number, which will be 01110.110 Then add 1 to the LSB of this result, i.e., 01110.110+1=01110.111 which is answer.

Example-3 − Find 2’s complement of each 3 bit binary number.

Simply invert each bit of given binary number, then add 1 to LSB of these inverted numbers,

Binary number	1’s complement	2’s complement
000	111	000
001	110	111
010	101	110
011	100	101
100	011	100
101	010	011
110	001	010
111	000	001

Uses of 2's Complement Binary Numbers

There are various uses of 2's complement of Binary numbers, mainly in signed Binary number representation and various arithmetic operations for Binary numbers, e.g., additions, subtractions, etc. Since 2's complement representation is unambiguous, so it very useful in Computer number representation.