*  **Introduction**

Binary number system, in mathematics , positional numeral system employing 2 as the base and so requiring only two different symbols for its digits, 0 and 1, instead of the usual 10 different symbols needed in the Decimal system. The numbers from 0 to 10 are thus in binary 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, and 1010. The importance of the binary system to information theory and computer technology derives mainly from the compact and reliable manner in which 0s and 1s can be represented in electro-mechanical devices with two states—such as “on-off,” “open-closed,” or “go–no go.”

In binary number system there are only 2 digits 0 and 1, and any number can be represented by these two digits. The arithmetic of binary numbers means the operation of addition, subtraction, multiplication and division. Binary arithmetic operation starts from the least significant bit i.e. from the right most side

**Binary Addition:** There are 3 basic rules for adding binary numbers:

* + 0 + 0 = 0
  + 0 + 1 = 1
  + 1 + 1 = 10.

 If the sum of 2 bits is greater than 1, we need to shift a column on the left. In decimal system, 1 + 1 = 2. Binary notation of 2 is 10 (1 \* 2^1 + 0 \* 2^0). So we keep 0 in the 1's column and shift (carry over) 1 to the 2's column.

Other rules are same as the decimal system, i.e. we add from right to left and the carry over get’s added to the digits in the next column.

Now let’s try adding 11 to 13. Binary for 11 is 1011 and that for 13 is 1101.

1011+ 1101

1's col. = 1 + 1 = 10. We keep 0 in 1's col. and carry over 1 to 2's col.

2's col. = 1 + 0 + 1 (carry over) = (1 + 0) + 1 = 1 + 1 = 10. Once again we keep 0 in 2's col. and carry over 1 to 4's col.

4's col. = 0 + 1 + 1 (carry over) = (0 + 1) + 1 = 1 + 1 = 10. Keep 0 in 4's col. and carry over 1 in 8's col.

8's col. = 1 + 1 + 1 (carry over) = (1 + 1) + 1 = 10 + 1 = 11. Keep 1 in 8's col. and carry over 1 in 16's col.

The sum is 11000. 11000 = 1 \* 2^4 + 1 \* 2^3 + 0 \* 2^2 + 0 \* 2^1 + 0 \* 2^0 = 16 + 8 + 0 + 0 + 0 = 24 = 11 + 13.

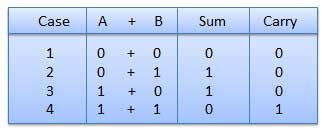
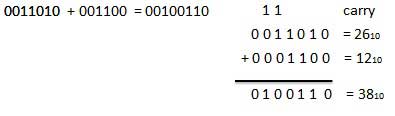


Fig [2.6]: Binary addition

**Example :**

Sol.



**Binary Multiplication:** Multiplication in binary is exactly as it is in decimal, i.e. multiply numbers right to left and multiply each digit of one number to every digit of the other number, them sum them up. The 3 basic binary multiplication rules are also similar to decimal.

* + 1 \* 1 = 1
  + 1 \* 0 = 0 \* 1 = 0
  + 0 \* 0 = 0

Also, remember that for every left shift of digit of the multiplier, an extra zero needs to be appended to the product. This is similar to the decimal system as well.

1011 X 1101

1011 \* 1 (multiplier 1's col) = 1011

1011 \* 0 (multiplier 2's col) = 00000 (one zero appended at the end)

1011 \* 1 (multiplier 4's col) = 101100 (two zero’s appended at the end)

1011 \* 1 (multiplier 8's col) = 1011000 (three zero’s appended at the end)

Sum up. 1011 + 00000 + 101100 + 1011000 = ((1011 + 00000) + 101100) + 1011000 = (01011 + 101100) + 1011000 = 110111 + 1011000 = 10001111

So the product is 10001111 which is = 1 \* 2^7 + 1 \* 2^3 + 1 \* 2^2 + 1 \* 2^1 + 1 \* 2^0 = 128 + 8 + 4 + 2 + 1 = 143 = 11 \* 13.

                                              Fig.[2.7]:Binary Multiplication

**Example :**

Sol.

**Binary Subtraction:** Before trying subtraction, we need to understand how negative numbers are represented in binary. Whatever system is used (i.e. 4-bit, 8-bit, 16-bit etc.), signed number must all have same number of bits. 0s are used to fill up empty bits. We’ll use 8-bit for this tutorial.

                                                         Fig [2.8]: Binary Subtraction

**Example :**

Sol.

**Binary Division:** Binary division is similar to decimal division. The only difference is that in decimal system, since we are dividing traditional numbers, the dividend (or portion of it) can be 0, 1 or more than 1 times of the divisor. However in binary, it can only be 0 or 1 times, i.e. the dividend (or portion of it) is >= or < than the divisor.

Let’s try dividing 6 by 3. Binary of 6 is 110 and that or 3 is 11.

Following decimal division convention

We check for a portion of dividend from its left that is >= the divisor.

Then we subtract the multiple of the divisor that is <= the portion of the dividend. The multiplier (1) gets appended to the quotient and result of the subtraction is the remainder.

We bring down 1 bit at the time (going left to right) for the remaining portion of the dividend and check if the expression (remainder + brought down bit) is >= the divisor. If not, we append 0 to the quotient, or else we follow same step again.

So the steps for 6 / 3 or 110 / 11 are

Is 1 (left most bit of 110) >= 11. No (we don’t need to add 0 to quotient here since 0s on the left are insignificant).

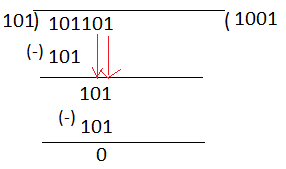
Is 11 (left 2 most bits of 110) >= 11. Yes. We add 1 (multiplier) to the quotient, and subtract 11 from 11. That gives us a remainder of 0. Note, we are subtracting “one-one from one-one NOT eleven from eleven”.

Now we bring down the remaining bit (0) from 110. Is 0 >= 11. No. So we append 0 to the quotient.

Since we have no more bits remaining in the dividend, we stop here and check. Our remainder is 0 and quotient is 10 (binary) = 2.

**Example:**

Sol.



**Applications:** These are basic applications **:**

* + Computer language and programming
  + Digital encoding
  + Digital binary clock,

Frequently used in everyday life in:

* + Accounting
  + Calendar systems
  + Financial systems or daily routine counting.
* **Books:**

[**https://www.amazon.in/Electronics-Analog-Digital-Nagrath-I-J/dp/8120348028**](https://www.amazon.in/Electronics-Analog-Digital-Nagrath-I-J/dp/8120348028)

[Electronics: Analog and Digital by Nagrath I.J-Amazon link](https://www.amazon.in/Electronics-Analog-Digital-Nagrath-I-J/dp/8120348028)

**Lecture Notes:**

[**https://www.tutorialspoint.com/digital\_electronics/index.asp**](https://www.tutorialspoint.com/digital_electronics/index.asp)

**Video Link:**

[**https://www.youtube.com/watch?v=Ri0VhKhw7dY**](https://www.youtube.com/watch?v=Ri0VhKhw7dY)

**References:**

[**https://www.google.com/searchq=number+system+in+digital+electronics&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjOsbqmnvPiAhXPZCsKHXrSBJYQ\_AUIESgC&biw=1366&bih=657#imgrc=DBR-RpBVZAmHDM:**](https://www.google.com/search?q=number+system+in+digital+electronics&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjOsbqmnvPiAhXPZCsKHXrSBJYQ_AUIESgC&biw=1366&bih=657#imgrc=DBR-RpBVZAmHDM:)