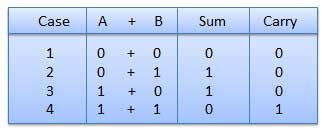
Binary Arithmetic

Binary arithmetic is essential part of all the digital computers and many other digital system.

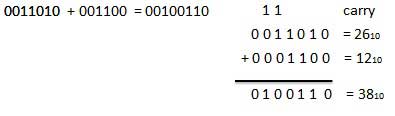
Binary Addition

It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.



In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

Example − Addition



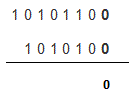
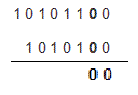
Binary Subtraction

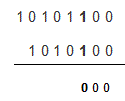
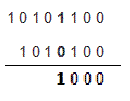
**Subtraction and Borrow**, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction.

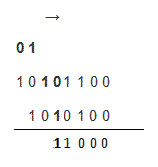
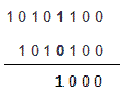
* 0 – 0 = 0.
* 0 – 1 = 1 ( with a borrow of 1)
* 1 – 0 = 1.
* 1 – 1 = 0.

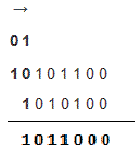
Example − Subtraction

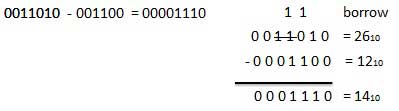
Suppose, A = 10101100 and B = 1010100 and we want to find out A – B.

Now implementing the rules of binary subtraction  
  
The first step is 0 – 0 = 0 and that’s what is written in the place for result  


Similarly again the last step is repeated as here the numbers are both 0 and from the table we know 0 – 0 = 0.  
  
From the table, we can find out that 1 – 1 = 0 and it is written  


The table shows that 1 – 0 = 1 and we have written exactly that in result  
  
Here 0 – 1 = 1 with the borrowing of 1 from the next significant bit and that’s what has been done. We will treat the next 1 as 0 in the next step as shown below.  


As the 1 was borrowed in the previous step we are treating the 1 as 0 and the result is 0 – 0 = 0 and that is written  
  
Again the last 1 has been borrowed because the operation done was 0 – 1 = 1 with borrow 1 from the next most significant bit and the final result of **binary subtraction**, we got is written in the place of result in the final step.



Another approach

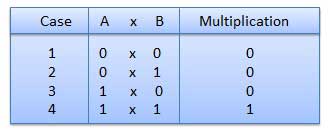
The above example is the most convenient way for us to do binary subtraction by hand. There is another approach however and this is the way that computers subtract binary digits. This approach is called Two's Complement.

Let's say we want to compute 1000 ( 8 ) - 11 ( 3 ).

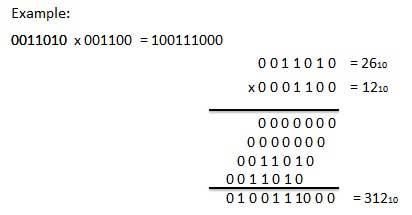
* Step 1: Write the equation out, padding the bottom number with 0's  
  **1000  
  0011 -**
* Step 2: Invert the digits of the lower number  
  **1000  
  1100**
* Step 3: Add 1 to the lower number  
  **1000  
  1101**
* Step 4: Add those two numbers together to get **10101**
* Step 5: Remove the leading 1 (and any 0's after it). You are left with **101** ( 5 ).

Binary Multiplication

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. There are four rules of binary multiplication.

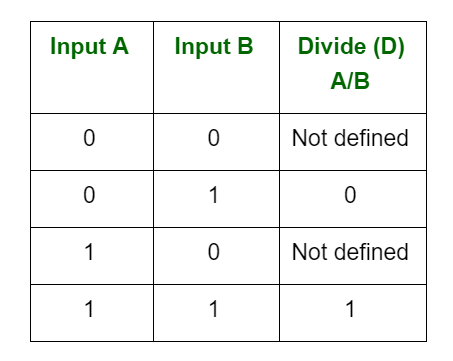


Example − Multiplication



Binary Division

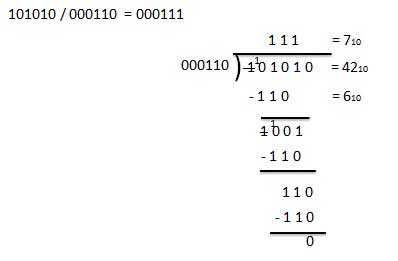
There are four parts in any division: Dividend, Divisor, quotient, and remainder.



The result is always not defined, whenever the divisor is 0.

Binary division is similar to decimal division. It is called as the long division procedure.

Example − Division



Questions:

**110011 + 110000**

**1100011 ( 99 )**

**10110 x 111**

**10011010 ( 154 )**

**1000111 – 1100**

**111011 ( 59 )**

**1011 / 101**

**111011 ( 59 )**