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.

Case	A + B	Sum	Carry
1	0 + 0	0	0
2	0 + 1	1	0
3	1 + 0	1	0
4	1 + 1	0	1

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

0011010 + 001100 = 00100110

```
    1 1          carry
  0 0 1 1 0 1 0  = 26₁₀
 +0 0 0 1 1 0 0  = 12₁₀
 ---------------
  0 1 0 0 1 1 0  = 38₁₀
```

Binary Subtraction

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

Case	A - B	Subtract	Borrow
1	0 - 0	0	0
2	1 - 0	1	0
3	1 - 1	0	0
4	0 - 1	0	1

Example − Subtraction

0011010 - 001100 = 00001110

```
        1 1      borrow
    0 0 1 1 0 1 0   = 26₁₀
   -0 0 0 1 1 0 0   = 12₁₀
   ---------------
    0 0 0 1 1 1 0   = 14₁₀
```

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.

Case	A	x	B	Multiplication
1	0	x	0	0
2	0	x	1	0
3	1	x	0	0
4	1	x	1	1

Example − Multiplication

Example:

0011010 x 001100 = 100111000

```
           0 0 1 1 0 1 0   = 26₁₀
          x0 0 0 1 1 0 0   = 12₁₀
   ------------------------
           0 0 0 0 0 0 0
         0 0 0 0 0 0 0
       0 0 1 1 0 1 0
     0 0 1 1 0 1 0
   ------------------------
     0 1 0 0 1 1 1 0 0 0     = 312₁₀
```

Binary Division

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

Example − Division

101010 / 000110 = 000111

111 = 7_{10}

000110) 101010 = 42_{10}

-110 = 6_{10}

1001

-110

110

-110

0

Octal Arithmetic

Octal Number System

Following are the characteristics of an octal number system.

- Uses eight digits, 0,1,2,3,4,5,6,7.
- Also called base 8 number system.
- Each position in an octal number represents a 0 power of the base (8). Example: 8^0
- Last position in an octal number represents an x power of the base (8). Example: 8^x where x represents the last position - 1.

Example

Octal Number – 12570_8

Calculating Decimal Equivalent –

Step	Octal Number	Decimal Number
Step 1	12570_8	$((1 \times 8^4) + (2 \times 8^3) + (5 \times 8^2) + (7 \times 8^1) + (0 \times 8^0))_{10}$
Step 2	12570_8	$(4096 + 1024 + 320 + 56 + 0)_{10}$
Step 3	12570_8	5496_{10}

Note – 12570_8 is normally written as 12570.

Octal Addition

Following octal addition table will help you to handle octal addition.

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

A (header row), B (first column), Sum (body)

To use this table, simply follow the directions used in this example: Add 6_8 and 5_8. Locate 6 in the A column then locate the 5 in the B column. The point in 'sum' area where these two columns intersect is the 'sum' of two numbers.

```
6₈ + 5₈ = 13₈.
```

Example – Addition

$456_8 + 123_8 = 601_8$

```
  1 1      carry
  4 5 6   = 302₁₀
+ 1 2 3   =  83₁₀
-------
  6 0 1   = 385₁₀
```

Octal Subtraction

The subtraction of octal numbers follows the same rules as the subtraction of numbers in any other number system. The only variation is in borrowed number. In the decimal system, you borrow a group of 10_{10}. In the binary system, you borrow a group of 2_{10}. In the octal system you borrow a group of 8_{10}.

Example – Subtraction

Example:

$456_8 - 173_8 = 333_8$

```
    8       borrow
  ³4 5 6   = 302₁₀
  -1 7 3   = 123₁₀
  -------
   2 6 3   = 179₁₀
```

Hexadecimal Arithmetic

Hexadecimal Number System

Following are the characteristics of a hexadecimal number system.

- Uses 10 digits and 6 letters, 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
- Letters represents numbers starting from 10. A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.
- Also called base 16 number system.
- Each position in a hexadecimal number represents a 0 power of the base (16). Example − 16^0
- Last position in a hexadecimal number represents an x power of the base (16). Example − 16^x where x represents the last position - 1.

Example

Hexadecimal Number − $19FDE_{16}$

Calculating Decimal Equivalent −

Step	Binary Number	Decimal Number
Step 1	$19FDE_{16}$	$((1 \times 16^4) + (9 \times 16^3) + (F \times 16^2) + (D \times 16^1) + (E \times 16^0))_{10}$
Step 2	$19FDE_{16}$	$((1 \times 16^4) + (9 \times 16^3) + (15 \times 16^2) + (13 \times 16^1) + (14 \times 16^0))_{10}$
Step 3	$19FDE_{16}$	$(65536 + 36864 + 3840 + 208 + 14)_{10}$
Step 4	$19FDE_{16}$	106462_{10}

Note − $19FDE_{16}$ is normally written as 19FDE.

Hexadecimal Addition

Following hexadecimal addition table will help you greatly to handle Hexadecimal addition.

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

X (top row); Y (left column); Sum (body)

To use this table, simply follow the directions used in this example − Add A_{16} and 5_{16}. Locate A in the X column then locate the 5 in the Y column. The point in 'sum' area where these two columns intersect is the sum of two numbers.

```
A16 + 516 = F16.
```

$A_{16} + 5_{16} = F_{16}$.

Example − Addition

$4A6_{16} + 1B3_{16} = 659_{16}$

```
 1        carry
 4 A 6  = 1190₁₀
+1 B 3  =  435₁₀
------
 6 5 9  = 1625₁₀
```

Hexadecimal Subtraction

The subtraction of hexadecimal numbers follow the same rules as the subtraction of numbers in any other number system. The only variation is in borrowed number. In the decimal system, you borrow a group of 10_{10}. In the binary system, you borrow a group of 2_{10}. In the hexadecimal system you borrow a group of 16_{10}.

Example - Subtraction

$4A6_{16} - 1B3_{16} = 2F3_{16}$

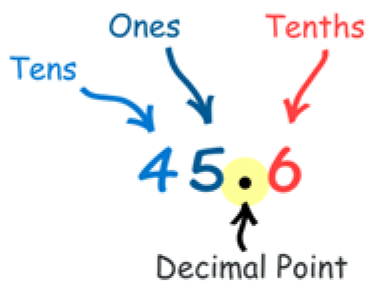
```
   16    borrow
 ³4 A 6  = 1190₁₀
 -1 B 3  =  435₁₀
 ------
  2 F 3  =  755₁₀
```

Floating Number Systems

Decimals

A Decimal Number (*based on the number **10***) contains a **Decimal Point.**

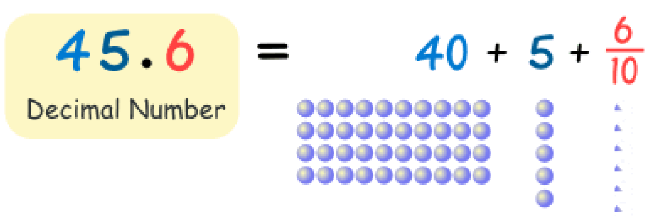
First, let's have an example:



Here is the number *"forty-five and six-tenths"* written as a decimal number:

The decimal point goes between Ones and Tenths.

45.6 has 4 Tens, 5 Ones and 6 Tenths, like this:



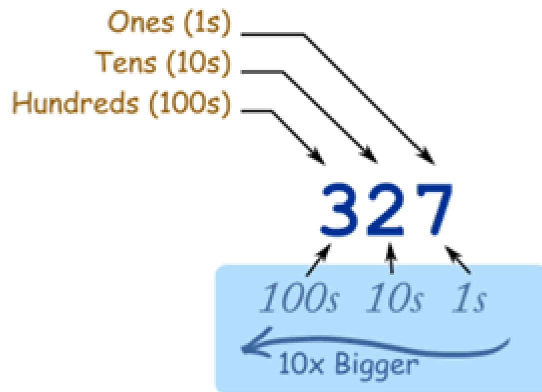
Place Value

It is all about Place Value !

When we write numbers, the **position** (or "**place**") of each digit
is important.

In the number 327:

- the "7" is in the **Ones** position, meaning 7 ones (which is 7),
- the "2" is in the **Tens** position meaning 2 tens (which is twenty),
- and the "3" is in the **Hundreds** position, meaning 3 hundreds.



"Three Hundred Twenty Seven"

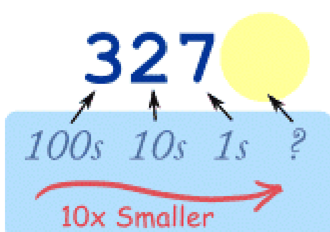
As we move left, each position is 10 times bigger!

Tens are 10 times bigger than **Ones**
Hundreds are 10 times bigger than **Tens**

... and ...

As we move right, each position is 10 times **smaller**.

From **Hundreds**, to **Tens**, to **Ones**

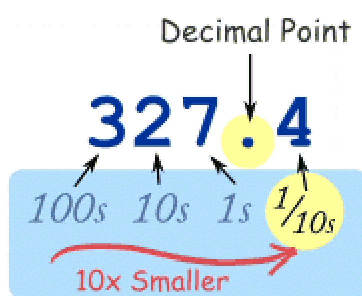


But what if we continue past Ones?

What is **10 times smaller** than Ones?

*1***10**ths (Tenths) are!

But we must first put a **decimal point**,
so we know exactly where the Ones position is:

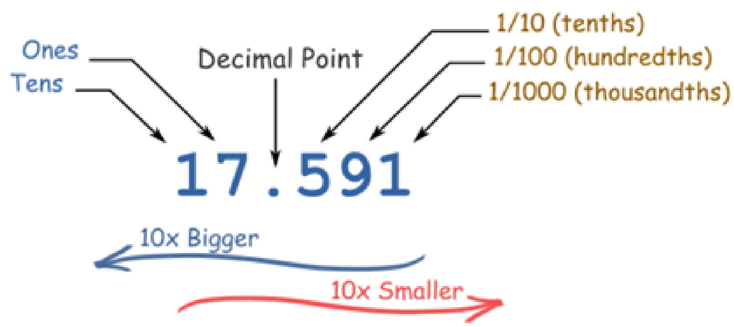


"three hundred twenty seven **and four tenths**"

but we usually just say "three hundred twenty seven **point four**"

And **that** is a Decimal Number!

We can continue with smaller and smaller values, from **tenths**, to **hundredths**, and so on, like in this example:



Large and Small

So, our Decimal System lets us write numbers as large or as small as we want, using the decimal point. Digits can be placed to the left or right of a decimal point, to show values greater than one or less than one.

The **decimal point** is the most important part of a Decimal Number. Without it we are lost, and don't know what each position means.

17.591

Ways to think about Decimal Numbers ...

... as a Whole Number Plus Tenths, Hundredths, etc

We can think of a decimal number as a whole number plus tenths, hundredths, etc:

Example 1: What is 2.3 ?

- On the left side is "2", that is the whole number part.
- The 3 is in the "tenths" position, meaning "3 tenths", or 3/10
- So, 2.3 is "2 and 3 tenths"

Example 2: What is 13.76 ?

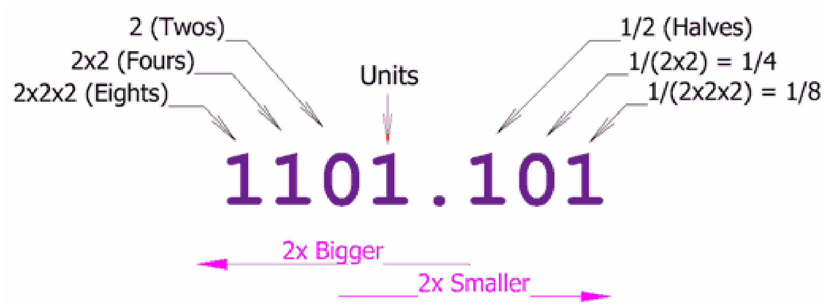
- On the left side is "13", that is the whole number part.
- There are two digits on the right side, the 7 is in the "tenths" position, and the 6 is the "hundredths" position
- So, 13.76 is "13 and 7 tenths and 6 hundredths"

Binary

Position

In the Decimal System there are Ones, Tens, Hundreds, etc

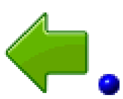
In **Binary** there are Ones, Twos, Fours, etc, like this:



This is 1×8 + 1×4 + 0×2 + 1 + 1×(1/2) + 0×(1/4) + 1×(1/8)
= **13.625 in Decimal**

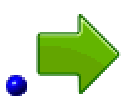
Numbers can be placed to the left or right of the point, to show values greater than one and less than one.

10.1



The number to the left of the point is a whole number (such as 10)

As we move further left, every number place gets **2 times bigger**.



The first digit on the right means **halves** (1/2).

As we move further right, every number place gets **2 times smaller** (half as big).

Example: 10.1

The "10" means 2 in decimal,

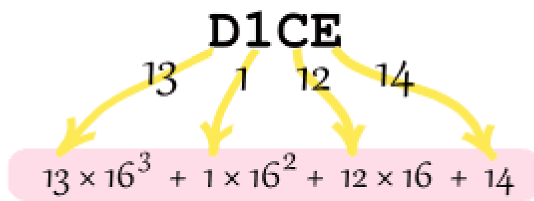
- The ".1" means half,
- So "10.1" in binary is 2.5 in decimal

Hexadecimals

- A Hexadecimal Number is based on the number **16**

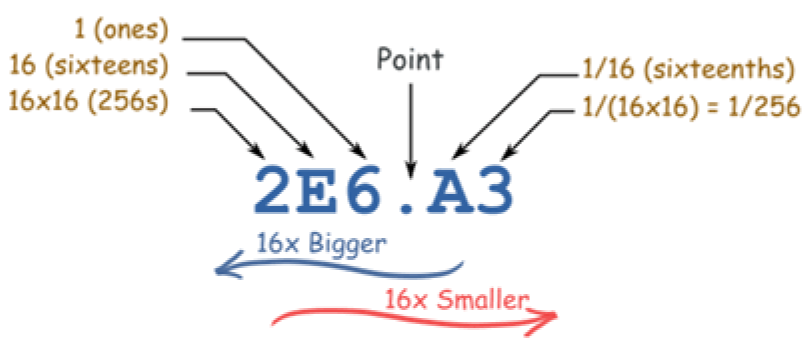
Remember:

Example: What is the decimal value of the hexadecimal number **"D1CE"**



= 53,248 + 256 + 192 + 14
= 53,710 in Decimal

Example: 2E6.A3



This is **2×16×16 + 14×16 + 6 + 10/16 + 3/(16×16)**
=742.63671875 in Decimal
Read below to find out why

Numbers can be placed to the left or right of the point, to show values greater than one or less than one:

The number just to the left of the point is a whole number.

As we move left, every number place is **16 times bigger**.

The first digit on the right of the point means **sixteenths** (1/16).

As we move further right, every number place is **16 times smaller** (one sixteenth as big).

More Examples

Example 1: What is 4B5 (Hexadecimal)?

- The "4" is in the "16×16" position, so that means 4 ×16×16
- The "B" (11) is in the "16" position, so that means 11 ×16
- The "5" is in the "1" position so that means 5.
- Answer: 4B5 = 4×16×16 + 11×16 + 5 (=1205) in Decimal

Example 2: What is 2.3 (Hexadecimal)?

- On the left side is "2", that is the whole number part.
- The 3 is in the "sixteenths" position, meaning "3 sixteenths", or 3/16
- So, 2.3 is "2 and 3 sixteenths" (=2.1875 in Decimal)