

Introduction ToNumber Systems

Collected by: Zainab Alkadhem

Digital systems have such a prominent role in everyday life that we refer to the present technological period as the digital age. Digital systems are used in communication, business transactions, traffic control, space guidance, medical treatment, weather monitoring, the Internet, and many other commercial, industrial, and scientific enterprises. We have digital telephones, digital television, digital versatile discs, digital cameras, and of course, digital computers. The most striking property of the digital computer is its generality. It can follow a sequence of instructions, called a program that operates on given data. The user can specify and change the program or the data according to the specific need. Because of this flexibility, general-purpose digital computers can perform a variety of information processing tasks that range over a wide spectrum of applications.

Digital systems manipulate discrete quantities of information that are represented in binary form.

Operands used for calculations may be expressed in the binary number system. Other discrete elements, including the decimal digits, are represented in binary codes. Data processing is carried out by means of binary logic elements using binary signals. Quantities are stored in binary storage elements.

A decimal number such as 7,392 represents a quantity equal to 7 thousands plus 3 hundreds, plus 9 tens, plus 2 units. The thousands, hundreds, etc. are powers of 10 implied by the position of the coefficients. To be more exact, 7,392 should be written as:

$$7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$$

However, the convention is to write only the coefficients and from their position deduce the necessary powers of 10. In general, a number with a decimal point is represented by a series of coefficients as follows:

$$d_5 d_4 d_3 d_2 d_1 d_0 . d_{-1} d_{-2} d_{-3}$$

The d_j coefficients are any of the 10 digits (0, 1, 2... 9), and the subscript value; gives the place value and, hence, the power of 10 by which the coefficient must be multiplied. This can be expressed as

$$10^5 d_5 + 10^4 d_4 + 10^3 d_3 + 10^2 d_2 + 10^1 d_1 + 10^0 d_0 + 10^{-1} d_{-1} + 10^{-2} d_{-2} + 10^{-3} d_{-3}$$

The decimal number system is said to be of base, or radix, 10 because it uses 10 digits and the coefficients are multiplied by powers of 10.

Binary System

The binary system is a different number system. The coefficients of the binary numbers system have only two possible values: 0 or 1. Each coefficient d is multiplied by 2^n . For example, the decimal equivalent of the binary number 11010.11 is 26.75, as shown from the multiplication of the coefficients by powers of 2:

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$$

As noted before, the digits in a binary number are called bits. When a bit is equal to 0, it does not contribute to the sum during the conversion. Therefore, the conversion from binary to decimal can be obtained by adding the numbers with powers of two corresponding to the bits that are equal to 1. For example,

$$(110101)_2 = 32 + 16 + 4 + 1 = (53)_{10}$$

There are four 1's in the binary number. The corresponding decimal number is the sum of the four powers of two numbers. In computer work, 2^{10} is referred to as K (kilo), 2^{20} as M (mega), 2^{30} as G (giga), and 2^{40} as T (tera). Thus $4K = 2^{12} = 4096$ and $16M = 2^{24} = 16,777,216$. Computer capacity is usually given in bytes. A byte is equal

to eight bits and can accommodate one keyboard character. A computer hard disk with 4 gigabytes of storage has a capacity of $4G = 2^{32}$ bytes (approximately 10 billion bytes).

Example 1

Express the number 101001_2 as a series of coefficients multiplied by powers of 2?

Solution:

$$101001_2 = 1x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0$$

Binary to Decimal Conversion

A binary number can be converted to decimal by forming the sum of powers of 2 of those coefficients whose value is 1.

Example 2

Convert the binary number $(1101001)_2$ to decimal.

Solution:

$$\begin{aligned}(1101001)_2 &= 1x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 \\ &= 64 + 32 + 0 + 8 + 0 + 0 + 1 \\ &= 105\end{aligned}$$

Binary Fractions

In the decimal number system, each digit of a number represents an increasing power of ten. This is true for all digits to the left of the decimal point ... for numbers to the right; each digit represents a decreasing power of ten.

In binary, the concept is the same, except that digits to the right of the "binary point" represent a decreasing power of two.

Example 3:

Convert the $(110.001)_2$ to decimal.

Solution:

$$\begin{aligned}(110.001)_2 &= 1x2^2 + 1x2^1 + 0x2^0 + 0x2^{-1} + 0x2^{-2} + 1x2^{-3} \\ &= 4 + 2 + 0 + 0 + 0 + 0.125 \\ &= 6.125\end{aligned}$$

Example 4

Convert $(0.11101)_2$ to decimal.

Solution:

$$\begin{aligned}(0.11101)_2 &= 1x2^{-1} + 1x2^{-2} + 1x2^{-3} + 0x2^{-4} + 1x2^{-5} \\ &= 0.5 + 0.25 + 0.125 + 0 + 0.03125 \\ &= 0.90625\end{aligned}$$

Decimal to Binary Conversion

Computer use binary number in its operation...it mean every data enter from computer's input devices such as decimal number will be converted by computer into binary number.

Algorithm 1

To convert from a base-10 integer numeral to its base-2 (binary) equivalent, the number is divided by two, and the remainder is the least-significant bit. The (integer) result is again divided by two, its remainder is the next least significant bit. This process repeats until the quotient becomes zero.

Example 5

Convert 23_{10} to binary number.

		Quotient		Remainder
1.	$23 \div 2 =$	11		1
2.	$11 \div 2 =$	5		1
3.	$5 \div 2 =$	2		1
4.	$2 \div 2 =$	1		0
5.	$1 \div 2 =$	0		1
6.	$0 \div 2 =$	0		0
7.	$0 \div 2 =$	0		0
	sign bit			0

The answer is found by reading "up" from the bottom.

Therefore, $23_{10} = 0001\ 0111_2$

Example 6:

Convert 46_{10} to base 2.

Solution:

		Quotient		Remainder
1.	$46 \div 2 =$	23		0
2.	$23 \div 2 =$	11		1
3.	$11 \div 2 =$	5		1
4.	$5 \div 2 =$	2		1
5.	$2 \div 2 =$	1		0
6.	$1 \div 2 =$	0		1
7.	$0 \div 2 =$	0		0
	sign bit			0

Therefore, $46_{10} = 0010\ 1110_2$

Decimal Fractions to Binary Fractions Conversions

to convert the fractional part successive multiplications are done instead of divisions. In each case the remaining fractional part is used in the succeeding multiplication...

Example 7

Convert the decimal fraction 0.59375_{10} to binary fraction.

Solution:

To convert the fractional part (.59375), successive multiplications are done instead of divisions. In each case the remaining fractional part is used in the succeeding multiplication ...

		Integer	Fraction
1.	$.59375 \times 2 =$	1	.1875
2.	$.1875 \times 2 =$	0	.375
3.	$.375 \times 2 =$	0	.75
4.	$.75 \times 2 =$	1	.5
5.	$.5 \times 2 =$	1	.0

The process ends when the fraction becomes zero. The answer is obtained by reading the integer numbers from the top down ...

Therefore $0.59375_{10} = 0.10011_2$

Example 8

Convert 46.59375_{10} to base 2.

Solution:

First, convert the whole number (46) using the previous method ...

		Quotient	Remainder
1.	$46 \div 2 =$	23	0
2.	$23 \div 2 =$	11	1
3.	$11 \div 2 =$	5	1
4.	$5 \div 2 =$	2	1
5.	$2 \div 2 =$	1	0
6.	$1 \div 2 =$	0	1
7.	$0 \div 2 =$	0	0
	sign bit		0

Therefore, $46_{10} = 0010\ 1110_2$

Next, to convert the fractional part (.59375), successive multiplications are done instead of divisions. In each case the remaining fractional part is used in the succeeding multiplication ...

		Integer	Fraction
1.	.59375 x 2 =	1	.1875
2.	.1875 x 2 =	0	.375
3.	.375 x 2 =	0	.75
4.	.75 x 2 =	1	.5
5.	.5 x 2 =	1	.0

The process ends when the fraction becomes zero. The answer is obtained by reading the integer numbers from the top down ...

Therefore $.59375_{10} = .10011_2$

Therefore, $46.59375_{10} = 0010\ 1110.10011_2$

Arithmetic in the Binary System

Binary Addition

The process for adding binary numbers is the same in any number system, except that you must be aware of when (and what) to “carry”.

In the decimal system, a carry occurs when the sum of 2 digits is 10 or more. For example,

$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \hline
 1
 \end{array}$$

In binary, a carry occurs when the sum of 2 binary digits is 2 or more. This leaves only four possibilities:

Addition table

+	0	1
0	0	1
1	1	10

$$\begin{aligned}
 0 + 0 &= 0_2 \\
 0 + 1 &= 1_2 \\
 1 + 1 &= 10_2 \text{ (therefore, 0 with a carry)} \\
 1 + 1 + 1 &= 11_2 \text{ (therefore, 1 with a carry)}
 \end{aligned}$$

Example9:

Add the binary numbers

$$0011\ 0010_2 + 0011\ 0111_2:$$

0	1	1	1	0	0	1	0	=	50
+	0	0	1	1	0	1	1	=	55
									105
	0	1	1	0	1	0	0		

Example10:

Add the binary numbers

$$1011.01_2 + 11.011_2$$

$$\begin{array}{r}
 1101 \\
 + 11.011 \\
 \hline
 1110.101
 \end{array}$$

Binary Subtraction

For binary subtraction, there are *four* facts instead of one hundred:

- $0 - 0 = 0$
- $1 - 0 = 1$
- $1 - 1 = 0$
- $10 - 1 = 1$

The first three are the same as in decimal. The fourth fact is the only new one; it is the borrow case. It applies when the “top” digit in a column is 0 and the “bottom” digit is 1. (Remember: in binary, 10 is pronounced “one-zero” or “two.”)

Example 11:

Subtract:

$$10101.101 - 1011.11$$

Solution:

Step 1: $1 - 0 = 1.$

$$\begin{array}{r} 10101.101 \\ - 1011.11 \\ \hline 1 \end{array}$$

Step 2: Borrow to make $10 - 1 = 1.$

$$\begin{array}{r} 10101.101 \\ - 1011.11 \\ \hline 11 \end{array}$$

Step 3: Borrow to make $10 - 1 = 1.$

$$\begin{array}{r} 10101.101 \\ - 1011.11 \\ \hline 111 \end{array}$$

Step 4: Cascaded borrow to make $10 - 1 = 1.$

$$\begin{array}{r} 10101.101 \\ - 1011.11 \\ \hline 1111 \end{array}$$

Step 5: $1 - 1 = 0.$

$$\begin{array}{r} 01.111 \\ 101 \\ - 1011.11 \\ \hline 001.111 \end{array}$$

Step 6: $0 - 0 = 0.$

$$\begin{array}{r} 001.111 \\ 101 \\ - 1011.11 \\ \hline 1001.111 \end{array}$$

Step 7: Borrow to make $10 - 1 = 1.$

$$\begin{array}{r} 1.111 \\ 101 \\ - 1011.11 \\ \hline 1.111 \end{array}$$

Since there are lots of 0s in binary numbers, there can be lots of borrows — and lots of messy looking cross-outs.

Checking the Answer

You can check the answer in a few ways. One way is to add the result (1001.111) to the subtrahend (1011.11), and check that that answer matches the minuend (10101.101):

$$\begin{array}{r} 1\ 111\ 1 \\ 1001.111 \\ + 1011.11 \\ \hline 10101.101 \end{array}$$

Subtracting a Bigger Number From a Smaller Number

To subtract a bigger number from a smaller number, just swap the numbers, do the subtraction, and negate the result.

Binary Multiplication

Binary multiplication uses the same algorithm as in decimal, but uses just three order-independent facts:

$$0 \times 0 = 0,$$

$$1 \times 0 = 0,$$

$$1 \times 1 = 1$$

(these work the same as in decimal). If you perform the multiplication phase with these facts, you'll notice two things: there are never any carries, and the partial products will either be zeros or a shifted copy of the multiplicand.

Example 12:

Multiply

$$1011.01 \times 110.1$$

Solution:

$$\begin{array}{r} 1011.01 \\ \times 110.1 \\ \hline 101101 \\ 0 \\ 101101 \\ \hline 1001001.001 \end{array}$$

↓

$$\begin{array}{r} 1011.01 \\ \times 110.1 \\ \hline 101101 \\ 0 \\ 101101 \\ \hline 1001001.001 \end{array}$$

↗

$$\begin{array}{r} 1011.01 \\ \times 110.1 \\ \hline 101101 \\ 0 \\ 101101 \\ \hline 1001001.001 \end{array}$$

↓

$$\begin{array}{r} 1011.01 \\ \times 11110.1 \\ \hline 101101 \\ 0 \\ 101101 \\ 101101 \\ 101101 \\ \hline 1001001.001 \end{array}$$

Exercises

1. Convert the following binary numbers to decimal system:

(1) 1110_2

(2) 10.11_2

(3) 0.0000011_2

2. Convert the following decimal numbers to binary system:

(1) 542

(2) 13.34

(3) 0.5635

3. Add:

(1) $1101_2 + 101_2$

(2) $1101.01_2 + 101.1_2$

4. Subtract:

(1) $1010111_2 - 11111_2$

(2) $100101_2 - 10101_2$

5. Multiply:

(1) $11_2 \times 101_2$

(2) $1101.1_2 \times 11.1$

OCTAL NUMBER SYSTEM

The octal, or base 8, number system is a common system used with computers. Because of its relationship with the binary system, it is useful in programming some types of computers.

Octal is fancy for Base Eight meaning eight symbols are used to represent all the quantities. They are 0, 1, 2, 3, 4, 5, 6, and 7.

Octal	0	1	2	3	4	5	6	7	10	11	12 ...	17	20 ...	30 ...	77	100
Decimal	0	1	2	3	4	5	6	7	8	9	10 ...	15	16 ...	24 ...	63	64

Just like how we used powers of ten in decimal and powers of two in binary, to determine the value of a number we will use powers of 8 since this is Base Eight.

Octal to decimal conversion

An octal number can be converted to decimal by forming the sum of powers of 8 of the coefficients.

Example1

Convert 764_8 to decimal:

Solution

$$764_8 = 7 \times 8^2 + 6 \times 8^1 + 4 \times 8^0 = 448 + 48 + 4 = 500_{10}$$

Example2:

$$65_8 = 6 \times 8 + 5 = 53_{10}$$

Example 2

Convert 0.235_8 to decimal:

Solution:

$$\begin{aligned} 0.235_8 &= 2 \times 8^{-1} + 3 \times 8^{-2} + 5 \times 8^{-3} \\ &= 2 \times 0.125 + 3 \times 8^{-2} + 5 \times 8^{-3} \\ &= 0.30664 \end{aligned}$$

Decimal to octal conversion

Conversion of decimal fraction to octal fraction

To convert a decimal fraction to octal, multiply by 8; the integer part of the result is the first digit of the octal fraction. Repeat the process with the fractional part of the result, until it is null or within acceptable error bounds.

Example3:

Convert 0.1640625 to octal:

$$0.1640625 \times 8 = 1.3125 = \mathbf{1} + 0.3125$$

$$0.3125 \times 8 = 2.5 = \mathbf{2} + 0.5$$

$$0.5 \times 8 = 4.0 = \mathbf{4} + 0$$

Therefore, $0.1640625_{10} = 0.124_8$.

Example4:

convert $(0.523)_{10}$ to octal equivalent up to 3 decimal places.

Solution

$$0.523 \times 8 = 4.184, \text{ its integer part is } 4$$

$$0.184 \times 8 = 1.472, \text{ its integer part is } 1$$

$$0.472 \times 8 = 3.776, \text{ its integer part is } 3$$

So the answer is $(0.413..)_{8}$

Conversion of decimal to octal (base 10 to base 8)

Example5:

convert $(177)_{10}$ to octal equivalent

$$177 / 8 = 22 \text{ remainder is } 1$$

$$22 / 8 = 2 \text{ remainder is } 6$$

$$2 / 8 = 0 \text{ remainder is } 2$$

Answer = 261

Note: the answer is read from bottom to top as $(261)_8$, the same as with the binary case.

Octal to binary conversion

Numeral Systems Conversion Table

Decimal Base-10	Binary Base-2	Octal Base-8	Hexadecimal Base-16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12

19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F
32	100000	40	20

To convert octal to binary, replace each octal digit by its binary representation in 3 bits, so add zeros to the left if necessary.

Example 6:

Convert 51_8 to binary:

Solution

$$5_8 = 101_2$$

$$1_8 = 001_2$$

Therefore, $51_8 = 101\ 001_2$.

Binary to Octal conversion

The Binary numbers base is represented by 2 and the Octal numbers base is represented by 8. The third power of binary numbers is called as Octal numbers. In order to convert the Binary number into its equivalent octal numbers, split the given binary number into groups and each group should contain ***three binary bits*** (because $2^3=8$), add zeros to the left if necessary, and then converting each group into its equivalent octal number.

Example 7:

convert binary 1010111100 to octal.

Solution:

001	010	111	100
1	2	7	4

Therefore, $1010111100_2 = 1274_8$.

Convert binary 11100.01001 to octal:

011	100	.	010	010
3	4	.	2	2

Therefore, $11100.01001_2 = 34.22_8$.

Arithmetic in octal system

Octal Addition

Addition of the octal number is carried out in the same way as the decimal addition is performed. The steps are given below:

1. First, add the two digits of the unit column of the octal number in decimal.
2. This process is repeated for each larger significant digit of the octal number.
3. During the process of addition, if the sum is less than or equal to 7, then it can be directly written as an octal digit.
4. If the sum is greater than 7, then subtract 8 from the digit and carry 1 to the next digit position.
5. Note that in this addition the largest octal digit is 7.

Example 8:

Evaluate:

(i) $(162)_8 + (537)_8$

1 1 <---- carry

1 6 2

$$\begin{array}{r} \underline{537} \\ 721 \end{array}$$

Therefore, sum = 721_8

(ii) $(136)_8 + (636)_8$

Solution:

1 <---- carry

1 3 6

$$\begin{array}{r} \underline{636} \\ 774 \end{array} \quad \leftarrow \text{---- } 6+6=12>8 \text{ in decimal, so in octal } 6+6=12-8=4 \text{ (4 and carry 1)}$$

Therefore, sum = 774_8

(iii) $(25.27)_8 + (13.2)_8$

1 <---- carry

2 5 . 2 7

$$\begin{array}{r} \underline{13.2} \\ 40.47 \end{array} \quad \begin{array}{l} \uparrow \\ \text{---} \end{array} \quad 5+3=8>7 \text{ in decimal, so in octal } 5+3=8-8=0 \text{ (0 and carry 1)}$$

Therefore, sum = $(40.47)_8$

(iv) $(67.5)_8 + (45.6)_8$

1 1 <---- carry

6 7 . 5

$$\begin{array}{r} \underline{45.6} \\ 135.3 \end{array} \quad \begin{array}{l} \uparrow \\ \text{---} \\ \uparrow \\ \text{---} \\ \uparrow \\ \text{---} \end{array} \quad \begin{array}{l} 5+6=11>8 \text{ in decimal, so in octal } 5+6=11-8=3 \\ 1+7+5=13>8, \text{ so in octal } 13=13-8=5 \text{ (5 and carry 1)} \\ 1+6+4=11>8, \text{ so in octal } 11=11-8=3 \text{ (3 and carry 1)} \end{array}$$

Therefore, sum = $(135.3)_8$

Subtraction of Octal Numbers

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

Consider the subtraction of 1 from 10 in decimal, binary, and octal number systems:

<u>DECIMAL</u>	<u>BINARY</u>	<u>OCTAL</u>
$\begin{array}{r} 10_{10} \\ - 1_{10} \\ \hline 9_{10} \end{array}$	$\begin{array}{r} 10_2 \\ - 1_2 \\ \hline 1_2 \end{array}$	$\begin{array}{r} 10_8 \\ - 1_8 \\ \hline 7_8 \end{array}$

In each example, you cannot subtract 1 from 0 and have a positive difference. You must use a borrow from the next column of numbers. Let's examine the above problems and show the borrow as a *decimal* quantity for clarity:

¹⁰	²	⁸ Borrow
$\begin{array}{r} \cancel{10}_{10} \\ - 1_{10} \\ \hline 9_{10} \end{array}$	$\begin{array}{r} \cancel{10}_2 \\ - 1_2 \\ \hline 1_2 \end{array}$	$\begin{array}{r} \cancel{10}_8 \\ - 1_8 \\ \hline 7_8 \end{array}$

When you use the borrow, the column you borrow from is reduced by 1, and the amount of the borrow is added to the column of the minuend being subtracted.

Example 9

Subtract $532_8 - 174_8$

Solution:

$$\begin{array}{r} \\ +8 \\ +8 \\ \\ - 174_8 \\ \hline 336 \end{array}$$

Steps:

1. Since $2 < 4$ then borrow 1 from 3 and add 8 to 2.
2. $2+8=10$ in decimal so $10-4=6$.
3. In the second column we have 2 after borrowing but $2 < 7$, so we need to borrow 1 from 5 and add 8 to 2.
4. $2+8=10$, $10-7=3$.
5. We have 4 after borrowing so we have $4-1=3$.

Exercises

1. Convert to decimal:

- (1) 502_8
- (2) 0.24_8
- (3) 105.105_8

2. Convert to octal system:

- (1) 525
- (2) 726
- (3) 0.125
- (4) 1.25

3. Convert to octal system:

- (1) 100101_2
- (2) 11101.11_2
- (3) 11100.0001_2

4. Find:

- (1) $3502_8 + 1243_8$
- (2) $43324_8 + 2015_8$
- (3) $4204_8 - 3131_8$
- (4) $1667_8 - 4006_8$

Hexadecimal System

Hexadecimal is the name given to a special number system which uses "16" as a base.

In a hexadecimal (base 16) system, we need 16 single digits. We could use 0-9, then invent six more. More conveniently, we use the letters A - F for the remaining digits (where A = 10, B = 11, C = 12, D = 13, E = 14 and F = 15).

Hexadecimal to Decimal Conversion

The principle of converting a base 16 number to decimal is the same as previously discussed, except that each column now represents an increasing power of 16 .

Example 1

Convert $D30C_{16}$ to decimal.

Solution

$$\begin{aligned} D30C_{16} &= 13 \times 16^3 + 3 \times 16^2 + 0 \times 16^1 + 12 \times 16^0 \\ &= 13 \times 4096 + 3 \times 256 + 0 + 12 \\ &= 53248 + 768 + 12 \\ &= 54028 \end{aligned}$$

Decimal To Hexadecimal Conversion

Similarly, any decimal number can be converted to hexadecimal by successive divisions by 16, keeping track of the remainder.

Example 2

Convert $2,563_{10}$ to base 16,

Solution

		Quotient	Remainder
1.	$2,563 \div 16 =$	160	3
2.	$160 \div 16 =$	10	0
3.	$10 \div 16 =$	0	10

The process stops when the quotient becomes zero. The answer is found by reading "up" from the bottom.

Therefore, $2,563_{10} = A03_{16}$

Binary To Hexadecimal Conversion

Hexadecimal has another important property. Since there are *exactly* 16 hexadecimal digits, it requires *exactly* 4 bits to represent *every* hexadecimal digit (since $2^4 = 16$). In order to convert the Binary number into its equivalent octal numbers, split the given binary number into groups and each group should contain four binary bits (because $2^4=16$), add zeros to the left if necessary, and then converting each group into its equivalent octal number. This can be shown by the following table on the right:

Binary	Hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Example 3:

Convert $(100000111001110)_2$ to hexadecimal.

Solution:

- $0010\ 0001\ 1100\ 1110_2 = 41CE_{16}$.

Hexadecimal To Binary Conversion

To convert Hexadecimal to binary, replace each Hexadecimal digit by its binary representation in 3 bits, so add zeros to the left if necessary.

Example 4:

Convert $F2D3_{16}$ to binary.

Solution: $F2D3_{16} = 1111\ 0010\ 1101\ 0011_2$

Octal to Hexadecimal Conversion

When converting from octal to hexadecimal, it is often easier to first convert the octal number into binary and then from binary into hexadecimal.

Example 5

Convert 345 octal into hex.

Solution:

Octal = 3 4 5

Binary = 011 100 101 = 011100101

Drop any leading zeros or pad with leading zeros to get groups of four binary digits (bits):

Binary 011100101 = 1110 0101

Then, look up the groups in a table to convert to hexadecimal digits.

Binary = 1110 0101

Hexadecimal = E 5 = E5

Hexadecimal To Octal Conversion

When converting from hexadecimal to octal, it is often easier to first convert the hexadecimal number into binary and then from binary into octal.

Example 6

Convert A2DE hex into octal:

Solution:

Hexadecimal = A 2 D E

Binary = 1010 0010 1101 1110 = 1010001011011110 binary

Add leading zeros or remove leading zeros to group into sets of three binary digits.

Binary: 1010001011011110 = 001 010 001 011 011 110

Then, look up each group in a table:

Binary:	000	001	010	011	100	101	110	111
Octal:	0	1	2	3	4	5	6	7

Binary = 001 010 001 011 011 110

Octal = 1 2 1 3 3 6 = 121336 octal

Hexadecimal Addition:

Use the following steps to perform hexadecimal addition:

1. Add one column at a time.
2. Convert to decimal and add the numbers.
 - 3a. If the result of step two is 16 or larger subtract the result from 16 and carry 1 to the next column.
 - 3b. If the result of step two is less than 16, convert the number to hexadecimal

Example 7:

Add AC5A9+ED694

Solution:

Carry Over:
1. Add one column at a time 2. Convert to decimal & add ($9 + 4 = 13$) 3. Follow less than 16 rule Decimal 13 is hexadecimal D

A	C	5	A	9	
E	D	6	9	4	
				D	

Carry Over:
1. Add next column 2. Convert to decimal & add ($10 + 9 = 19$) 3. Follow 16 or larger than 16 rule ($19 - 16 = 3$ carry a 1)

			1		
A	C	5	A	9	
E	D	6	9	4	
			3	D	

Carry Over:
1. Add next column 2. Convert to decimal & add ($1 + 5 + 6 = 12$) 3. Follow less than 16 rule, convert to hex Decimal 12 is hexadecimal C

			1		
A	C	5	A	9	
E	D	6	9	4	
		C	3	D	

Carry Over:
1. Add next column 2. Convert to decimal & add ($12 + 13 = 25$) 3. Follow 16 or larger than 16 rule ($25 - 16 = 9$ carry a 1)

1					
A	C	5	A	9	
E	D	6	9	4	
9	C	3	D		

Carry Over:
1. Add next column 2. Convert and add ($1 + 10 + 11 = 22$) 3. Follow 16 or larger than 16 rule ($22 - 16 = 6$ carry a 1)

1					
A	C	5	A	9	
B	D	6	9	4	
6	9	C	3	D	

Carry Over:	
1.	Add next column
2.	Convert and add ($1 + 0 + 0 = 1$)
3.	Follow less than 16 rule

1						
0	A	C	5	A	9	
0	B	D	6	9	4	
1	6	9	C	3	D	

Hexadecimal Subtraction

Subtraction in hexadecimal works similar to subtraction in decimal except that we occasionally have a digit larger than 9.

Example:

Subtract $A8D2 - 3EAC$ (hexadecimal)

Solution:

- We'll align our numbers:

$$\begin{array}{r} A\ 8\ D\ 2 \\ -\ 3\ E\ A\ C \\ \hline \end{array}$$

- Now in the ones place, we can't subtract C (12) from 2 so we borrow 1 from the sixteens place.

$$\begin{array}{r} \ 8\ \overset{12}{\cancel{D}}\ \overset{18}{2} \\ -\ 3\ E\ A\ C \\ \hline \ 2\ 6 \end{array}$$

(D = 13 minus the 1 we borrowed) and gives us 18 ones (2 plus the 16 we got from the borrow), then subtract $18 - 12 = 6$.

- Now we don't need to borrow because we can subtract 10 (A) from 12:

$$\begin{array}{r} \ 8\ \overset{12}{\cancel{D}}\ \overset{18}{2} \\ -\ 3\ E\ A\ C \\ \hline \ 2\ 6 \end{array}$$

- In the 256's place, we again need to borrow. We'll borrow 1 from the 4096's place and exchange it for sixteen 256's (one 4096 equals sixteen 256's). This leaves us 9 in the 4096's place ($A = 10$ minus the 1 that we borrowed), and gives us 24 in the 256's place (8 plus the 16 from the borrow). We then can subtract $24 - 14 = 10 = A$. So we have:

$$\begin{array}{r} \overset{9}{\cancel{A}}\ \overset{24}{\cancel{8}}\ \overset{12}{\cancel{D}}\ \overset{18}{2} \\ -\ 3\ E\ A\ C \\ \hline A\ 2\ 6 \end{array}$$

- Finally, we subtract $9 - 3 = 6$ in the 4096's place:

$$\begin{array}{r} \overset{9}{\cancel{A}}\ \overset{24}{\cancel{8}}\ \overset{12}{\cancel{D}}\ \overset{18}{2} \\ -\ 3\ E\ A\ C \\ \hline 6\ A\ 2\ 6 \end{array}$$

Exercises

1. Convert to binary system:

- (1) $A13_{16}$
- (2) $100A_{16}$

2. Convert to octal system:

- (1) $A9B_{16}$
- (2) $ABCDE_{16}$

3. Convert to decimal:

- (1) $21E_{16}$
- (2) $AEF94_{16}$

4. Convert to hexadecimal:

- (1) 87
- (2) 9876

**5. Convert the following binary numbers to
(a) decimal (b) octal and (c) hexadecimal:**

- (1) 10010_2
- (2) 111110.11111_2

References

1. <http://www.wikihow.com/Convert-from-Binary-to-Decimal>
2. http://www.cci-compeng.com/Unit_1_Representing_Data/1309_Fractions.htm
3. <http://syedatnsu.tripod.com/chap1.pdf>
4. **Digital logic and computer design, Morris mano:**
5. http://www.4shared.com/office/NKpFEyey/Digital_Logic_And_Computer_Des.htm
6. <http://www.exploringbinary.com/binary-addition/>
7. **The Pearson Guide to MCA Entrance Examinations by Edgar Thorpe**
8. <http://www.robotroom.com>