Module 4: Number Systems and Binary Arithmetic

4.1 Number Systems: Decimal, Binary, Octal, Hexadecimal
4.2 Conversions between Number System
4.3 Binary Arithmetic: Addition and Subtraction (Using 1's and 2's complement)
4.4 Introduction to Binary code, BCD Code, Gray Code and its code conversions.

Self-Learning Topics: Practice and understand the process of code conversion through more examples.

4.1 Number Systems: Decimal, Binary, Octal, Hexadecimal

A number system or numeral system is defined as an elementary system to express numbers and figures. It is the unique way of representing of numbers in arithmetic and algebraic structure. The writing system for denoting numbers using digits or symbols in a logical manner is defined as a Number system. The numeral system represents a useful set of numbers, reflects the arithmetic and algebraic structure of a number, and provides standard representation.
For representing the information, we use the number system in the digital system.
The digit value in the number system is calculated using:

1. The digit
2. The index, where the digit is present in the number.
3. Finally, the base numbers, the total number of digits available in the number system.

In digital electronics, the number system is *used for representing the information.* The number system has different bases and the most common of them are the decimal, binary, octal, and hexadecimal. The base or radix of the number system is the total number of the digit used in the number system. *Suppose if the number system representing the digit from 0 – 9 then the base of the system is the 10.*

Types of Number Systems

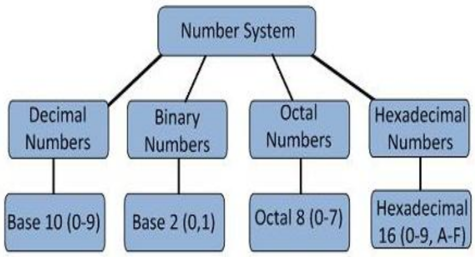
Some of the important types of number system are:

Decimal Number System

Binary Number System

Octal Number System

Hexadecimal Number System



Decimal Number System

The decimal numbers are used in our day-to-day life. The decimal number system contains ten digits from 0 to 9 (base 10). Here, the successive place value or position, left to the decimal point holds units, tens, hundreds, thousands, and so on. The position in the *decimal number system specifies the power of the base (10).* The 0 is the minimum value of the digit, and 9 is the maximum value of the digit.
For example, the decimal number 2541 consist of the digit 1 in the unit position, 4 in the tens position, 5 in the hundreds position, and 2 in the thousand positions and the value will be written as:

$= (2\times1000) + (5\times100) + (4\times10) + (1\times1)$

$= (2\times10^3) + (5\times10^2) + (4\times10^1) + (1\times10^0)$

$=2000 + 500 + 40 + 1$

$=2541$

Binary Number System

Generally, a binary number system is used in the digital computers. In this number system, it carries only two digits, either 0 or 1. There are two types of electronic pulses present in a binary number system. The first one is the absence of an electronic pulse representing '0' and second one is the presence of electronic pulse representing '1'. Each digit is known as a bit. A four-bit collection (1101) is known as a nibble, and a collection of eight bits (11001010) is known as a byte. The location of a digit in a binary number represents a specific power of the base (2) of the number system.

Characteristics:

It holds only two values, i.e., either 0 or 1. It is also known as the base 2 number system. The position of a digit represents the 0 power of the base (2). Example: 2^0

The position of the last digit represents the x power of the base 2.

Example: 2^x, where x represents the last position, i.e., 1

Examples: $(10100)_2$, $(11011)_2$, $(11001)_2$, $(000101)_2$, $(011010)_2$.

Octal Numbers

The octal number system has base 8 (means it has only eight digits from 0 to 7). There are only *eight possible digit values to represent a number*. With the help of only three bits, an octal number is represented. Each set of bits has a distinct value between 0 and 7.

Characteristics:

An octal number system carries eight digits starting from 0, 1, 2, 3, 4, 5, 6, and 7. *It is also known as the base 8 number system*. The position of a digit represents the 0 power of the base 8. Example: 8^0

The position of the last digit represents the x power of the base 8.

Example: 8^x, where x represents the last position, i.e., 1

Examples: $(273)_8$, $(5644)_8$, $(0.5365)_8$, $(1123)_8$, $(1223)_8$.

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

Hexadecimal Numbers

It is another technique to represent the number in the digital system called the hexadecimal number system. The number system has a base of 16 means there are total 16 symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) used for representing a number. The single-bit representation of decimal values10, 11, 12, 13, 14, and 15 are represented by A, B, C, D, E, and F. Only *4 bits are required for representing a number* in a hexadecimal number. Each set of bits has a distinct value between 0 and 15.

Characteristics

It has ten digits from 0 to 9 and 6 letters from A to F.

The letters from A to F defines numbers from 10 to 15. It is also known as the base 16 number system.

In hexadecimal number, the position of a digit represents the 0 power of the base(16).

Example: 16^0

In hexadecimal number, the position of the last digit represents the x power of the base(16).

Example: 16^x, where x represents the last position, i.e., 1

Examples: $(FAC2)_{16}$, $(564)_{16}$, $(0ABD5)_{16}$, $(1123)_{16}$, $(11F3)_{16}$.

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A

The table shows the decimal, binary, octal, and hexadecimal numbers from 0 to 15 and their equivalent binary number.

Conversions between Number Systems

Number system conversions deal with the operations to change the base of the numbers. For example, to change a decimal number with base 10 to binary number with base 2. We can also perform the arithmetic operations like addition, subtraction, multiplication on the number system. The representation of number system base conversion in general form for any base number is;

$(Number)_b = d_{n-1}d_{n-2}$——.$d_1d_0$.$d_{-1}d_{-2}$—- d_{-m}

In the above expression, $d_{n-1}d_{n-2}$——.d_1d_0 represents the value of integer part and $d_{-1}d_{-2}$ —d_{-m}represents the fractional part.

Also, $d_{n-1}$is the Most significant bit (MSB) and d_{-m} is the Least significant bit (LSB).

Binary to Decimal Conversion Methods

The decimal number system is represented by the digit 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
It consists ten digits and hence the base of the system is 10.
Their base is increased by the factor of 10.
On the other hand, the binary number system consists only two digits 1 and 0.
The base of the binary number system is 2, and it is increased by the factor of two.
The first digit has 2^0 weights, the second has 2^1 weights, the weight of the third digit is $2^2$and so on.
Consider a decimal number 3285 it can be written as:

$3285 = 3000 + 200 + 80 + 5$
$3285 = 3(10^3) + 2(10^2) + 8(10^1) + 5(10^0)$

Thus, the decimal number system is an example of positional notation where *each digit position has a weight in terms of powers of ten.* In the binary number system each digit position has a weight regarding *powers of two.*

Examples of a binary digit

2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
↓	↓	↓	↓	↓	↓	↓	↓
8	4	2	1	0.5	0.25	0.125	0.0625

In binary to decimal conversion when there is one in a digit position of a binary number, then the weight of the position is added. But when there is the zero in a binary position the weight of the position is disregarded.

Example: Considered the conversion of the binary number 10101 into its equivalent decimal numbers.

Solution:

1	0	1	0	1
2^4	$\cancel{2}^3$	2^2	$\cancel{2}^1$	2^0
Adding the remaining number				
2^4	+	2^2	+	2^0 =21

On disregarding the weight 2^3 and 2^1 and summing up the remaining weights, we get the equivalent decimal number 21.

Ans: $(10101)_2 = (21)_{10}$

Example: Convert $(1101)_2$ into a decimal number.

Solution: Now, multiplying each digit from MSB to LSB with reducing the power of the base number 2.

$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

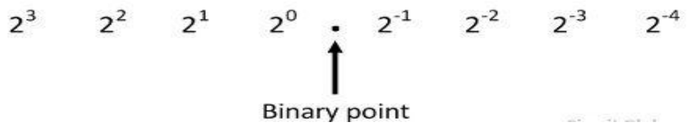
$= 8 + 4 + 0 + 1$

$= 13$

Ans: $(1101)_2 = (13)_{10}$

Binary to Decimal Conversion for Mixed Numbers

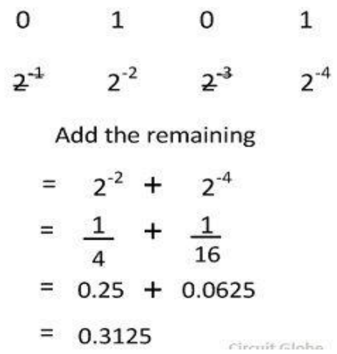
The number with an integer and fractional part are called mixed number.
The weight for a mixed number can be written as:



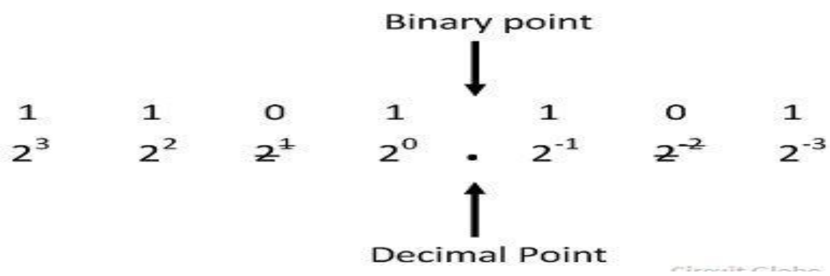
For a fractional number, the weights of the digit position to the right of binary point are given by: 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}… etc.

Example: Convert the fractional binary number .0101 into its equivalent decimal numbers.
Solution:

After disregarding the weight $2^{-1}$and 2^{-3} and summing up the remaining weights, i.e., we get the decimal number 0.3125.



Example: Convert of a binary number 1101.101 into its equivalents decimal number.
Solution:



After disregarding the weight $2^1$and 2^{-2} and on summing up the remaining weights the required decimal number is 13.625.

Example: Convert the Binary number $(10110.001)_2$ to Decimal equivalent number

Solution: $(10110.001)_2$ to Decimal

$(10110.001)_2=(1\times2^4)+(0\times2^3)+(1\times2^2)+(1\times2^1)+(0\times2^0)+(0\times2^{-1})+(0\times2^{-2})+(1\times2^{-3})$

$(10110.001)_2=(1\times16)+(0\times8)+(1\times4)+(1\times2)+(0\times1)+(0\times1/2)+(0\times1/4)+(1\times1/8)$

$(10110.001)_2=16+0+4+2+0+0+0+0.125$

$(10110.001)_2=(22.125)_{10}$

Practice Example: Convert $(1101001)_2$ into an equivalent decimal number.
Ans:$(1101001)_2 = (105)_{10}$
Convert $(11110111)_2$ into base-10 number system.
Ans: ($11110111)_2 = (247)_{10}$
Convert binary number 10111 into an equivalent decimal number.
Convert $(111)_2$ in decimal number.
What is $(101010)_2$ in decimal number?

Ans: $(10111)_2 = 23$, $(111)_2 = 7$, $(101010)_2 = 42$
Practice Examples: Convert to decimal number $(110.01)_2 = ?$, $(111.1)_2 = ?$, $(10.111)_2 = ?$, $(0.1010)_2 = ?$

Ans: $(110.01)_2 = 6.25$, $(111.1)_2 = 7.5$, $(10.111)_2 = 2.875$, $(0.1010)_2 = 0.625$

Decimal to Binary Conversion

We use Repeated Division-by-2 Method (double dabble method). Write down the decimal number and to continually divide-by-2 (two) to give a result and a remainder of either a “1” or a “0” until the final result equals zero.

Step – 1 Divide the decimal number which is to be converted by two which is the base of the binary number.

Step – 2 The remainder which is obtained from step 1 is the least significant bit of the new binary number.

Step – 3 Divide the quotient which is obtained from the step 2 and the remainder obtained from this is the second least significant bit of the binary number.

Step – 4 Repeat the process until the quotient remains zero.

Step – 5 The last remainder obtained from the division is the most significant bit of the binary number.

Hence arrange the number from most significant bit to the least significant bit (i.e., from bottom to top).

Example: Convert 25 in decimal to binary
Solution:

2	25	1	←First remainder
2	12	0	←Second remainder
2	6	0	← Third remainder
2	3	1	← Fourth remainder
2	1	1	← Fifth remainder
	0		

Read up↑
Binary Number = 11001

Example: Convert Decimal 17 to Binary number system
Solution:

2	17	1	←First remainder
2	8	0	←Second remainder
2	4	0	← Third remainder
2	2	0	← Fourth remainder
	1	1	← Fifth remainder

Read up↑
Binary Number = 10001

Conversion of Decimal to Binary for Fraction Number

For fractional decimal numbers, multiply it by 2 and record the carry in the integral position. The carries when read down produces the equivalent binary fraction.
Consider the fractional binary number 0.35

Multiplication	Result	Carry
0.35 × 2	0.70	0
0.70 × 2	0.40	1
0.40 × 2	0.80	0
0.80 × 2	0.60	1
0.60 × 2	0.20	1

Read down
Thus, the fractional binary number is 0.01011.
The process of multiplication by 2 will continue till the desired accuracy is achieved.

Example: Convert decimal fractional number 0.8125 into binary number.
Solution:

Multiplication	Resultant integer part (R)
0.81252 × 2 = 1.625	1
0.6252 × 2 = 1.25	1
0.252 × 2 = 0.50	0
0.52 × 2 = 1.0	1
0 × 2 = 0	0

Ans: The equivalent binary fractional number of decimal fractional 0.8125 integer part is 0.11010

Conversion of Decimal to Binary for Mixed Number

To convert a decimal mixed number into the binary number, the same approach is used, as was done in integer and fractional parts of the number.

Example: Convert $(152.25)_{10}$ to Binary Number
Solution: Step 1: Divide the number 152 and its successive quotients with base 2.

Operation	Quotient	Remainder
152 ÷ 2	76	0 (LSB)
76 ÷ 2	38	0
38 ÷ 2	19	0
19 ÷ 2	9	1
9 ÷ 2	4	1
4 ÷ 2	2	0
2 ÷ 2	1	0
1 ÷ 2	0	1 (MSB)

$(152)_{10} = (10011000)_2$

Step 2: Now, Perform the multiplication of 0.25 and successive fraction with base 2.

Operation	Result	Carry
0.25 × 2	0.50	0
0.50 × 2	0	1

(0.25)10 = (.01)2

Ans: $(152.25)_{10} = (1\ 0011000.\ 01)_2$

Example: Convert the decimal number 13.40 into binary equivalent.

The binary equivalent of 13 is 1101
The binary equivalent of 0.40 is. 011001.
Ans: Therefore 1101.011001 is the required binary number.

Practice Problems

1. 25.5 → $(11001.1)_2$, 2. 18.75 → $(10010.11)_2$, 3. 36.25 → $(100100.01)_2$, 4. 52.625 → $(110100.101)_2$, 5. 41.875 → $(101001.111)_2$, 6. 60.5 → $(111100.1)_2$, 7. 33.25 → $(100001.01)_2$, 8. 44.125 → $(101100.001)_2$

Decimal to Octal Conversion

Step 1: Divide the integer part by 8 and record remainders (bottom to top).

Step 2: Multiply the fractional part by 8 and record the integer part of each result (top to bottom).

Operation	Quotient	Remainder
152 ÷ 8	19	0
19 ÷ 8	2	3
2 ÷ 8	0	2

Example: Convert $(152.25)_{10}$ to Octal

Solution:
Step 1: Divide the number 152 and its
successive quotients with base 8.

$(152)_{10}=(230)_8$

Step 2: Now perform the multiplication of

Operation	Result	Carry
0.25 × 8	0	2

0.25 and successive fraction with base 8.

$(0.25)_{10}=(2)_8$

So, the octal number of the decimal number 152.25 is 230.2

Practice Examples:

1. 25.375 → $(31.3)_8$
   - 25 (decimal) = 31 (octal)
   - 0.375 × 8 = 3.0 → 0.375 = .3 (octal)
2. 18.625 → $(22.5)_8$
   - 18 → 22
   - 0.625 × 8 = 5.0 → .5
3. 45.75 → $(55.6)_8$
   - 45 → 55
   - 0.75 × 8 = 6.0 → .6
4. 12.5 → $(14.4)_8$
   - 12 → 14
   - 0.5 × 8 = 4.0 → .4
5. 7.125 → $(7.1)_8$
   - 7 → 7
   - 0.125 × 8 = 1.0 → .1
6. 63.875 → $(77.7)_8$
   - 63 → 77
   - 0.875 × 8 = 7.0 → .7

Octal to Decimal Conversion

The process of converting octal to decimal is the same as binary to decimal. The process starts from multiplying the digits of octal numbers with its corresponding positional weights. And lastly, we add all those products.

Example: Convert $(152.25)_8$ to Decimal

Solution: We multiply each digit of 152.25 with its respective positional weight, and last we add the products of all the bits with its weight.

$(152.25)_8 = (1\times8^2)+(5\times8^1)+(2\times8^0)+(2\times8^{-1})+(5\times8^{-2})$

$(152.25)_8=64+40+2+(2\times1/8)+(5\times1/64)$

$(152.25)_8=64+40+2+0.25+0.078125$

$(152.25)_8=106.328125$

Ans: So, the decimal number of the octal number 152.25 is 106.328125

Practice Examples:

$(127.4)_8 = 87.5_{10}$
- $1\times64 + 2\times8 + 7 = 64 + 16 + 7 = 87$
- $4\times8^{-1} = 0.5$ → 87.5

2. $(245.6)_8 = 165.75_{10}$
   - $2\times64 + 4\times8 + 5 = 128 + 32 + 5 = 165$
   - $6\times8^{-1} = 0.75$ → 165.75
3. $(314.2)_8 = 204.25_{10}$

- $3 \times 64 + 1 \times 8 + 4 = 192 + 8 + 4 = 204$
- $2 \times 8^{-1} = 0.25 \rightarrow 204.25$

4. $(512.5)_8 = 330.625_{10}$
   - $5 \times 64 + 1 \times 8 + 2 = 320 + 8 + 2 = 330$
   - $5 \times 8^{-1} = 0.625 \rightarrow 330.625$
5. $(764.3)_8 = 500.375_{10}$
   - $7 \times 64 + 6 \times 8 + 4 = 448 + 48 + 4 = 500$
   - $3 \times 8^{-1} = 0.375 \rightarrow 500.375$

Decimal to Hexadecimal Conversion

Step 1: Convert the Integer Part:

- Repeatedly divide the integer part by 16.
- Record the remainders after each division.
- The hexadecimal number is formed by writing the remainders in reverse order (from last to first).
- Use hexadecimal digits:
  10 = A, 11 = B, 12 = C, 13 = D, 14 = E, 15 = F.

Step 2: Convert the Fractional Part (if any)

- Repeatedly multiply the fractional part by 16.
- Record the integer part of each product.
- Use the new fractional part for the next multiplication.
- Stop when the fractional part becomes 0 or you reach the desired number of digits.

Example: Convert $(152.25)_{10}$ to Hexadecimal

Step 1: Divide the number 152 and its successive quotients with base 16.

Operation	Quotient	Remainder
152 ÷ 16	9	8
9 ÷ 16	0	9

$(152)_{10} = (98)_{16}$

Step 2: Now perform the multiplication of 0.25 and successive fraction with base 16.

Operation	Quotient	Remainder
152 ÷ 16	9	8
9 ÷ 16	0	9

$(0.25)_{10} = (4)_{16}$

Ans: So, the hexadecimal number of the decimal number 152.25 is 98.4.

Practice Examples:

1023.5 → $(3FF.8)_{16}$

- 1023 = 3FF
- 0.5 × 16 = 8.0 → .8

2. 4095.75 → $(FFF.C)_{16}$
   - 4095 = FFF
   - 0.75 × 16 = 12 → C
3. 2048.625 → $(800.A)_{16}$
   - 2048 = 800
   - 0.625 × 16 = 10 → A
4. 8192.25 → $(2000.4)_{16}$
   - 8192 = 2000
   - 0.25 × 16 = 4 → .4
5. 5000.125 → $(1388.2)_{16}$
   - 5000 = 1388
   - 0.125 × 16 = 2 → .2
6. 12345.375 → $(3039.6)_{16}$
   - 12345 = 3039
   - 0.375 × 16 = 6 → .6
7. 10000.5 → $(2710.8)_{16}$
   - 10000 = 2710
   - 0.5 × 16 = 8 → .8
8. 16384.875 → $(4000.E)_{16}$
   - 16384 = 4000
   - 0.875 × 16 = 14 → E

Hexadecimal to Decimal Conversion Method

The base of the hexadecimal number system is 16.

The weights corresponding to various positions of the digits will be as shown below.

$$16^2 \quad 16^1 \quad 16^0 \quad . \quad 16^{-1} \quad 16^{-2} \quad 16^{-3}$$

↑

Hexadecimal Point

Example: Convert the hexadecimal number E8F6.27 into its equivalent decimal number.

Solution:

$$E8F6.27 = E(16^3) + 8(16^2) + F(16^1) + 6(16^0) + 2(16^{-1}) + 7(16^{-2})$$

$$E8F6.27 = 14(4096) + 8(256) + 15(16) + 6(1) + 2 \times \frac{1}{16} + 7 \times \frac{1}{16^2}$$

$$E8F6.27 = 59638.1523437$$

Example: Convert hexadecimal number 1F.01B into decimal number.

Solution: Since value of Symbols: B and F are 11 and 15 respectively.

Therefore, equivalent decimal number is:

$(1F.01B)_{16}$

$= (1 \times 16^1 + 15 \times 16^0 + 0 \times 16^{-1} + 1 \times 16^{-2} + 11 \times 16^{-3})_{10}$

$= (31.0065918)_{10}$ which is answer.

Practice Examples:

1. Convert $(1A)_{16}$ to decimal
Answer: $1\times16 + 10 = 26$

2. Convert $(2F)_{16}$ to decimal
Answer: $2\times16 + 15 = 47$

3. Convert $(FF.A)_{16}$ to decimal
Answer: $(15\times16 + 15) + (10\times16^{-1}) = 255 + 0.625 = 255.625$

4. Convert $(64)_{16}$ to decimal
Answer: $6\times16 + 4 = 100$

5. Convert $(3B.4)_{16}$ to decimal
Answer: $(3\times16 + 11) + (4\times16^{-1}) = 48 + 11 + 0.25 = 59.25$

6. Convert $(100.4)_{16}$ to decimal
Answer: $1\times256 + 0 + 0 + (4\times16^{-1}) = 256 + 0.25 = 256.25$

7. Convert $(A5.2)_{16}$ to decimal
Answer: $(10\times16 + 5) + (2\times16^{-1}) = 160 + 5 + 0.125 = 165.125$

8. Convert $(12F)_{16}$ to decimal
Answer: $(1\times256 + 2\times16 + 15) = 256 + 32 + 15 = 303$

Decimal to Hexadecimal Number Conversion

Step 1: Convert Integer Part

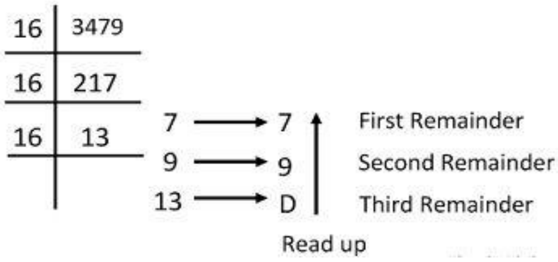
- Divide the decimal integer by 16.
- Record the remainder.
- Repeat the division with the quotient until it becomes 0.
- Write remainders in reverse order (bottom to top).
- Use A–F for remainders 10–15.

Step 2: Convert Fractional Part (if any)

- Multiply the fractional part by 16.
- Write down the integer part of the result.
- Repeat with the new fractional part.
- Stop when the fractional part becomes 0 or you've reached required precision.

Example: Convert the decimal number 3749 into its hexadecimal equivalent number.

Solution:



The third reminder 13 is equivalent to D in a hexadecimal number system.
Thus the equivalent hexadecimal number D97

Practice Examples:

$(3749)_{10} = (EA5)_{16}$

$(2543)_{10} = (9EF)_{16}$

$(43981)_{10} = (ABCD)_{16}$

$(1600)_{10} = (640)_{16}$

$(9999)_{10} = (270F)_{16}$

$(12345)_{10} = (3039)_{16}$

$(2048)_{10} = (800)_{16}$

$(50000)_{10} = (C350)_{16}$

Hexadecimal to Binary & Binary to Hexadecimal Conversion Methods

The base of the hexadecimal number is 16 and it consist the numbers from 0 -15. It uses the number system from 0-9 and alphabets from A- F. The hexadecimal number system is also called hex. It is one of the convenient ways of describing the binary digit number. The base of the binary number system is two because it is represented by two digits, i.e., 0 and 1. It is difficult to represent a large number in the form of binary digits and hence the hexadecimal system is used in digital electronics. The hexadecimal number systems are easily converted into a binary system by using the method explained below:

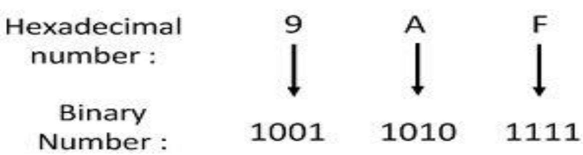
Number	0	1	2	3	4	5	6	7
Binary	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal	0	1	2	3	4	5	6	7

Number	8	9	10	11	12	13	14	15
Binary	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal	8	9	A	B	C	D	E	F

Hexadecimal to Binary Conversion Method

To convert a hexadecimal number to a binary number, convert each hexadecimal digit to its four-digit equivalent.

For example, consider the hexadecimal number 9AF which is converted into a binary digit.



Therefore, the equivalent binary number is 1001 1010 1111.

Example: Convert $(152A.25)_{16}$ to Binary

Solution: We write the four-bit binary digit for 1, 5, A, 2, and 5.

$(152A.25)_{16}$ = $(0001\ 0101\ 0010\ 1010.\ 0010\ 0101)_2$

So, the binary number of the hexadecimal number 152.25 is $(1010100101010.00100101)_2$

Practice Examples

$(2F3B.9A)_{16} = (0010111100111011.10011010)_2$

$(1A3C.4F)_{16} = (0001101000111100.01001111)_2$

$(FFAC.8D)_{16} = (1111111110101100.10001101)_2$

$(3B7E.D2)_{16} = (0011101101111110.11010010)_2$

$(C9AF.35)_{16} = (1100100110101111.00110101)_2$

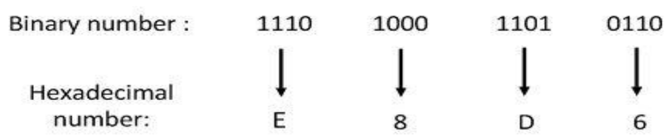
$(7E3D.6B)_{16} = (0111111000111101.01101011)_2$

$(1F2A.9C)_{16} = (0001111100101010.10011100)_2$

$(B45D.E3)_{16} = (1011010001011101.11100011)_2$

Binary to Hexadecimal Conversion Methods

To convert the given binary number into its equivalent hexadecimal number, rewrite the binary number of the sets of four digits and then place the hexadecimal digit in front of each four digit set of a binary number as explained by the following number.



Thus, the equivalent hexadecimal number is E8D6.

Example: Covert $(11110101011.0011)_2$ into Hexadecimal

1. Firstly, we make pairs of four bits on both sides of the binary point.

111 1010 1011.0011

On the left side of the binary point, the first pair has three bits. To make it a complete pair of four bits, add one zero on the extreme side.

0111 1010 1011.0011

2. Then, we write the hexadecimal digits, which correspond to each pair.

$(011110101011.0011)_2=(7AB.3)_{16}$

Practice Examples

$(152A.25)_{16} = (0001\ 0101\ 0010\ 1010.0010\ 0101)_2$

$(3F.9)_{16} = (00111111.1001)_2$

$(AB.CD)_{16} = (10101011.11001101)_2$

$(10E.75)_{16} = (000100001110.01110101)_2$

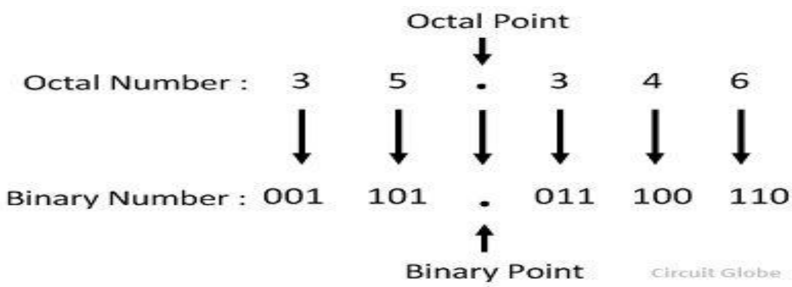
$(7B2.AC)_{16} = (011110110010.10101100)_2$

Octal to Binary & Binary to Octal Conversion Methods

The base of the octal number system is 8 because it consists the digit from 0 – 7. The octal number system is converting into binary number system by grouping the binary digit in the group of three from the right of the binary number. On the other hand, the base of the binary digit is 2 and it consists only two digits 0 and 1.

Octal to Binary Conversion

The base of the octal number system (8) is the third power of base of a binary system (2)
The inter-conversion of octal and a binary number is very simple and direct.
Convert the octal number 35.346 to its equivalent binary number



Thus, the required binary number is 011 101. 011 100 110.

Example: Convert the $(152.25)_8$ to Binary.

Solution: We write the three-bit binary digit for 1, 5, 2, and 5.

$(152.25)_8 = (001101010.010101)_2$

So, the binary number of the octal number 152.25 is $(001101010.010101)_2$

Practice Examples:

$(35.346)_8 = (011101.011100110)_2$

$(152.25)_8 = (001101010.010101)_2$

$(467.3)_8 = (100110111.011)_2$

$(720.65)_8 = (111010000.110101)_2$

$(17.4)_8 = (001111.100)_2$

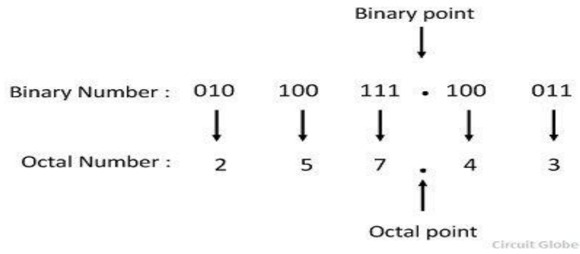
$(523.56)_8 = (101010011.101110)_2$

$(640.12)_8 = (110100000.001010)_2$

$(775.07)_8 = (111111101.000111)_2$

Binary to Octal Conversion

The conversion of binary to octal is a reversal of the above procedure. For example, the binary number 010100111.100011 can be converted into an octal number by first written the bits in the group of three and then awarding the decimal number of to each of the group of three bits.



Thus, the required octal number is 257.43

Example: $(111110101011.0011)_2$

1. Firstly, we make pairs of three bits on both sides of the binary point.

    111 110 101 011. 001 1

On the right side of the binary point, the last pair has only one bit. To make it a complete pair of three bits, we added two zeros on the extreme side.

    111 110 101 011. 001 100

2. Then, write the octal digits, which correspond to each pair.

    $(111110101011.0011)_2 = (7653.14)_8$

Practice Examples:

$(010100111.100011)_2 = (247.43)_8$

$(111110101011.0011)_2 = (7653.14)_8$

$(1010111.1011)_2 = (127.54)_8$

$(11010011.111)_2 = (323.7)_8$

$(1001010.01)_2 = (112.2)_8$

$(101100111.111101)_2 = (547.75)_8$

$(11101.10101)_2 = (161.52)_8$

$(100110.011)_2 = (46.3)_8$

Hexadecimal to Octal Conversion

For converting hexadecimal to octal, there are two steps required to perform, which are as follows:

In the first step, we will find the binary equivalent of the hexadecimal number. Next, we have to make the pairs of three bits on both sides of the binary point. If there will be one or two bits left in a pair of three bits pair, we add the required number of zeros on extreme sides and write the octal digits corresponding to each pair.

Example: Convert $(152A.25)_{16}$ to octal

Step 1: We write the four-bit binary digit for 1, 5, 2, A, and 5.

$(152A.25)_{16}=(0001\ 0101\ 0010\ 1010.0010\ 0101)_2$

So, the binary number of hexadecimal number 152A.25 is $(0011010101010.010101)_2$

Step 2: Then, we make pairs of three bits on both sides of the binary point.

001 010 100 101 010.001 001 010

Then, we write the octal digit, which corresponds to each pair.

$(001010100101010.001001010)_2=(12452.112)_8$

Ans: So, the octal number of the hexadecimal number 152A.25 is 12452.112

Practice Examples:

$(1F3.4A)_{16} = (0763.224)_8$

$(ACD.5)_{16} = (12635.24)_8$

$(7E4.9)_{16} = (07620.44)_8$

$(B2F.3)_{16} = (13137.14)_8$

$(3A7.7D)_{16} = (07231.372)_8$

Octal to Hexadecimal conversion

For converting octal to hexadecimal, there are two steps required to perform, which are as follows:

In the first step, we will find the binary equivalent of number. Next, we have to make the pairs of four bits on both sides of the binary point. If there will be one, two, or three bits left in a pair of four bits pair, we add the required number of zeros on extreme sides and write the hexadecimal digits corresponding to each pair.

Example: $(152.25)_8$ to hexadecimal

Step 1: We write the three-bit binary digit for 1, 5, 2, and 5.

$(152.25)_8$ $= (001101010.010101)_2$

So, the binary number of the octal number 152.25 is **$(001101010.010101)_2$**

Step 2: Now, we make pairs of four bits on both sides of the binary point.

0 0110 1010.0101 01

On the left side of the binary point, the first pair has only one digit, and on the right side, the last pair has only two-digit. To make them complete pairs of four bits, **add zeros** on extreme sides.

0000 0110 1010.0101 0100

Now, we write the hexadecimal digits, which correspond to each pair.

$(0000 \quad 0110 \quad 1010.0101 \quad 0100)_2 = (6A.54)_{16}$

Practice Examples:

$(467.3)_8 = (1B7.6)_{16}$

$(720.65)_8 = (1D0.D4)_{16}$

$(17.4)_8 = (F.8)_{16}$

$(523.56)_8 = (153.B8)_{16}$

$(640.12)_8 = (1A0.28)_{16}$

$(775.07)_8 = (1FD.1C)_{16}$

4.3 Binary Arithmetic: Addition and Subtraction (Using 1's and 2's complement)

The addition and subtraction of the binary number system are similar to that of the decimal number system. The only difference is that the decimal number system consists the digit from 0-9 and their base is 10 whereas the binary number system consists only two digits (0 and 1) which make their operation easier.

Addition in decimal

	1	
	3	4
+	5	7
	9	1

The binary number system uses only two digits 0 and 1 due to which their addition is simple.

There are four basic operations for binary addition.

0 + 0 = 0
0 + 1 =1
1 + 0 =1
1 + 1= 0 carry 1

Question: Addition of 11101 and 11011.

		1	1	1	1		← carry
		1	1	1	0	1	
(+)		1	1	0	1	1	
	1	1	1	0	0	0	

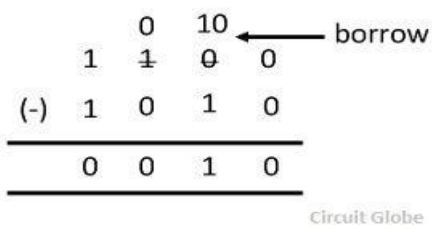
Circuit Globe

Binary Subtraction

The subtraction of the binary digit depends on the four basic operations

0 – 0 = 0
1 – 0 = 1
1 – 1 = 0
10 – 1 = 1 (The fourth operation is logic two minus one is one.)

Question: Subtract 1010 from 1100



The above subtraction is carried out through the following steps.

0 – 0 = 0

For 0 – 1 = 1, taking borrow 1 and then 10 – 1 = 1

For 1 – 0, since 1 has already been given, it becomes 0 – 0 = 0

1 – 1 = 0

Therefore, the result is 0010.

1’s Complement and 2’s Complement of a number

1’s complement of a binary number is another binary number obtained by toggling all bits in it, i.e., transforming the 0 bit to 1 and the 1 bit to 0.

2’s complement of a binary number is 1 added to the 1’s complement of the binary number.

1's complement of "0111" is "1000"

2's complement of "0111" is 1000 + 1 = 1001 "

Question: Find the 1's complement of the number "1100 "?

Find the 2's complement of the number "1100 "?

Solution: 1's complement of "1100" is "0011 "

2's complement of "1100" is "0100 "

Arithmetic operations such as addition and subtraction using 1's complement

Addition using 1's complement

Case 1: Addition of the positive number with a negative number when the positive number has a greater magnitude.

Initially, calculate the 1's complement of the given negative number.

Sum up with the given positive number.

If we get the end-around carry 1, it gets added to the LSB.

Case 1: Example: 1101 and -1001

First, find the 1's complement of the negative number 1001. So, for finding 1's complement, change all 0 to 1 and all 1 to 0.

The 1's complement of the number 1001 is 0110

Now, add both the numbers, i.e., 1101 and 0110;

1101

+ 0110

1 0011

By adding both numbers, we get the end-around carry 1.

We add this end around carry to the LSB of 0011.

```
  0011
+    1
------
  0100
```

Case 2: Adding a positive value with a negative value in case the negative number has a higher magnitude.

Initially, calculate the 1's complement of the negative value.

Sum it with a positive number.

In this case, we did not get the end-around carry.

So, take the 1's complement of the result to get the final result.

Note: The resultant is a negative value.

Example: 1101 and -1110

First find the 1's complement of the negative number 1110.

So, for finding 1's complement, we change all 0 to 1, and all 1 to 0.

1's complement of the number 1110 is 0001.

Now, add both the numbers, i.e., 1101 and 0001;

```
  1101
+ 0001
------
  1110
```

Now, find the 1's complement of the result 1110 that is the final result.

So, the 1's complement of the result 1110 is 0001, and we add a negative sign before the number so that we can identify that it is a negative number.

Answer: - 0001

Case 3: Addition of two negative numbers

In this case, first find the 1's complement of both the negative numbers, and then we add both these complement numbers.

In this case, we always get the end-around carry, which get added to the LSB, and for getting the final result, we take the 1's complement of the result.

Note: The resultant is a negative value.

Example: -1101 and -1110 in five-bit register

Firstly, find the 1's complement of the negative numbers 01101 and 01110.

So, for finding 1's complement, we change all 0 to 1, and all 1 to 0.

1's complement of the number 01110 is 10001, and 01101 is 10010.

Now, we add both the complement numbers, i.e., 10001 and 10010;

$$\begin{array}{r} 10001 \\ +\ \ 10010 \\ \hline 1\ \ 00011 \end{array}$$

By adding both numbers, we get the end-around carry 1.

We *add this end-around carry* 0to the LSB of 00011.

$$\begin{array}{r} 00011 \\ +\ \ \ \ \ \ 1 \\ \hline 00100 \end{array}$$

Now, find the 1's complement of the result 00100 that is the final answer.

So, the 1's complement of the result 00100 is 11011, and add a negative sign before the number so that we can identify that it is a negative number.

Ans: - 11011

Practice examples with answers

Q1. Add 1010 and –0011
Answer = 0111 (positive)
Q2. Add 1110 and –0100
Answer = 1010 (positive)
Q3. Add 1001 and –1110
Answer = –0101 (negative)
Q4. Add 0101 and –1111
Answer = –1010 (negative)
Q5. Add –1001 and –0110 (5-bit register)
Answer = –01111 (negative)
Q6. Add –1010 and –1000 (5-bit register)
Answer = –10010 (negative)
Q7. 1010 + (–0011)
Answer: 0111
Q8. 1110 + (–0100)
Answer: 1010
Q9. 1001 + (–1110)
Answer: –0101
Q10. 0101 + (–1111)
Answer: –1010
Q11. (–1001) + (–0110) [5-bit register]
Answer: –01111
Q12. (–1010) + (–1000) [5-bit register]
Answer: –10010
Q13. 1100 + (–0101)
Answer: 1001
Q14. 1011 + (–1101)
Answer: –0010

Q15. (–0111) + (–0101) [5-bit register]
Answer: –11001
Q16. 1111 + (–1000)
Answer: 0111

Subtraction using 1's complement

These are the following steps to subtract two binary numbers using 1's complement

In the first step, find the 1's complement of the subtrahend.

Next, add the complement number with the minuend.

If got a carry, add the carry to its LSB.

Else take 1's complement of the result which will be negative

Note: *The subtrahend value always gets subtracted from minuend.*

Example: 10101 - 00111

We take *1's complement of subtrahend 00111*, which comes out 11000.

Now, sum them. So,

```
      10101
  +   11000
  ---------
    1 01101
```

In the above result, we get the carry bit 1, so add this to the LSB of a given result, i.e.,

```
Answer: 01101
     +      1
     --------
        01110
```

Example:10101 - 10111

We take 1's complement of subtrahend 10111, which comes out 01000.

Now, add both of the numbers.

```
     10101
  +  01000
  --------
     11101
```

In the above result, we didn't get the carry bit.

So, calculate the 1's complement of the result, i.e., 00010, which is the negative number and the final answer.

Answer: - 00010

Addition and Subtraction using 2's complement

Case 1: Addition of the positive number with a negative number when the positive number has a greater magnitude.

Initially find the 2's complement of the given negative number.

Sum up with the given positive number.

If we get the end-around carry 1 then the number will be a positive number and the carry bit will be discarded and remaining bits are the final result.

Example: 1101 and -1001

First, find the 2's complement of the negative number 1001.

So, for finding 2's complement, change all 0 to 1 and all 1 to 0 or find the 1's complement of the number 1001.

The 1's complement of the number 1001 is 0110, and add 1 to the LSB of the result 0110.

So, the 2's complement of number 1001 is 0110+1=0111

Add both the numbers, i.e., 1101 and 0111;

```
   1101
 + 0111
 ------
 1 0100
```

By adding both numbers, we get the end-around carry 1.

We discard the end-around carry.

So, the addition of both numbers is 0100.

Case 2: Adding of the positive value with a negative value when the negative number has a higher magnitude.

Initially, add a positive value with the 2's complement value of the negative number.

Here, no end-around carry is found.

So, we take the 2's complement of the result to get the final result.

The resultant is a negative value.

Example: 1101 and -1110

First, find the 2's complement of the negative number 1110.

So, for finding 2's complement, add 1 to the LSB of its 1's complement value 0001.
0001+1=0010

Add both the numbers, i.e., 1101 and 0010;

```
   1101
 + 0010
 ------
   1111
```

Find the 2's complement of the result 1111 that is the final result.

So, the 2's complement of the result 1111 is 0001, and add a negative sign before the number so that we can identify that it is a negative number.

Ans: 0001

Case 3: Addition of two negative numbers

In this case, first, find the 2's complement of both the negative numbers

Add both these complement numbers.

In this case, we will always get the end-around carry, which will be added to the LSB,

Forgetting the final result, take the2's complement of the result.

Example: -1101 and -1110 in five-bit register

Firstly, find the 2's complement of the negative numbers 01101 and 01110.

So, for finding 2's complement, we add 1 to the LSB of the 1's complement of these numbers.

2's complement of the number 01110 is 10010, and 01101 is 10011.

We add both the complement numbers, i.e., 10001 and 10010;

```
   10010
+  10011
 1 00101
```

By adding both numbers, we get the end-around carry 1.

This carry is discarded and the final result is the 2.s complement of the result 00101.

So, the 2's complement of the result 00101 is 11011, and we add a negative sign before the number so that we can identify that it is a negative number.

Answer: - 11011

Subtraction using 2's complement

These are the following steps to subtract two binary numbers using 2's complement

In the first step, find the 2's complement of the subtrahend.

Add the complement number with the minuend.

If we get the carry by adding both the numbers, then we discard this carry and the result is positive else take 2's complement of the result which will be negative.

Example: 10101 - 00111

We take 2's complement of subtrahend 00111, which is 11001. Now, sum them. So,

```
   10101
+  11001
 1 01110
```

In the above result, we get the carry bit 1.

So, we discard this carry bit and remaining is the final result and a positive number.

Answer: 01110

Example: 10101 - 10111

We take 2's complement of subtrahend 10111, which comes out 01001.

Now, we add both of the numbers. So,

```
   10101
+  01001
   11110
```

In the above result, we didn't get the carry bit.

So, calculate the 2's complement of the result, i.e., 00010. It is the negative number and the final answer.

Answer: - 00010

Addition Using 2's Complement

Case 1: Positive + Negative (positive > negative)

Q1. 1011 + (–0010)

- 2's complement of 0010 → 1110
- Add: 1011 + 1110 = 1 1001 → discard carry → **1001**
  Answer = **1001** (positive)

Q2. 1101 + (–0100)

- 2's complement of 0100 → 1100
- Add: 1101 + 1100 = 1 1001 → discard carry → **1001**
  Answer = **1001** (positive)

Case 2: Positive + Negative (negative > positive)

Q3. 0101 + (–1110)

- 2's complement of 1110 → 0010
- Add: 0101 + 0010 = 0111
- No carry → 2's complement of 0111 = 1001 → negative
  Answer = **–1001**

Q4. 1010 + (–1101)

- 2's complement of 1101 → 0011
- Add: 1010 + 0011 = 1101
- No carry → 2's complement of 1101 = 0011 → negative
  Answer = **–0011**

Case 3: Negative + Negative

Q5. (–1011) + (–0110) [5-bit register]

- 2's complement of 01011 → 10101
- 2's complement of 00110 → 11010
- Add: 10101 + 11010 = 101111
- Discard carry → 01111
- Take 2's complement → 10001 → negative
  Answer = **–10001**

Q6. (–1100) + (–1010) [5-bit register]

- 2's complement of 01100 → 10100
- 2's complement of 01010 → 10110
- Add: 10100 + 10110 = 101010

- Discard carry → 01010
- Take 2's complement → 10110 → negative
  Answer = **–10110**

Subtraction Using 2's Complement

Q7. 10101 – 00111

- 2's complement of 00111 → 11001
- Add: 10101 + 11001 = 1 01110 → discard carry → **01110**
  Answer = **01110** (positive)

Q8. 10101 – 10111

- 2's complement of 10111 → 01001
- Add: 10101 + 01001 = 11110
- No carry → take 2's complement of 11110 = 00010 → negative
  Answer = **–00010**

Q9. 11100 – 01010

- 2's complement of 01010 → 10110
- Add: 11100 + 10110 = 1 10010 → discard carry → **10010**
  Answer = **10010** (positive)

Q10. 10011 – 11001

- 2's complement of 11001 → 010111 (in 6-bit register)
- Add: 010011 + 010111 = 101010
- No carry → 2's complement of 101010 = 010110 → negative
  Answer = **–010110**

Introduction to Binary code, BCD Code, Gray Code and its code conversions.

Binary code

A binary code is the value of a data-encoding convention represented in a binary notation that usually is a sequence of 0s and 1s; sometimes called a *bit string*.

A binary code is the value of a data-encoding convention represented in a binary notation that usually is a sequence of 0s and 1s; sometimes called a *bit string*.

Binary code is a system of representing data or instructions using only two symbols: **0 and 1**. It is the foundation of all digital systems, including computers, communication devices, and embedded systems.

- **Base**: Binary uses **base-2** number system (only two digits: 0 and 1).
- **Bit**: Each digit (0 or 1) is called a **bit** (binary digit).
- **Byte**: A group of 8 bits forms a **byte**, which can represent 256 (2^8) different values.
- **Working Principle**: Computers and digital circuits operate on binary because electronic components (like transistors) have two stable states: **ON (1)** and **OFF (0)**.
- **Applications**:
  - Data representation (numbers, text, images, audio, video).
  - Instruction encoding in machine language.

- Error detection and correction codes.
- Communication protocols and cryptography.

Types of Binary Codes

1. **Weighted Codes** – e.g., Binary Coded Decimal (BCD).
2. **Non-Weighted Codes** – e.g., Gray Code.
3. **Alphanumeric Codes** – e.g., ASCII, Unicode.
4. **Error Detecting/Correcting Codes** – e.g., Parity Code, Hamming Code.

BCD Code (8421 Code)

In BCD 8421 code, each decimal digit is represented using a 4-bit binary number. The 4-bit binary numbers have their weights attached as 8, 4, 2, 1 from MSB to LSB side. Since the weights are attached to it comes in the category of weighted codes and is also sequential.

The BCD rendition of the base-10 number 1895 is 0001 1000 1001 0101

The binary equivalents of 1, 8, 9, and 5, always in a four-digit format, go from left to right.

The BCD representation of a number is not the same, in general, as its simple binary representation. In binary form, for example, the decimal quantity 1895 appears as 11101100111

Disadvantage of BCD Code

BCD codes are more inefficient than usual binary codes. Usually, in binary numbers, we represent $\mathbf{(13)_{10}}$ = $(1101)_2$ i.e., we require 4-bits but in BCD notation $\mathbf{(13)_{10}}$ is represented as **(0001 0011)**.

Here, we require 8-bits to represent the same **13**.

Another disadvantage is that arithmetic operations become more complex as compared to the usual binary numbers because, in BCD numbers, we have **6 illegal** states as **1010, 1011, 1100, 1101, 1110 and 1111** which are not part of **8421 BCD system**.

Example: Represent $(28)_{10}$ and $(53)_{10}$ in 8421 BCD notation

Solution:

$\mathbf{(28)_{10}}$ in BCD notation can be represented as **(0010 1000)**.

Similarly, $\mathbf{(53)_{10}}$ in BCD notation can be represented as **(0101 0011)**.

The BCD system offers relative ease of conversion between machine-readable and human-readable numerals.

As compared to the simple binary system, however, BCD increases the circuit complexity.

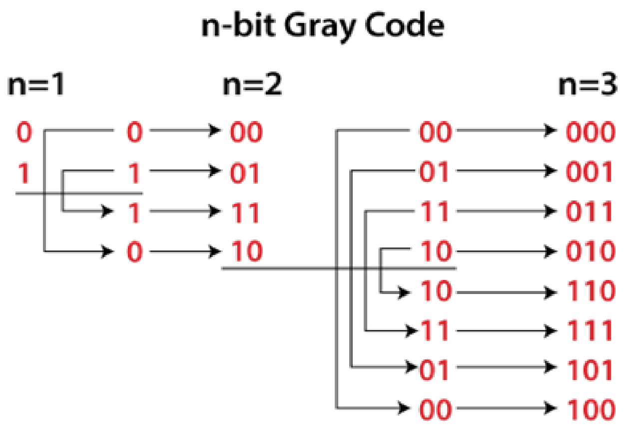
The BCD system is not as widely used today as it was a few decades ago, although some systems still employ BCD in financial applications.

Gray code

The Gray Code is a sequence of binary number systems, which is also known as reflected binary code. The reason for calling this code as reflected binary code is the first N/2 values compared with those of the last N/2 values in reverse order. In this code, two consecutive values are differed by one bit of binary digits. Gray codes are used in the general sequence of hardware-generated binary numbers. These numbers cause ambiguities or errors when the transition from one number to its successive is done. This code simply solves this problem by changing only one bit when the transition is between numbers is done. The Gray code is a very light weighted code because it doesn't depend on the value of the digit specified by the position. This code is also called a cyclic variable code as the transition of one value to its successive value carries a change of one bit only. The prefix and reflect method are recursively used to generate the gray code of a number. For generating gray code:

We find the number of bits required to represent a number. Next, we find the code for 0, i.e., 0000, which is the same as binary.

Now, we take the previous code, i.e., 0000, and change the most significant bit of it. We perform this process reclusively until all the codes are not uniquely identified. If by changing the most significant bit, we find the same code obtained previously, then the second most significant bit will be changed, and so on.



Self-Learning Topics: Practice and understand the process of converting Binary numbers to Gray code through simple examples.

Decimal Number	Binary Number	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Easy Way to Convert Binary to Gray Code

1. Write the binary number.
2. First Gray digit = First Binary digit.
3. For the rest:
    - Compare each binary digit with the one just before it.
    - If they are the **same**, write **0**.
    - If they are **different**, write **1**.

Example 1: Convert Binary 1011

- Binary: 1 0 1 1
- Step 1: First digit is same → Gray = **1**
- Step 2: Compare 1 and 0 → different → **1**
- Step 3: Compare 0 and 1 → different → **1**
- Step 4: Compare 1 and 1 → same → **0**

Gray Code = 1110

Example 2: Convert Binary 1101

- Binary: 1 1 0 1
- First digit = **1**
- Compare 1 and 1 → same → **0**
- Compare 1 and 0 → different → **1**
- Compare 0 and 1 → different → **1**

Gray Code = 1011

Example 3: Convert Binary 0100

- Binary: 0 1 0 0
- First digit = **0**
- Compare 0 and 1 → different → **1**
- Compare 1 and 0 → different → **1**
- Compare 0 and 0 → same → **0**

Gray Code = 0110

Practice problems

Convert the following **Binary numbers** into **Gray code**:

1. 0101 **2.** 0110 **3.** 0111 **4.** 1000 **5.** 1001 **6.** 1010 **7.** 1100 **8.** 1101 **9**.1110 **10.** 1111

Answers (for self-checking)

0101 → 0111, 0110 → 0101, 0111 → 0100, 1000 → 1100, 1001 → 1101, 1010 → 1111, 1100 → 1000, 1101 → 1001, 1110 → 1011, 1111 → 1010