Al-Rasheed University College

Dept. of Computer Eng. Techniques

Digital Logic – First Year

References

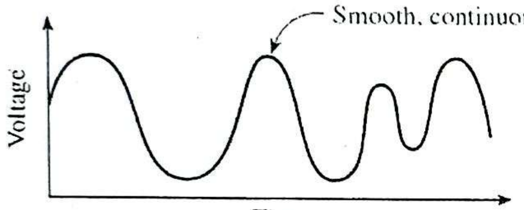
1. Digital Design — By: Morris Mano
2. Digital Fundamentals — By: F. Floyd

Sheet No. 1

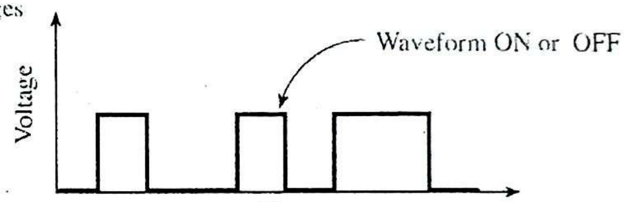
1. Introduction to digital electronics

Digital Electronics is the sub-branch of electronics which deals with digital signals for processing and controlling various systems and sub-systems. In various applications like sensors and actuators, usage of digital electronics is increasing extensively.

Digital electronics is entirely the field in which digital signals is used. Digital signals are discretization of analog signals. A signal carries information. In digital signals the values in a particular band is same i.e. constant. Digital signals form the basis of digital circuit and digital electronics.



(a)



(b)

Digital signals can be represented with two numbers or states, in most cases, the number of these states is two, and they are represented by two voltage bands: one near a reference value (typically termed as "ground" or zero volts), and the other a value near the supply voltage. These correspond to the "false" ("0") and "true" ("1") values of the Boolean domain respectively, named after its inventor, George Boole, yielding binary code.

Digital techniques are useful because it is easier to get an electronic device to switch into one of a number of known states than to accurately reproduce a continuous range of values. Digital electronic circuits are usually made from large assemblies of logic gates, simple electronic representations of Boolean logic functions. A digital circuit is typically constructed from small electronic circuits called logic gates that can be used to create combinational logic. Each logic gate is designed to perform a function of Boolean logic when acting on logic signals. A logic gate is generally created from one or more electrically controlled switches, usually transistors but thermionic valves have seen historic use. The output of a logic gate can, in turn, control or feed into more logic gates. Integrated circuits consist of multiple transistors on one silicon chip, and are the least expensive way to make large number of interconnected logic gates. Integrated circuits are usually designed by engineers using electronic design automation software (see below for more information) to perform some type of function.

الأنظمة العددية

2. Numbering systems

2.1. Decimal Numbering System (Base 10)

نظام الأعداد العشرية

In the decimal numbering system, each position contains 10 different possible digits. These digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Each position in a multidigit number will have a weighting factor based on a power of 10.

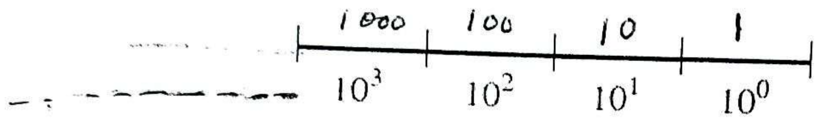


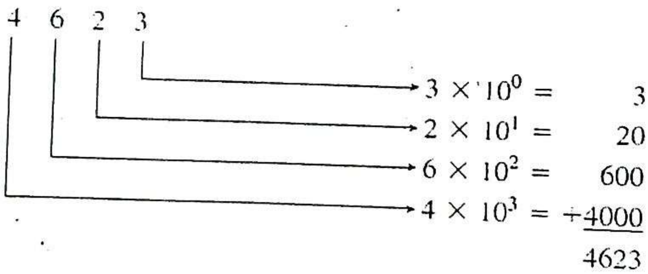
Figure 1 Weighting factors of four-digit decimal number.

The least significant position has a weighting factor of 10^0

The most significant position (leftmost) has a weighting factor of 10^3:

Example: To evaluate the decimal number 4623, the digit in each position is multiplied by the appropriate weighting factor:

Answer:



For fractional numbers,the weights,are negative powers of ten that decrease from left to right beginning with 10^{-1}

$\frac{1}{10}$ $\frac{1}{100}$ $\frac{1}{1000}$

......................10^2 10^1 10^0 • 10^{-1} 10^{-2} 10^{-3}..............

العدد الصحيح

Decimal point

Example: Express the decimal number 568.23

Answer:

$568.23=(5x10^2)+6(10^1)+8(10^0)+2(10^{-1})+3(10^{-2})$

$=500+60+8+0.2+0.03$

2.2. Binary Numbering System (Base 2)

Digital electronics use the **binary** numbering system because it uses only the digits 0 and 1, which can be represented simply in a digital system by two distinct voltage levels, such as +5 V = 1 and 0 V = 0.

In Binary numbering system, the counting will be as follows:

0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, ……………

2.2.1. Binary to Decimal conversion

The weighting factors for binary positions are the powers of 2 shown below:-

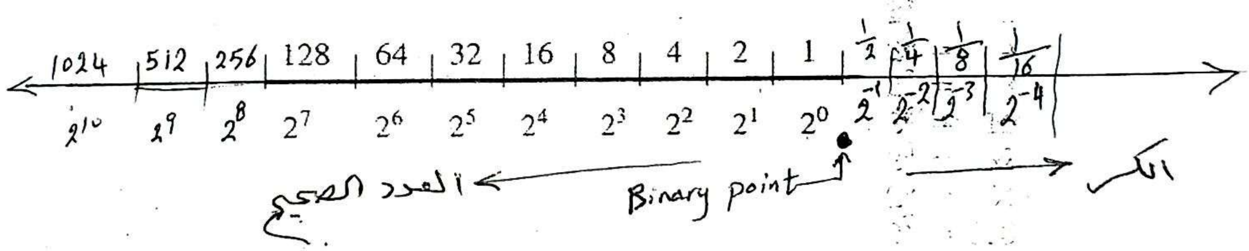
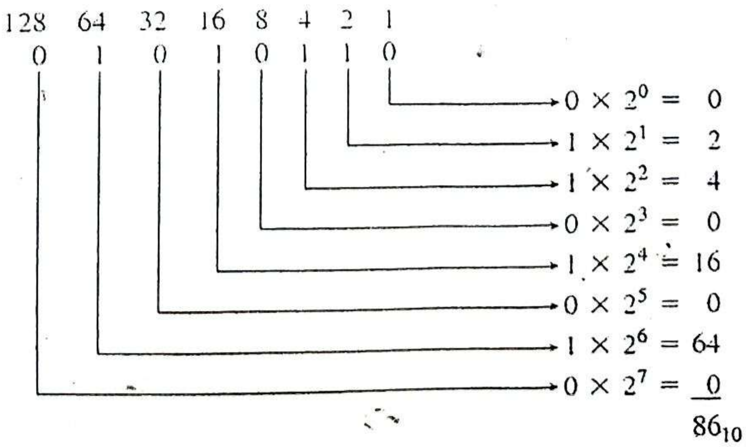


Figure 2 Weighting factors of digit binary number.

Example: Convert the binary number **(01010110)₂** to decimal?

Answer:



$(01010110)_2 = (86)_{10}$

Example: Convert the fractional binary number **(1011.1010)₂** to decimal?

Answer:

1 0 1 1 . 1 0 1 0

$$1 \times 2^{-3} = 0.125$$
$$1 \times 2^{-1} = 0.500$$
$$1 \times 2^{0} = 1$$
$$1 \times 2^{1} = 2$$
$$1 \times 2^{3} = 8$$
$$11.625_{10}$$

$\therefore$ **$(1011.1010)_2 = (11.625)_{10}$**

2.2.2. Decimal to Binary conversion

The binary equivalent can be found by successively dividing the integer part of the number by 2 and recording the remainders until the quotient becomes '0'. The remainders written in reverse order constitute the binary equivalent.

Example: Convert **$(152)_{10}$** to binary number?

Answer:

1. First Method

$152 \div 2 = 76$	remainder 0	(LSB) ↑ (أوطأ مرتبة)
$76 \div 2 = 38$	remainder 0	
$38 \div 2 = 19$	remainder 0	
$19 \div 2 = 9$	remainder 1	
$9 \div 2 = 4$	remainder 1	
$4 \div 2 = 2$	remainder 0	
$2 \div 2 = 1$	remainder 0	
$1 \div 2 = 0$	remainder 1	(MSB) (أعلى مرتبة)

$\therefore$ **$(152)_{10} = (10011000)_2$**

2. Second Method

128	64	32	16	8	4	2	1
1	0	0	1	1	0	0	0

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

Example: Convert $(87)_{10}$ to binary number?

Answer:

2	87	1
2	43	1
2	21	1
2	10	0
2	5	1
2	2	0
2	1	1

↑

$(87)_{10} = (1010111)_2$

Example: Convert $(20.625)_{10}$ to binary number?

Answer:

العدد الصحيح

2	20	0
2	10	0
2	5	1
2	2	0
2	1	1

↑

الكسر

$0.625 \times 2 = 1.25$ = .25	1
$0.25 \times 2 = 0.5$	0
$0.5 \times 2 = 1.0$ = .0	1

↓

$(20.625)_{10} = (10100.101)_2$

2.3. Octal Numbering System (Base 8)

The **octal** numbering system is a method of grouping **binary numbers in groups of three**. The eight allowable digits are 0, 1, 2, 3, 4, 5, 6, and 7. **The octal numbering system is us**

To count above 7, begin another column and start over:

10, 11, 12, 13, 14, 15, 16, and 17.

20, 21, 22, 23, 24, 25, 26, and 27.

Decimal	Binary	Octal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	10
9	1001	11
10	1010	12
⋮	⋮	⋮
15	1111	17

A- Octal – to – Decimal conversion:

Weight … 8^3 | 8^2 | 8^1 | 8^0 . 8^{-1} | 8^{-2} | 8^{-3} | …

Example: Convert $(137.21)_8$ to decimal number?

Answer:

1 3 7 . 2 1

$1 \times 8^{-2} = 0.01562$

$2 \times 8^{-1} = 0.25$

$7 \times 8^{0} = 7$

$3 \times 8^{1} = 24$

$1 \times 8^{2} = 64$

95.26562

$(137.21)_8 = (95.26562)_{10}$

B- Decimal – to – Octal Conversion:

Example: Convert $(73.75)_{10}$ to octal number?

Answer:

8	73	1	↑
8	9	1	
8	1	1	

$0.75 \times 8 = 6 \simeq 0$	6	↓

$(73.75)_{10} = (111.6)_8$

2.3.1. Binary–Octal and Octal–Binary Conversions

An octal number can be converted into its binary equivalent by replacing each octal digit with its three-bit binary equivalent. We take the three-bit equivalent because the base of the octal number system is 8 and it is the third power of the base of the binary number system, i.e. 2. All we have then to remember is the three-bit binary equivalents of the basic digits of the octal number system. A binary number can be converted into an equivalent octal number by splitting the integer and fractional parts into groups of three bits, starting from the binary point on both sides. The 0s can be added to complete the outside groups if needed.

Example: Convert $(624)_8$ to binary number?

Answer:

$$\overbrace{1\ 1\ 0}^{6}\ \ \overbrace{0\ 1\ 0}^{2}\ \ \overbrace{1\ 0\ 0}^{4} = 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0_2$$

$(624)_8 = (110010100)_2$

Example: Convert $(10111001)_2$ to octal number?

Answer:

$$\underbrace{1\ 0}\ \ \underbrace{1\ 1\ 1}\ \ \underbrace{0\ 0\ 1}$$

add a leading zero → $\underbrace{0\ 1\ 0}_{2}$ $\quad 7 \quad 1 \quad = 271_8$

$(10111001)_2 = (271)_8$

Example 1.15. Convert 47.321_8 into an equivalent binary number.

Solution. The octal number given is 4 7 . 3 2 1

3-bit binary equivalent 100 111.011 010 001

Hence the binary number is $(100111.011010001)_2$.

Example 1.13. Convert 1101.0111_2 into an equivalent octal number.

Solution. The binary number given is 1101.0111

Grouping 3 bits 001 101.011 100

Octal equivalent: 1 5 3 4

Hence the octal number is $(15.34)_8$.

2.4. Hexadecimal Numbering System (Base 16)

The hexadecimal numbering system, like the octal system, is a method of grouping bits to simplify entering and reading the instructions or data present in digital computer systems. Hexadecimal (hex) uses 16 different digits and is a method of grouping binary numbers in groups of four.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	1 0000	1 0
17	1 0001	1 1
18	1 0010	1 2
19	1 0011	1 3
20	1 0100	1 4

The 16 allowable hex digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

1.2.9 Conversion from a Binary to Hexadecimal Number and Vice Versa

To convert from binary to hexadecimal, group the binary number in groups of four (starting in the least significant position) and write down the equivalent hex digit.

Example: Convert $(01101101)_2$ to hexadecimal number?

Answer:

$$\underbrace{0\ 1\ 1\ 0}_{6}\quad \underbrace{1\ 1\ 0\ 1}_{D} = 6D_{16}$$

$(01101101)_2 = (6D)_{16}$

Example: Convert $(A9)_{16}$ to binary number?

Answer:

$$\overbrace{1\ 0\ 1\ 0}^{A}\quad \overbrace{1\ 0\ 0\ 1}^{9} = 10101001_2$$

$(A9)_{16} = (10101001)_2$

Example 1.18. Convert $111011{,}011_2$ into an equivalent hexadecimal number.

Solution. The binary number given is 111011.011

Grouping 4 bits 0011 1011. 0110

Hexadecimal equivalent 3 B 6

Hence the hexadecimal equivalent number is $(3B.6)_{16}$.

Example 1.20. Convert $9E{,}AF2_{16}$ into an equivalent binary number.

Solution. The hexadecimal number given is 9 E A F 2

4-bit binary equivalent 1001 1110 1010 1111 0010

Hence the equivalent binary number is $(10011110{,}101011110010)_2$.

Hexadecimal - Decimal and Decimal - Hexadecimal Conversions

Example: Convert $(2A6)_{16}$ to decimal number?

Answer:

2 A 6

$6 \times 16^0 = 6 \times 1 = 6$

$A \times 16^1 = 10 \times 16 = 160$

$2 \times 16^2 = 2 \times 256 = \underline{512}$

678_{10}

$(2A6)_{16} = (678)_{10}$

Example: Convert $(82.25)_{10}$ to hexadecimal number?

Answer:

16	82	2 ↑
16	5	5

$0.25 \times 16 = 4$	4 ↓

$(82.25)_{10} = (52.4)_{16}$

H.W: Convert $(1A.8)_{16}$ to Decimal

1 A . 8

$8 \times 16^{-1} = \frac{8}{16} = \frac{1}{2} = .5$

$10 \times 16^0 = 10$

$1 \times 16^1 = 16$

26.5

$(1A.8)_{16} = (26.5)_{10}$

1.2.10 Conversion from an Octal to Hexadecimal Number and Vice Versa

Conversion from octal to hexadecimal and vice versa is sometimes required. To convert an octal number into a hexadecimal number the following steps are to be followed:

(*i*) First convert the octal number to its binary equivalent (as already discussed above).
(*ii*) Then form groups of 4 bits, starting from the LSB.
(*iii*) Then write the equivalent hexadecimal number for each group of 4 bits.

Similarly, for converting a hexadecimal number into an octal number the following steps are to be followed:

(*i*) First convert the hexadecimal number to its binary equivalent.
(*ii*) Then form groups of 3 bits, starting from the LSB.
(*iii*) Then write the equivalent octal number for each group of 3 bits.

Example 1.21. Convert the following hexadecimal numbers into equivalent octal numbers.

(a) A72E *(b) 4.BF85*

Solution:

(*a*) Given hexadecimal number is A 7 2 E
Binary equivalent is 1010 0111 0010 1110
= 1010011100101110
Forming groups of 3 bits from the LSB 001 010 011 100 101 110
Octal equivalent 1 2 3 4 5 6
Hence the octal equivalent of $(A72E)_{16}$ is $(123456)_8$.

(*b*) Given hexadecimal number is 4.B F 8 5
Binary equivalent is 0100.1011 1111 1000 0101
= 0100.1011111110000101
Forming groups of 3 bits 100. 101 111 111 000 010 100
Octal equivalent 4 . 5 7 7 0 2 4

Hence the octal equivalent of $(4.BF85)_{16}$ is $(4.577024)_8$.

Example 1.22. Convert $(247)_8$ into an equivalent hexadecimal number.

Solution. Given octal number is 2 4 7
Binary equivalent is 010 100 111
= 010100111
Forming groups of 4 bits from the LSB 1010 0111
Hexadecimal equivalent A 7
Hence the hexadecimal equivalent of $(247)_8$ is $(A7)_{16}$.

Example 1.23. Convert $(36.532)_8$ into an equivalent hexadecimal number.

Solution. Given octal number is 3 6 . 5 3 2
Binary equivalent is 011 110 .101 011 010
=011110.101011010
Forming groups of 4 bits 0001 1110. 1010 1101
Hexadecimal equivalent 1 E. A D
Hence the hexadecimal equivalent of $(36.532)_8$ is $(1E.AD)_{16}$.

Summary

عشري Decimal	ثنائي Binary	ثماني Octal	سداسي عشر Hexadecimal
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

ملاحظة :- لكل digit بنظام Octal نحتاج الى 3-bit للتمثيل بالنظام Binary .

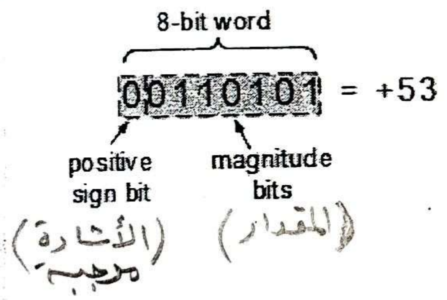
بينما لكل digit بنظام hexadecimal نحتاج الى 4-bit للتمثيل بالنظام Binary .

SIGNED BINARY NUMBERS

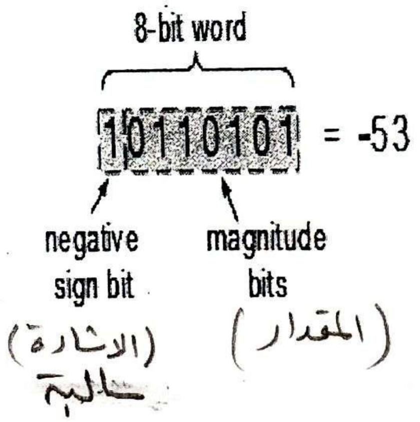
In mathematics, positive numbers (including zero) are represented as unsigned numbers. That is we do not put the +ve sign in front of them to show that they are positive numbers. However, when dealing with negative numbers we do use a -ve sign in front of the number to show that the number is negative in value and different from a positive unsigned value, and the same is true with **signed binary numbers**. However, in digital circuits there is no provision made to put a plus or even a minus sign to a number, since digital systems operate with binary numbers that are represented in terms of "0's" and "1's".

So to represent a positive (N) and a negative (-N) binary number we can use the binary numbers with sign. For signed binary numbers the most significant bit (MSB) is used as the sign. If the sign bit is "0", this means the number is positive. If the sign bit is "1", then the number is negative. The remaining bits are used to represent the magnitude of the binary number in the usual unsigned binary number format.

Positive Signed Binary Numbers.



Negative Signed Binary Numbers



3. Arithmetic Operations

3.1. Addition

The following tables illustrate the rules of addition in Binary, Octal and hexadecimal.

Binary addition rules.

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

Example: Add (11010)$_2$ to (1100)$_2$?

Answer:

```
   1 1      ←— Carry
   1 1 0 1 0
+
   0 1 1 0 0
  ___________
 1 0 0 1 1 0
```

Example: Find the result of (11010.1101)$_2$ + (111101.111)$_2$?

Answer:

```
 1 1 1 1 1 1 1   1      ←— Carry
   0 1 1 0 1 0 . 1 1 0 1
+
   1 1 1 1 0 1 . 1 1 1 0
 ________________________
 1 0 1 1 0 0 0 . 1 0 1 1
```

Ex: Find the result of $(11001.101)_2 + (1101)_2$

```
    1   1
   11001.101
+  01101.000
 ___________
  100110.101
```

3.2. Subtraction

3.2.1. 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	1	1

Figure 6 Binary subtraction rules

Example: Find the results of $(0011010)_2$ - $(001100)_2$?

Answer:

```
          10 10     <----- borrow
        0  0        <----- new value
  0 0 1 1 0 1 0
-   0 0 1 1 0 0
  -------------
  0 0 0 1 1 1 0
```

Example: Find the results of $(110.1101)_2$ - $(11.1011)_2$?

Answer:

```
    10 10      10     <----- borrow
  0  0        0       <----- new value
  1  1  0 . 1 1 0 1
- 0  1  1 , 1 0 1 1
  -----------------
  0  1  1 , 0 0 1 0
```

4. Binary system complements المتممات الثنائية

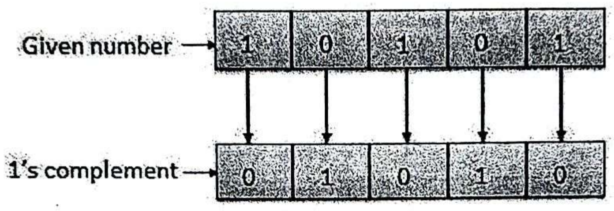
As the binary system has base r = 2. So, the two types of complements for the binary system are 2's complement and 1's complement.

4.1. 1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement. Example of 1's Complement is as follows.

Example: Find the 1's complement of $\mathbf{(10101)_2}$?

Answer:



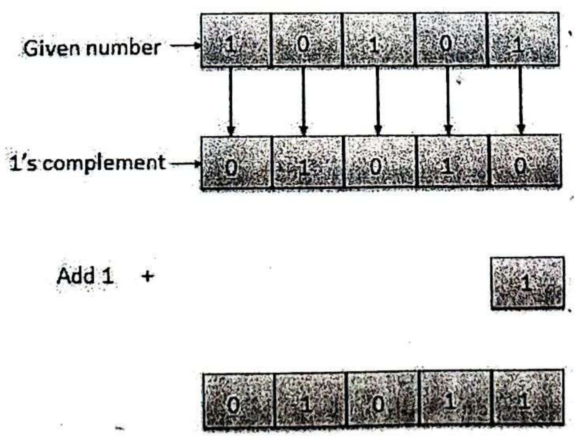
1's complement of $\mathbf{(10101)_2 = (01010)_2}$

4.2. 2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

2's complement = 1's complement + 1

Example: Find the 1's complement of $\mathbf{(10101)_2}$?



2's complement of $\mathbf{(10101)_2 = (01011)_2}$

Ex: Perform the following operation using 1's & 2's complements

1011 – 11101 = ?

	1's	2's
01011	01011	01011
– 11101 →	+ 00010	+ 00011
	01101 (carry)	01110 (carry)
	↓ 1's	↓ 2's
	–10010	**–10010**

Ex: Perform the following operation using 1's & 2's complements

1010 – 110.1 = ?

	1's	2's
1010.0	1010.0	1010.0
– 0110.1 →	1001.0	1001.1
	10011.0 (carry +1)	10011.1 (carry)
	+0011.1	**+0011.1**

Hw: Perform the following operation using 1's & 2's complements

1101 – 101.11 = ?

1's | 2's

1101.00

– 0101.11

3.3. Multiplication

In this section we will discuss the multiplication rules only in binary numbers.

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 × B	Multiplication
1	0 × 0	0
2	0 × 1	0
3	1 × 0	0
4	1 × 1	1

Figure 7 Binary multiplication rules

Example: Find the results of $(10011)_2 \times (0101)_2$?

Answer:

```
      1 0 0 1 1
×       0 1 0 1
---------------
      1 0 0 1 1
    0 0 0 0 0
  1 0 0 1 1
0 0 0 0 0
---------------
0 1 0 1 1 1 1 1
```

Example: Find the results of $(100.111)_2 \times (010.11)_2$?

Answer:

```
          1 0 0 . 1 1 1
            0 1 0 . 1 1
-----------------------
            1 0 0 1 1 1
          1 0 0 1 1 1
        0 0 0 0 0 0
      1 0 0 1 1 1
    0 0 0 0 0 0
-----------------------
    0 1 1 0 1 . 0 1 1 0 1
```

3.14 Division

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

Example: Find the results of $(11011)_2 \div (11)_2$?

Answer:

```
          1 0 0 1
      ___________
  1 1 | 1 1 0 1 1
       -1 1
        -----
          0 0
         -0 0
          -----
            0 1
           -0 0
            -----
              1 1
             -1 1
              -----
              0 0
```

Example: Find the results of $(101011.10)_2 \div (110)_2$?

Answer:

```
            1 1 1 . 0 1
       ________________
  1 1 0 | 1 0 1 0 1 1 . 1 0
         -1 1 0
          -------
            1 0 0 1
           -  1 1 0
            --------
              0 1 1 1
             -  1 1 0
              --------
                0 0 1 1
               -  0 0 0
                --------
                    1 1 0
                   -1 1 0
                    ------
                    0 0 0
```

Binary Coded Decimal

BINARY CODED DECIMAL (BCD)

Binary coded decimal (BCD) is a way to express each of the decimal digits with a binary code. There are only ten code groups in the BCD system, so it is very easy to convert between decimal and BCD. Because we like to read and write in decimal, the BCD code provides an excellent interface to binary systems. Examples of such interfaces are keypad inputs and digital readouts.

There are two types of BCD codes:

1) Weighted BCD
2) Unweighted BCD

rt

Weighted BCD

The 8421 Code

The 8421 code is a type of **BCD** (binary coded decimal) code. Binary coded decimal means that each decimal digit, 0 through 9, is represented by a binary code of four bits. The designation 8421 indicates the binary weights of the four bits ($2^3, 2^2, 2^1, 2^0$). The ease of conversion between 8421 code numbers and the familiar decimal numbers is the main advantage of this code. All you have to remember are the ten binary combinations that represent the ten decimal digits as shown in Table 2–5. The 8421 code is the predominant BCD code, and when we refer to BCD, we always mean the 8421 code unless otherwise stated.

DECIMAL DIGIT	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Invalid Codes You should realize that, with four bits, sixteen numbers (0000 through 1111) can be represented but that, in the 8421 code, only ten of these are used. The six code combinations that are not used—1010, 1011, 1100, 1101, 1110, and 1111—are invalid in the 8421 BCD code.

To express any decimal number in BCD, simply replace each decimal digit with the appropriate 4-bit code, as shown by Example

Convert each of the following decimal numbers to BCD:

(a) 35 (b) 98 (c) 170 (d) 2469

Solution

(a) 3 5 → $(00110101)_{BCD}$

(b) 9 8 → $(10011000)_{BCD}$

(c) 1 7 0 → $(000101110000)_{BCD}$

(d) 2 4 6 9 → $(0010010001101001)_{BCD}$

Convert each of the following BCD codes to decimal:

(a) $(10000110)_{BCD}$ (b) $(001101010001)_{BCD}$ (c) $(1001010001110000)_{BCD}$

Solution (a) 10000110 → $(8\ 6)_{10}$

(b) 001101010001 → $(3\ 5\ 1)_{10}$

(c) 1001010001110000 → $(9\ 4\ 7\ 0)_{10}$

Unweighted BCD

1) Excess-3 Code

The Excess-3 code is another unweighted code.The code assingnment is obtained from the corresponding value of BCD code after the addition of 3.

The table below shows the BCD &Excess-3 code for the decimal digits.

Decimal digit	(BCD) 8421	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Note: The six combinations code that are not used (0000, 0001, 0010, 1101, 1110, 1111) and called invalid in the Gray code

2) ## The Gray Code

The **Gray code** is unweighted and is not an arithmetic code; that is, there are no specific weights assigned to the bit positions. The important feature of the Gray code is that *it exhibits only a single bit change from one code word to the next in sequence.* This property is important in many applications, such as shaft position encoders, where error susceptibility increases with the number of bit changes between adjacent numbers in a sequence.

Table 2–6 is a listing of the 4-bit Gray code for decimal numbers 0 through 15. Binary numbers are shown in the table for reference. Like binary numbers, *the Gray code can have any number of bits.* Notice the single-bit change between successive Gray code words. For instance, in going from decimal 3 to decimal 4, the Gray code changes from 0010 to 0110, while the binary code changes from 0011 to 0100, a change of three bits. The only bit change is in the third bit from the right in the Gray code; the others remain the same.

DECIMAL	BINARY	GRAY CODE	DECIMAL	BINARY	GRAY CODE
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Binary-to-Gray Code Conversion Conversion between binary code and Gray code is sometimes useful. The following rules explain how to convert from a binary number to a Gray code word:

1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.
2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

For example, the conversion of the binary number 10110 to Gray code is as follows:

$$\begin{array}{ccccccccccccccc} 1 & - & + & \rightarrow & 0 & - & + & \rightarrow & 1 & - & + & \rightarrow & 1 & - & + & \rightarrow & 0 & \text{Binary} \\ \downarrow & & & & \downarrow & & & & \downarrow & & & & \downarrow & & & & \downarrow & \\ 1 & & & & 1 & & & & 1 & & & & 0 & & & & 1 & \text{Gray} \end{array}$$

The Gray code is 11101.

Gray-to-Binary Conversion To convert from Gray code to binary, use a similar method; however, there are some differences. The following rules apply:

1. The most significant bit (left-most) in the binary code is the same as the corresponding bit in the Gray code.
2. Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.

For example, the conversion of the Gray code word 11011 to binary is as follows:

$$\begin{array}{cccccccccccc} 1 & & 1 & & 0 & & 1 & & 1 & \text{Gray} \\ \downarrow & \nearrow + & \downarrow & \nearrow + & \downarrow & \nearrow + & \downarrow & \nearrow + & \downarrow & \\ 1 & & 0 & & 0 & & 1 & & 0 & \text{Binary} \end{array}$$

The binary number is 10010.

(a) Convert the binary number 11000110 to Gray code.

(b) Convert the Gray code 10101111 to binary.

Solution (a) Binary to Gray code:

```
1−+→1−+→0−+→0−+→0−+→1−+→1−+→0
↓   ↓   ↓   ↓   ↓   ↓   ↓   ↓
1   0   1   0   0   1   0   1
```

(b) Gray code to binary:

```
1    0    1    0    1    1    1    1
↓ + ↗↓ + ↗↓ + ↗↓ + ↗↓ + ↗↓ + ↗↓ + ↗↓
1 ╱  1 ╱  0 ╱  0 ╱  1 ╱  0 ╱  1 ╱  0
```