Chapter 1: Number Systems and Codes

1.1 The Numbers

1.1.1 Decimal Numbers

In the decimal number system each of the ten digits 0 through 9, represents a certain quantity. These ten symbols (digits) don't limit you to expressing only ten different quantities, because you use the various digits in appropriate positions within a number to indicate the magnitude of the quantity.

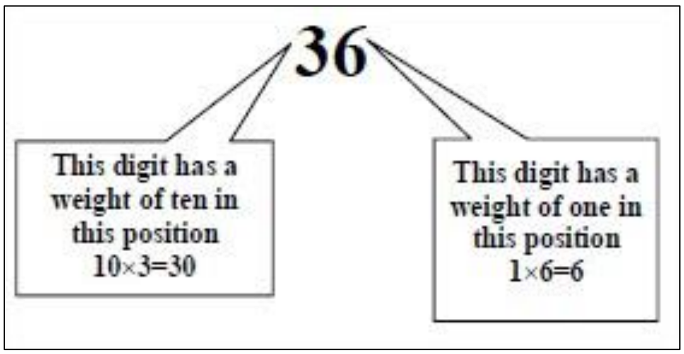

Fig. 1: Show the Decimal digit Wight.

The position of each digit in a decimal number indicates the magnitude of the quantity represented, and can be assigned a weight. 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. In general, a number with a decimal point is represented by a series of coefficients:

$$a_4\ a_3\ a_2\ a_1\ a_0.\ a_{-1}\ a_{-2}\ a_{-3}$$

The coefficients a_j are any of the 10 digits (0, 1, 2… 9), and the subscript value j gives the place value and, hence, the power of 10 by which the coefficient must be multiplied.

$$\boxed{N=10^4\times a_4+10^3\times a_3+10^2\times a_2+10^1\times a_1+10^0\times a_0.\ 10^{-1}\times a_{-1}+10^{-2}\times a_{-2}+10^{-3}\times a_{-3}}$$

1.1.2 Binary Numbers

This is another way to represent quantities. The binary system is less complicated than the decimal system because it has only two digits. It's a base 2 system.

A binary digit, called a bit, has two values 0 & 1. Each coefficient a_j is multiplied by 2_j, and the results are added to obtain the decimal equivalent of the number. For example,

(11010.11) is equal to (26.75) as follows:

$$1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times 2^1 + 0\times2^0 + 1\times 2^{-1} + 1\times 2^{-2}$$

$$16 + 8 + 0 + 2 + 0 + 0.5 + 0.25 = (26.75)_{10}$$

In general, a number expressed in a base-r system has coefficients multiplied by powers of r.

$$\boxed{N=2^n\times a_n + 2^{n-1}\times a_{n-1}+ \ldots..+ 2^2\times a_2 + 2^1\times a_1 + 2^0 \times a_0+ 2^{-1}\times a_{-1} + 2^{-2}\times a_{-2} +\ldots..+ 2^{-m}\times a_{-m}}$$

Now, let us begin to count in the binary system:

One bit 2	values	0, 1
Two bits	4 values	00, 01, 10, 11
Three bits	8 values	000, 001, 010, 011, 100, 101, 110, 111.
Four bits	16 values	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

And so on……..

Table 1.1: Show the decimal number equivalent to the binary number.

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

(LSB) least significant bit

(MSB) most significant bit

LSB (right-most bit) has a weight of $2^0 = 1$.

MSB (left- most bit) has a weight of $2^2 = 4$.

In general, with n – bit, you can count up to a number equal to:

Largest decimal number = $2^n - 1$

When n = 5, you can count from 0 – 31, (32 values). Max number = $(31)_{10}$.

The weight structure of a fractional binary number is

$2^{n-1} \ldots 2^3\ 2^2\ 2^1\ 2^0 .\ 2^{-1}\ 2^{-2}\ 2^{-3} \ldots\ 2^{-n}$

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

1.1.3 Hexadecimal Numbers

It has (16) digits and is used primarily as a compact way of displaying or writing binary numbers, it's very easy to convert between binary and hexadecimal numbers.

Table 1.2: Show the hexadecimal number equivalent to the decimal number and the binary number.

Decimal number	Binary number	Hexadecimal number
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

The general form is:

$$N = (16^{n}\times a_{n}) + (16^{n-1}\times a_{n-1}) + \ldots + (16^{2}\times a_{2}) + (16^{1}\times a_{1}) + (16^{0}\times a_{0}) + (16^{-1}\times a_{-1}) + (16^{-2}\times a_{-2}) + \ldots + (16^{-m}\times a_{-m})$$

How do you count in hexadecimal once you get to F? Simply start over with another column and continue as follows:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F

With two hexadecimal digits, you can count up to FF which in decimal 255.

$(100)_{16} = (256)_{10}$, $(101)_{16} = (257)_{10}$

$(FFF)_{16} = (4095)_{10}$, $(FFFF)_{16} = (65535)_{10}$

1.1.4 Octal Numbers

This system provides a convenient way to express binary numbers and codes (like Hex. Number system), it is used less frequently than hexadecimal.

The general form is:

$$N=(8^n\times a_n)+(8^{n-1}\times a_{n-1})+\ldots+(8^2\times a_2)+(8^1\times a_1)+(8^0\times a_0)+(8^{-1}\times a_{-1})+(8^{-2}\times a_{-2})+\ldots+(8^{-m}\times a_{-m})$$

The octal number system is composed of eight digits, which are:

0 1 2 3 4 5 6 7

To count above 7, begin another column and start over

10 11 12 13 14 15 16 17

20 21 22 23 24 25 26 27

30 31 32 33 34 35 36 37……

$(15)_8 = (13)_{10} = (D)_{16}$

Table 1.3: Show the octal number equivalent to the decimal number and the binary number.

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

1.1.5 Binary Coded Decimal (BCD)

The 8421 code is a type of binary coded decimal. It means that each decimal digit 0 through 9 is represented by a binary code of four bits:

Table 1.4: Show the binary number (BCD 4 bit) equivalent to the decimal number.

Decimal number	Binary number (BCD)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

{1010, 1011, 1100, 1101, 1110, 1111} are invalid in 8421 BCD code.

10 0001 0000: 32 = 0011 0010

1.1.6 The Gray Code

- This code is often used in digital systems because it has the advantage that only one bit in the numerical representation changes between successive numbers. For example, 0111 represent 5 and 0101 represent 6 in Gray code.
- Its primary application is in the location of angles on a rotating shaft.
- This is an un-weighted code, which means that there is <u>NO SPECIFC WEIGHT</u> assigned code.
- The Gray code is NOT an arithmetic code.

Table 1.5: Show the gray code number equivalent to the decimal number and the binary number.

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
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

1.1.7 Exess-3 Code

Is un-weighted code in which each code is obtained from the corresponding binary value plus 3. It is particularly significant for arithmetic operations as it overcomes the shortcomings encountered while using the 8421 BCD code to add two decimal digits whose sum exceeds 9. The excess-3 code has no such limitation, and it considerably simplifies arithmetic operations. Table (1.7) lists the excess-3 code for the decimal numbers 0–9. The excess-3 code for a given decimal number is determined by adding ‘3’ to each decimal digit in the given number and then replacing each digit of the newly found decimal number by its four-bit binary equivalent.

Table (1.7) lists the binary and excess-3 code for the decimal numbers 0–9.

Decimal	***Binary***	***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

1.2 Numbers and Codes Converter

1.2.1 Decimal Number Converter

1.2.1.1 Decimal – to – Binary Conversion

There are two methods to do the conversion

- The first one ***Sum – of – Weights*** Method: determine the set of binary weights whose sum is equal to the decimal number.

  Example: Convert 9 to binary.

  Solution: 9= 8 + 1 “choosing numbers with power of 2”

Binary weights are:	32	16	8	4	2	1
			1	0	0	1

  So $(9)_{10} = (1001)_2$

  Example

  Convert the following decimal number to binary number: 12, 25 and 82.

  12 = 8 + 4 = 1100

  25 = 16 + 8 + 1 = 11001

  82 = 64 + 16 + 2 = 1010010.

  Homework: Convert the following decimal number to binary number: 23, 45, 66 and 578.

- The second method ***Repeated Division -by- 2***:

  Example: Convert 13 to binary.

  Solution:

  13/2 = 6 reminder = 1 this is the LSB

  6/2 = 3 reminder = 0

  3/2 = 1 reminder = 1

  1/2 = 0 reminder = 1 this is the MSB

  Stop when the whole number quotient is 0. So $(13)_{10} = (1101)_2$

Example: convert 45 to binary.

Solution:

45/2 = 22 reminder = 1 this is the LSB

22/2 = 11 reminder = 0

11/2 = 5 reminder = 1

5/2 = 2 reminder = 1

2/2 = 1 reminder = 0

1/2 = 0 reminder = 1 this is the MSB

$(45)_{10} = (101101)_2$

Homework: Convert the following decimal number to binary number: 23, 45, 66 and 578.

What about decimal numbers with fractions?

Binary weights. 0.5 0.25 0.125 0.0625

$(0.625)_{10} = 0.5 + 0.125 = (0.101)_2$ (by using sum – of – weight method)

OR by using repeated ***MULTIPLECTION*** - by – 2 method

$0.625 \times 2 = 1.25$ 1 this is the MSB

$0.25 \times 2 = 0.5$ 0

$0.5 \times 2 = 1.00$ 1 this is the LSB

So $(0.625)_{10} = (0.101)_2$

MSB LSB

Homework: Convert the following decimal number to binary number: 77.375 and 185.1875.

1.2.1.2 Decimal – to – Hexadecimal Conversion

Repeated division of a decimal number by 16 will produce the equivalent hexadecimal number, formed by the reminders of the division. The firs reminder produced is the LSD. Each successive division by 16 yields a reminder that becomes a digit in the equivalent hexadecimal number.

Note: when a quotient has a fractional part, the fractional part is multiplied by the divisor to get the remainder.

Example: Convert the following decimal number to hexadecimal number $(650)_{10}$

Solution:

650/16 = 40	reminder = 10	= A this is the LSD
40/16 = 2	reminder = 8	= 8
2/16 = 0	reminder = 2	= 2 this is the MSD

So $(650)_{10} = (28A)_{16}$

Homework: Convert the following decimal number to hexadecimal number: 10, 253, and 315.

1.2.1.3 Decimal – to – Octal Conversion

Repeated division of a decimal number by 8 will produce the equivalent octal number, formed by the reminders of the division. The firs reminder produced is the LSD. Each successive division by 8 yields a reminder that becomes a digit in the equivalent octal number.

Example: Convert the following decimal number to octal number (359)10 = (?)8

Solution:

359/8 = 44	reminder = 7	this is the LSD
44/8 = 5	reminder = 4	
5/8 = 0	reminder = 5	this is the MSD

So $(359)_{10} = (547)_8$

Homework: Convert the following decimal number to octal number: 7, 15, 20 and 283.

1.2.2 Binary Number Converter

1.2.2.1 Binary – to – Decimal Conversion

The decimal value of any binary number can be found by adding the weights of all bits that are 1 and discarding the weights of all bits that are 0.

Example: Convert the binary number 1101101 to decimal.

Solution:

$2^6 \times 1 + 2^5 \times 1 + 2^4 \times 0 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1$

$64 + 32 + 0 + 8 + 4 + 0 + 1 = (109)_{10}$

Example: Convert 0.1011 to decimal.

Solution:

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

$0 + 0.5 + 0.125 + 0 + 0.0625 = (0.6875)_{10}$

Homework: Convert the following binary number to decimal number: 01, 10, 10010 and 10101010.

1.2.2.2 Binary – to – Hexadecimal Conversion

Simply break the binary number into 4-bit groups, starting at the rightmost bit and ***replace each 4-bit group*** with the equivalent hexadecimal symbol.

Example: Convert the binary number 1100101001010111 to hexadecimal.

Solution:

$1100101001010111 = \underbrace{1100}_{C}\ \underbrace{1010}_{A}\ \underbrace{0101}_{5}\ \underbrace{0111}_{7}$

Note: each group must be four bits.

Homework: Convert the following binary number to hexadecimal number: 1010, 110, 11101, 11101111 and 11010101010101.

1.2.2.3 Binary – to – Octal Conversion

Start with the right – most group of three bits and moving from right to left, ***convert each 3-bit group*** to the equivalent octal digit.

Example: Convert the binary number 11010000100 to octal.

Solution:

11010000100 = 011 010 000 100

3 2 0 4

$(11010000100)_2 = (3204)_8$

Homework: Convert the following binary number to octal number: 1010, 110, 11101, 11101111 and 11010101010101.

1.2.3 Hexadecimal Number Converter

1.2.3.1 Hexadecimal – to – Binary Conversion

Replace each hexadecimal symbol with the appropriate four bits.

Example: Convert the hexadecimal number 10AF to Binary.

Solution:

10AF = 1 0 A F

0001 0000 1010 1111

So $(10AF)_{16} = (0001000010101111)_2$

Homework: Convert the following hexadecimal number to binary number: 1234, A3C and EF.

1.2.3.2 Hexadecimal – to – Decimal Conversion

Convert each hexadecimal digit to equivalent decimal digit (see table 1:2) and multiplying each hexadecimal digit by its weight and then taking the sum of these products. The weights of a hexadecimal number are increasing powers of 16 (from right to left).

Table 2.1: For hexadecimal weight.

16^4	16^3	16^2	16^1	16^0
65536	4096	256	16	1

Example: Convert the hexadecimal number FEA1 to decimal number.

Solution: $F_{16}=15_{10}$,

$E_{16}=14_{10}$,

$A_{16}=10_{10}$,

$1_{16}=1_{10}$

$(F*16^3 + E*16^2 + A*16^1 + 1*16^0)_{16} = (15*16^3 + 14*16^2 + 10*16^1 + 1*16^0)_{16} = 65185_{10}$

Homework: Convert the following hexadecimal number to decimal number: 1234, A3C and EF.

1.2.3.3 Hexadecimal – to – Octal Conversion

There are two methods to do this conversion:

- The first method is to convert the hexadecimal number to decimal number then convert decimal number to octal number.
- The second method is to convert the hexadecimal number to binary number then convert binary number to octal number.

Homework: Convert the following hexadecimal number to octal number: 6, 77, 4321 and BDF.

1.2.4 Octal Number Converter

1.2.4.1 Octal – to – Decimal Conversion

Convert each octal digit to equivalent decimal digit (see table 1:3) and multiplying each octal digit by its weight and then taking the sum of these products. The weights of an octal number are increasing powers of 8 (from right to left).

Example: Convert the octal number 2374 to decimal number.

Solution: $(2374)_8 = 2 \times 8^3 + 3 \times 8^2 + 7 \times 8^1 + 4 \times 8^0$

$= 1024 + 192 + 56 + 4$

$= (1276)_{10}$

Homework: Convert the following octal number to decimal number: 45, 676 and 7145.

1.2.4.2 Octal – to – Binary Conversion

Replace each octal digit with the appropriate three binary bits.

Example: Convert the octal number 25 to binary number.

Solution:

$$(25)_8 = \begin{array}{cc} 2 & 5 \\ \downarrow & \downarrow \\ 010 & 101 \end{array}$$

$(25)_8 = (010101)_2$

$$(7526)_8 = \begin{array}{cccc} 7 & 5 & 2 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \underline{111} & \underline{101} & \underline{010} & \underline{110} \end{array}$$

$(7526)_8 = (111101010110)_2$

Homework: Convert the following octal number to binary number: 45, 677 and 7245.

1.2.4.3 Octal – to – Hexadecimal Conversion

There are two methods to do this conversion:

- The first method is to convert the octal number to decimal number then convert decimal number to hexadecimal number.
- The second method is to convert the octal number to binary number then convert binary number to hexadecimal number.

Homework: Convert the following octal number to hexadecimal number: 6, 77 and 432.

1.2.5 The Gray Cod

1.2.5.1 Binary – to – Gray Conversion

To convert from binary to gray we have to take the following steps:

A. The MSB of binary and gray code are the same.
B. The second bit of the gray code is the compare of first bit and second bit of the binary number if they are same, the result is 0, if they are different, the result is 1.
C. The third bit of the gray code is the compare of second bit and third bit of the binary number if they are same, the result is 0, if they are different, the result is 1 and so on.

	MSB				LSB
Binary	1	0	1	1	0
Gray	1	1	1	0	1

1.2.5.2 Gray – to – Binary Conversion

To convert from gray to binary we have to take the following steps:

A. The MSB of gray code and binary are the same.

B. The second bit of binary number is the compare of MSB bit of gray code and second bit of the binary number. If they are same, the result is 0, if they are different, the result is 1.

C. The third bit of binary number is the compare of second bit of gray code and third bit of the binary number. If they are same, the result is 0, if they are different, the result is 1. Repeat until both code are the same.

	MSP				LSB
Gray	1	1	1	0	1
Binary	1	0	1	1	0

1.2.6 Excess -3 code – to – Decimal

The excess-3 code for a given decimal number is determined by adding '3' to each decimal digit in the given number and then replacing each digit of the newly found decimal number by its four-bit binary equivalent.

Example:

Convert the following dicimal number to excess-3 code: A) 25, B) 63, C) 709

Solution

A) First, add 3 to each digit in the decimal number 25

2(0010)+3(0011)=5 (0101) 5(0101)+3(0011)=8 (1000)

Then put the representation of all digits together: $25=(01011000)_{\text{Excess-3 code}}$

B) First, add 3 to each digit in the decimal number 63

6(0110)+3(0011)=9 (1001) 3(0011)+3(0011)=6 (0110)

Then put the representation of all digits together: $63=(10010110)_{\text{Excess-3 code}}$

C) First, add 3 to each digit in the decimal number 709

7(0010)+3(0011)=10 (1010) 0(0101)+3(0011)=3 (0011)

9(0101)+3(0011)=12 (1100)

Then put the representation of all digits together: $709=(01011000)_{\text{Excess-3 code}}$

Homework: Convert each of the decimal number to excess-3 and add as indicated: 14, 67 and 157.

1.3 Mathematical Operation

1.3.1 Binary Addition

Binary arithmetic is performed according to the same rules as decimal arithmetic. When adding two numbers, each column of digits is added in sequence from right to left and, if the sum of any column is greater than the value of the highest digit, a carry is added to the next column. In binary, the largest digit is 1, so any sum greater than 1 will result in a carry. The table below show add two binary number.

Table 3.1: Add two binary number.

Operation	Result
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	10 (zero with carry =1)

In case if there are three number binary bits to be add the following table can be considered:

Table 3.2: Add three binary number.

Operation	Result	How to read it
1 + 0 + 0	1	1 with no carry
1 + 1 + 0	10	Zero with carry = 1
1 + 0 + 1	10	Zero with carry = 1
1 + 1 + 1	11	One with carry = 1

Example: Add the following binary number (11+11=?), (110+100=?) and (101+110+11=?).

Solution:

A- 11 + 11 = 110 because

$$\begin{array}{rr} & 11 \\ + & 11 \\ \hline & 110_2 \end{array} \qquad \begin{array}{rr} & 3 \\ + & 3 \\ \hline & 6_{10} \end{array}$$

B- 110 + 100 = 1010

$$\begin{array}{rr} & 110 \\ + & 100 \\ \hline & 1010_2 \end{array} \qquad \begin{array}{rr} & 6 \\ + & 4 \\ \hline & 10_{10} \end{array}$$

C- 101 + 110 + 11= 1110

$$\begin{array}{rr} & 101 \\ + & 110 \\ + & 011 \\ \hline & 1110_2 \end{array} \qquad \begin{array}{rr} & 5 \\ + & 6 \\ + & 3 \\ \hline & 14_{10} \end{array}$$

Classwork: Add the following binary number: (110 + 10), (101 +111) and (1010 +1111).

Homework: Convert the decimal number to binary number and then do the addition operation: (13+20=?), (17+16=?) and (11+8=?).

1.3.2 Binary Subtraction

Table 3.3: Subtract two binary number.

Operation	Result
0 – 0 = 0	with a barrow = 0
0 – 1 = 1	with a barrow = 1 10 – 1 = 1
1 – 0 = 1	with a barrow = 0
1 – 1 = 0	with a barrow = 0

Example: Subtract the following binary number (11–01), (111–100) and (1011–101)

Solution:

A- 11 – 01 = 10 because

$$\begin{array}{r} 11 \\ -\ 01 \\ \hline 10_2 \end{array} \qquad \begin{array}{r} 3 \\ -\ 1 \\ \hline 2_{10} \end{array}$$

B- 111 – 100 = 011

$$\begin{array}{r} 111 \\ -\ 100 \\ \hline 011_2 \end{array} \qquad \begin{array}{r} 7 \\ -\ 4 \\ \hline 3_{10} \end{array}$$

C- 1011 – 101 = 110

$$\begin{array}{r} 1011 \\ -\ 0101 \\ \hline 0110_2 \end{array} \qquad \begin{array}{r} 11 \\ -\ 5 \\ \hline 6_{10} \end{array}$$

Classwork: Subtract the following binary number: (110–10), (111–101) and (1001–111).

Homework: Convert the decimal number to binary number and then do the subtraction operation: (13–5=?), (4–3=?) and (9–7=?).

1.3.3 Binary Multiplication

Table 3.4: Multiplication two binary number.

Operation	Result
0 × 0	0
0 × 1	0
1 × 0	0
1 × 1	1

It involves forming partial products, shifting each successive partial product left one place and then adding all the partial products.

Example: Multiplication the following number (12×12= 144).

Solution:

Operation in Decimal	Operation in Binary
1 2 × 1 2 2 4 + 1 2 1 4 4	1 1 0 0 × 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 + 1 1 0 0 1 0 0 1 0 0 0 0

Classwork: Multiplication the following binary number: (11×11), and (100×101).

Homework: Convert the decimal number to binary number and then do the multiplication operation: (11×9=?), (10×13=?) and (6×7=?).

1.3.4 Binary Division

This operation follows the same procedure as division in decimal number system.

Example: Division the following number:

Solution:

#	Operation in Decimal	Operation in Binary
	3 3 ⟌ 9 9 0	11 11 ⟌ 1001 011 0011 0011 0000
	2.5 6 ⟌ 15 12 30 30 00	10.1 110 ⟌ 1111 110 00110 00110 00000

Homework	
7 ⟌ 11	
6 ⟌ 25	
4 ⟌ 18	

1.3.5 1's and 2's Complement of Binary Numbers

The l's complement and the 2's complement of a binary number are important because they permit the representation of negative numbers. The method of 2's complement arithmetic is commonly used in computers to handle negative numbers.

Finding the 1's Complement

The l's complement of a binary number is found by changing all 1s to 0s and all 0s to 1s, as illustrated below:

10110010	Binary number
↓↓↓↓↓↓↓↓	
01001101	1's complement

1's Complement Subtraction

Subtraction of binary number can be accomplished by using 1's complement method. Which allows us to subtract using only addition.

Subtracting small number from large number

Is done as follows steps:

A. Determine the 1's complement of the smaller number.
B. Add the 1's complement to the large number.
C. Remove the last carry and add it to the result.

Example: Subtract the following binary number (11–01), (111–100) and (1011–101) by using 1's complement method.

Solution:

A- 11 – 01 = 10 because

$$\begin{array}{r} 11 \\ -\ 01 \\ \hline 10_2 \end{array} \quad \xrightarrow{\text{1's complement}} \quad \begin{array}{r} 11 \\ +\ 10 \\ \hline 1\ \ 01 \\ \text{Adding the carry} \searrow \ \ 1 \\ \hline 10_2 \end{array}$$

B- 111 – 100 = 011 because

$$
\begin{array}{r} 111 \\ -\ 100 \\ \hline 011_2 \end{array}
\quad \xrightarrow{\text{1's complement}} \quad
\begin{array}{r} 111 \\ +\ 011 \\ \hline 1\ \ 010 \\ \text{Adding the carry} \searrow \quad 1 \\ \hline 011_2 \end{array}
$$

C- 1011 – 0101 = 0110 because

$$
\begin{array}{r} 1011 \\ -\ 0101 \\ \hline 0110_2 \end{array}
\quad \xrightarrow{\text{1's complement}} \quad
\begin{array}{r} 1011 \\ +\ 1010 \\ \hline 1\ \ 0101 \\ \text{Adding the carry} \searrow \quad 1 \\ \hline 0110_2 \end{array}
$$

Classwork: Subtract the following binary number by using 1's complement method: (110–10), (111–101) and (1001–111).

Homework: Subtract the following binary number by using 1's complement method: (11001–10011=?) and (1000–0011=?).

Subtracting large number from small number

Is done as follows steps:

A. Determine the 1's complement of the large number.
B. Add the 1's complement to the small number.
C. There is no carry, so we take the 1's complement for the last result and change the sign to negative.

A- 1001 – 1101 = – 0100 because

$$
\begin{array}{r} 1001 \\ -\ 1101 \\ \hline -\ 0100_2 \end{array}
\quad \xrightarrow{\text{1's complement}} \quad
\begin{array}{r} 1001 \\ +\ 0010 \\ \hline 1011 \\ \text{1's complement} \Downarrow \\ \hline -\ 0100_2 \end{array}
$$

B- 10011 – 11101 = – 01010 because

$$\begin{array}{rcr} 10011 & & 10011 \\ -\ 11101 & \xrightarrow{\text{1's complement}} & +\ 00010 \\ \hline -\ 01010_2 & & 10101 \\ & \text{1's complement} \downarrow & \\ \hline & & -\ 01010_2 \end{array}$$

Classwork: Subtract the following binary number by using 1's complement method: (101–011), (1011–1001) and (11111–10101).

Homework: Subtract the following binary number by using 1's complement method: (111–10100=?) and (11–1010=?).

Finding the 2's Complement

The 2's complement of a binary number is found by adding 1 to the LSB of the l's complement.

$$\boxed{\textbf{2's complement = (l's complement) + 1}}$$

Example: Find the 2's complement of 10110010.

Solution:

$$\begin{array}{rl} 10110010 & \text{Binary number} \\ 01001101 & \text{l's complement} \\ +\quad\quad\quad 1 & \text{add 1} \\ \hline 01001110 & \text{2's complement} \end{array}$$

2's Complement Subtraction

As was shown in previous section, there are two cases to the subtraction:

Subtracting small number from large number

Is done as follows steps:

A. Determine the 2's complement of the smaller number.
B. Add the 2's complement to the large number.

C. Discard the carry (there is always carry in this case).

Example: Subtract the following binary number (1100–1011) by using 2's complement method.

Solution:

A- 1100 – 1011 = 0001 because

$$\begin{array}{rcr} 1100 & & 1100 \\ -\ 1011 & \xrightarrow{\text{2's complement}} & +\ 0101 \\ \hline 0001 & & 1\ \ 0001 \\ & \boxed{\text{Discarded}} & \end{array}$$

Subtracting large number from small number

Is done as follows steps:

- A- Determine the 2's complement of the large number.
- B- Add the 2's complement to the small number.
- C- There is no carry, so we take the 2's complement for the last result and change the sign to negative.

A-1001 – 1101 = – 0100 because

$$\begin{array}{rcr} 1001 & & 1001 \\ -\ 1101 & \xrightarrow{\text{2's complement}} & +\ 0011 \\ \hline -\ 0100 & & 1100 \\ & \text{2's complement} & \Downarrow \\ \hline & & -\ 0100_2 \end{array}$$

Classwork: Subtract the following binary number by using 2's complement method: (110–10), (1001–111), (1011–1001) and (11111–10101).

Homework: Subtract the following binary number by using 2's complement method: (11001–10011=?), (1000–0011=?), (11–1010=?) and (111–10100=?).

1.3.6 9’s and 10’s Complement of Decimal Numbers

Finding the 9’s Complement

The 9’s complement of a decimal number is found by subtraction each digit from 9.

Table 3.5: 9’s complement for decimal number digit.

Decimal Digit	9’s Complement
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

Example: Find the 9’s complement for the following decimal numbers: 45, 19 and 2378.

Solution:

1- 45 $\xrightarrow{\text{9's complement}}$ 54

2- 19 $\xrightarrow{\text{9's complement}}$ 80

3- 2378 $\xrightarrow{\text{9's complement}}$ 7621

9's Complement Subtraction

Subtraction of decimal number can be accomplished by using 9’s complement method. Which allows us to subtract using only addition.

Subtracting small number from large number

Is done as follows steps:

A. Determine the 9's complement of the smaller number.
B. Add the 9's complement to the large number.
C. Remove the last carry and add it to the result.

Example: Subtract the following decimal number (8 – 3=?) by using 9's complement method.

Solution:

A- 8 – 3 = 5 because

	8		8
–	3	$\xrightarrow{\text{9's complement}}$ +	6
	5	Adding the carry	1 4
			1
			5

Subtracting large number from small number

Is done as follows steps:

A. Determine the 9's complement of the large number.
B. Add the 9's complement to the small number.
C. There is no carry, so we take the 9's complement for the last result and change the sign to negative.

Example: Subtract the following decimal number (8 – 3=?) by using 9's complement method.

Solution:

A- 15 – 28 = – 13 because

	15		15
–	28	$\xrightarrow{\text{9's complement}}$ +	71
–	13		86
		9's complement	⇓
			13

Homework: Subtract the following decimal number by using 9's complement method: (19–18), (46–23), (35–77) and (847–942).

Finding the 10's Complement

The 10's complement of a decimal number is equal to the 9's complement plus1.

10's complement = (9's complement) + 1

Table 3.5: 9's complement for decimal number digit.

Decimal Digit	9's Complement	10's Complement
0	9	9 + 1 = ***0***
1	8	8 + 1 = ***9***
2	7	7 + 1 = ***8***
3	6	6 + 1 = ***7***
4	5	5 + 1 = ***6***
5	4	4 + 1 = ***5***
6	3	3 + 1 = ***4***
7	2	2 + 1 = ***3***
8	1	1 + 1 = ***2***
9	0	0 + 1 = ***1***

Example: Find the 10's complement for the following decimal numbers: 45, 19 and 2378.

Solution:

1- 45 $\xrightarrow{\text{9's complement}}$ 54 $\xrightarrow{\text{10's complement}}$ 54+1=55

2- 19 $\xrightarrow{\text{9's complement}}$ 80 $\xrightarrow{\text{10's complement}}$ 80+1=81

3- 2378 $\xrightarrow{\text{9's complement}}$ 7621 $\xrightarrow{\text{10's complement}}$ 7621+1=7622

10's Complement Subtraction

Subtraction of decimal number can be accomplished by using 10’s complement method. Which allows us to subtract using only addition.

Subtracting small number from large number

Is done as follows steps:

A- Determine the 10’s complement of the small number.

B- Add the 10’s complement to the large number.

C- Discard the carry.

Example: Subtract the following decimal number (8 – 3=?) by using 10’s complement method.

Solution:

A- 8 – 3 = 5 because

$$\begin{array}{rcr} 8 & & 8 \\ -\ 3 & \xrightarrow{\text{10's complement}} & +\ 7 \\ \hline 5_{10} & & 1\ 05_{10} \end{array}$$

$\boxed{\text{Discarded}}$

Subtracting large number from small number

Is done as follows steps:

A- Determine the 10’s complement of the large number.

B- Add the 10’s complement to the small number.

C- There is no carry, so we take the 9’s complement for the last result and change the sign to negative.

Example: Subtract the following decimal number (8 – 3=?) by using 10’s complement method.

Solution:

A- 15 – 28 = – 13 because

$$\begin{array}{rcr} 15 & & 15 \\ -\ 28 & \xrightarrow{\text{10's complement}} & +\ 72 \\ \hline -\ 13_{10} & & 87 \\ & \text{10's complement} \downarrow & \\ \hline & & -\ 13_{10} \end{array}$$

Homework: Subtract the following decimal number by using 10’s complement method: (54–21), (196–155) and (7653–92).

1.3.7 Excess-3 Code Addition and Subtraction

Excess-3 code addition

The operation of addition can be done by very simple method we will illustrate the operation in a simple way using three rules:

1- Add the excess-3 number using the rules for binary addition.

2- If there is no a carry from a four-bit group, subtract 3 (0011) from that group to get the excess-3 code for the digit.

3- If there is a carry from a four-bit group, add 3(0011) to that group to get the excess-3 code for the digit and add 3 to any new column (digit) generated by the last carry.

Example:- Convert each of the decimal number to excess-3 and add as indicated:

A) 8+1=?

8	1011	← Excess-3 for 8
+ 1	+ 0100	← Excess-3 for 1
9	1111	← no carry
	- 0011	← Subtract 3(0011)
	1100	← Excess-3 for 9

B) 35+24=?

35	0110 1000	← Excess-3 for 35
+ 24	+ 0101 0111	← Excess-3 for 24
59	1011 1111	← no carry
	-0011 -0011	← Subtract 3(0011)
	1000 1100	← Excess-3 for 59

C) 273+126=?

273	0101 1010 0110	Excess-3 for 273
+ 126	+ 0100 0101 1001	Excess-3 for 24
399	1 1001 1111 1111	no carry
	+0011 - 0011 -0011 -0011	
	0110 1100 1100	Excess-3 for 399

D) 7+6=?

7	1010	Excess-3 for 7
+ 6	+ 1001	Excess-3 for 6
13	1 0011	there is a carry
	+0011 +0011	add 3(0011) to both group
	0100 0110	Excess-3 for 13

E) 98+86=?

98	1100 1011	Excess-3 for 98
+ 86	+ 1011 1001	Excess-3 for 86
184	1 1000 0100	
	+0011 +0011 +0011	
	0100 1011 0111	Excess-3 for 184

F) 15+15=?

15	0100 1000	Excess-3 for 15
+ 15	+ 0100 1000	Excess-3 for 15
30	1001 0000	carry out of right-most group only
	-0011 +0011	Subtract 3 from left, add 3 to right
	0110 0011	Excess-3 for 30

G) 29+39=?

```
   29            0101 1100   ←  Excess-3 for 29
+  39       +    0110 1100   ←  Excess-3 for 39
------      --------------
   68            1100   1000 ←  carry out of right-most group only
              -0011  +0011  ←  Subtract 3 from left, add 3 to right
            --------------
                 1001 1011   ←  Excess-3 for 68
```

Subtraction in BCD and Excess-3 codes

Example:

A)

```
   7
 – 2
------
 + 5
```

BCD

```
   0111
 - 0010     9's com
 ------
   0111
   + 1
 ------
   1000  →  10's com
 + 0111
 ------
   1111     > 9
 + 0110     Add 6
 ------
 1 0101     BCD for 5
   + 5₁₀
```

Discarded ← 1

Excess-3 code

```
   1010
 - 0101     1's com
 ------
   1010
   + 1
 ------
   1011  →  2's com
 + 1010
 ------
 1 0101     There is a carry
 + 0011     +3
 ------
   1000     Excess-3 code for 5
   + 5₁₀
```

B)

$$
\begin{array}{r}
6 \\
-\ 8 \\
\hline
-\ 2
\end{array}
$$

<u>BCD</u>

$$
\begin{array}{rl}
0110 & \\
-\ 1000 & \text{9's com} \\
\hline
0001 & \\
+\ 1 & \\
\hline
0010 & \text{10's com} \\
+\ 0110 & \\
\hline
\cancel{1}\ 1000 & \text{Take 10's comp} \\
0010 & \\
\hline
-\ 2_{10} &
\end{array}
$$

<u>Excess-3 code</u>

$$
\begin{array}{rl}
1001 & \\
-\ 1011 & \text{1's com} \\
\hline
0100 & \\
+\ 1 & \\
\hline
0101 & \text{2's comp} \\
+\ 1001 & \\
\hline
1110 & \text{No carry} \\
+\ 1101 & \text{+3 (2's comp)} \\
\hline
\cancel{1}\ 1011 & \text{2's comp} \\
0101 & \text{Excess-3 code for } \mathbf{2} \\
\hline
-\ 2_{10} &
\end{array}
$$