Module No.1

Number Systems and Codes

- Review of Binary, Octal and Hexadecimal Number Systems
- Conversion of number systems
- Binary code, Gray code and BCD code
- Binary Arithmetic, Addition, Subtraction using 1's and 2's Complement

System

We can define a system as a group of components or subsystems that integrate and function together in order to achieve a specific goal.

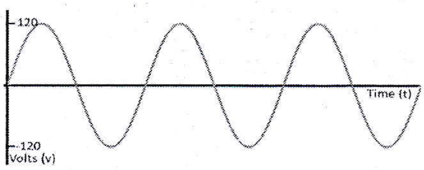
Analog System : The system which deals with analog signals called as analog system.

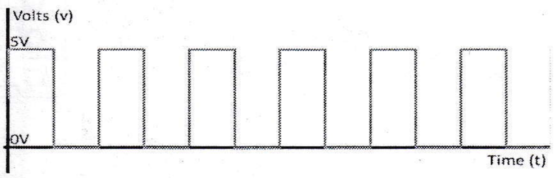
Ex. analog watches, Analog voltmeter , Ammeter etc.

Digital System : The system which deals with digital signals called as digital system.

Ex. Digital watches, computer, smartphones, scanners and digital ticket readers

Analog Signal : continuously variable signal **Digital Signal :** signals usually take only two levels





The common examples of analog signals are temperature, current, voltage, voice, pressure, speed, etc.

The common example of digital signal is the data store in a computer memory.

Digital Systems

- Digital circuits are electronic circuits designed to operate with a fixed number of discrete voltage values.
- Truth values, true and false, denoted T and F, respectively.
- Boolean values, 1 and 0.
- Voltage, the electrical potential between a point in the circuit and a common reference point, called "ground." We call these values "high" and "low," denoted H and L, respectively

Digital Systems Application

Digital computers :

- General purposes
- Many scientific, industrial and commercial applications

Digital systems:

- Telephone switching exchanges
- Digital camera
- Electronic calculators, PDA's
- Digital TV

Analog signal	Digital signal
- continuous & time varying	It is discrete in nature
- It has infinite values	It has only two States
- Easily affected by noise	less prone to noise
- Accuracy is affected by noise	More immune to noise
- Use more power	Use less power
- Ex. Temperature, voltage etc	ex. Data store in PC

Digital Signal

- For digital systems, the variable takes on discrete values.
  - Two level
- Binary values are represented abstractly by:
  - Digits 0 and 1
  - Words (symbols) False (F) and True (T)
  - Words (symbols) Low (L) and High (H)
  - And words On and Off
- Binary values are represented by values or ranges of values of physical quantities.

Number system classification

- Positional/weighted
  - Decimal
  - Binary
  - Octal
  - Hexadecimal
  - BCD
- Non-positional / Non-weighted
  - Excess-3
  - Roma code
  - Cyclic code
  - Gray code

Number system

The ***Number system*** is the group of symbols used to represent the quantity.

Numbering System		
System	Base	Digits
Binary	2	0, 1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

1. Decimal Number System

- The number of ***base*** or ***radix*** ten is called decimal numbers
- Other number system is derived from this number.
- It is generated with the combination of
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Base (also called radix) = 10
  - 10 digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
- Digit Position
  - Integer & fraction
- Digit Weight
  - Weight = $(Base)^{Position}$
- Magnitude
  - Sum of "*Digit* x *Weight*"
- Formal Notation

Decimal Positional Value

10^3	10^2	10^1	10^0 .	10^{-1}	10^{-2}	10^{-3}
MSB			Decimal point			LSB

$$d_2*B^2+d_1*B^1+d_0*B^0+d_{-1}*B^{-1}+d_{-2}*B^{-2}$$

2. Binary Number System

- Binary number is a number expressed in the binary numeral system or base-2
- which represents numeric values using two different symbols: typically 0 (zero) and 1 (one)

2^2	2^1	2^0 .	2^{-1}	2^{-2}
MSB				LSB

3. Octal Number System

- The octal number system, or oct for short, is the base-8 number system.
- Uses the digits 0 to 7.
- Octal numbers can be made from binary numbers by grouping consecutive binary digits into groups of three

octal positional value

8^3	8^2	8^1	8^0 .	8^{-1}	8^{-2}	8^{-3}
MSB			octal point			LSB

4. Hexadecimal Number System

- The hexadecimal number system, also known as just hex, is a number system made up of 16 symbols (base 16).
- The standard number system is called decimal (base 10) and uses ten symbols: 0,1,2,3,4,5,6,7,8,9.
- Hexadecimal uses the decimal numbers and includes six extra symbols
- 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E and F

Decimal Numbers	Binary Numbers	Octal Numbers	Hexa-decimal Numbers
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	-	8
9	1001	-	9
10	1010	-	A
11	1011	-	B
12	1100	-	C
13	1101	-	D
14	1110	-	E
15	1111	-	F

Hexadecimal Positional Value

16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}

MSB Hexadecimal Point LSB

Number System Conversion

1. Decimal to binary conversion

An easy method of converting decimal to binary number equivalents is to 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.

Ex.

1. Convert $(11)_{10}$ into binary

- Given No. $(11)_{10}$ in decimal divide it by 2

2	11	1
2	5	1
2	2	0
	1 →	1

LSB

Write from bottom to up

MSB $(1011)_2$

$\therefore \boxed{(11)_{10} = (1011)_2}$

2. Convert $(29)_{10}$ into binary

2	29	1
2	14	0
2	7	1
2	3	1
	1 →	1

LSB

MSB $(11101)_2$

$\boxed{(29)_{10} = (11101)_2}$

3. Convert $(30)_{10}$ into binary

Given Decimal number : $(30)_{10}$
Divide it by '2' until final result is zero or quotient becomes '1'

2	30	0	LSB
2	15	1	
2	7	1	
2	3	1	
	1 →	1	MSB

$(11110)_2$

$\therefore \boxed{(30)_{10} = (11110)_2}$

4. Convert $(50)_{10}$ into binary

Given Decimal number = $(50)_{10}$
Divide it by '2'.

2	50	0	LSB
2	25	1	
2	12	0	
2	6	0	
2	3	1	
	1 →	1	MSB

$(110010)_2$

$\therefore \boxed{(50)_{10} = (110010)_2}$

2. Fractional decimal to binary conversion

Write down the fractional number and to continually multiply-by-2 (two) till the fractional part of product is zero

Ex.

1. Convert $(0.652)_{10}$ into binary

- First write down the fractional part & multiply by '2' till it becomes zero.

· fractional part of given no. is 0.625.

~~0.625 X 2 =~~

Fractional Part	Product	Integer bit	
0.625 X 2	= 1.25	1	MSB
0.25 X 2	= 0.5	0	
0.5 X 2	= 1	1	LSB

∴ Integer bit write from up to bottom i.e → 101

$\therefore \boxed{(0.652)_{10} = (0.101)_2}$

2. Convert $(0.6875)_{10}$ into binary

Fractional Part	Product	Integer bit	
0.6875 X 2	= 1.375	1	MSB
0.375 X 2	= 0.75	0	
0.75 X 2	= 1.5	1	
0.5 X 2	= 1	1	LSB

∴ Integer bit → 1011

$\therefore \boxed{(0.6875)_{10} = (0.1011)_2}$

3. Convert $(3.4375)_{10}$ into binary

Given number 3.4375

Integer part ↙ ↘ fractional part

Divide integer part by '2' and multiply fractional part by '2'

2	3	1
	1	→ 1

↑ LSB ... MSB

$(11)_2$

$0.4375 \times 2 = 0.875 \rightarrow 0$

$0.875 \times 2 = 1.75 \rightarrow 1$

$0.75 \times 2 = 1.5 \rightarrow 1$

$0.5 \times 2 = 1 \rightarrow 1$

$(0.0111)_2$

Then combine integer & fractional part equivalent binary

$(11.0111)_2$

$\therefore \boxed{(3.4375)_{10} = (11.0111)_2}$

4. Convert $(11.6875)_{10}$ into binary

Given number 11.6875

Integer part ↙ ↘ fractional part

2	11	1
2	5	1
2	2	0
	1	→ 1

$(1011)_2$

$0.6875 \times 2 = 1.375 \rightarrow 1$

$0.375 \times 2 = 0.75 \rightarrow 0$

$0.75 \times 2 = 1.5 \rightarrow 1$

$0.5 \times 2 = 1 \rightarrow 1$

$(0.1011)_2$

Add Integer & fractional part equivalent binary

$(1011.1011)_2$

$\therefore \boxed{(11.6875)_{10} = (1011.1011)_2}$

3. Binary to Decimal conversion

Any binary number can be converted to its equivalent decimal number by adding together the weights of various positions in the binary number which contains 1.

Ex.

1. Convert $(1011)_2$ into decimal

Given no. 1 0 1 1

$2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$1\times2^3 \quad 0\times2^2 \quad 1\times2^1 \quad 1\times2^0$

$8 \quad 0 \quad 2 \quad 1$

$8+0+2+1 = 11$

Add all weights

$\boxed{(1011)_2 = (11)_{10}}$

2. Convert $(110110)_2$ into decimal

1 1 0 1 1 0

$1\times2^5 \quad 1\times2^4 \quad 0\times2^3 \quad 1\times2^2 \quad 1\times2^1 \quad 0\times2^0$

$32 \quad 16 \quad 0 \quad 4 \quad 2 \quad 0$

Add all weights together

$32+16+0+4+2+0$

$= 54$

$\therefore \boxed{(110110)_2 = (54)_{10}}$

3. Convert $(0111.11)_2$ into decimal

0 1 1 1 . 1 1

$0\times2^3 \quad 1\times2^2 \quad 1\times2^1 \quad 1\times2^0 \;.\; 1\times2^{-1} \quad 1\times2^{-2}$

$0+4+2+1 + 0.5 + 0.25$

$4+2+1+0.5+0.25$

$= 7.75$

$\therefore \boxed{(0111.11)_2 = (7.75)_{10}}$

4. Convert $(1100.010)_2$ into decimal

Given number $(1100.010)_2$

$$\begin{array}{ccccccc} 1 & 1 & 0 & 0 & . & 0 & 1 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 1\times2^3 & 1\times2^2 & 0\times2^1 & 0\times2^0 & & 0\times2^{-1} & 1\times2^{-2} & 0\times2^{-3} \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 8 & 4 & 0 & 0 & & 0 & 0.25 & 0 \end{array}$$

$8+4+0.25$

$= 12.05$

$\therefore \boxed{(1100.010)_2 = (12.05)_{10}}$

4. Octal to Binary conversion

3-bits binary numbers are written for each octal digit.

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

Ex.

1. Convert $(652)_8$ into binary

To convert given octal no. into binary, write 3-bit binary no. of each digit.

$$\begin{array}{ccc} 6 & 5 & 2 \\ \downarrow & \downarrow & \downarrow \\ 110 & 101 & 010 \end{array}$$

$\boxed{(652)_8 = (110\ 101\ 010)_2}$

2. Convert $(5432.76)_8$ into binary

Given no. $(5432.76)_8$

- write 3-bit equivalent binary no. of each digit

$$\begin{array}{cccccc} 5 & 4 & 3 & 2 & . & 7 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 101 & 100 & 011 & 010 & & 111 & 110 \end{array}$$

$\therefore \boxed{(5432.76)_8 = (101\ 100\ 011\ 010.111\ 110)_2}$

3. Convert $(437.21)_8$ into binary

Given no. $(437.21)_8$ in octal

$$\begin{array}{cccccc} 4 & 3 & 7 & . & 2 & 1 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 100 & 011 & 111 & . & 010 & 001 \end{array}$$

$\therefore \boxed{(437.21)_8 = (100\ 011\ 111 . 010\ 001)_2}$

5. Binary to Octal conversion

- Make 3-bits grouping from last bit.
- Convert each group into decimal numbers.

Ex.

1. Convert $(1011010011)_2$ into octal number.

- Make group of 3-bits from LSB

MSB 1 011 010 011 LSB

∵ As MSB bit only one

∴ To make group of 3 add extra bits (0) to its left side.

$$\begin{array}{cccc} 001 & 011 & 010 & 011 \\ 1 & 3 & 2 & 3 \end{array}$$

$\therefore \boxed{(1011010011)_2 = (1323)_8}$

2. Convert $(1101010)_2$ into octal number.

1101010 → Given no.

Make group of 3 bits from LSB

$$\begin{array}{ccc} 001 & 101 & 010 \\ \downarrow & \downarrow & \downarrow \\ 1 & 5 & 2 \end{array}$$

→ Equivalent decimal value

$\therefore \boxed{(1101010)_2 = (152)_8}$

3. Convert $(1101.11011)_2$ into octal number.

1 101 . 110 11

Make group of 3-bits starting from binary point in left & right direction

∵ To make group of 3 bits of fractional part add extra bits (0) to its right side

$$\begin{array}{ccccc} 001 & 101 & . & 110 & 110 \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 1 & 5 & & 6 & 6 \end{array}$$

$\therefore \boxed{(1101.11011)_2 = (15.66)_8}$

4. Convert $(101010111)_2$ into octal number.

Given no. $(101010111)_2$

Make group of 3-bits

$$\begin{array}{ccc} 101 & 010 & 111 \\ \downarrow & \downarrow & \downarrow \\ 5 & 2 & 7 \end{array}$$

$\therefore \boxed{(101010111)_2 = (527)_8}$

6. Octal to Decimal conversion

An octal number can be converted to its decimal equivalent by multiplying each octal digit by its equivalent position weight.

Ex.

1. Convert $(250)_8$ into decimal

write ~~pos~~ octal positional value of each digit and add together

250

$2 \times 8^2 \quad 5 \times 8^1 \quad 0 \times 8^0$

$128 \quad 40 \quad 0$

$128 + 40 = 168$

$\therefore \boxed{(250)_8 = (168)_{10}}$

2. Convert $(150)_8$ into decimal

→ write octal positional value of each digit and add together

150

$1 \times 8^2 \quad 5 \times 8^1 \quad 0 \times 8^0$

$64 \quad 40 \quad 0$

$64 + 40 = 104$

$\therefore \boxed{(150)_8 = (104)_{10}}$

3. Convert $(37.7)_8$ into decimal

→ write octal positional value of each digit and add together

37.7

$3 \times 8^1 \quad 7 \times 8^0 \quad 7 \times 8^{-1}$

$24 \quad 7 \quad 0.875$

$24 + 7 + 0.875$

$= 31.875$

$\therefore \boxed{(37.7)_8 = (31.875)_{10}}$

4. Convert $(52.3)_8$ into decimal

→ write octal positional value of each digit and add together

52.3

$5 \times 8^1 \quad 2 \times 8^0 \quad 3 \times 8^{-1}$

$40 \quad 2 \quad 0.375$

$= 40 + 2 + 0.375$

$= 42.375$

$\therefore \boxed{(52.3)_8 = (42.375)_{10}}$

9. Decimal to Octal conversion

To convert a decimal number to octal, decimal number is divided by 8 till the quotient becomes zero

Ex.

1. Convert $(6458)_{10}$ to octal

Divide given decimal no. by base of octal i.e '8'.

8	6458	2	LSB
8	807	7	
8	100	4	
8	12	4	
	1	1	MSB

$\therefore \boxed{(6458)_{10} = (14472)_8}$

2. Convert $(255)_{10}$ to octal

8	255	7
8	31	7
	3	3

$(377)_8$

$\therefore \boxed{(255)_{10} = (377)_8}$

3. Convert $(211)_{10}$ to octal

8	211	3	LSB
8	26	2	
	3	3	MSB

$(323)_8$

$\therefore \boxed{(211)_{10} = (323)_8}$

10. Fractional decimal to hexadecimal conversion

Write down the fractional number and to continually multiply-by-16 till the fractional part of product is zero

Ex.

1. Convert $(0.652)_{10}$ into Hexadecimal

→ Write down the fractional part and continually multiply by '16' till fractional part of product is zero.

→ $0.652 \times 16 = 10.432 \rightarrow 10(A)$ MSB

$0.432 \times 16 = 6.912 \rightarrow 6$

$0.912 \times 16 = 14.592 \rightarrow 14(E)$

$0.592 \times 16 = 9.472 \rightarrow 9$ LSB

$\boxed{(0.652)_{10} = (0.A6E9)_{16}}$

2. Convert $(2001.43)_{10}$ into Hexadecimal

2001 . 43 → fractional part

↓

Integer part

First convert integer part by dividing by '16'

16	2001	1
16	125	13 (D)
	7	→ 7

↑ LSB … MSB

7D1

Convert fractional part by multiplying by 16

$0.43 \times 16 = 6.88 \rightarrow 6$

$0.88 \times 16 = 14.08 \rightarrow 14 (E)$

$0.08 \times 16 = 1.28 \rightarrow 1$

$= 0.6E1$

Add integer & fractional part

7D1 . 6E1

$\therefore \boxed{(2001.43)_{10} = (7D1.6E1)_{16}}$

11. Hexadecimal to Binary conversion

4-bits binary numbers are written for each hexadecimal digit.

Ex.

1. Convert $(D283)_{16}$ into binary

- convert each digit into 4-bit equivalent binary number.

D → 1101, 2 → 0010, 8 → 1000, 3 → 0011

$\therefore \boxed{(D283)_{16} = (1101\ 0010\ 1000\ 0011)_2}$

2. Convert $(57EB.AD)_{16}$ into binary

5 → 0101, 7 → 0111, E → 1110, B → 1011, . , A → 1010, D → 1101

$\therefore \boxed{(57EB.AD)_{16} = (0101\ 0111\ 1110\ 1011 . 1010\ 1101)_2}$

12. Binary to Hexadecimal conversion

- Make 4-bits grouping from last bit.
- Convert each group into hexadecimal numbers.

Ex.

1. Convert $(1011010011)_2$ into hexadecimal number.

→ Given no. $(1011010011)_2$

Make group of 4-bits from LSB

10 1101 0011

↓

To make group of 4-bits add extra '0' to leftside

$\therefore$ 0010 1101 0011

0010 → 2, 1101 → D, 0011 → 3

$\therefore \boxed{(1011010011)_2 = (2D3)_{16}}$

2. Convert $(1101010)_2$ into hexadecimal number.

→ 1101010

Add one '0' left side to make group of 4-bits

0110 1010

6 A

$\therefore \boxed{(1101010)_2 = (6A)_{16}}$

3. Convert $(1101.11011)_2$ into hexadecimal number.

→ Make group of 4-bits from ~~hex~~ Binary point in left & Right direction

1101 . 1101 1

To make group of 4 bits add '0' to right side

1101 . 1101 1000

D D 8

$\therefore \boxed{(1101.11011)_2 = (DD8)_{16}}$

4. Convert $(101010111)_2$ into hexadecimal number.

→ Make group of 4-bits from LSB

1 0101 0111

Add extra '0' to left side.

0001 0101 0111

1 5 7

$\therefore \boxed{(101010111)_2 = (157)_{16}}$

13. Octal to Hexadecimal conversion

- Represent the given octal number using equivalent 3 bit binary number.
- Make 4-bits grouping from last bit.
- Convert each group into hexadecimal numbers.

Ex.

1. Convert $(537)_8$ into hexadecimal number.

→ Represent given octal no. into 3-bit binary no.

$(537)_8 = \cancel{100}(101\ 011\ 111)_2$

→ Now make group of 4-bits from LSB

101011111

0001 0101 1111

1 5 F

$\therefore \boxed{(537)_8 = (15F)_{16}}$

2. Convert $(146)_8$ into hexadecimal number.

→ Represent given octal no. into 3-bit binary no.

$(146)_8 = (001\ 100\ 110)_2$

→ Now make group of 4-bits from LSB

0 0110 0110

∴ 0000 0110 0110

0 6 6

$\therefore \boxed{(146)_8 = (66)_{16}}$

14. Hexadecimal to Octal conversion

- Represent the given hexadecimal number using equivalent 4 bit binary number.
- Make 3-bits grouping from last bit.
- Convert each group into octal numbers.

Ex.

1. Convert $(3DB)_{16}$ into octal number.

→ Represent given hexadecimal no. into 4-bit binary no.

$(3DB)_{16} = (0011\ 1101\ 1011)_2$

Now make group of 3-bits from LSB

$\underbrace{001}_{1}\ \underbrace{111}_{7}\ \underbrace{011}_{3}\ \underbrace{011}_{3}$

$\therefore \boxed{(3DB)_{16} = (1733)_8}$

2. Convert $(5CB.12E)_{16}$ into octal number.

$(5CB.12E)_{16}$

0101 1100 1011 0001 0010 1110

$\therefore 0101\,1100\,1011 . 0001\,0010\,1110$

Now make group of 3 bits from binary point in left & right direction

$\underbrace{010}_{2}\ \underbrace{111}_{7}\ \underbrace{001}_{1}\ \underbrace{011}_{3} . \underbrace{000}_{0}\ \underbrace{100}_{4}\ \underbrace{101}_{5}\ \underbrace{110}_{6}$

$\therefore \boxed{(5CB.12E)_{16} = (2713.0456)_8}$

Floating-point number

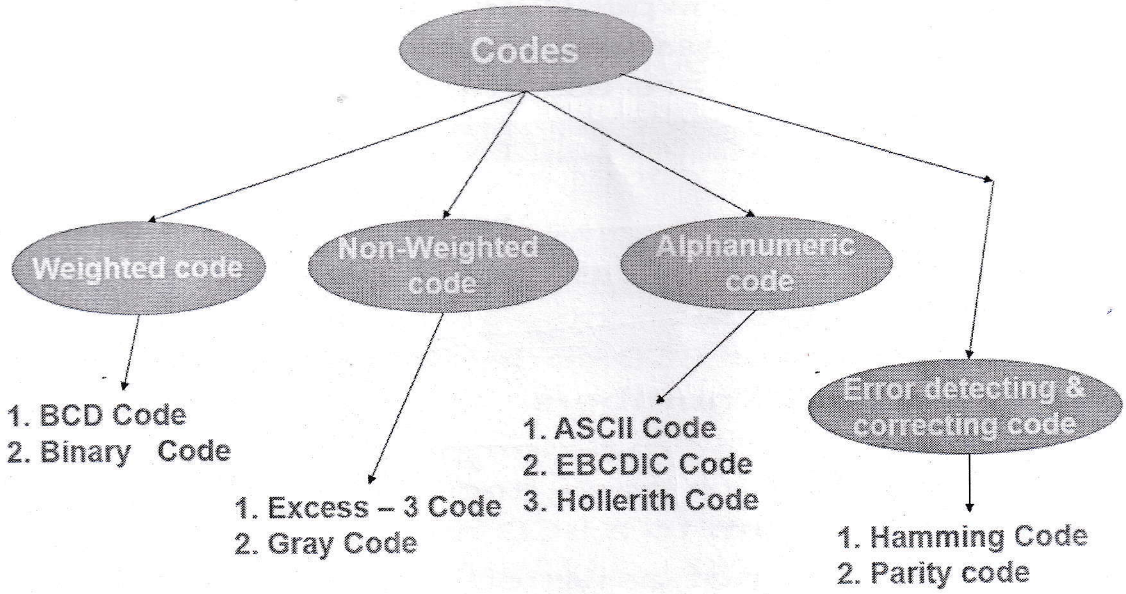
A real number (that is, a number that can contain a fractional part). The following are floating-point numbers:

Ex.

3.0

-111.5

Different Types of Code



BCD or 8421 code:-

It is composed of four bits representing the decimal digits 0 through 9. The 8421 indicates the binary weights of the four bits(2^3,2^2,2^1,2^0).

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

Non-weighted code:-

In non-weighted code, there is no positional weight i.e. each position within the binary number is not assigned a prefixed value. No specific weights are assigned to bit position in non –weighted code.

The non-weighted codes are:-

a) The Gray code

b) The Excess-3 code

The Excess-3 code:-

It is an important BCD code , is a 4 bit code and used with BCD numbers

To convert any decimal numbers into its excess-3 form ,add 3 to each decimal digit and then convert the sum to a BCD number

As weights are not assigned, it is a kind of non weighted codes.

Decimal	BCD 8 4 2 1	Excess-3 BCD + 0011
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0

The Gray code:-

It is non weighted code in which each number differs from previous number by a single bit.

Decimal	Binary	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

ASCII Code

ASCII stands for the "American Standard Code for Information Interchange". It was designed in the early 60's, as a standard character set for computers and electronic devices. ASCII is a 7-bit character set containing 128 characters.

It contains the numbers from 0-9, the upper and lower case English letters from A to Z, and some special characters. The character sets used in modern computers, in HTML, and on the Internet, are all based on ASCII.

The following tables list the 128 ASCII characters and their equivalent number.

	MSBs							
LSBs	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	'	p
0001	SOH	DC_1	1	1	A	Q	a	q
0010	STX	DC_2	"	2	B	R	b	r
0011	ETX	DC_3	#	3	C	S	c	s
0100	EOT	DC_4	S	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	*	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VI	ESC	+	;	K	[	k	{
1100	FF	FS	, <	L	\	/	|	
1101	CR	GS	-	=	M	]	m	}
1110	SQ	RS	, >	N	↑	n	~	
1111	SI	US	/	?	O	←	o	DEL

Binary Arithmetics

1. Binary Addition

There are four rules of binary addition.

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

0011010 + 001100 = 00100110

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

2. Binary Subtraction

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

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

eg.

```
  1110
- 1011
-----------
  0011
```

3. Binary Multiplication

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. There are four rules of binary multiplication.

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

```
     1 0 1 1  (A)
    ×1 0 1 0  (B)
   ---------
     0 0 0 0   ← Corresponds to the rightmost 'zero' in B
+   1 0 1 1   ← Corresponds to the next 'one' in B
+ 0 0 0 0
+1 0 1 1
 ---------------
=1 1 0 1 1 1 0
```

4. Binary Division

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

```
       1 1 1
    ---------
10 | 1 1 1 0
     1 0
    -----
       1 1
       1 0
      -----
         1 0
         1 0
        -----
           0
```

```
        1 0 1
      --------
1 0 1 ) 1 1 0 1 1
      - 1 0 1
      -----
          1 1 1
        - 1 0 1
          -----
            1 0
```

Binary subtraction using 1's complement

1. Find 1's complement of the subtrahend B
2. Add 1's complement of B to A
3. If final carry is 1, then add it to the result(Result is positive)
4. If carry produced in step 2 is 0, then the answer is 1's complement of the result and negative.

Ex.

1. To subtract 10101 from 11101

→ Let A = 11101 & B = 10101

Find 1's complement of B

i.e 01010 → 1's complement of B

Add 1's complement of B to A

```
              A = 1 1 1 0 1
1's comp. of B =  0 1 0 1 0
                 ----------
 Final Carry 1   0 0 1 1 1
```

If final carry is '1' then add it to the result and result is positive

```
  0 0 1 1 1
+         1
 ----------
  0 1 0 0 0
```

$\therefore \boxed{(11101)_2 - (10101)_2 = (1000)_2}$

2. To subtract 1100 from 1000

→ A = 1000 & B = 1100 (subtrahend)

→ Find 1's complement of 'B'

i.e 0011 → 1's comp. of B

→ Add 1's complement of 'B' with A

$$
\begin{array}{lr}
A \rightarrow & 1000 \\
\text{1's comp. of B} \rightarrow & +\,0011 \\
\hline
& 1011 \rightarrow \text{result}
\end{array}
$$

Here final carry is '0', therefore obtain 1's complement of result and it is negative

∴ 1's complement of result '1011' is → 0100

and it is negative i.e $(-0100)_2 \Rightarrow \underline{\underline{-(4)_{10}}}$

$A = (1000)_2 = (8)_{10}$
$B = (1100)_2 = (12)_{10}$
$A - B = (8)_{10} - (12)_{10}$
$= \underline{\underline{-(4)_{10}}}$

3. Perform $(10)_{10}$ - $(05)_{10}$

→ convert given no.s in binary first.

$$
\begin{array}{c|c|c}
2 & 10 & 0 \\
\hline
2 & 5 & 1 \\
\hline
2 & 2 & 0 \\
\hline
 & 1 & 1
\end{array}
\uparrow \quad (10)_{10} = (1010)_2
$$

$$
\begin{array}{c|c|c}
2 & 5 & 1 \\
\hline
2 & 2 & 0 \\
\hline
 & 1 & 1
\end{array}
\uparrow \quad (5)_{10} = (101)_2
$$

→ let $A = (10)_{10} \Rightarrow (1010)_2$

$B = (05)_{10} \Rightarrow (0101)_2$

→ Now find 1's complement of 'B'

i.e 1010 → 1's comp. of 'B'.

→ Add 1's complement of 'B' with 'A'

$$
\begin{array}{lr}
A \rightarrow & 1010 \\
\text{1's comp. of B} \rightarrow & 1010 \\
\hline
\text{final carry } \boxed{1} & 0100 \rightarrow \text{result}
\end{array}
$$

→ As final carry is '1', Add it to the result

$$
\begin{array}{r}
0100 \\
+\quad 1 \\
\hline
0101
\end{array}
\Rightarrow (5)_{10}
$$

$\boxed{(10)_{10} - (05)_{10} = (5)_{10}}$

Binary subtraction using 2's complement

(i) At first, 2's complement of the subtrahend is found.

(ii) Then it is added to the minuend.

(iii) If the final carry over of the sum is 1, it is dropped and the result is positive.

(iv) If there is no carry over, the two's complement of the sum will be the result and it is negative.

Ex.

1. Perform $(12)_{10}$ - $(8)_{10}$

first convert given no.s into binary

2	12	0
2	6	0
2	3	1
	1	1

$(12)_{10} = (1100)_2$

2	8	0
2	4	0
2	2	0
	1	1

$(8)_{10} = (1000)_2$

→ Let $A = (12)_{10} \Rightarrow (1100)_2$

$B = (8)_{10} \Rightarrow (1000)_2$

→ Find 2's complement of 'B'.

1's comp of B ⇒ 0111

Add '1' into it to obtain 2's comp.

$$\begin{array}{r} 0111 \\ +\quad 1 \\ \hline 1000 \end{array} \rightarrow \text{2's compt. of 'B'}$$

→ Add 2's compt. of 'B' with 'A'

A → 1100

2's comt of B → 1000

$\boxed{1}$ 0100 → Result.

final carry (discard it)

→ If final carry is '1', discard it and, result is positive

∴ $(0100)_2 \Rightarrow (4)_{10}$

∴ $\boxed{(12)_{10} - (8)_{10} = (4)_{10}}$

2. To subtract 1011 from 1001

→ Let A = 1001

B = 1011

1's compt of B ⇒ 0100

(Add '1') to obtain 2's compt.

$$\begin{array}{r} 0100 \\ +\quad 1 \\ \hline 0101 \end{array}$$

2's compt. of 'B' is 0101

add it with 'A'.

A → 1001

2's compt of B → 0101

$\boxed{0}$ 1110 → result

final carry

Here final carry is '0', hence obtain 2's complement of result & it is negative

1's complement of result is 0001

$$\begin{array}{r} 0001 \\ +\quad 1 \\ \hline 0010 \end{array}$$

2's compt. of result ← 0010

∴ $-(0010) \Rightarrow -(2)_{10}$//

$(1001)_2 \Rightarrow (9)_{10}$

$(1011)_2 \Rightarrow (11)_{10}$

$(9)_{10} - (11)_{10} = (-2)_{10}$

BCD addition

Add two numbers as same as binary addition

Case 1: If the result is less than or equals to 9 and carry is zero then it is valid BCD.

Case 2: If result is greater than 9 and carry is zero then add 6 in four bit combination.

Case 3: If result is less than or equals to 9 but carry is 1 then add 6 in four bit combination.

1. Add $(3)_{10} + (4)_{10}$ in BCD.

$(3)_{10} \Rightarrow 0011$

$(4)_{10} \Rightarrow 0100$

$$
\begin{array}{rl}
A \rightarrow & 0011 \\
+\ B \rightarrow & 0100 \\
\hline
& (0111)_2 \Rightarrow (7)_{10}
\end{array}
$$

Here result is less than '9' and carry is zero.

∴ As per case 1 it is valid BCD

∴ $\boxed{(3)_{10} + (4)_{10} = (7)_{10}}$

2. Add $(709)_{10} + (698)_{10}$ in BCD.

$A = (709)_{10} \Rightarrow 0111\ 0000\ 1001$

$B = (698)_{10} \Rightarrow 0110\ 1001\ 1000$

$$
\begin{array}{rccc}
+ & 0111 & 0000 & 1001 \\
 & 0110 & 1001 & 1000 \\
\hline
 & 1101 & 1010 & 0001
\end{array}
$$

1101 ↓ Invalid

1010 ↓ greater than '9' it is Not valid

0001 ↓ less than '9' but carry is '1' therefor it is Not valid BCD

In order to make valid BCD add '6' i.e $(0110)_2$ in each invalid digit

$$
\begin{array}{rccc}
 & 1101 & 1010 & 0001 \\
 & 0110 & 0110 & 0110 \\
\hline
\boxed{1} & 0100 & 0000 & 0111
\end{array}
$$

∴ $\underline{1}\ \underline{0100}\ \underline{0000}\ \underline{0111}$
$\ \ \ 1 \quad 4 \quad 0 \quad 7$

∴ $\boxed{709 + 698 = 1407}$