UNIT - I

Number systems are the technique to represent numbers in the computer system architecture, every value that you are saving or getting into/from computer memory has a defined number system.

Computer architecture supports following number systems.

- **Binary number system**
- **Octal number system**
- **Decimal number system**
- **Hexadecimal (hex) number system**

1) Binary Number System

A Binary number system has only two digits that are **0 and 1**. Every number (value) represents with 0 and 1 in this number system. The base of binary number system is 2, because it has only two digits.

2) Octal number system

Octal number system has only eight (8) digits from **0 to 7**. Every number (value) represents with 0,1,2,3,4,5,6 and 7 in this number system. The base of octal number system is 8, because it has only 8 digits.

3) Decimal number system

Decimal number system has only ten (10) digits from **0 to 9**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8 and 9 in this number system. The base of decimal number system is 10, because it has only 10 digits.

4) Hexadecimal number system

A Hexadecimal number system has sixteen (16) alphanumeric values from **0 to 9** and **A to F**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F in this number system. The base of hexadecimal number system is 16, because it has 16 alphanumeric values. Here **A is 10**, **B is 11**, **C is 12**, **D is 13**, **E is 14** and **F is 15**.

Table of the Numbers Systems with Base, Used Digits, Representation, C language representation:

Number system	Base	Used digits	Example	C Language assignment
Binary	2	0,1	$(11110000)_2$	int val=0b11110000;
Octal	8	0,1,2,3,4,5,6,7	$(360)_8$	int val=0360;
Decimal	10	0,1,2,3,4,5,6,7,8,9	$(240)_{10}$	int val=240;
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	$(F0)_{16}$	int val=0xF0;

Number System Conversions

There are three types of conversion:

- **Decimal Number System to Other Base**

  [for example: Decimal Number System to Binary Number System]
- **Other Base to Decimal Number System**

  [for example: Binary Number System to Decimal Number System]
- **Other Base to Other Base**

  [for example: Binary Number System to Hexadecimal Number System]

Decimal Number System to Other Base

To convert Number system from **Decimal Number System** to **Any Other Base** is quite easy; you have to follow just two steps:

A) Divide the Number (Decimal Number) by the base of target base system (in which you want to convert the number: Binary (2), octal (8) and Hexadecimal (16)).

B) Write the remainder from step 1 as a Least Signification Bit (LSB) to Step last as a Most Significant Bit (MSB).

Decimal to Octal Conversion Result

Deci

Decimal to Binary Conversion Result

8	12345	1	LSB
8	1543	7	
8	192	0	
8	24	0	
	3	3	MSB

$(30071)_8$

Decimal Number is : **(12345)**$_{10}$

Binary Number is **(11000000111001)**$_{2}$

2	12345	1	**LSB**
2	6172	0	
2	3086	0	
2	1543	1	
2	771	1	
2	385	1	
2	192	0	
2	96	0	
2	48	0	
2	24	0	
2	12	0	
2	6	0	
2	3	1	
	1	1	**MSB**

Decimal to Hexadecimal Conversion

Result

Example 1

Decimal Number is : **(12345)**$_{16}$

Hexadecimal Number is **(3039)**$_{16}$

16	12345	9	**LSB**
16	771	3	
16	48	0	
8	3	3	**MSB**

Example 2

Decimal Number is : **(725)**$_{10}$

Hexadecimal Number is **(2D5)**$_{16}$

Convert
10, 11, 12, 13, 14, 15
to its equivalent...
A, B, C, D, E, F

16	725	5	5	**LSB**
16	45	13	D	
	2	2	2	**MSB**

Other Base System to Decimal Number Base

To convert Number System from **Any Other Base System** to **Decimal Number System**, you have to follow just three steps:
A) Determine the base value of source Number System (that you want to convert), and also determine the position of digits from LSB (first digit's position – 0, second digit's position – 1 and so on).
B) Multiply each digit with its corresponding multiplication of position value and Base of Source Number System's Base.
C) Add the resulted value in step-B.

Explanation regarding examples:
Below given exams contains the following rows:
A) Row 1 contains the **DIGITs** of number (that is going to be converted).
B) Row 2 contains the **POSITION** of each digit in the number system.
C) Row 3 contains the multiplication: **DIGIT* BASE^POSITION**.
D) Row 4 contains the calculated result of **step C**.
E) And then add each value of **step D**, resulted value is the Decimal Number.

Binary to Decimal Conversion

Binary Number is : **(11000000111001)**$_2$

1	1	0	0	0	0	0	0	1	1	1	0	0	1
13	12	11	10	9	8	7	6	5	4	3	2	1	0
$1x2^{13}$	$1x2^{12}$	$0x2^{11}$	$0x2^{10}$	$0x2^9$	$0x2^8$	$0x2^7$	$0x2^6$	$1x2^5$	$1x2^4$	$1x2^3$	$0x2^2$	$0x2^1$	$1x2^0$
8192	4096	0	0	0	0	0	0	32	16	8	0	0	1

=8192+4096+32+16+8+1
=12345

Octal to Decimal Conversion

Octal Number is : **(30071)**$_8$

3	0	0	7	1
4	3	2	1	0
$3*8^4$	$0*8^3$	$0*8^2$	$7*8^1$	$1*8^0$
12288	0	0	56	1

Hexadecimal to Decimal Conversion Result

Hexadecimal Number is : **(2D5)$_{16}$**

2	D (13)	5
2	1	0
$2*16^2$	$13*16^1$	$5*16^0$
512	208	5

=512+208+5
=725
Decimal Number is: **(725)$_{10}$**

Binary Arithmetic

Binary arithmetic is essential part of all the digital computers and many other digital system.

Binary Addition

It is a key for binary subtraction, multiplication, division. 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

In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

Example − Addition

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
```

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

Example − Subtraction

0011010 - 001100 = 00001110

```
      1 1        borrow
  0 0 1 1 0 1 0  = 26₁₀
 -0 0 0 1 1 0 0  = 12₁₀
 ---------------
  0 0 0 1 1 1 0  = 14₁₀
```

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

Example − Multiplication

Example:

0011010 x 001100 = 100111000

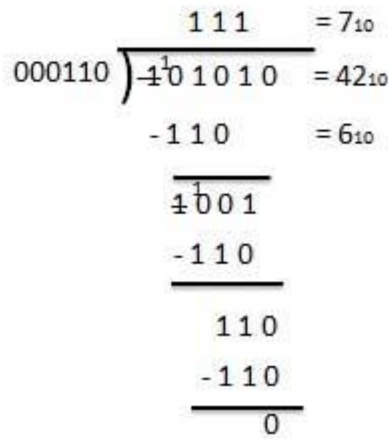
```
          0 0 1 1 0 1 0   = 26₁₀
         x0 0 0 1 1 0 0   = 12₁₀
    ----------------------
          0 0 0 0 0 0 0
        0 0 0 0 0 0 0
      0 0 1 1 0 1 0
    0 0 1 1 0 1 0
    ----------------------
    0 1 0 0 1 1 1 0 0 0   = 312₁₀
```

Binary Division

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

Example − Division

101010 / 000110 = 000111



What are Binary Codes?

By encoding, an explicit group of symbols are used for representing a number or word or letter. Code implies to the explicit group of symbols. Binary code represents the stored and transmitted digital data. Numbers and alphanumeric letters are used for representing the Binary codes.

What are the advantages of Binary Code?

Some of the advantages offered by the binary code are as follows:

- Computer applications mostly use Binary codes.
- Digital communications mostly use Binary codes.
- Digital circuits can be analyzed and designed by using the binary codes.
- Implementation of binary codes is easy as only 0 and 1 are used.

How Binary codes are classified?

Binary codes are categorized into -

- Weighted Codes
- Non-Weighted Codes
- Binary Coded Decimal Code
- Alphanumeric Codes
- Error Detecting Codes
- Error Correcting Codes

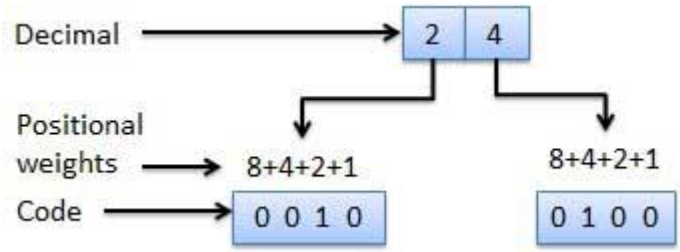
What are Weighted Binary Codes?

When the positional weight balance principle is applied by the binary codes, they are known as Weighted Binary Codes. Explicit weight is represented by each position. A group of four bits represent a decimal digit in this Weighted Binary Code.

8 4 2 1 code

- The weights of this code are 8, 4, 2 and 1.
- This code has all positive weights. So, it is a **positively weighted code**.
- This code is also called as **natural BCD** BinaryCodedDecimal **code**.

Example



2 4 2 1 code

- The weights of this code are 2, 4, 2 and 1.
- This code has all positive weights. So, it is a **positively weighted code**.
- It is an **unnatural BCD** code. Sum of weights of unnatural BCD codes is equal to 9.
- It is a **self-complementing** code. Self-complementing codes provide the 9's complement of a decimal number, just by interchanging 1's and 0's in its equivalent 2421 representation.

Example

Let us find the 2421 equivalent of the decimal number 786. This number has 3 decimal digits 7, 8 and 6. From the table, we can write the 2421 codes of 7, 8 and 6 are 1101, 1110 and 1100 respectively.

Therefore, the 2421 equivalent of the decimal number 786 is **110111101100**.

What are Non-Weighted Binary Codes?

The positional weights are not assigned in Non-weighted Binary codes. Excess – 3 code and Gray code are illustrations of Non-weighted Binary codes.

Excess-3 code

The decimal numbers are expressed by a non-weighted code, known as Excess-3 code or XS-3 code. An example of Excess-3 code is as follows:

Example

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 following table shows the various binary codes for decimal digits 0 to 9.

Decimal Digit	8421 Code	2421 Code	84-2-1 Code	Excess 3 Code
0	0000	0000	0000	0011
1	0001	0001	0111	0100
2	0010	0010	0110	0101
3	0011	0011	0101	0110
4	0100	0100	0100	0111
5	0101	1011	1011	1000
6	0110	1100	1010	1001
7	0111	1101	1001	1010
8	1000	1110	1000	1011

9	1001	1111	1111	1100

Gray Code

In this code, each time when a decimal number is incremented, only one bit is changed and hence is also called as unit distance code. Arithmetic operations cannot use Gray code as it being a cyclic code. The bit positions are not assigned with any specific weights.

Application of Gray code

- Shaft position encoders widely use Gray code.
- The angular position of the shaft is represented by the code word produced by the shaft position encoder.

Decimal	BCD	Gray
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1

What is Binary Coded Decimal (BCD) code?

Under this BCD code, a 4-bit binary number represents each of the decimal digit. BCD enables to represent 16 numbers by using 4 bits but BCD facilitates in using only the first 10 code combinations, while the rest are considered invalid.

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

Advantages of BCD Codes

- It is very similar to decimal system.

- The binary equivalent of decimal numbers from 0 to 9 is to be remembered.

Disadvantages of BCD Codes

- There are different rules under BCD for addition and subtraction.
- The BCD arithmetic is complicated.
- The decimal numbers are represented by using more bits than binary and hence are considered less efficient than binary.

Reflective codes:

A code is reflective when the code is self complementing. In other words, when the code for 9 is the complement the code for 0, 8 for 1, 7 for 2, 6 for 3 and 5 for 4.

2421, Excess-3 code are reflective codes.

Self Complementing Codes*:*

Self-complementing binary codes are those whose members complement on themselves. For a binary code to become a self-complementing code, the following two *conditions* must be satisfied:

1. The complement of a binary number should be obtained from that number by replacing 1's with 0's and 0's with 1's (already stated procedure).
2. The sum of the binary number and its complement should be equal to decimal 9.

What are Alphanumeric codes?

Many more symbols apart from 0 and 1 are required for communication between two computers. These symbols should represent all the 26 alphabets (capital and small), all the numbers from 0 to 9 and punctuation marks.

The numbers and alphabetic characters are represented by alphanumeric codes. Symbols and other instructions used for communicating the information are also represented by the alphanumeric codes. The different alphanumeric codes that are most widely and commonly used for representing data are:

- American Standard Code for Information Interchange (ASCII).
- Extended Binary Coded Decimal Interchange Code (EBCDIC).
- Five bit Baudot Code.

Among the three the most commonly used is ASCII which is a 7-bit code and EBCDIC is used for IBM computers which are huge and is an 8-bit code.