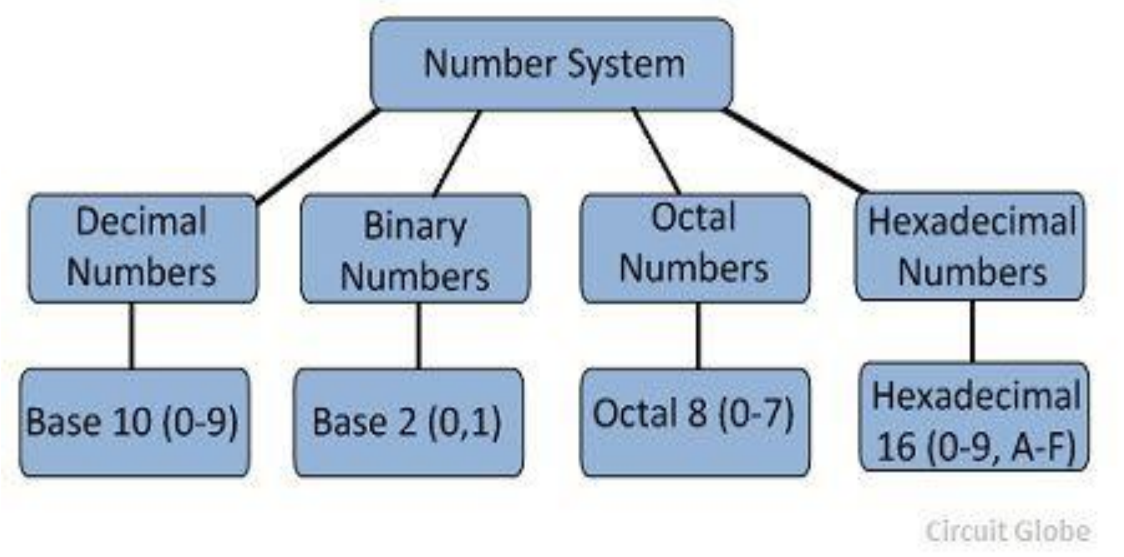
Unit-4:
Number System(Conversion and codes)

Number System and Code

Digital system process binary digits 0 and 1

Base 10 is important for everyday business
Base 2 is important for processing of digital circuit
Base 8 and Base 16 provide convenient shortened representation for multibit number in a digital system



Binary	*Decimal*	*Octal*	*3-Bit String*	*Hexadecimal*	*4-Bit String*
0	0	0	000	0	0000
1	1	1	001	1	0001
10	2	2	010	2	0010
11	3	3	011	3	0011
100	4	4	100	4	0100
101	5	5	101	5	0101
110	6	6	110	6	0110
111	7	7	111	7	0111
1000	8	10	—	8	1000
1001	9	11	—	9	1001
1010	10	12	—	A	1010
1011	11	13	—	B	1011
1100	12	14	—	C	1100
1101	13	15	—	D	1101
1110	14	16	—	E	1110
1111	15	17	—	F	1111

Number conversion

Methods or techniques used to convert numbers from one base to another

<u>Decimal to Other</u>

Step 1 – Divide the decimal number to be converted by the value of the other base.

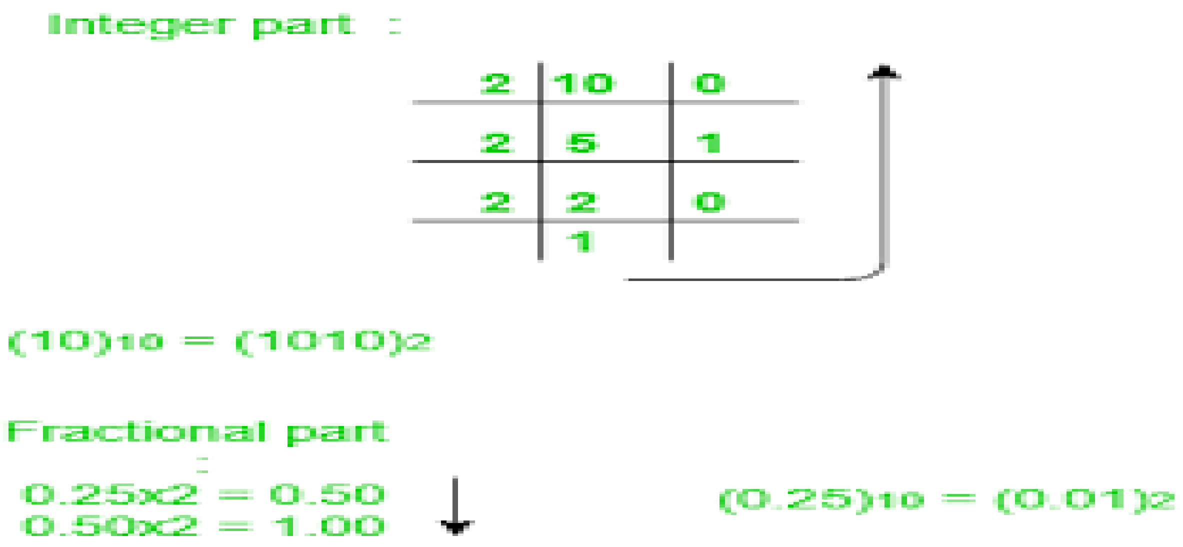
Step 2 – Get the remainder from Step 1 as (least significant digit) of new base number

Step 3 – Divide the quotient of the previous divide by the new base.

Step 4 – Record the remainder from Step 3 as the next digit

Repeat Steps 3 and 4, getting remainders until the quotient becomes zero

The last remainder thus obtained will be the Most Significant bit(MSB) of the new base number.



Decimal to Hexadecimal

$$(3509)_{10} = (DB5)_{16}$$

Divisor	Quotient	Remainder
16	3509	5
16	219	11
16	13	13
	0	

Decimal to Octal

$$(569)_{10} = (1071)_8$$

		Remainders
8	569	
8	71	1
8	8	7
8	1	0
	0	1

Read in reverse order

$0.342_{10} = ?_8$

$0.342 \times 8 = 2.736 \quad (.2_8)$

$0.736 \times 8 = 5.888 \quad (.25_8)$

$0.888 \times 8 = 7.104 \quad (.257_8)$

$0.104 \times 8 = 0.832 \quad (.2570_8)$

$0.342_{10} \approx 0.2570_8$ it's an approximation

Other Base System to Decimal System

Step 1 – Determine positional value of each digit

Step 2 – Multiply the obtained position values by the digits in the corresponding columns.

Step 3 – Sum the products calculated in Step 2.

32 16 8 4 2 1

1 0 0 1 1 1 =32 + 4 +2 +1 = (39)10

Binary Number – 11101_2

Calculating Decimal Equivalent –

Step	Binary Number	Decimal Number
Step 1	11101_2	$((1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0))_{10}$
Step 2	11101_2	$(16 + 8 + 4 + 0 + 1)_{10}$
Step 3	11101_2	29_{10}

Octal to Decimal

$$\begin{aligned}(2754)_8 &= (2\times 8^3) + (7\times 8^2) + (5\times 8^1) + (4\times 8^4)\\ &= 1024 + 448 + 40 + 4\\ &= 1516_{10}\end{aligned}$$

Hexadecimal to Decimal

$$\begin{aligned}(54.D2)^{16} &= (5\times 16^1) + (4\times 16^0) + (13\times 16^{-1}) + (2\times 16^{-2})\\ &= 80 + 4 + 0.8125 + 0.0078125\\ &= 84.8203125_{10}\end{aligned}$$

Binary to Octal

Step 1 − Divide the binary digits into groups of three (starting from the right).
Step 2 − Convert each group of three binary digits to one octal digit.

Binary Number − 10101_2

Calculating Octal Equivalent −

Step	Binary Number	Octal Number
Step 1	10101_2	010 101
Step 2	10101_2	2_8 5_8
Step 3	10101_2	25_8

Octal to Binary

Step 1 − Convert each octal digit to a 3 digit binary number.
Step 2 − Combine all the resulting binary groups (of 3 digits each) into a single binary number

Octal Number − 25_8

Calculating Binary Equivalent −

Step	Octal Number	Binary Number
Step 1	25_8	010_2 101_2
Step 2	25_8	010101_2

Binary to Hexadecimal

Step 1 − Divide the binary digits into groups of four (starting from the right).
Step 2 − Convert each group of four binary digits to one hexadecimal symbol.

Binary Number − 10101_2

Calculating hexadecimal Equivalent −

Step	Binary Number	Hexadecimal Number
Step 1	10101_2	0001 0101
Step 2	10101_2	15_{16}

Hexadecimal to Binary

Step 1 − Convert each hexadecimal digit to a 4 digit binary number.
Step 2 − Combine all the resulting binary groups (4 digits each) into a single binary number.

Hexadecimal Number − 15_{16}

Calculating Binary Equivalent −

Step	Hexadecimal Number	Binary Number
Step 1	15_{16}	0001_2 0101_2
Step 2	15_{16}	00010101_2

$$8421$$

$$(1101)_2 = 8+4+1 = (13)_{10}$$

$$(0111)_2 = 7_{10}$$

$$(1010)_2 = (10)_{10} = A_{16}$$

$$(10101)_2$$

Practice Question

$(4021.5)_5 = (\underline{\hspace{4cm}})_{10}$

$(B65F)_{16} = (\underline{\hspace{4cm}})_{10}$

$(630.4)_8 = (\underline{\hspace{4cm}})_{10}$

$(0.6875)_{10} = (\underline{\hspace{5cm}})_2$

$(0.513)_{10} = (\underline{\hspace{5cm}})_8$

$(306.D)_{16} = (\underline{\hspace{4cm}})_8$

$(10110001101011.11110010)_2 = (\underline{\hspace{4cm}})_{16}$

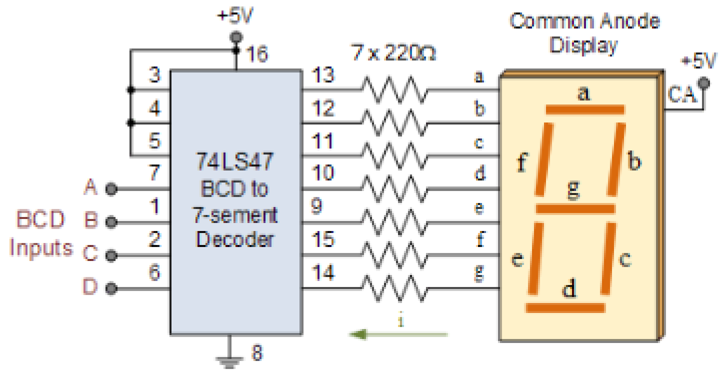
$(108)_{10} = (\underline{\hspace{2cm}})_{16}$

$(\underline{\hspace{4cm}})_2 = (\underline{\hspace{3cm}})_{10} = (576)_8 = (\underline{\hspace{2.5cm}})_{16}$

BCD (Binary Coded Decimal)

- Do not get confused, *BCD* is not the same as hexadecimal.
- Main advantage of BCD - Easy conversion between decimal and binary form.
- However, disadvantage - BCD code is wasteful as the states between 1010 (decimal 10), and 1111 (decimal 15) are not used.

- Nevertheless, BCD has many important applications especially using digital displays.



Convert each of the following decimal numbers to BCD:

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

(a) $\begin{array}{cc} 3 & 5 \\ \downarrow & \downarrow \\ \overbrace{\mathbf{0011}} & \overbrace{\mathbf{0101}} \end{array}$

(b) $\begin{array}{cc} 9 & 8 \\ \downarrow & \downarrow \\ \overbrace{\mathbf{1001}} & \overbrace{\mathbf{1000}} \end{array}$

(c) $\begin{array}{ccc} 1 & 7 & 0 \\ \downarrow & \downarrow & \downarrow \\ \overbrace{\mathbf{0001}} & \overbrace{\mathbf{0111}} & \overbrace{\mathbf{0000}} \end{array}$

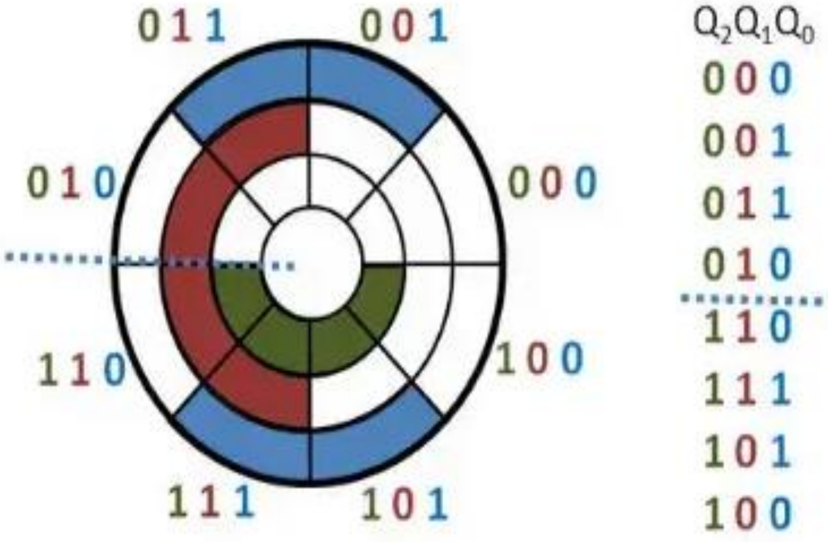
(d) $\begin{array}{cccc} 2 & 4 & 6 & 9 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \overbrace{\mathbf{0010}} & \overbrace{\mathbf{0100}} & \overbrace{\mathbf{0110}} & \overbrace{\mathbf{1001}} \end{array}$

Gray Code (Cyclic Code, or Reflected Binary Code)

- Ordering of the binary number system such that each incremental value can only differ by one bit.
- Gray code is not weighted that means it does not depends on positional value of digit.
- it is not suitable for arithmetic operations.

Some other applications of gray code:

- Boolean circuit minimization.
- Error correction in communication system



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:

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

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:

1	1	0	1	1	Gray
↓	↓	↓	↓	↓	
1	0	0	1	0	Binary

The binary number is 10010.

Binary Addition

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (which is 0 carry 1)

0011010 + 001100 = 00100110

1 1	carry
0 0 1 1 0 1 0	$= 26_{10}$
+ 0 0 0 1 1 0 0	$= 12_{10}$
0 1 0 0 1 1 0	$= 38_{10}$

$$(10010)_2 + (1001)_2 = ?$$

Binary Subtraction

A - B	Subtract	Borrow
0 - 0	0	0
1 - 0	1	0
1 - 1	0	0
0 - 1	1	1

0011010 - 001100 = 00001110

$$
\begin{array}{rl}
1\;\,1 & \text{borrow} \\
0\,0\,\cancel{1}\,\cancel{1}\,0\,1\,0 & = 26_{10} \\
-\,0\,0\,0\,1\,1\,0\,0 & = 12_{10} \\
\hline
0\,0\,0\,1\,1\,1\,0 & = 14_{10}
\end{array}
$$

$$(1100)_2 - (1010)_2 = ?$$