Mahavir Polytechnic, Nashik

Department of Artificial Intelligence and Machine Learning

Year: SY **Subject: DTE (313303)**

UNIT 1: Number Systems **Marks: 08**

Course Outcome 1: Apply number system and codes concept to interprete working of digital systems.

Syllabus:

1.1 Number Systems: Types of Number Systems (Binary, Octal, Decimal, Hexadecimal), conversion of number Systems

1.2 Binary Arithematic: Addition, Subtraction, Multiplication and Division

1.3 Subtraction using 1's and 2's complement method

1.4 Codes: BCD, Gray code, Excess-3 and ASCII code, Code conversions, Applications of codes.

1.5 BCD Arithematic: BCD Addition, Subtraction using 9's and 10's complement

---

Introduction

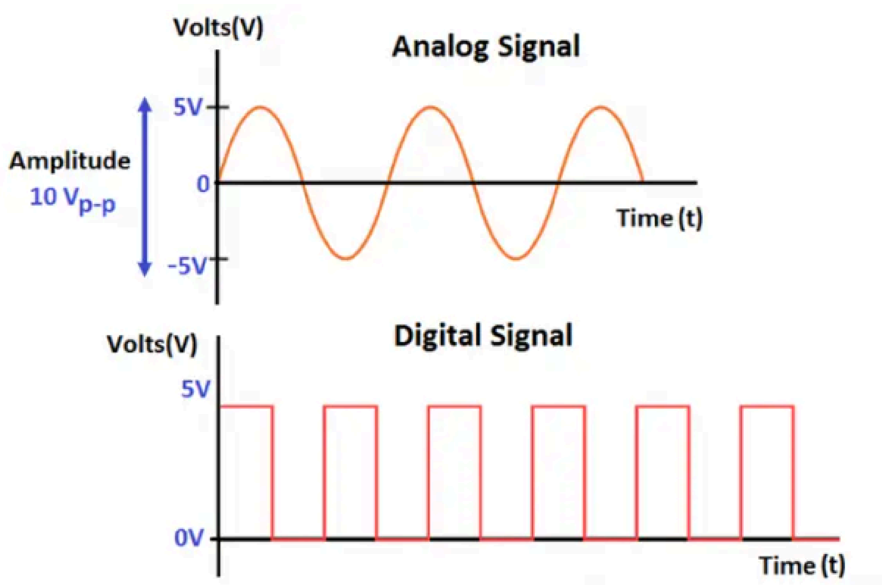
Digital electronics, digital circuits, and digital technology are electronics that are operated on digital signals. Digital techniques are much easier for getting the electronic device. These devices are used to switch into one of the known states apart from reproducing a continuous range of values. Digital circuits are made from a large collection of logic gates and a simple electronic representation of the Boolean logic function.

What are Analog Signals?

The analog signals were used in many systems to produce signals to carry information. These signals are continuous in both values and time. The use of analog signals has been declined with the arrival of digital signals. In short, to understand analog signals – all signals that are natural or come naturally are analog signals.

What are Digital Signals?

Unlike analog signals, digital signals are not continuous, but signals are discrete in value and time. These signals are represented by binary numbers and consist of different voltage values.



Binary Information Representation and Groups

Bit: A bit is the smallest unit of data measurement, representing a single binary value of either 0 or 1.

Nibble: A group of 4 bit

Example: 1010, 1100

Byte: A group of 8 bit

Example: 1010 0010, 1100 1111

Word: A group of 16 bit

Example: 1010 0010 1100 1111

Table 1: Binary Information Group Representations and Terms

Number of Bits	Common Representation Terms
1	Bit / Digit / Flag
4	Nybble / Nibble
8	Byte / Octet / Character
16	Double Byte / Word
32	Double Word / Long Word
64	Very Long Word

1.1 Number Systems:

Number systems are mathematical systems for representing numbers. The most common number systems include decimal, binary, octal, and hexadecimal. Each system has a base that defines the number of unique digits it uses and how it represents numbers.

1. Decimal Number System (Base-10)

Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base: 10

Usage: Commonly used in everyday life.

Example: 345 (Decimal)

2. Binary Number System (Base-2)

Digits: 0, 1

Base: 2

Usage: Used in computers and digital systems.

Example: 1011 (Binary)

3. Octal Number System (Base-8)

Digits: 0, 1, 2, 3, 4, 5, 6, 7

Base: 8

Usage: Sometimes used in computing.

Example: 345 (Octal)

4. Hexadecimal Number System (Base-16)

Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Base: 16

Usage: Used in computing to represent binary data more compactly.

Example: 1A3 (Hexadecimal)

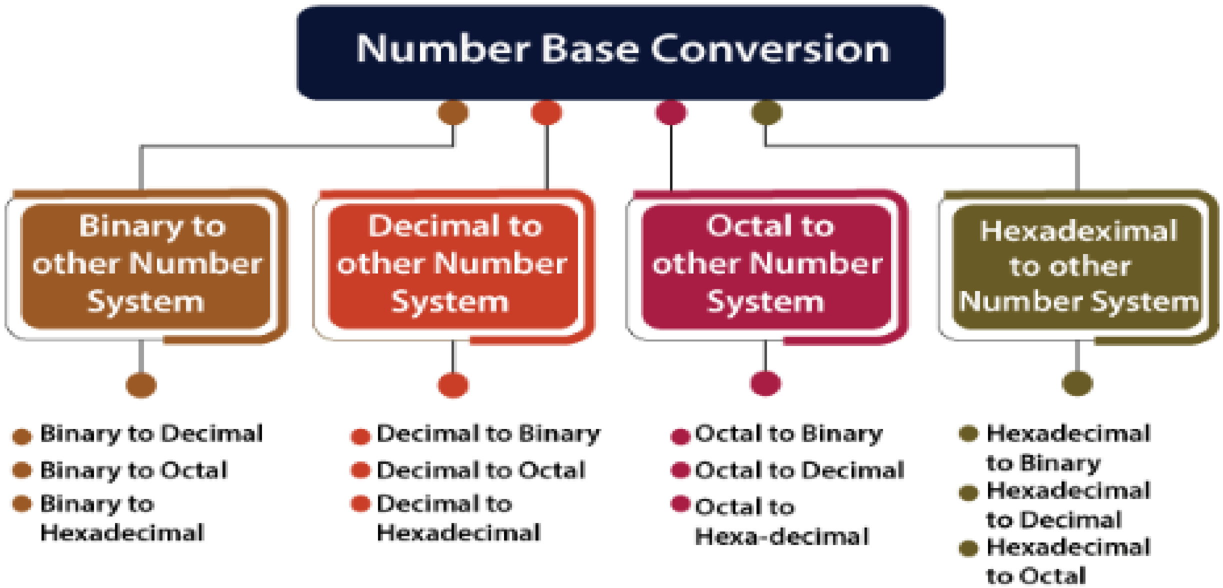
- **Relation between number systems:**

Decimal Base-10	Binary Base-2	Octal Base-8	Hexa Decimal Base-16
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

1.1.1 Conversion of number Systems:

In the previous section, we learned different types of number systems, including binary, decimal, octal, and hexadecimal. Now, we'll learn how to convert a number from one system to another.

Since we have four types of number systems, each can be converted into any of the other three. This means we can make the following conversions:



- **Decimal to other Number System:**

The decimal number can be an integer or floating-point integer. When the decimal number is a floating-point integer, then we convert both part (integer and fractional) of the decimal number in the isolated form (individually).

1. Decimal to Binary:

1. Divide the number by 2.
2. Write down the remainder.
3. Repeat with the quotient until the quotient is 0.
4. The binary representation is the remainders read in reverse.

Example 1: $(152.25)_{10}$

Divide the number 152 and its successive quotients with base 2.

Operation	Quotient	Remainder
152/2	76	0 (LSB)
76/2	38	0
38/2	19	0
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1(MSB)

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

Now, perform the multiplication of 0.27 and successive fraction with base 2

Operation	Result	carry
0.25×2	0.50	0
0.50×2	0	1

$(0.25)_{10}$= $(.01)_2$

2. Decimal to Octal:

1. Divide the number by 8.
2. Write down the remainder.
3. Repeat with the quotient until the quotient is 0.
4. The octal representation is the remainders read in reverse.

Example: Convert 83 to octal

Answer: Octal - 123

Example 1: $(152.25)_{10}$

Divide the number 152 and its successive quotients with base 8.

Operation	Quotient	Remainder
152/8	19	0
19/8	2	3
2/8	0	2

$(152)_{10}$= $(230)_8$

Now perform the multiplication of 0.25 and successive fraction with base 8.

Operation	Result	carry
0.25×8	0	2

$(0.25)_{10}=(2)_8$

So, the octal number of the decimal number 152.25 is 230.2

3. Decimal to Hexadecimal:

1. Divide the number by 16.
2. Write down the remainder.
3. Repeat with the quotient until the quotient is 0.
4. The hexadecimal representation is the remainders read in reverse.

Example: Convert 255 to hexadecimal

Answer: Hexadecimal - FF

Example 1: $(152.25)_{10}$

Divide the number 152 and its successive quotients with base 8.

Operation	Quotient	Remainder
152/16	9	8
9/16	0	9

$(152)_{10}=(98)_{16}$

Now perform the multiplication of 0.25 and successive fraction with base 16.

Operation	Result	carry
0.25×16	0	4

$(0.25)_{10}=(4)_{16}$

So, the hexadecimal number of the decimal number 152.25 is 230.4.

- **<u>Binary to other Number Systems</u>**

There are three conversions possible for binary number, i.e., binary to decimal, binary to octal, and binary to hexadecimal. The conversion process of a binary number to decimal differs from the remaining others

1. Binary to Decimal:

1. Multiply each bit by 2 raised to the position of the bit, starting from 0 on the right.

2. Sum the results.

Example: Convert 1101 to decimal

Answer: Decimal - 13

Example 1: $(10110.001)_2$

We multiplied each bit of $(10110.001)_2$ with its respective positional weight, and last we add the products of all the bits with its weight.

$(10110.001)_2$= $(1\times2^4)+(0\times2^3)+(1\times2^2)+(1\times2^1)+(0\times2^0)+ (0\times2^{-1})+(0\times2^{-2})+(1\times2^{-3})$

$(10110.001)_2$= $(1\times16)+(0\times8)+(1\times4)+(1\times2)+(0\times1)+(0\times1/2)+(0\times1/4)+(1\times1/8)$

$(10110.001)_2$=16+0+4+2+0+0+0+0.125

$(10110.001)_2=(22.125)_{10}$

2. Binary to Octal:

1. Group the binary digits into sets of three, starting from the right. Add leading zeros if necessary.
2. Convert each set to its octal equivalent.

Example: Convert 110101 to octal

Answer: Octal - 65

Example 1: $(111110101011.0011)_2$

1. Firstly, we make pairs of three bits on both sides of the binary point.

111 110 101 011.001 1

On the right side of the binary point, the last pair has only one bit. To make it a complete pair of three bits, we added two zeros on the extreme side.

111 110 101 011.001 100

2. Then, we wrote the octal digits, which correspond to each pair.

$(111110101011.0011)_2= (7653.14)_8$

3. Binary to Hexadecimal:

1. Group the binary digits into sets of four, starting from the right. Add leading zeros if necessary.
2. Convert each set to its hexadecimal equivalent.

Example: Convert 11010110 to hexadecimal

Answer: Hexadecimal - D6

Example 1: $(10110101011.0011)_2$

1. Firstly, we make pairs of four bits on both sides of the binary point.

111 1010 1011.0011

On the left side of the binary point, the first pair has three bits. To make it a complete pair of four bits, add one zero on the extreme side.

0111 1010 1011.0011

2. Then, we write the hexadecimal digits, which correspond to each pair.

0111 1010 1011 . 0011

7 A B 3

$(011110101011.0011)_2 = (7AB.3)_{16}$

- **Octal to other Number System**

Like binary and decimal, the octal number can also be converted into other number systems. The process of converting octal to decimal differs from the remaining one

1. Octal to Decimal Conversion

The process of converting octal to decimal is the same as binary to decimal. The process starts from multiplying the digits of octal numbers with its corresponding positional weights. And lastly, we add all those products.

Example 1: $(152.25)_8$

We multiply each digit of 152.25 with its respective positional weight, and last we add the products of all the bits with its weight.

$(152.25)_8=(1\times8^2)+(5\times8^1) + (2\times8^0)+(2\times8^{-1})+(5\times8^{-2})$

$(152.25)_8=64+40+2 + (2\times1/8)+(5\times1/64)$

$(152.25)_8=64+40+2 + 0.25+0.078125$

$(152.25)_8=106.328125$

So, the decimal number of the octal number 152.25 is **106.328125**

2. Octal to Binary Conversion

The process of converting octal to binary is the reverse process of binary to octal. We write the three bits binary code of each octal number digit.

Example 1: $(152.25)_8$

We write the three-bit binary digit for 1, 5, 2, and 5.

$(152.25)_8$= $(001\ \ 101\ \ 010.010\ 101)_2$

So, the binary number of the octal number 152.25 is **$(001101010.010101)_2$**

3. Octal to hexadecimal conversion

For converting octal to hexadecimal, there are two steps required to perform, which are as follows:

1. In the first step, we will find the binary equivalent of number 25.
2. Next, we have to make the pairs of four bits on both sides of the binary point.
   If there will be one, two, or three bits left in a pair of four bits pair, we add the required number of zeros on extreme sides and write the hexadecimal digits corresponding to each pair.

Example 1: $(152.25)_8$

Step 1:

We write the three-bit binary digit for 1, 5, 2, and 5.

$(152.25)_8$= $(001\ \ 101\ \ 010.\ 010\ 101)_2$

So, the binary number of the octal number 152.25 is **$(001101010.010101)_2$**

Step 2:

1. Now, we make pairs of four bits on both sides of the binary point.

0 0110 1010. 0101 01

On the left side of the binary point, the first pair has only one digit, and on the right side, the last pair has only two-digit. To make them complete pairs of four bits, add zeros on extreme sides.

0000 0110 1010. 0101 0100

0 6 A . 5 4

2. Now, we write the hexadecimal digits, which correspond to each pair.

$(0000\ \ 0110\ \ 1010.\ 0101\ \ 0100)_2 = (6A.54)_{16}$

$(152.25)_8 = (6A.54)_{16}$

- **Hexa-decimal to other Number System**

Like binary, decimal, and octal, hexadecimal numbers can also be converted into other number systems. The process of converting hexadecimal to decimal differs from the remaining one.

1. Hexa-decimal to Decimal Conversion

The process of converting hexadecimal to decimal is the same as binary to decimal. The process starts from multiplying the digits of hexadecimal numbers with its corresponding positional weights. And lastly, we add all those products.

Example 1: $(152A.25)_{16}$

Step 1:

We multiply each digit of **152A.25** with its respective positional weight, and last we add the products of all the bits with its weight.

$(152A.25)_{16}=(1\times16^3)+(5\times16^2)+(2\times16^1) \quad + \quad (A\times16^0)+(2\times16^{-1})+(5\times16^{-2})$

$(152A.25)_{16}=(1\times4096)+(5\times256)+(2\times16) \quad + \quad (10\times1)+(2\times16^{-1})+(5\times16^{-2})$

$(152A.25)_{16}=4096+1280+32+10 \quad + \quad (2\times1/16)+(5\times1/256)$

$(152A.25)_{16}=5418 \quad + \quad 0.125+0.125$

$(152A.25)_{16}=5418.14453125$

So, the decimal number of the hexadecimal number 152A.25 is **5418.14453125**

2. Hexadecimal to Binary Conversion

The process of converting hexadecimal to binary is the reverse process of binary to hexadecimal. We write the four bits binary code of each hexadecimal number digit.

Example 1: $(152A.25)_{16}$

We write the four-bit binary digit for 1, 5, A, 2, and 5.

$(152A.25)_{16}= (0001\ 0101\ 0010\ 1010.0010\ 0101)_2$

So, the binary number of the hexadecimal number 152.25 is **$(1010100101010.00100101)_2$**

3. Hexadecimal to Octal Conversion

For converting hexadecimal to octal, there are two steps required to perform, which are as follows:

1. In the first step, we will find the binary equivalent of the hexadecimal number.
2. Next, we have to make the pairs of three bits on both sides of the binary point.
   If there will be one or two bits left in a pair of three bits pair, we add the required number of zeros on extreme sides and write the octal digits corresponding to each pair.

Example 1: $(152A.25)_{16}$

Step 1:

We write the four-bit binary digit for 1, 5, 2, A, and 5.

$(152A.25)_{16}$= $(0001\ 0101\ 0010\ 1010.0010\ 0101)_2$

So, the binary number of hexadecimal number 152A.25 is **$(0011010101010.010101)_2$**

Step 2:

3. Then, we make pairs of three bits on both sides of the binary point.

001 010 100 101 010.001 001 010

4. Then, we write the octal digit, which corresponds to each pair.

$(001010100101010.001001010)_2$= $(12452.112)_8$

So, the octal number of the hexadecimal number 152A.25 is **12452.112**

1.2 Binary Arithmetic: (Addition, Subtraction, Multiplication and Division)

Binary arithmetic involves performing arithmetic operations (addition, subtraction, multiplication, and division) on binary numbers.

1. Binary Addition

Binary addition is similar to decimal addition but follows binary rules where each digit is either 0 or 1.

Rules for Binary Addition:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 1 0 (which is 0 with a carry of 1)

1 + 1 + 1 = 1 1 (which is 1 with a carry of 1)

Example: Add 1011 and 1101

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

2. Binary Subtraction

Binary subtraction is similar to decimal subtraction but follows binary rules. When you subtract 1 from 0, you need to borrow from the next higher bit.

Rules for Binary Subtraction:

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 - 1 = 1 (borrow 1 from the next higher bit)

Example: Subtract 1011 from 1101

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

3. Binary Multiplication

Binary multiplication is similar to decimal multiplication and involves adding shifted versions of the original number.

Rules for Binary Multiplication:

0 * 0 = 0

0 * 1 = 0

1 * 0 = 0

1 * 1 = 1

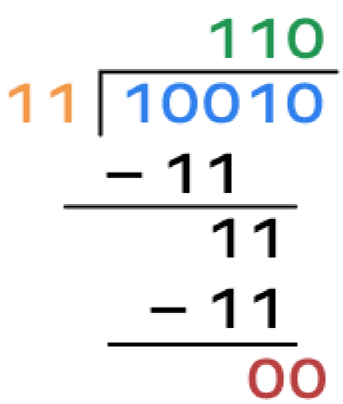
Example: Multiply 101 by 11

```
   101
  X 11
----------
   101
  1010
----------
  1111
```

4. Binary Division

Binary division is similar to decimal division. It involves repeatedly subtracting the divisor from the dividend and recording the quotient and remainder.

Example: Divide 10010 by 11



- **1's complement:**

The one's complement of a binary number is a method of representing negative numbers in binary form by inverting all the bits in the number (changing 0s to 1s and 1s to 0s). This method can also be used to perform binary subtraction.

Finding One's Complement

To find the one's complement of a binary number:

Change each 0 to a 1.

Change each 1 to a 0.

Example: Find the one's complement of 1010

Original: 1010

One's Complement: 0101

- **2's complement**

Two's complement is a widely used method for representing signed integers in binary and performing binary subtraction. It simplifies binary arithmetic by allowing subtraction to be performed as addition

Finding Two's Complement

To find the two's complement of a binary number:

Find the one's complement of the number (invert all bits).

Add 1 to the least significant bit (LSB) of the one's complement.

Example: Find the two's complement of 1010

Find the one's complement of 1010:

1010 -> 0101

Add 1 to the one's complement:

```
0101
+   1
---------
 0110
```

1.3 Subtraction using 1's and 2's complement method

- **Subtraction using 1's complement**

These are the following steps to subtract two binary numbers using 1's complement

1. In the first step, find the 1's complement of the subtrahend.
2. Next, add the complement number with the minuend.
3. If got a carry, add the carry to its LSB. Else take 1's complement of the result which will be negative

<u>**Note:**</u> The subtrahend value always get subtracted from minuend.

Example 1: 10101 - 00111

We take 1's complement of subtrahend 00111, which comes out 11000. Now, sum them. So,
10101+11000 =1 01101.
In the above result, we get the carry bit 1, so add this to the LSB of a given result, i.e., 01101+1=01110, which is the answer.

Example 2: 10101 - 10111
We take 1's complement of subtrahend 10111, which comes out 01000. Now, add both of the numbers. So,
10101+01000 =11101.
In the above result, we didn't get the carry bit. So calculate the 1's complement of the result, i.e., 00010, which is the negative number and the final answer.

- **Subtraction using 2's complement**

These are the following steps to subtract two binary numbers using 2's complement

1. In the first step, find the 2's complement of the subtrahend.
2. Add the complement number with the minuend.
3. If we get the carry by adding both the numbers, then we discard this carry and the result is positive else take 2's complement of the result which will be negative.

Example 1: 10101 - 00111
We take 2's complement of subtrahend 00111, which is 11001. Now, sum them. So,
10101+11001 =1 01110.
In the above result, we get the carry bit 1. So we discard this carry bit and remaining is the final result and a positive number.

Example 2: 10101 - 10111
We take 2's complement of subtrahend 10111, which comes out 01001. Now, we add both of the numbers. So,
10101+01001 =11110.
In the above result, we didn't get the carry bit. So calculate the 2's complement of the result, i.e., 00010. It is the negative number and the final answer.

1.4 **Codes:**

BCD, Gray code, Excess-3 and ASCII code, Code conversions, Applications of codes

In digital systems, various codes are used to represent data, control devices, and facilitate communication. Here are some common codes:

1.4.1 BCD (Binary-Coded Decimal)

BCD is a binary-encoded representation of decimal numbers where each decimal digit is represented by a fixed number of binary digits, usually four. This allows for easy conversion between binary and decimal systems.

Example: Representing 27 in BCD

Decimal 2 = 0010

Decimal 7 = 0111

So, the BCD representation of 27 is: 0010 0111

BCD Encoding Table:

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

1.4.2 Gray Code

Gray code is a binary numeral system where two successive values differ in only one bit. This minimizes errors in digital systems during transitions between values.

Example: Gray Code for Decimal 0 to 3

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010

Example: Convert binary 1010 to Gray code

MSB: 1

Next bit: 1 XOR 0 = 1

Next bit: 0 XOR 1 = 1

Last bit: 1 XOR 0 = 1

Gray code for 1010 is: 1111

1.4.3 Excess-3 Code

Excess-3 (XS-3) is a self-complementary code that is a variation of BCD where each decimal digit is represented by its BCD equivalent plus a bias of 3 (or 0011 in binary).

Example: Representing 2 in Excess-3

Decimal 2 = BCD 0010

Excess-3 = BCD + 0011 = 0101

So, the Excess-3 representation of 2 is: 0101

3.10 Convert the following numbers to Excess-3 code.
(a) 87 **(b) 159**

Solution:

(a)

8	7
+3	+3
11	10
↓	↓
1011	1010

(b)

1	5	9
+3	+3	+3
4	8	12
↓	↓	↓
0100	1000	1100

Excess-3 Encoding Table:

Decimal	BCD	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

1.4.4 ASCII Code

ASCII (American Standard Code for Information Interchange) is a character encoding standard used for text in computers and communication devices. Each character is represented by a 7-bit binary number (extended ASCII uses 8 bits).

Example: ASCII Code for Characters

Character	ASCII (Decimal)	ASCII (Binary)
A	65	01000001
B	66	01000010
C	67	01000011
a	97	01100001
b	98	01100010
c	99	01100011

- **Code Conversion:**

1. BCD to Decimal Conversion

Binary-Coded Decimal (BCD) represents each decimal digit as a separate 4-bit binary number. To convert BCD to decimal:

1. Group the BCD digits into 4-bit segments.
2. Convert each 4-bit segment to its decimal equivalent.
3. Combine these decimal values to get the final decimal number.

Example: Convert BCD 0010 0111 to decimal

0010 (BCD) = 2 (Decimal)

0111 (BCD) = 7 (Decimal)

So, BCD 0010 0111 = Decimal 27.

2. Decimal to BCD Conversion

To convert a decimal number to BCD:

1. Separate the decimal number into individual digits.
2. Convert each digit to its 4-bit BCD representation.
3. Combine these BCD values to get the final BCD number.

Example: Convert Decimal 45 to BCD

Decimal 4 = 0100 (BCD)

Decimal 5 = 0101 (BCD)

So, Decimal 45 = BCD 0100 0101.

3. Binary to Gray Code Conversion

To convert binary to Gray code:

1. Keep the most significant bit (MSB) the same.
2. Generate each subsequent bit by XORing the current bit with the previous bit.

Example: Convert Binary 1101 to Gray Code

MSB: 1

Next bit: 1 XOR 1 = 0

Next bit: 1 XOR 0 = 1

Last bit: 0 XOR 1 = 1

Gray code for Binary 1101 is: 1011

4. Gray Code to Binary Conversion

To convert Gray code to binary:

1. Keep the MSB the same.
2. Generate each subsequent bit by XORing the previous binary bit with the corresponding Gray code bit.

Example: Convert Gray Code 1011 to Binary

MSB: 1

Next bit: 1 XOR 0 = 1

Next bit: 1 XOR 1 = 0

Last bit: 0 XOR 1 = 1

Binary for Gray Code 1011 is: 1101

5. Excess-3 Code to Decimal Conversion

To convert Excess-3 code to decimal:

1. Subtract 3 from each 4-bit Excess-3 group.
2. Convert the result to decimal.

Example: Convert Excess-3 Code 0101 to Decimal

Excess-3 code 0101 = BCD 0010 (Subtract 0011 from 0101)

Decimal 2 (after subtracting 3)

So, Excess-3 0101 = Decimal 2.

6. Decimal to Excess-3 Code Conversion

To convert decimal to Excess-3:

1. Convert the decimal number to BCD.
2. Add 3 (0011) to each BCD digit.

Example: Convert Decimal 4 to Excess-3 Code

Decimal 4 = BCD 0100

Add 0011: 0100 + 0011 = 0111

So, Decimal 4 = Excess-3 Code 0111.

7. ASCII to Binary Conversion

To convert ASCII characters to binary:

1. Find the ASCII code of the character.
2. Convert the ASCII code to its 7-bit binary equivalent.

Example: Convert ASCII Character 'A' to Binary

ASCII code for 'A' = 65

Binary of 65 = 01000001

So, ASCII 'A' = Binary 01000001.

8. Binary to ASCII Conversion

To convert binary to ASCII:

1. Group the binary digits into 7-bit segments.
2. Convert each 7-bit segment to its decimal ASCII code.
3. Find the corresponding ASCII character.

Example: Convert Binary 01000001 to ASCII

Binary 01000001 = Decimal 65

ASCII character for 65 = 'A'

So, Binary 01000001 = ASCII 'A'.

- **Applications of codes:**

1. BCD (Binary-Coded Decimal)
   - Digital Displays: For showing decimal numbers directly.
   - Financial Applications: Accurate decimal calculations.
2. Gray Code
   - Encoders: Reduces errors in position detection.
   - Data Transmission: Minimizes bit transition errors.
3. Excess-3 Code
   - Digital Systems: Simplifies arithmetic operations.
   - Error Detection: Self-complementary property helps in error checking.

4. ASCII Code

- Text Encoding: Standard for character representation in computing.
- Data Storage & Communication: Used in programming and data exchange.

1.5 BCD Arithematic:

BCD (Binary-Coded Decimal) arithmetic involves performing arithmetic operations on numbers that are represented in binary-coded decimal format. In BCD, each digit of a decimal number is represented by its own binary sequence. For example, the decimal number 45 would be represented in BCD as 0100 0101 (where 4 is 0100 and 5 is 0101).

Operations in BCD Arithmetic:

BCD Addition:

Example: Perform the addition 30 + 15 in BCD scheme.

Solution − Given decimal numbers and their equivalent BCD representation is,

$(30)_{10} = (0011\ 0000)_{BCD}$

$(15)_{10} = (0001\ 0101)_{BCD}$

The BCD addition of the given numbers is as below

30			0011	0000
+ 15		+	0001	0101
45			0100	0101

1.5 BCD Subtraction using 9's complement method

There are various methods of BCD subtraction. Among all these methods, 9's complement method or 10's complement method is more familiar.

3.4.1 1st Method of BCD Subtraction

(BCD subtraction by 9's complement method)

- **Steps : (i) :** First find the decimal equivalent of the given BCD codes.

(ii) Carry 9's complement of the subtrahend

(iii) Add this result to the number from which the subtraction is to be done.

(iv) If there is any **carry on bit**, then the carry-on bit may be added to the result of the subtraction.

Ex. 3.4.1 : Find (0 1 0 1 0 0 0 1) – (0 0 1 0 0 0 0 1)

Soln. :

- **Step (I) :** Note that

$(0\,1\,0\,1\,0\,0\,0\,1)_2 = (51)_{10}$ and

$(0\,0\,1\,0\,0\,0\,0\,1)_2 = (21)_{10}$ are decimal values of BCD codes

- **Step (II) :** Subtrahend is (21)

The 9's complement of the subtrahend is

$99 - 21 = 78$

- **Step (III) :** We add the complemented value to 51; i.e.

$51 + 78 = 129.$

- **Step (IV) :** In this result the MSB i.e. 1 is the carry on

We add this carry on to 29

$\therefore$ $29 + 1 = 30$ which is the final answer of BCD.

$\therefore$ The final result of BCD subtraction is

$(0\,1\,0\,1\,0\,0\,0\,1)_{BCD} - (0\,0\,1\,0\,0\,0\,0\,1)_{BCD}$

$= (0\,0\,1\,1\,0\,0\,0\,0)_{BCD}$

****