Computer Fundamental

UNIT : 04

NUMBERING SYSTEM AND CODES

-By Hiral Pandya

WHAT ARE BINARY CODES ?

- A ***BINARY CODE*** represents text, computer processor instructions, or any other data using a two-symbol system.
- The two-symbol system used is often ***"0"*** and ***"1"*** from the binary number system.
- The binary code assigns a pattern of binary digits, also known as bits, to each character, instruction, etc.

- By : Hiral Pandya

BINARY CODES :

- ***NIBBLE :*** A *NIBBLE* is four bits wide, half the width of an eight-bit byte. It can represent exactly 16 unique values, ranging from 0000 (0 in decimal and in hexadecimal) through 1111 (15 in decimal, F in hexadecimal).
- ***BIT*** : A BIT is a basic unit of information or is the smallest unit of data in the computer and digital communications, which stands for binary digit. Either 1 or a 0 (off or on, low or high, false or true) is used to represent each bit.
- ***BYTE*** : A *BYTE* is currently considered a unit of storage that is eight bits wide. This is also known as an octet. It can represent exactly 256 unique values, ranging from **00000000** (0 in decimal) through **11111111** (255 in decimal). Two hexadecimal (base 16) digits can be used as a shorthand to represent a byte's value. In hexadecimal, the unique values in a byte range from 00 through FF.

BINARY CODES :

- ***CARRY BIT :*** It is also known as *CARRY FLAG* (usually indicated as the C flag) is a single bit in a system status register/flag register used to indicate when an arithmetic CARRY OR BORROW has been generated out of the most significant arithmetic logic unit (ALU) bit position.
- ***PARITY BIT*** : It is also known as a ***check bit***, is a single bit that can be appended to a binary string. It is set to either 1 or 0 to make the total number of 1-bits either even ("even parity") or odd ("odd parity").
- ***SIGN BIT*** : It is a bit in a signed number representation that indicates the sign of a number. Although only signed numeric data types have a sign bit, it is invariably located in the most significant bit position, so the term may be used interchangeably with "most significant bit" in some contexts.

STORAGE MEASUREMENTS :

- ***KB :*** KB stands for KiloByte. A kilobyte (KB or Kbyte) is a unit of measurement for computer memory or data storage used by mathematics and computer scienc. 1KB = 1024 Bytes
- ***MB*** : MB stands for MegaByte. The megabyte is a multiple of the unit byte for digital information. Its recommended unit symbol is MB. Therefore, one megabyte is one million bytes of information. This definition has been incorporated into the International System of Quantities. 1MB = 1024 KB
- ***GB*** : GB Stands for GigaByte. The gigabyte is a multiple of the unit byte for digital information. The prefix GIGA means 109 in the International System of Units. Therefore, one gigabyte is one billion bytes. 1GB = 1024 MB

By : Hiral Pandya

STORAGE MEASUREMENTS :

- ***TB :*** TB stands for TerraByte. A terabyte (TB) is a unit of digital data that is equal to about 1 trillion bytes. In decimal notation (base 10), a terabyte is exactly 1 trillion bytes. 1TB = 1024 GB
- **Processor / Virtual Storage Values**
  - **1 Bit = Binary Digit**
  - **8 Bits = 1 Byte**
  - **1024 Bytes = 1 Kilobyte**
  - **1024 Kilobytes = 1 Megabyte**
  - **1024 Megabytes = 1 Gigabyte**
  - **1024 Gigabytes = 1 Terabyte**
  - **1024 Terabytes = 1 Petabyte**
  - **1024 Petabytes = 1 Exabyte**
  - **1024 Exabytes = 1 Zettabyte**
  - **1024 Zettabytes = 1 Yottabyte**
  - **1024 Yottabytes = 1 Brontobyte**
  - **1024 Brontobytes = 1 Geopbyte**

TYPES OF NUMBERING SYSTEM :

- ***BINARY :*** The BINARY NUMBER SYSTEM, also called the BASE-2 NUMBER SYSTEM, is a method of representing numbers that counts by using combinations of only two numerals: ZERO (0) AND ONE (1). Computers use the binary number system to manipulate and store all of their data including numbers, words, videos, graphics, and music. Using binary numbers, **1 + 1 = 10** because "2" does not exist in this system.
- ***OCTAL :*** The Octal Number System has a base of eight and uses the number from 0 to 7. The octal numbers, in the number system, are usually represented by binary numbers when they are grouped in **PAIRS OF THREE**. For example, 540 is expressed as 101100000.
  - $(540)_8$
  - $(101\ 100\ 000)_2$
  - $(101100000)_2$

- By : Hiral Pandya

TYPES OF NUMBERING SYSTEM :

- ***DECIMAL :*** This number system has ten digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) so its base is 10. In this number system, the maximum value of a digit is 9 and the minimum value of a digit is 0. The position of each digit in decimal number represents a specific power of the base (10) of the number system.
- **E.G.: There are three numbers : 207, 971 , 734**

- The value of 7 in all three numbers is different.
- In 207, value 0f 7 is 7 UNITS or 7 or

  7×1 or 7×10^0
- In 971, value of 7 is 7 TENS or 70 or

  7×10 or 7×10^1
- In 734, value of 7 is 7 HUNDREDS or

  700 or 7×100 or 7×10^2

- By : Hiral Pandya

TYPES OF NUMBERING SYSTEM :

- ***HEXADECIMAL :*** This number system has 16 digits that ranges from ***0 to 9 and A to F***. So, its base is 16. The ***A to F*** alphabets represent ***10 to 15 decimal numbers***. The position of each digit in a hexadecimal number represents a specific power of base (16) of the number system. It is also known as ALPHANUMERIC NUMBER SYSTEM as it uses both numeric digits and alphabets.
- Each Hexadecimal number can be represented using only 4 bits, with each group of bits having a distinct values between 0000 (for 0) and 1111 (for F = 15 = 8+4+2+1). The equivalent binary number of Hexadecimal number are as given below.

Hex digit	0	1	2	3	4	5	6	7
Binary	0000	0001	0010	0011	0100	0101	0110	0111

Hex digit	8	9	A = 10	B = 11	C = 12	D = 13	E = 14	F = 15
Binary	1000	1001	1010	1011	1100	1101	1110	1111

By : Hiral Pandya

TYPES OF CODES :

- ***ASCII :*** ASCII stands for **AMERICAN STANDARD CODE FOR INFORMATION INTEREXCHANGE**, It is a standard that assigns letters, numbers, and other characters in the 256 slots available in the 8-bit code.
- ASCII was first developed and published in ***1963*** by the X3 committee, a part of ***the ASA (American Standards Association)***.
- **The ASCII table is divided into three different sections**
  - **NON-PRINTABLE**: System codes between 0 and 31.
  - **LOWER ASCII :** between 32 and 127. This table originates from the older, American systems, which worked on 7-bit character tables.
  - **HIGHER ASCII :** between 128 and 255. This portion is programmable; characters are based on the language of your operating system or program you are using. Foreign letters are also placed in this section.

TYPES OF CODES :

- ***BCD :*** BCD stands for **Binary Coded Decimal**.
- It is a form of binary encoding where each digit in a decimal number is represented in the form of bits.
- This encoding can be done in either 4-bit or 8-bit (usually 4-bit is preferred).
- It is a fast and efficient system that converts the decimal numbers into binary numbers as compared to the existing binary system.
- These are generally used in digital displays where is the manipulation of data is quite a task.
- Thus BCD plays an important role because the manipulation is done treating each digit as a separate single sub-circuit.

- By : Hiral Pandya

TYPES OF CODES :

- ***EBCDIC :*** *E*BCDIC stands for EXTENDED BINARY-CODED DECIMAL INTERCHANGE CODE.
- It is data-encoding system, developed by **IBM** and used mostly on its computers, that uses a unique eight-bit binary code for each number and alphabetic character as well as punctuation marks and accented letters and non-alphabetic characters.
- EBCDIC differs in several respects from Unicode and ASCII, the most widely used systems of encoding text, dividing the eight bits for each character into two four-bit zones, with one zone indicating the type of character, digit, punctuation mark, lowercase letter, capital letter, and so on, and the other zone indicating the value—that is, the specific character within this type.

TYPES OF CODES :

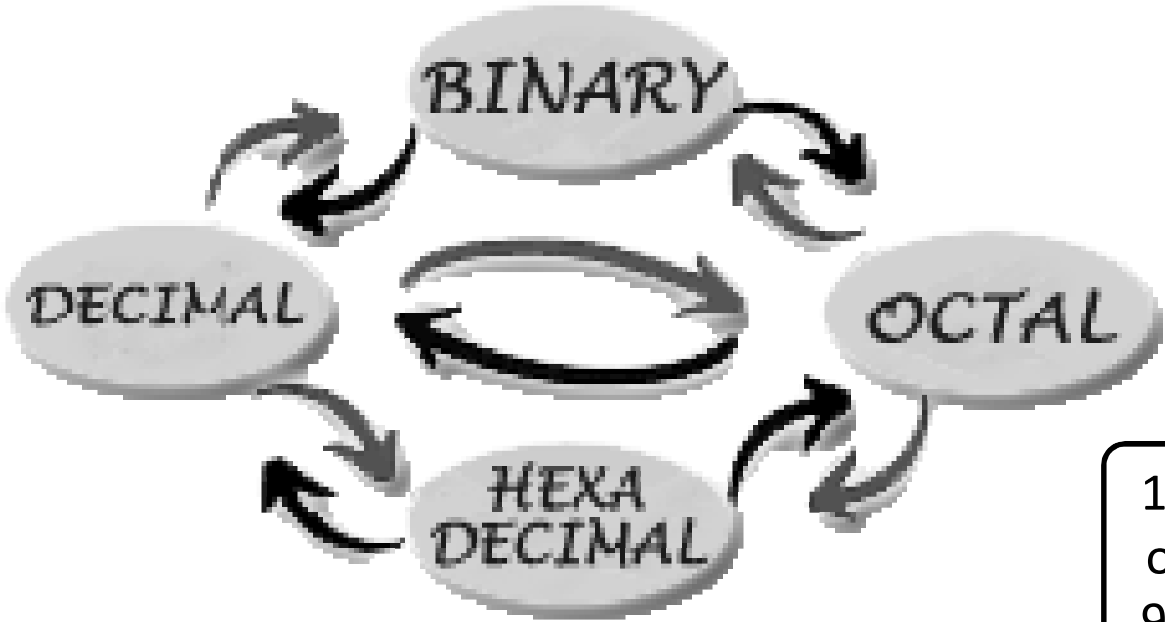
- ***UNICODE :*** Unicode is a universal encoding system to provide a comprehensive character set and was created by the ***Unicode Consortium (a group of multilingual software manufacturers)***.
- Unicode simplifies software localization and improves multilingual text processing. It overcomes the difficulty inherent in ASCII and extended ASCII.
- Unicode has standardizes script behavior which allows any combination of characters, drawn from any combination of scripts and languages, to co-exist in a single document.
- Unicode defines multiple encodings of its single character set: UTF-7, UTF-8, UTF-16, and UTF-32.
- Conversion of data among these encodings is lossless.

Computer Fundamental

be
Alert
now

Computer Fundamental

Let's start conversion



CONVERSION: BINARY TO OCTAL

- ***STEPS :***
  1) First, recognize the given is a binary number
  2) Then binary number to decimal number system
  3) Multiply each digit by $\mathbf{2^{n-1}}$, where **n** is the total number of the digits and make sum of numbers.
  4) We will get the decimal number for the given binary number
  5) Divide the decimal number by 8
  6) Write down the remainder
  7) Continue the above two steps with the quotient till the quotient is zero
  8) Write the remainder in the reverse order
  9) The result is the required octal number for the given binary number

CONVERSION: BINARY TO OCTAL

- ***EXAMPLE :*** Convert the BINARY number (**1010101)$_2$** to the OCTAL number.

➢ First, we will convert given binary number to decimal number

$(1010101)_2$

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

$+ (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

$= 64 + 0 + 16 + 0 + 4 + 0 + 1$

$= 85$

$(1010101)_2 = (85)_{10}$

➢ Now we will convert this decimal number to octal number

➢ Let us divide **85** by **8** we will get remainder **5** and quotient **10**.

➢ Again divide **10** by **8** we will get remainder **2** and quotient **1**

➢ Again dived **1** by **8** we will get remainder **1** and quotient **0**.

➢ Now collecting the remainders in **REVERSE ORDER,** we get **1, 2** and **5**

➢ **Therefore, $(1010101)_2 = (125)_8$**

By : Hiral Pandya

CONVERSION: BINARY TO OCTAL

- ***EXAMPLE :*** Convert the BINARY number **$(1101011)_2$** to the OCTAL number.

➢ First, we will convert given binary number to decimal number

$(1101011)_2$

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

$= 64 + 32 + 0 + 8 + 0 + 2 + 1$

$= 107$

$(1101011)_2 = (107)_{10}$

➢ Now we will convert this decimal number to octal number

➢ Let us divide **107** by **8** we will get remainder **3** and quotient **13**.

➢ Again divide **13** by **8** we will get remainder **5** and quotient **1**

➢ Again dived **1** by **8** we will get remainder **1** and quotient **0**.

➢ Now collecting the remainders in **REVERSE ORDER,** we get **1, 5** and **3**

➢ **Therefore, $(1101011)_2 = (153)_8$**

By : Hiral Pandya

CONVERSION: BINARY TO DECIMAL

- ***STEPS :***

1) First, recognize the given is a binary number
2) Then binary number to decimal number system
3) Multiply each digit by $\mathbf{2^{n-1}}$, where **n** is the total number of the digits.
4) Add Each Numbers together.
5) We will get the decimal number for the given binary number

- By : Hiral Pandya

CONVERSION: BINARY TO DECIMAL

- ***EXAMPLE :*** Convert the BINARY number (**1010101)$_2$** to the DECIMAL number.

➢ First, we will convert given binary number to decimal number

$$(1010101)_2$$

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

$$= 64 + 0 + 16 + 0 + 4 + 0 + 1$$

$$= 85$$

$$(1010101)_2 = (85)_{10}$$

➢ **Therefore, $(1010101)_2 = (85)_{10}$**

CONVERSION: BINARY TO DECIMAL

- ***EXAMPLE :*** Convert the BINARY number $(\mathbf{1101011})_2$ to the DECIMAL number.

➢ First, we will convert given binary number to decimal number

$$(1101011)_2$$

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

$$= 64 + 32 + 0 + 8 + 0 + 2 + 1$$

$$= 107$$

$$(1101011)_2 = (107)_{10}$$

➢ **Therefore,** $(1101011)_2 = (107)_{10}$

- By : Hiral Pandya

CONVERSION: BINARY TO HEXADECIMAL

- ***STEPS :***
  1) First, recognize the given is a binary number
  2) Then binary number to decimal number system
  3) Multiply each digit by $\mathbf{2^{n-1}}$, where **n** is the total number of the digits and get sum of all.
  4) We will get the decimal number for the given binary number
  5) Divide the decimal number by **16**
  6) Write down the remainder
  7) Continue the above two steps with the quotient till the quotient is zero
  8) Write the remainder in the reverse order
  9) Consider Hex Table for number
  10) The result is the required HEXADECIMAL number for the given BINARY number

CONVERSION: BINARY TO HEXADECIMAL

- ***STEPS :***
  1) First, recognize the given is a BINARY Number.
  2) First make group the numbers in a SET OF 4.
  3) Every 4 digit in binary becomes one 1 digit in hexadecimal.
  4) If Needed, Add *ZEROS* to the LEFT OF THE LAST DIGIT, if there aren't enough digits to make a set of 4.
  5) Arrange the numbers together to get the final HEX Number.

- By : Hiral Pandya

CONVERSION: BINARY TO HEXADECIMAL

- ***HEX TABLE :***

Hex Number	Binary Number
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Hex Number	Binary Number
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

- By : Hiral Pandya

CONVERSION: BINARY TO HEXADECIMAL

- ***Example :*** Convert the BINARY number $(1101011)_2$ to the HEXADECIMAL number.
- $(1101011)_2$
- Here, Total Digits are 7. So, ZERO needs to be added at left most position.
- So It is 0110 1011
- 01101011
  - ➢ 0110 = 6
  - ➢ 1011 = B
- **Answer Is : $(1101011)_2$ = $(6B)_{16}$**

CONVERSION: BINARY TO HEXADECIMAL

- ***Example :*** Convert the BINARY number $(1010101)_2$ to the HEXADECIMAL number.
- $(1010101)_2$
- Here, Total Digits are 7. So, ZERO needs to be added at left most position.
- So It is 0101 0101
- 01010101
  - 0101 = 5
  - 0101 = 5
- **Answer Is : $(1010101)_2$ = $(55)_{16}$**

- By : Hiral Pandya

CONVERSION: DECIMAL TO BINARY

- ***STEPS :***
  1) First, recognize the given is a INTEGER DECIMAL number
  2) Take decimal number as dividend
  3) Divide this number by **2** (2 is base of binary so divisor here)
  4) Store the remainder (it will be either 0 or 1 because of divisor 2).
  5) Repeat the above two steps until the number is greater than zero.
  6) Write the remainder in the reverse order.
  7) This will be equivalent binary number of given decimal number.

CONVERSION: DECIMAL TO BINARY

- ***EXAMPLE :*** Convert the DECIMAL number **$(153)_{10}$** to the BINARY number.

➢ First, we will check given decimal number as INTEGER DECIMAL **$(153)_{10}$**

DIVISION	REMINDER
= 153 / 2 = 76	**1**
= 076 / 2 = 38	**0**
= 038 / 2 = 19	**0**
= 019 / 2 = 09	**1**
= 009 / 2 = 04	**1**
= 004 / 2 = 02	**0**
= 002 / 2 = 01	**0**
= 001 / 2 = 00	**1**

➢ **Answer Is : $(153)_{10}$ = $(10011001)_{2}$**

- By : Hiral Pandya

CONVERSION: DECIMAL TO BINARY

- ***EXAMPLE :*** Convert the DECIMAL number **$(119)_{10}$** to the BINARY number.

➢ First, we will check given decimal number as INTEGER DECIMAL **$(119)_{10}$**

DIVISION	REMINDER
= 119 / 2 = 59	1
= 059 / 2 = 29	1
= 029 / 2 = 14	1
= 014 / 2 = 07	0
= 007 / 2 = 03	1
= 003 / 2 = 01	1
= 001 / 2 = 00	1

➢ **Answer Is : $(119)_{10}$ = $(1110111)_2$**

- By : Hiral Pandya

CONVERSION: DECIMAL TO OCTAL

- ***STEPS :***
  1) First, recognize the given is a INTEGER DECIMAL number
  2) Take decimal number as dividend
  3) Divide this number by **8**
  4) Store the remainder
  5) Repeat the above two steps until the number is greater than zero.
  6) Write the remainder in the reverse order.
  7) This will be equivalent OCTAL number of given DECIMAL number.

CONVERSION: DECIMAL TO OCTAL

- ***EXAMPLE :*** Convert the DECIMAL number $(\mathbf{153})_{10}$ to the OCTAL number.

➢ First, we will check given decimal number as INTEGER DECIMAL $(\mathbf{153})_{10}$

DIVISION	REMINDER
= 153 / 8 = 19	1 ↑
= 019 / 8 = 02	3
= 002 / 8 = 00	2

➢ **Answer Is : $(\mathbf{153})_{10}$ = $(\mathbf{231})_{8}$**

- By : Hiral Pandya

CONVERSION: DECIMAL TO OCTAL

- ***EXAMPLE :*** Convert the DECIMAL number $(\mathbf{119})_{10}$ to the OCTAL number.

➢ First, we will check given decimal number as INTEGER DECIMAL $(\mathbf{119})_{10}$

DIVISION	REMINDER
= 119 / 8 = 14	7
= 014 / 8 = 01	6
= 001 / 8 = 00	1

➢ **Answer Is : $(119)_{10}$ = $(167)_8$**

- By : Hiral Pandya

CONVERSION: DECIMAL TO HEX-DECIMAL

- ***STEPS :***
  1) First, recognize the given is a INTEGER DECIMAL number
  2) Take decimal number as dividend
  3) Divide this number by **16**
  4) Store the remainder
  5) Repeat the above two steps until the number is greater than zero.
  6) Write the remainder in the reverse order.
  7) This is equivalent HEX-DECIMAL number of given DECIMAL number.

CONVERSION: DECIMAL TO HEX-DECIMAL

- ***EXAMPLE :*** Convert the DECIMAL number (**153**)$_{10}$ to the HEX-DECIMAL number.

➢ First, we will check given decimal number as INTEGER DECIMAL (**153**)$_{10}$

DIVISION	REMINDER
= **153 / 16 = 09**	**9** ↑
= **009 / 16 = 00**	**9**

➢ **Answer Is : (153)$_{10}$ = (99)$_{16}$**

- By : Hiral Pandya

CONVERSION: DECIMAL TO HEX-DECIMAL

- ***EXAMPLE :*** Convert the DECIMAL number $(\mathbf{119})_{10}$ to the HEX-DECIMAL number.

➢ First, we will check given decimal number as INTEGER DECIMAL $(\mathbf{119})_{10}$

DIVISION	REMINDER
= 119 / 16 = 07	**7** ↑
= 007 / 16 = 00	**7**

➢ **Answer Is :** $\mathbf{(119)_{10} = (77)_{16}}$

- By : Hiral Pandya

CONVERSION: OCTAL TO BINARY

Conversion of OCTAL to BINARY number is a two-step process. First, it need to converted the given octal number into its equivalent decimal number and then convert the decimal into binary.

- ***STEP 01 :* OCTAL *to* DECIMAL *Conversion***
  1) Count the number of digits present in the given number. Let the number of digits be 'n'.
  2) Now multiply each digit of the number with $\mathbf{8^{n-1}}$, when the digit is in the **n^{th}** position from the right end of the number.
  3) If the number has a decimal part, multiply each digit in the decimal part by "$\mathbf{8^{-m}}$" when the digit is in the **m^{th}** position from the decimal point.
  4) Add all the terms after multiplication.
  5) The obtained value is the equivalent decimal number.

CONVERSION: OCTAL TO BINARY

- ***STEP 02 :* DECIMAL *to* BINARY *Conversion***
  1) First, recognize the given is a INTEGER DECIMAL number
  2) Take decimal number as dividend
  3) Divide this number by **2** (2 is base of binary so divisor here)
  4) Store the remainder (it will be either 0 or 1 because of divisor 2).
  5) Repeat the above two steps until the number is greater than zero.
  6) Write the remainder in the reverse order.

This will be equivalent BINARY number of given OCTAL number.

CONVERSION: OCTAL TO BINARY

- ***EXAMPLE :*** Convert $(\mathbf{167})_8$ to the BINARY number.

➢ First $(\mathbf{167})_8$ will be converted to DECIMAL : Total Digits are **3**

➢ = $(\mathbf{167})_8$

➢ $= 1 \times 8^2 = 64$

$+ 6 \times 8^1 = 48$

$+ 7 \times 8^0 = 07$

$= 64 + 48 + 7$

$= (119)_{10}$

DIVISION	REMINDER
= 119 / 2 = 59	1
= 059 / 2 = 29	1
= 029 / 2 = 14	1
= 014 / 2 = 07	0
= 007 / 2 = 03	1
= 003 / 2 = 01	1
= 001 / 2 = 00	1

➢ Answer Is $(167)_8 = (1110111)_2$

- By : Hiral Pandya

CONVERSION: OCTAL TO BINARY

- ***EXAMPLE :*** Convert $(\mathbf{231})_8$ to the BINARY number.

➢ First $(\mathbf{231})_8$ will be converted to DECIMAL : Total Digits are **3**

➢ = $(\mathbf{231})_8$

➢ $= 2 \times 8^2 = 128$

$+ 3 \times 8^1 = 024$

$+ 1 \times 8^0 = 001$

$= 128 + 24 + 1$

$= (153)_{10}$

DIVISION	REMINDER
= 153 / 2 = 76	**1**
= 076 / 2 = 38	**0**
= 038 / 2 = 19	**0**
= 019 / 2 = 09	**1**
= 009 / 2 = 04	**1**
= 004 / 2 = 02	**0**
= 002 / 2 = 01	**0**
= 001 / 2 = 00	**1**

➢ Answer Is $(231)_8 = (10011001)_2$

- By : Hiral Pandya

CONVERSION: OCTAL TO DECIMAL

- ***STEP :***
  1) Count the number of digits present in the given number. Let the number of digits be 'n'.
  2) Now multiply each digit of the number with $\mathbf{8^{n-1}}$, when the digit is in the $\mathbf{n^{th}}$ position from the right end of the number.
  3) If the number has a decimal part, multiply each digit in the decimal part by "$\mathbf{8^{-m}}$" when the digit is in the $\mathbf{m^{th}}$ position from the decimal point.
  4) Add all the terms after multiplication.
  5) The obtained value is the equivalent decimal number.

CONVERSION: OCTAL TO DECIMAL

- ***EXAMPLE :*** Convert $(\mathbf{167})_8$ to the DECIMAL number.

➢ $(\mathbf{167})_8$

➢ $= 1 \times 8^2 = 64$

$+ 6 \times 8^1 = 48$

$+ 7 \times 8^0 = 07$

$= 64 + 48 + 7$

$= (119)_{\mathbf{10}}$

➢ Answer Is $(167)_{\mathbf{8}} = (119)_{\mathbf{10}}$

- By : Hiral Pandya

CONVERSION: OCTAL TO HEXADESIMAL

- ***STEP 01 : OCTAL** to **DECIMAL** Conversion*
  1) Count the number of digits present in the given number. Let the number of digits be 'n'.
  2) Now multiply each digit of the number with $\mathbf{8^{n-1}}$, when the digit is in the $\mathbf{n^{th}}$ position from the right end of the number.
  3) If the number has a decimal part, multiply each digit in the decimal part by "$\mathbf{8^{-m}}$" when the digit is in the $\mathbf{m^{th}}$ position from the decimal point.
  4) Add all the terms after multiplication.
  5) The obtained value is the equivalent decimal number.

- By : Hiral Pandya

CONVERSION: OCTAL TO HEXADECIMAL

- ***STEP 02 :*** **DECIMAL *To HEXADECIMAL Conversion***
  1) Take DECIMAL number as dividend
  2) Divide this number by **16**
  3) Store the remainder
  4) Repeat the above two steps until the number is greater than zero.
  5) Write the remainder in the reverse order.
  6) Convert any remainders bigger than 9 into hex letters

This is equivalent HEX-DECIMAL number of given OCTAL number.

By : Hiral Pandya

CONVERSION: OCTAL TO HEXADECIMAL

- ***EXAMPLE :*** Convert **$(167)_8$** to the HEXADECIMAL number.

$(167)_8$

- $= 1 \times 8^2 = 64$

  $+ 6 \times 8^1 = 48$

  $+ 7 \times 8^0 = 07$

$= 64 + 48 + 7$

$= (119)_{10}$

DIVISION	REMINDER
= 119 / 16 = 07	**7** ↑
= 007 / 16 = 0	**7**

- Answer Is $(167)_8 = (77)_{16}$

- By : Hiral Pandya

CONVERSION: OCTAL TO HEXADECIMAL

- ***EXAMPLE :*** Convert $(\mathbf{263})_8$ to the HEXADECIMAL number.

$(\mathbf{263})_8$

- $= 2 \times 8^2 = 128$

$+ 6 \times 8^1 = 048$

$+ 3 \times 8^0 = 003$

$= 128 + 48 + 3$

$= (179)_{10}$

DIVISION	REMINDER
= 179 / 16 = 11	**3** ↑
= 11 / 16 = 00	**11**

Here $(11)_{10} = (B)_{16}$

- Answer Is $(263)_8 = (B3)_{16}$

- By : Hiral Pandya

CONVERSION: HEXADECIMAL TO BINARY

- ***STEPS :***
  1) Note down the hex number. If there are any, change the hex values represented by letters to their DECIMAL equivalents.
  2) Consider Hex Table for equivalent Binary number.
  3) Write down binary number with respected decimal number.
  4) Skip ZERO(s) from left most position.
  5) This will be Binary number for HEXADECIMAL number.

- By : Hiral Pandya

CONVERSION: HEXADECIMAL TO BINARY

- ***EXAMPLE :*** Convert $(\mathbf{149})_{16}$ to the BINARY number.

$(\mathbf{149})_{16}$

➢ =

HEX-DECIMAL	1	4	9
BINARY	0001	0100	1001

➢ 0001 0100 1001

➢ Skip ZERO(s) from left most position.

Answer Is $(149)_{16} = (101001001)_2$

- By : Hiral Pandya

CONVERSION: HEXADECIMAL TO BINARY

- ***EXAMPLE :*** Convert $(\mathbf{A2C})_{16}$ to the BINARY number.

$(\mathbf{A2C})_{16}$

- =

HEX-DECIMAL	**A**	**2**	**C**
BINARY	**1010**	**0010**	**1100**

- 1010 0010 1100
- There is no ZERO at left most position.

Answer Is $(A2C)_{16} = (101000101100)_2$

- By : Hiral Pandya

CONVERSION: HEXADECIMAL TO OCTAL

- ***STEPS :***
  1) First count the number of digits in the number
  2) If n is the position of the digit from the right end then multiply each digit with $\mathbf{16^{n-1}}$
  3) Add the terms after multiplication
  4) Resultant is the equivalent decimal form
  5) Divide the decimal number with **8**
  6) Note down the remainder
  7) Repeat the previous two steps with the quotient, until the quotient is zero
  8) Write the remainders in reverse order
  9) The obtained number is the required result

- By : Hiral Pandya

CONVERSION :HEXADECIMAL TO OCTAL

- ***EXAMPLE :*** Convert $(\mathbf{A2C})_{16}$ to the OCTAL number.

$(\mathbf{A2C})_{16}$

Hex to Decimal

= A 2 C

= A X 16^2 = 10 X 256 = 2560

2 X 16^1 = 02 X 016 = 0032

C X 16^0 = 12 X 001 = 0012

= (2560 + 32 + 12)

= $\mathbf{(2604)_{10}}$

Decimal To Octal

Division	Reminder
= 2604 / 8 **= 325**	**4**
= 325 / 8 **= 40**	**5**
= 40 / 8 **= 5**	**0**
= 5 / 8 **= 00**	**5**

Answer Is $(\text{A2C})_{16}$ = $(5054)_{8}$

- By : Hiral Pandya

CONVERSION :HEXADECIMAL TO OCTAL

- ***EXAMPLE :*** Convert **(D92)**$_{16}$ to the OCTAL number.

(D92)$_{16}$

Hex to Decimal	Decimal To Octal
= D 9 2	

= D 9 2

$= D \times 16^2 = 13 \times 256 = 3328$

$9 \times 16^1 = 09 \times 016 = 0144$

$2 \times 16^0 = 02 \times 001 = 0002$

= (3328 + 144 + 2)

= **(3474)**$_{10}$

Division	Reminder
= 3474 / 8 **= 434**	**2**
= 434 / 8 **= 54**	**2**
= 54 / 8 **= 6**	**6**
= 6 / 8 **= 00**	**6**

Answer Is (D92)$_{16}$ = (6622)$_{8}$

- By : Hiral Pandya

CONVERSION: HEXADECIMAL TO DECIMAL

- ***STEPS :***
  1) First count the number of digits in the number
  2) If n is the position of the digit from the right end then multiply each digit with $\mathbf{16^{n-1}}$
  3) Add the terms after multiplication
  4) Resultant is the equivalent DECIMAL number

- By : Hiral Pandya

CONVERSION: HEXADECIMAL TO DECIMAL

- ***EXAMPLE :*** Convert ($\mathbf{A2C}$)$_{16}$ to the DECIMAL number.

$(\mathbf{A2C})_{16}$

- = A 2 C

$= A \times 16^2 = 10 \times 256 = 2560$

$2 \times 16^1 = 02 \times 016 = 0032$

$C \times 16^0 = 12 \times 001 = 0012$

$= (2560 + 32 + 12)$

$= \mathbf{(2604)_{10}}$

Answer Is $(A2C)_{\mathbf{16}} = (2604)_{\mathbf{10}}$

- By : Hiral Pandya

CONVERSION: HEXADECIMAL TO DECIMAL

- ***EXAMPLE :*** Convert (**D92)**$_{16}$ to the DECIMAL number.

(**D92)**$_{16}$

- = D 9 2

= D X 16^2 = 13 X 256 = 3328

9 X 16^1 = 02 X 016 = 0144

2 X 16^0 = 09 X 001 = 0002

= (3328 + 144 + 2)

= **(3474)**$_{10}$

Answer Is (D92)$_{16}$ = (3474)$_{10}$

- By : Hiral Pandya

CONVERSION EXERCISE : 01

- Convert **$(11001100)_2$** to the OCTAL number.
  - **$(11001100)_2 = (314)_8$**
- Convert **$(11001100)_2$** to the DECIMAL number.
  - **$(11001100)_2 = (204)_{10}$**
- Convert **$(11001100)_2$** to the HEX-DECIMAL number.
  - **$(11001100)_2 = (CC)_{16}$**

- Convert **$(217)_8$** to the BINARY number.
  - **$(217)_8 = (10001111)_2$**
- Convert **$(217)_8$** to the DECIMAL number.
  - **$(217)_8 = (143)_{10}$**
- Convert **$(217)_8$** to the HEX-DECIMAL number.
  - **$(217)_8 = (8F)_{16}$**

- By : Hiral Pandya

CONVERSION EXERCISE : 02

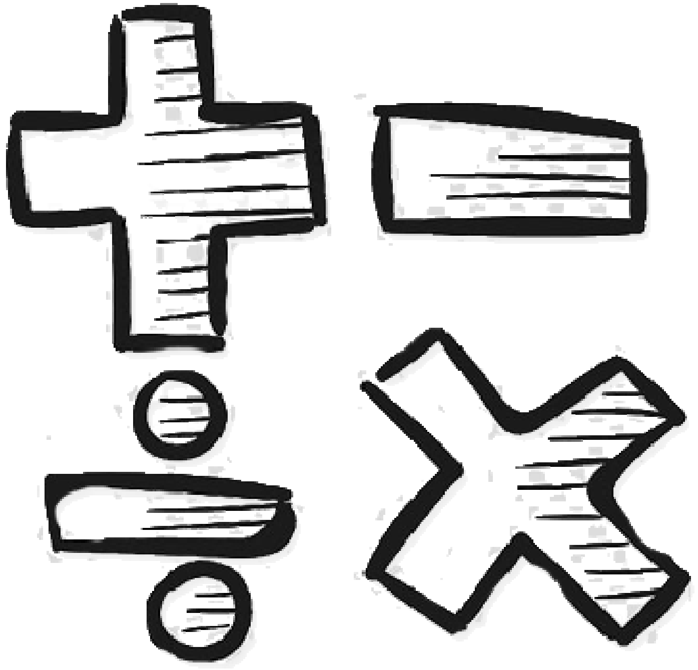
- Convert $(123)_{10}$ to the BINARY number.
  - $(123)_{10} = (1111011)_2$
- Convert $(123)_{10}$ to the OCTAL number.
  - $(123)_{10} = (173)_8$
- Convert $(217)_8$ to the HEX-DECIMAL number.
  - $(123)_{10} = (7B)_{16}$

- Convert $(6C)_{16}$ to the BINARY number.
  - $(6C)_{16} = (1101100)_2$
- Convert $(6C)_{16}$ to the OCTAL number.
  - $(6C)_{16} = (154)_8$
- Convert $(217)_8$ to the DECIMAL number.
  - $(6C)_{16} = (108)_{10}$

- By : Hiral Pandya

Computer Fundamental

Let's start BINARY ARITHMETIC



ADDITION:

- The binary addition operation works similarly to the base 10 decimal system, except that it is a base 2 system.
- The binary system consists of only two digits, ***1 and 0***.
- Most of the functionalities of the computer system use the binary number system.
- The binary code uses the digits 1's and 0's to make certain processes turn off or on.
- The process of the addition operation is very familiar to the decimal system by adjusting to the base 2.
- The bit ***0*** represents the **"OFF"** state, and the bit ***1*** represents the ***"ON"*** state.

ADDITION:

- ***Rules for Addition :***

➢ 0 + 0 = 0

➢ 0 + 1 = 1

➢ 1 + 0 = 1

➢ 1 + 1 = 10 (where 1 is carried over)

So, 1 + 1 = 0 **and 1 is carried over**

- By : Hiral Pandya

ADDITION:

- **Example :** 1 0 1 + 1 0 1
- **Step 1:** First consider the 1’s column, and add the one’s column,(1+1) and it gives the result 10 as per the Rule of binary addition.
- **Step 2:** Now, leave the 0 in the one’s column and carry the value 1 to the 10’s column.

Carry Over ==>>		**1**	
	1	0	0
+	1	0	1
			0

ADDITION:

- **Step 3:** Now add 10's place, 1+(0 + 0) = 1. So, nothing carries to the 100's place and leave the value 1 in the 10's place.

Carry Over ==>>		**1**	
	1	0	0
+	1	0	1
		1	0

- **Step 4:** Now add the 100's place (1 + 1) = 10. Leave the value 0 in the 100's place and carries 1 to the 1000's place.

		1	
	1	0	0
+	1	0	1
1	0	1	0

So, 1 0 1 + 1 0 1 = 1 0 1 0

ADDITION:

- **Example :** 1 0 0 1 0 1 + 1 0 1 0 1

			1		**1**	
	1	**0**	**0**	**1**	**0**	**1**
+		**1**	**0**	**1**	**0**	**1**
	1	**1**	**1**	**0**	**1**	**0**

So, 100101 + 10101 = 111010

- By : Hiral Pandya

ADDITION:

- **Example :** 1 0 1 0 0 1 + 1 1 0 1 0 1

					1	
	1	**0**	**1**	**0**	**0**	**1**
+	**1**	**1**	**0**	**1**	**0**	**1**
1	**0**	**1**	**1**	**1**	**1**	**0**

So, 101001 + 110101 = 1011110

- By : Hiral Pandya

SUBTRACTION :

- Binary SUBTRACTION is one of the four binary operations, where we perform the subtraction method for two binary numbers (comprising only two digits, 0 and 1).
- This operation is similar to the basic arithmetic subtraction performed on decimal numbers in Mathematics.
- When we subtract 1 from 0, we need to borrow 1 from the next higher order digit, to reduce the digit by 1 and the remainder left here is also 1.

SUBTRACTION :

- ***Rules for Subtraction :***

  - 0 - 0 = 0
  - 1 - 0 = 1
  - 1 - 1 = 0
  - 0 - 1 = 1 (Where 1 is Borrowed)

- By : Hiral Pandya

SUBTRACTION :

- **Example :** 1 0 1 0 – 1 0 1
- **Step 1:** First consider the 1's column, and subtract the one's column,(0 – 1) and it gives the result 1 as per the condition of binary subtraction with a borrow of 1 from the 10's place.
- **Step 2:** After borrowed 1 from the 10's column, the value 1 in the 10's column is changed into the value.

Borrow ➔➔➔			**1**	
	1	**0**	**1**	**0**
-		**1**	**0**	**1**
				1

- By : Hiral Pandya

SUBTRACTION :

- **Step 3:** Now, Subtract the value in the 10's place, (0 – 0) = 0.

Borrow ➔➔➔			**1**	
	1	**0**	**1**	**0**
-		**1**	**0**	**1**
			0	**1**

- **Step 4:** Perform same subtraction for next TWO Columns.

	1		**1**	
	1	**0**	**1**	**0**
-		**1**	**0**	**1**
Answer	**0**	**1**	**0**	**1**

SUBTRACTION :

- **Example :** 1 0 0 1 0 1 - 1 0 1 0 1

	1					
	1	**0**	**0**	**1**	**0**	**1**
-	**0**	**1**	**0**	**1**	**0**	**1**
	0	**1**	**0**	**0**	**0**	**0**

So, 100101 - 10101 = 10000

- By : Hiral Pandya

SUBTRACTION :

- **Example :** 1 1 0 1 0 1 - 1 0 1 0 0 1

		1				
	1	**1**	**0**	**1**	**0**	**1**
-	**1**	**0**	**1**	**0**	**0**	**1**
		0	**1**	**1**	**0**	**0**

So, 110101 - 101001 = 01100

- By : Hiral Pandya

1’s Complement :

- To get 1’s complement of a binary number, simply *INVERT THE GIVEN NUMBER*.
- There are various uses of 1’s complement of Binary numbers.
- Mainly in signed Binary number representation and various arithmetic operations for Binary numbers.

1’s Complement :

- **Example :** Find 1’s complement of binary number 10101110.

1 ↓ **0**	**0** ↓ **1**	**1** ↓ **0**	**0** ↓ **1**	**1** ↓ **0**	**1** ↓ **0**	**1** ↓ **0**	**0** ↓ **1**

- By : Hiral Pandya

1's Complement :

- **Example :** Find 1's complement of binary number 10111011.

1	**0**	**1**	**1**	**1**	**0**	**1**	**1**
↓	↓	↓	↓	↓	↓	↓	↓
0	**1**	**0**	**0**	**0**	**1**	**0**	**0**

- By : Hiral Pandya

2's Complement :

- To get 2's complement of binary number is 1's complement of given number plus 1 to the least significant bit (LSB).
- Example 2's complement of binary number 10010
- 0 1 1 0 1 (1's Complement of 10010)

+ 1

=0 1 1 1 0

- By : Hiral Pandya

2’s Complement :

- **Example :** Find 2’s complement of binary number 10101110.

1	0	1	0	1	1	1	0
↓	↓	↓	↓	↓	↓	↓	↓
0	1	0	1	0	0	0	1

	0	1	0	1	0	0	0	1
+								1
	0	1	0	1	0	0	1	0

2's Complement :

- **Example :** Find 2's complement of binary number 10111011.

1 ↓ 0	0 ↓ 1	1 ↓ 0	1 ↓ 0	1 ↓ 0	0 ↓ 1	1 ↓ 0	1 ↓ 0

	0	1	0	0	0	1	0	0
+								1
	0	1	0	0	0	1	1	0

- By : Hiral Pandya

SUBTRACTION USING 1'S COMPLEMENT:

- **Step 1:** In the first step, find the 1's complement of the subtrahend.
- **Step 2:** Add the complement number with the minuend.
- **Step 3:** If got a carry, add the carry to its LSB. Else take 1's complement of the result which will be negative
- **NOTE:** The SUBTRAHEND value always get subtracted from MINUEND.

- By : Hiral Pandya

SUBTRACTION USING 1'S COMPLEMENT:

- **Example :** 10101 - 10111

1 ↓ **0**	**0** ↓ **1**	**1** ↓ **0**	**1** ↓ **0**	**1** ↓ **0**

	1	**0**	**1**	**0**	**1**
+	**0**	**1**	**0**	**0**	**0**
	1	**1**	**1**	**0**	**1**

- By : Hiral Pandya

SUBTRACTION USING 1'S COMPLEMENT:

- **Example :** 10101 – 00111

0 ↓ **1**	**0** ↓ **1**	**1** ↓ **0**	**1** ↓ **0**	**1** ↓ **0**

	1	**0**	**1**	**0**	**1**
+	**1**	**1**	**0**	**0**	**0**
1	**0**	**1**	**1**	**0**	**1**

- In the above result, we get the carry bit 1, so add this to the LSB of a given result.

0 1 1 0 1 + 1 = 0 1 1 1 0

- By : Hiral Pandya

SUBTRACTION USING 2’S COMPLEMENT:

- **Step 1:** In the first step, find the 2's complement of the subtrahend.
- **Step 2:** Add the complement number with the minuend.
- **Step 3:** If we didn’t get the carry by adding both the numbers, then take 2's complement of the result which will be negative.
- **Step 4:** If we get the carry by adding both the numbers, then we discard this carry and the result is positive.

- By : Hiral Pandya

SUBTRACTION USING 2'S COMPLEMENT:

- **Example :** 10101 – 10111

1 ↓ **0**	**0** ↓ **1**	**1** ↓ **0**	**1** ↓ **0**	**1** ↓ **0**

- Now, Let's Find 2's Complement of SUBTRAHEND

	0	**1**	**0**	**0**	**0**
+	**0**	**0**	**0**	**0**	**1**
2's Comp	**0**	**1**	**0**	**0**	**1**

- By : Hiral Pandya

SUBTRACTION USING 2'S COMPLEMENT:

- Now, Let's Add Minuend and 2's Complement

	1	**0**	**1**	**0**	**1**
+	**0**	**1**	**0**	**0**	**1**
Result	**1**	**1**	**1**	**1**	**0**

- There is no **CARRY BIT,** So we need to find 2's Complement of ***RESULT.***

1	**1**	**1**	**1**	**0**
↓	↓	↓	↓	↓
0	**0**	**0**	**0**	**1**

	0	**0**	**0**	**0**	**1**
+	**0**	**0**	**0**	**0**	**1**
Answer	**0**	**0**	**0**	**1**	**0**

- By : Hiral Pandya

SUBTRACTION USING 2'S COMPLEMENT:

- **Example :** 10101 – 00111

0 ↓ **1**	**0** ↓ **1**	**1** ↓ **0**	**1** ↓ **0**	**1** ↓ **0**

- Now, Let's Find 2's Complement of SUBTRAHEND

	1	**1**	**0**	**0**	**0**
+	**0**	**0**	**0**	**0**	**1**
2's Comp	**1**	**1**	**0**	**0**	**1**

– By : Hiral Pandya

SUBTRACTION USING 2'S COMPLEMENT:

- Now, Let's Add Minuend and 2's Complement

Carry1	**1**	**0**	**1**	**0**	**1**
+	**1**	**1**	**0**	**0**	**1**
Result	**0**	**1**	**1**	**1**	**0**

- Here we got a **CARRY,** So just discard carry and result is POSITIVE**.**

Answer	**0**	**1**	**1**	**1**	**0**

MULTIPLICATION:

- BINARY MULTIPLICATION is one of the four binary arithmetic.
- The other three fundamental operations are addition, subtraction and division.
- In the case of a binary operation, we deal with only two digits, i.e. 0 and 1.
- The binary multiplication operation is actually a process of addition and shifting operation.
- This process has to be continued until all the multiplier is done, and finally, the addition operation is made.

MULTIPLICATION:

- ***Rules for Multiplication:***

➢ 0 X 0 = 0

➢ 0 X 1 = 0

➢ 1 X 0 = 0

➢ 1 X 1 = 1

- By : Hiral Pandya

MULTIPLICATION:

- ***Example :*** **1 0 1 1 X 1 0 1**

				1	**0**	**1**	**1**
			X	**0**	**1**	**0**	**1**
				1	**0**	**1**	**1**
+			**0**	**0**	**0**	**0**	**0**
+		**1**	**0**	**1**	**1**	**0**	**0**
+	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	0	**1**	**1**	**0**	**1**	**1**	**1**

MULTIPLICATION:

- ***Example :*** **1 1 0 1 X 1 1 0**

				1	**1**	**0**	**1**
			X	**0**	**1**	**1**	**0**
				0	**0**	**0**	**0**
+			**1**	**1**	**0**	**1**	**0**
+		**1**	**1**	**0**	**1**	**0**	**0**
+	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	1	**0**	**0**	**1**	**1**	**1**	**0**

- By : Hiral Pandya

DIVISION:

- The binary division operation is similar to the base 10 decimal system, except the base 2.
- The division is probably one of the most challenging operations of the basic arithmetic operations.
- There are different ways to solve division problems using binary operations.
- Long division is one of them and the easiest and the most efficient way.

DIVISION:

- ***Example :*** **1 1 0 1 % 1 0 1**

1 0 1	**1 1 0 1** **- 1 0 1** ↓	**1 0**
	0 0 1 1	

- **1 1 0 1 % 1 0 1 = 10, Reminder 0011**

- By : Hiral Pandya

DIVISION:

- ***Example :*** **1 0 1 1 % 1 1 0**

1 1 0	1 0 1 1	0 1
	- 0 1 1 0	
	0 1 0 1	

- **1 0 1 1 % 1 1 0 = 01, Reminder 0101**

- By : Hiral Pandya

PARITY :

- A parity bit, also known as a check bit, is a single bit that can be appended to a binary string.
- It is set to either 1 or 0 to make the total number of 1-bits either even ("even parity") or odd ("odd parity").
- THE TWO TYPES OF PARITY BIT :
  - **EVEN PARITY :** Here the total number of bits in the message is made even.
  - **ODD PARITY :** Here the total number of bits in the message is made odd.

- By : Hiral Pandya

EVEN PARITY SYSTEM :

- Even parity refers to a parity checking mode in asynchronous communication systems in which an extra bit, called a parity bit, is set to zero if there is an even number of one bits in a one-byte data item.
- If the number of one bits adds up to an odd number, the parity bit is set to one.
- **Example:** Consider the transmitted message 1010001, which has three ones in it. This is turned into even parity by adding a one, making the sequence 1 1010001, so that there are four ones (**AN EVEN NUMBER**).
- If the transmitted message has the form 1101001, which is already an even number, a zero is added to sustain the EVEN PARITY.
- The resulting message is 0 1101001, so that an even number of ones remains in the transmitted message.

ODD PARITY SYSTEM :

- In asynchronous communication systems, odd parity refers to parity checking modes, where each set of transmitted bits has an odd number of bits.
- If the total number of ones in the data plus the parity bit is an odd number of ones, it is called odd parity. If the data already has an odd number of ones, the value of the added parity bit is 0, otherwise it is 1.
- **Example:** Consider the transmitted message **1010001**, which has three ones in it. This is turned into odd parity by adding a zero, making the sequence **0 1010001**. Thus, the total number of ones remain at three, an odd number. If the transmitted message has the form **1101001**, which has four ones in it, this can be turned into odd parity by adding a one, making the sequence

1 1101001.

END OF UNIT - 04

-By Hiral Pandya