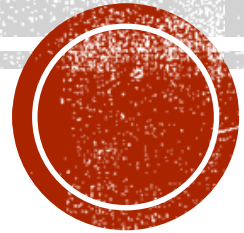
DIGITAL LOGIC

BSC. CSIT.- I/I

COURSE CODE: CSC111



UNIT 1
INTRODUCTION

1.1 Digital Signals and wave forms

- A signal is a function, that represents the variation of a physical quantity with respect to any parameter.
- In electronics circuits and systems, signal can be divided into two broad categories:
  I. ANALOG SIGNAL
  II. DIGITAL SIGNAL

i. **ANALOG SIGNAL: smooth, continuous voltage variations such as voice or video.**
ii. **DIGITAL SIGNAL: binary pulses or codes.**

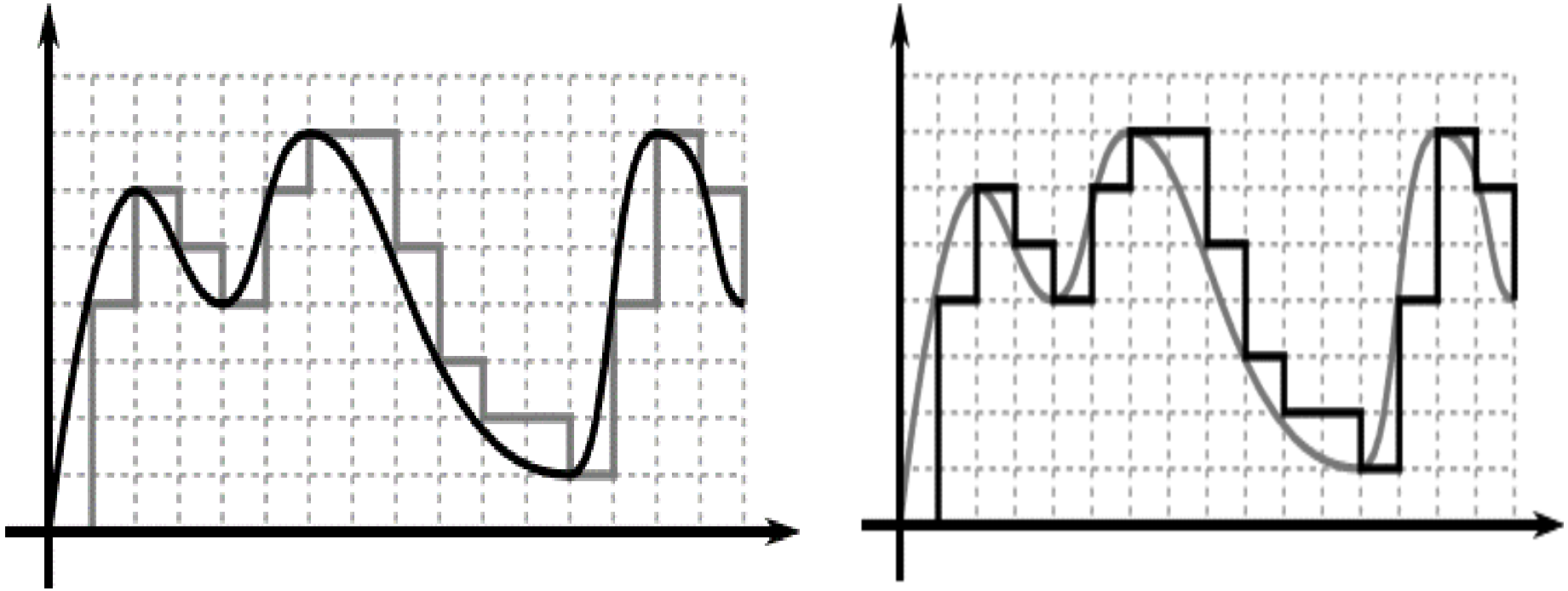


Fig: (a) Analog Signal **(b) Digital Signal**

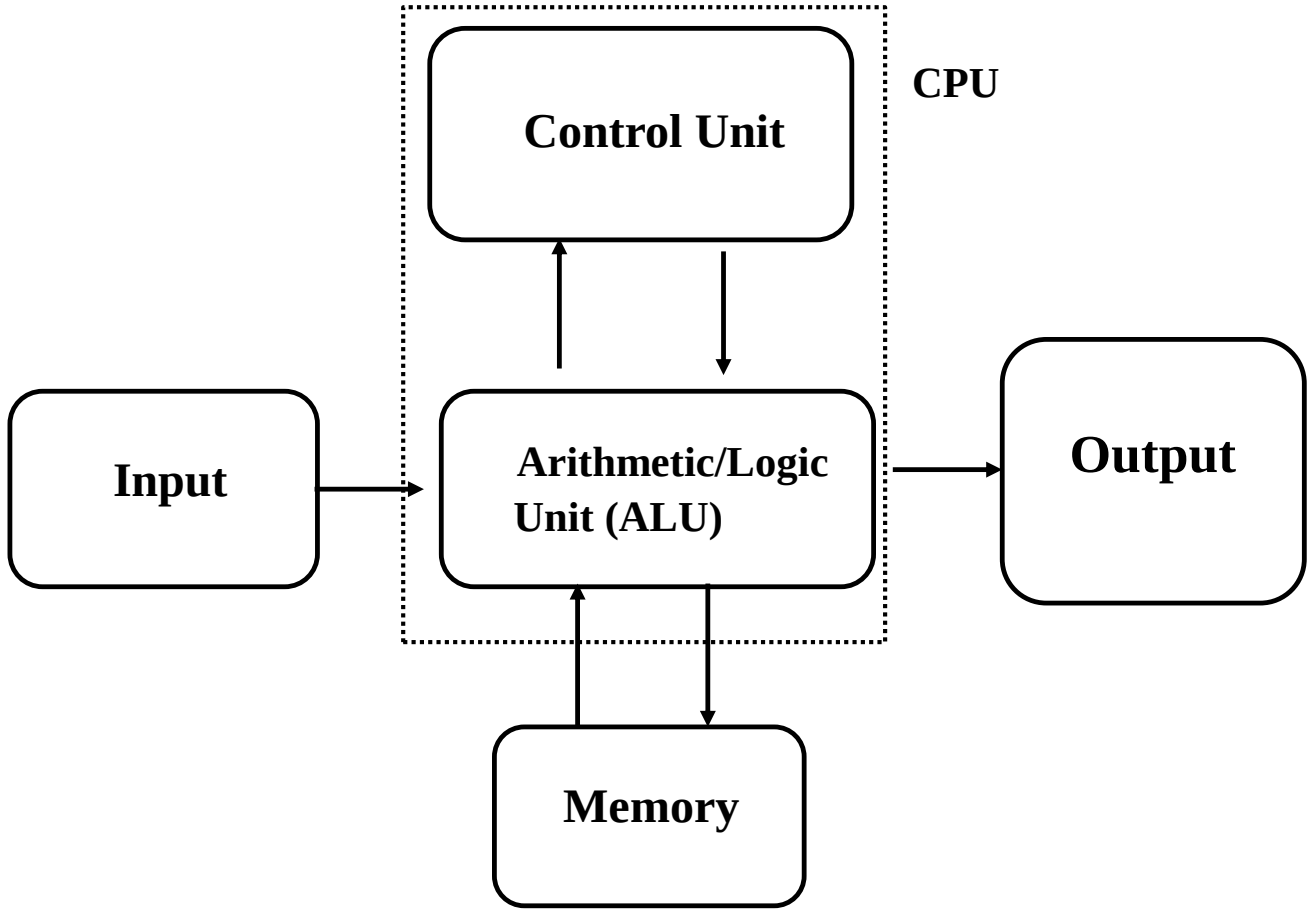
- **Any quantity that changes with time either can be represented as *an analog signal* or it can be treated as *a digital signal*.**
- **For example, the temperature recorded over the 24 hours continuously, and it changes smoothly from 2°C to 20°C is *an analog signal* as all possible value can be represented.**

- In other case, the temperature can be recorded over 24 hours at specific interval which gives a ***discrete point values***.
- When a quantity is measured as a *series of distinct points*, it is said to be *sampled*.
- This is a *digital signal*.

Digital computers

- Electronic device.
- Integration of number of IC units called CPU, registers, memory, and I/O, designed specifically to carry out specific task.
- Takes the analog signals from the outside world, process them digitally and provides the output.
- The output can be digital or anaolog, depends on users choice.

Block diagram of digital computer



Digital computers and integrated ICs

Integrated circuit (IC) and signal levels:

- An electronic device that gathers (or integrates) a number of electronic components on a small semiconductor chip.
- Usually, has a particular functionality, such as amplifying the voltage of a signal or applying a logic AND on 2/3 inputs and it could be broad as a microprocessor.
- Can be digital or analog.
- Usually, analog ICs handle continuous signals, such as audio signals and digital ICs handle discrete signals such as binary values.

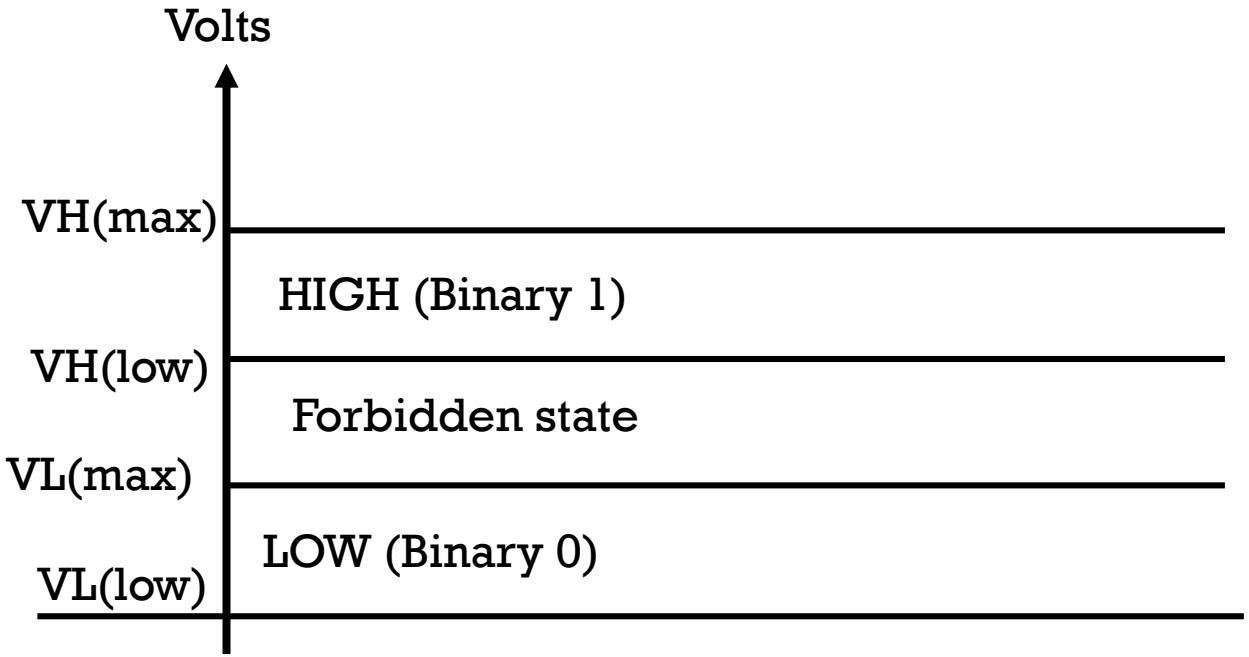


Fig: logic level profile

- In digital system, two digits '0' and '1' are used, called bits (binary digits).
- In digital circuit, two voltage levels are used:
  - High == logic '1'
  - Low == logic '0'
- High is normally +5 volt and low is 0 volt in positive logic and opposite in case of negative logic system.

Clock waveforms

- A waveform is a graphical representation of a signal in the form of a wave.
- Digital waveforms consist of voltage levels that are changing back and forth between the HIGH and LOW levels as states.

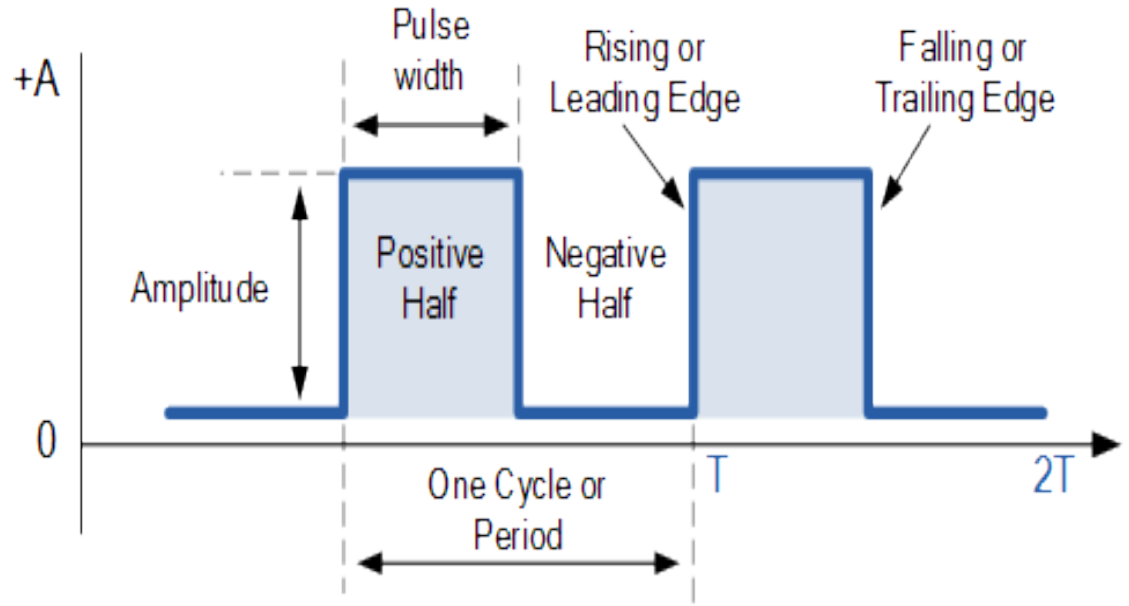


Fig: Digital Waveform

Advantage/disadvantages of digital system

Advantages:

- Cheaper and smaller.
- Speech, video and other data can be merged and transmitted over a single channel using *Multiplexing.*
- Data encryption, high security.
- High tolerance to noise.
- Due to channel coding technique, errors can be detected and corrected.
- Can be conveyed in all direction simultaneously.

Disadvantage:

- Requires high bandwidth.
- Requires synchronization in case of synchronous modulation.

1.2 NUMBER SYSTEM

- Computer communicates and operates in binary digit '0' and '1'.
- Human beings generally use decimal system (0–9).
- Other system can be used: Octal (0-7), Hexadecimal (0-15) where 10-15, we use A-F letters.
- A positional scheme is used to represent a number in any of the number system.
- Radix or base; for example 2 for binary, 10 for decimal etc.

DIFFERENT NUMBER SYSTEM

1. **Decimal number system (base or radix 10)**
   - Used 10-digits (0-9).
   - Means 10 different symbols to represent numbers.
   - E.g.: 2 1 0
     4 3 6=4x10^2+3x10^1+6x10^0
     2 1 0 -1 -2
     8 0 1 2 7=8x10^2+0x10^1+1x10^0+2x10^-1+7x10^-2
2. **Binary number system (base or radix 2)**
   - Used 2-digits (0 and 1).
   - Means two different symbols to represent numbers.
   - E.g.: 2 1 0
     1 0 1=1x2^2+0x2^1+1x2^0
     2 1 0 -1 -2
     1 0 1 1 1=1x2^2+0x2^1+1x2^0+1x2^-1+1x2^-2

3. Octal number system (base or radix 8)

- Used 8-digits (0-7).
- Means 8 different symbols to represent numbers.
- E.g.: 2 1 0

  4 3 6=4x8^2+3x8^1+6x8^0

  2 1 0 -1 -2

  4 0 1 2 7=4x8^2+0x8^1+1x8^0+2x8^-1+7x8^-2

4. Hexa-decimal number system (base or radix 16)

- Used 16-digits (0-15).
- Means 16 different symbols to represent numbers.
- First 10-digits are same as that of decimal number system and rest 10,11,12,13,14 and 15 are represented using letters A,B,C,D,E and F respectively.
- E.g.: 2 1 0

  4 E 6=4x16^2+14x16^1+6x16^0 (here, E=14)

Decimal	Binary	Ocatal	Hexadecimal
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

BASE CONVERSION METHODS

1. DECIMAL NUMBER TO OTHER BASES CONVERSION

A. Decimal number contains both ***integer part*** and ***fractional part***.

B. Convert each parts of decimal number into other base individually as their base (r) mentioned.

   I. Perform division of integer part of given decimal number by base r.

   II. Note the remainders successively until the remainder is zero.

   III. Place the remainders in ***reverse order*** to get the integer part of equivalent number of base 'r'.

   IV. First and last remainders denote the least significant digit and most significant digit respectively.

   V. Do multiplication of fractional part of decimal number and successive fractions with base 'r' and note down the carry till the result is zero.

   VI. Consider the ***normal sequence of carry*** in order to get the fractional part of equivalent number of base 'r'.

1. Decimal to Binary Conversion

- Division of integer part and successive quotients with base 2.
- Multiplication of fractional part and successive fractions with base 2.

Example

Consider the decimal number ***58.25.***
Here, the integer part is ***58*** and fractional part is ***0.25***.
Step 1 – **Division of 58 and successive quotients with base 2.**

Operation	Quotient	Remainder
58/2	29	0 LSB
29/2	14	1
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0	1M SB

$\Rightarrow (58)_{10} = (111010)_{2}$

Step 2 – Multiplication of 0.25 and successive fractions with base 2.

Operation	**Result**	**Carry**
0.25 x 2	0.5	**0**
0.5 x 2	1.0	**1**
-	**0.0**	**-**

⇒ $(.25)_{10} = (.01)_{2}$

Therefore, the binary equivalent of decimal number 58.25 is 111010.01.

2. DECIMAL TO OCTAL CONVERSION

- Division of integer part and successive quotients with base 8.
- Multiplication of fractional part and successive fractions with base 8.

Example

Consider the decimal number 58.25. Here, the integer part is 58 and fractional part is 0.25.

Step 1 – Division of 58 and successive quotients with base 8.

Operation	Quotient	Remainder
58/8	7	2 ↑
7/8	0	7

⇒ (58) $_{10}$ = (72) $_{8}$

Therefore, the integer part of equivalent octal number is 72.

Step 2 – Multiplication of 0.25 and successive fractions with base 8.

Operation	Result	Carry
0.25 x 8	2.00	2
-	0.00	-

⇒ .25 = .2

Therefore, the octal equivalent of decimal number 58.25 is 72.2.

3. DECIMAL TO HEXA-DECIMAL CONVERSION

- Division of integer part and successive quotients with base 16.
- Multiplication of fractional part and successive fractions with base 16.
- Example: Consider the decimal number 58.25. Here, the integer part is 58 and decimal part is 0.25.
- ***Step 1*** – Division of 58 and successive quotients with base 16.

Operation	Quotient	Remainder
58/16	3	10=A
3/16	0	3

⇒ 58 = 3A

- Therefore, the integer part of equivalent Hexa-decimal number is 3A.

Step 2 – Multiplication of 0.25 and successive fractions with base 16.

Operation	Result	Carry
0.25 x 16	4.00	4
-	0.00	-

⇒ $.25_{10} = .4_{16}$

Therefore, the fractional part of equivalent Hexa-decimal number is .4.

⇒$58.25_{10} = 3A.4_{16}$

Therefore, the Hexa-decimal equivalent of decimal number 58.25 is 3A.4.

2. BINARY TO OCTAL CONVERSION

- **The bases of binary and octal number systems are 2 and 8 respectively.**
- **Three bits of binary number is equivalent to one octal digit, since 2^3 = 8.**
- **Two steps:**

  step-1: **Start from the binary point and make the groups of 3 bits on both sides of binary point. If one or two bits are less while making the group of 3 bits, then include required number of zeros on extreme sides.**

  Step 2: **Write the octal digits corresponding to each group of 3 bits.**

- **Example: Consider the binary number 101110.01101.**

  Step 1 – Make the groups of 3 bits on both sides of binary point.

  101 110.011 01

  Here, on right side of binary point, the last group is having only 2 bits. So, include one zero on extreme side in order to make it as group of 3 bits.

  ⇒ 101 110.011 010

- **Step 2 – Write the octal digits corresponding to each group of 3 bits.**
  - **⇒ 101110.011010 2 = 56.32 8**
- **Therefore, the octal equivalent of binary number 101110.01101 is 56.32.**

3. BINARY TO HEXA-DECIMAL CONVERSION

- **The bases of binary and Hexa-decimal number systems are 2 and 16 respectively.**
- **Four bits of binary number is equivalent to one Hexa-decimal digit, since 2^4 = 16.**
- **Two steps:**

Step 1 : **Start from the binary point and make the groups of 4 bits on both sides of binary point. If some bits are less while making the group of 4 bits, then include required number of zeros on extreme sides.**

Step 2: **Write the Hexa-decimal digits corresponding to each group of 4 bits.**

- **Example:** Consider the binary number 101110.01101

Step 1 – Make the groups of 4 bits on both sides of binary point.

10 1110.0110 1

Here, the first group is having only 2 bits. So, include two zeros on extreme side in order to make it as group of 4 bits. Similarly, include three zeros on extreme side in order to make the last group also as group of 4 bits.

⇒ 0010 1110.0110 1000

Step 2 – Write the Hexa-decimal digits corresponding to each group of 4 bits.

⇒ $00101110.01101000_2 = 2E.68_{16}$

- **Therefore, the Hexa-decimal equivalent of binary number 101110.01101 is $2E.68_{16}$.**

COMPLEMENT

- Used in digital computers to simplify subtraction operation and logical manipulation.
- For each radix r, there are two types:
  1. r's Complement (Radix Complement)
  2. (r-1)'s Complement (Diminished Radix Complement)
- **r's Complement:** For a given positive number N in base r with integer part n, the r's complement of N is defines as,

  r^n***-N for N≠0 and 0 for N=0***

- **Ex: find the 10's complement of 2345_{10}**

  Solution: Here, r=10, N=2345 and n=4

  $\therefore$ r's complement is $r^n - N = 10^4 - 2345 = 7655$

- ***For fraction,*** r^0***-N=1-N, where N is the fraction.***

(R-1)'S COMPLEMENT

- For a given positive number N in base r with integer part n and fractional part m, the (r-1)'s complement of N is defines as,

$$r^n - r^{-m} - N$$

- **Ex: find the 9's complement of 2345_{10}**

  Solution: Here, r=10, N=2345 and n=4

  $\therefore$ (r-1)'s complement is $(r^n - 1) - N = (10^4 - 1) - 2345 = 7654$

- **Ex: find the 9's complement of 2345.55_{10}**

  Solution: Here, r=10, N=2345 and n=4 and m=2

  $\therefore$ (r-1)'s complement is $r^n - r^{-m} - N = 10^4 - 10^{-2} - 2345.55 = 7654.44$

BINARY NUMBER REPRESENTATION

- **Two Types:**
  1. **Signed number representation**
  2. **Unsigned Number representation**

1. Signed Numbers

- Contain both sign and magnitude of the number.
- The sign is placed in front of number. So, we have to consider the **positive sign** for positive numbers and **negative sign** for negative numbers.

2. Unsigned Numbers

- Contain only magnitude of the number.
- They don't have any sign. That means all unsigned binary numbers are **positive** including Zero.

REPRESENTATION OF UNSIGNED BINARY NUMBERS:

- The bits present in the un-signed binary number holds the magnitude of a number.
- That means, if the un-signed binary number contains ***'N' bits, then all N bits represent the magnitude*** of the number, since it doesn't have any sign bit.
- Example: $(7)_{10} = (0111)_2$
  - It has 4-bits and all represents its magnitude.

REPRESENTATION OF SIGNED BINARY NUMBERS:

- The Most Significant Bit (MSB) of signed binary numbers is used to indicate the sign of the numbers.
- "+" is represented by '0'
- "-" is represented by '1'
- For 'N' bits length number , only N − 1 bits represent the magnitude of the number as 1-bit MSB is reserved for representing sign of the number.
- Three types of representations:
  i. ***Sign-Magnitude form***
  ii. ***1's complement form***
  iii. ***2's complement form***
- Representation of a positive number in all these 3 forms is same. But, only the representation of negative number will differ in each form.

1. SIGN-MAGNITUDE FORM

- The MSB is used for representing sign of the number and the remaining bits represent the magnitude of the number.
- Similar to signed decimal number representation.
- Example: Consider the negative decimal number -108.

  Magnitude = 108.

  Unsigned binary representation = 1101100.

  All these bits represent the magnitude.
- Since the given number is negative, consider the ***sign bit as one***, which is ***placed on left most side of magnitude.***

  −108 = **1**1101100
- Therefore, the sign-magnitude representation of -108 is **1**1101100.

2. 1's Complement Form

- The 1's complement of a number is obtained by complementing all the bits of signed binary number., i.e., '1' is changed to '0' and vice versa.
- 1's complement of positive number gives a negative number.
- Similarly, 1's complement of negative number gives a positive number.
- Example: Consider the negative decimal number -108.

  The magnitude of this number is 108.

  We know the signed binary representation of 108 is 01101100.
- It is having 8 bits with MSB zero, i.e., the number is positive.
- Complement of '0' is '1' and vice-versa. So, replace zeros by ones and ones by zeros in order to get the negative number.

$$-108_{10} = 10010011_{2}$$

- Therefore, the 1's complement of 108_{10} is 10010011_{2}.

3. 2's Complement Form

- Obtained by adding one to the 1's complement of signed binary number.
- So, 2's complement of positive number gives a negative number. Similarly, 2's complement of negative number gives a positive number.
- ***That means, if you perform two times 2's complement of a binary number including sign bit, then you will get the original signed binary number.***
- Example: Consider the negative decimal number -108.

  We know the 1's complement of $(108)_{10}$ is $(10010011)_2$
- 2's compliment of 108_{10} = 1's compliment of 108 + 1.

  = 10010011 + 1

  = 10010100
- Therefore, the 2's complement of -108_{10} is 10010100_2.

2. BINARY NUMBER TO OTHER BASES CONVERSION

1. Binary to Decimal conversion

- Multiply the bits of binary number with the respective positional weights and then add all those products.
- Example: Consider the binary number (1101.11).

  Mathematically, we can write it as,

  1101.11 = (1 × 2^3) + (1 × 2^2) + (0 × 2^1) + (1 × 2^0) + (1 × 2^-1) + (1 × 2^-2)

  ⇒ 1101.11 = 8 + 4 + 0 + 1 + 0.5 + 0.25 = 13.75

  ⇒ 1101.11 = 13.75 10
- Therefore, the decimal equivalent of binary number 1101.11 is 13.75.

BINARY ARITHMETIC

1. Signed Binary Arithmetic:

- **Addition of two signed binary numbers:**
  - **Example 1:** Let us perform the addition of two decimal numbers +7 and +4 using 2's complement method.
  - The 2's complement representations of +7 and +4 with 5 bits each are shown below.

$$+7_{10} = 00111_2$$
$$+4_{10} = 00100_2$$

  - The addition of these two numbers is

$$+7 \ +(+4) = 00111 \ +00100$$
$$\Rightarrow +7 \ +(+4) = 01011$$

  - The resultant sum contains 5 bits. So, there is no carry out from sign bit. The sign bit '0' indicates that the resultant sum is positive. So, the magnitude of sum is 11 in decimal number system.
  - Therefore, ***addition of two positive numbers*** will give another ***positive number***.

Example 2:

- Let us perform the addition of two decimal numbers -7 and -4 using 2's complement method.
- The 2's complement representation of -7 and -4 with 5 bits each are shown below.

$$-7_{10} = 11001_2$$

$$-4_{10} = 11100_2$$

- The addition of these two numbers is

$$-7_{10} + -4_{10} = 11001_2 + 11100_2$$

$$\Rightarrow -7 + -4 = \mathbf{1}10101$$

- The resultant sum contains 6 bits. In this case, carry is obtained from sign bit. So, we can remove it.
- Resultant sum after removing carry is $-7 + -4 = 10101$.
- The sign bit '1' indicates that the resultant sum is negative.

2. Subtraction of two Signed Binary Numbers:

- Consider the two signed binary numbers A & B, which are represented in 2's complement form.
- 2's complement of positive number gives a negative number.
- subtract a number B from number A, then take 2's complement of B and add it to A.

$$A - B = A + 2's\ complement\ of\ B$$

- subtract a number A from number B, then take 2's complement of A and add it to B.

$$B - A = B + 2's\ complement\ of\ A$$

- So, the subtraction of two signed binary numbers is similar to the addition of two signed binary numbers. But, we have to take 2's complement of the number, which is supposed to be subtracted.

Example 3

- **Let us perform the subtraction of two decimal numbers +7 and +4 using 2's complement method. The subtraction of these two numbers is,**

$$+7_{10} - +4_{10} = +7_{10} + -4_{10}.$$

- **The 2's complement representation of +7 and -4 with 5 bits each are shown below:**

$$+7_{10} = 00111_2$$
$$-4_{10} = 11100_2$$
$$\Rightarrow +7_{10} + -4_{10} = 00111_2 + 11100_2 = 100011_2$$

Here, the carry obtained from sign bit. So, we can remove it. The resultant sum after removing carry is,

$$+7_{10} + -4_{10} = 00011_2$$

- **The sign bit '0' indicates that the resultant sum is positive. So, the magnitude of it is 3 in decimal number system. Therefore, subtraction of two decimal numbers +7 and +4 is +3.**

DIGITAL CIRCUITS - CODES

- The group of symbols is called as code.
- Binary codes can be classified into two types.
  1. ***Weighted codes***
  2. ***Unweighted codes***
- If the code has positional weights, then it is said to be weighted code. Otherwise, it is an unweighted code.
- Weighted codes can be further classified as positively weighted codes and negatively weighted codes.

Binary codes for decimal digits:

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

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 known as natural ***Binary Coded Decimal*** code.

Example

Let us find the BCD equivalent of the decimal number **786**.

This number has 3 decimal digits 7, 8 and 6.

From the table, we can write the BCD 8421 codes of **7, 8** and **6 are 0111, 1000 and 0110 respectively.**

$\therefore 786_{10} = \mathbf{011110000110}_{BCD}$

- There are 12 bits in BCD representation, since each BCD code of decimal digit has 4 bits.

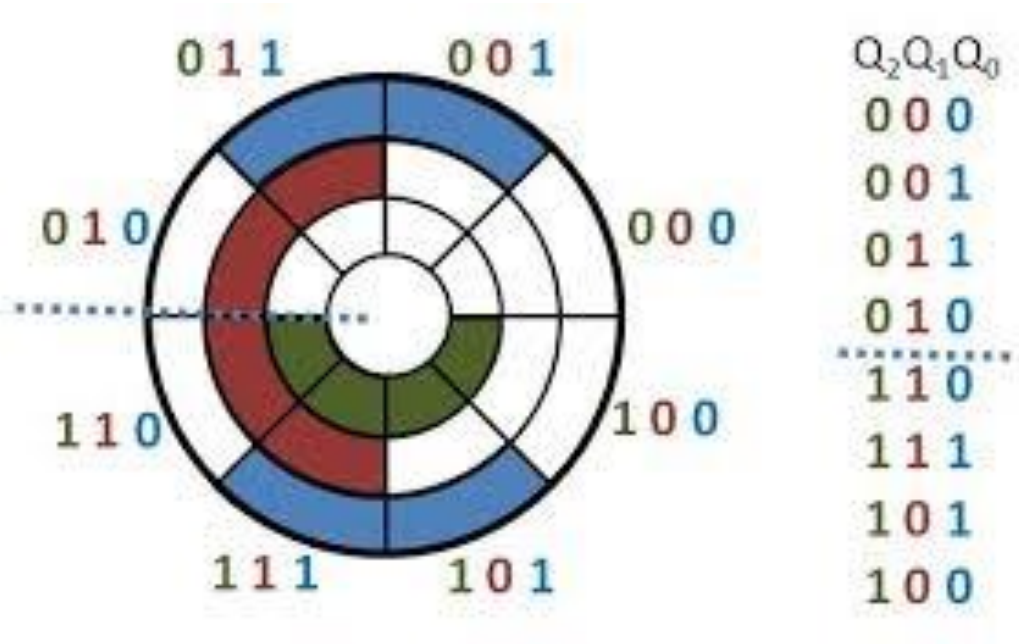
EXCESS 3 CODE

- This code doesn't have any weights. So, it is an **un-weighted code**.
- We will get the Excess 3 code of a decimal number by adding three 0011 to the binary equivalent of that decimal number. Hence, it is called as Excess 3 code.
- It is a **self-complementing** code.
- **Example**: Excess 3 equivalent of the decimal number 786.
  - This number has 3 decimal digits 7, 8 and 6.
  - From the table, we can write the Excess 3 codes of 7, 8 and 6 are 1010, 1011 and 1001 respectively.
  - Therefore, the Excess 3 equivalent of the decimal number 786 is **101010111001.**

GRAY CODE

- This code doesn't have any weights. So, it is an **un-weighted code**.
- The successive Gray codes are differed in one bit position only. Hence, this code is called as **unit distance** code.

Decimal Number	Binary Code	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



BINARY CODE TO GRAY CODE CONVERSION

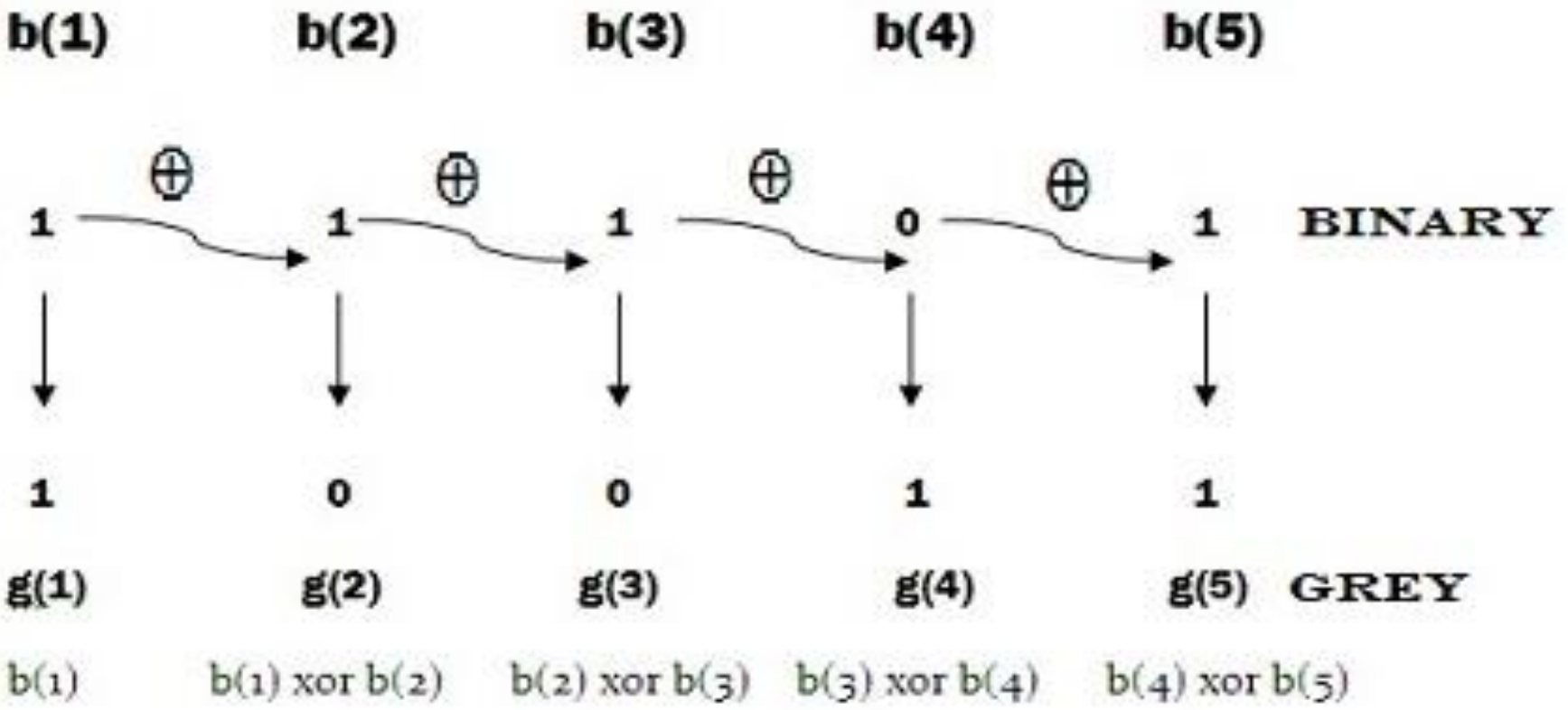
- **Follow these steps:**
  1. **Consider the given binary code and place a zero to the left of MSB.**
  2. **Compare the successive two bits starting from zero. If the 2 bits are same, then the output is zero. Otherwise, output is one.**
  3. **Repeat the above step till the LSB of Gray code is obtained.**
- **Example: Convert binary code 11101 to Grey.**

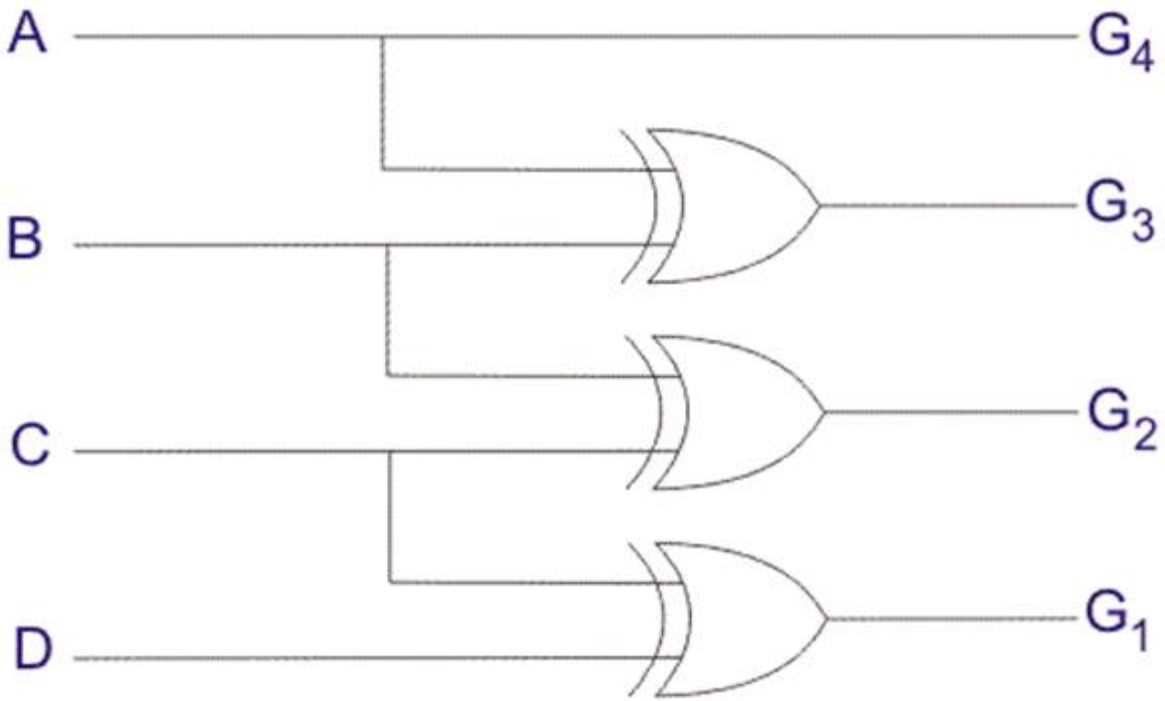
  Step 1 – By placing zero to the left of MSB, the binary code will be 11101.

  Step 2 – By comparing successive two bits of new binary code, we will get the gray code as **10011**.

Binary to Grey Code Conversion

Convert the binary 11101_2 to its equivalent Grey code





Binary to Gray Code Conversion

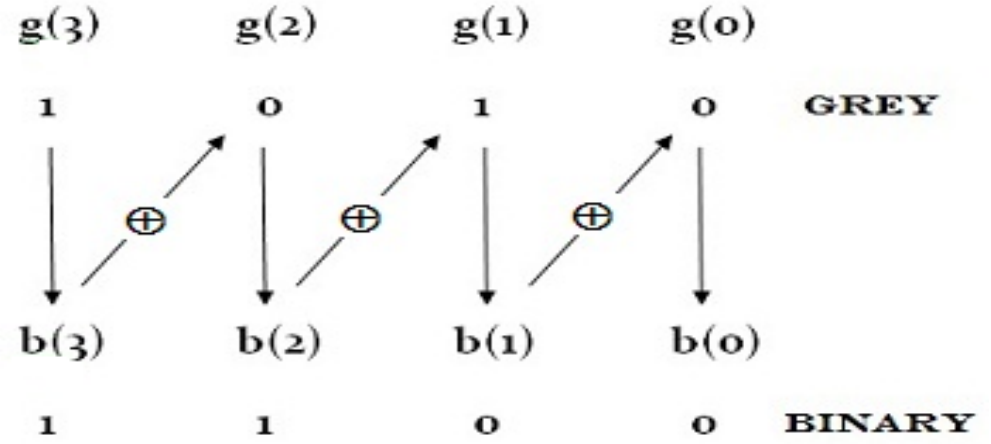
GRAY TO BINARY CONVERSION

Follow these steps:

1. **Consider the given Gray code and place the MSB as it is to the result.**
2. **Compare the successive bit and result. If the 2 bits are same, then the output is zero. Otherwise, output is one.**
3. **Repeat the above step till the LSB of Gray code is obtained.**

Grey Code to Binary Conversion

Convert the Grey code 1010 to its equivalent Binary



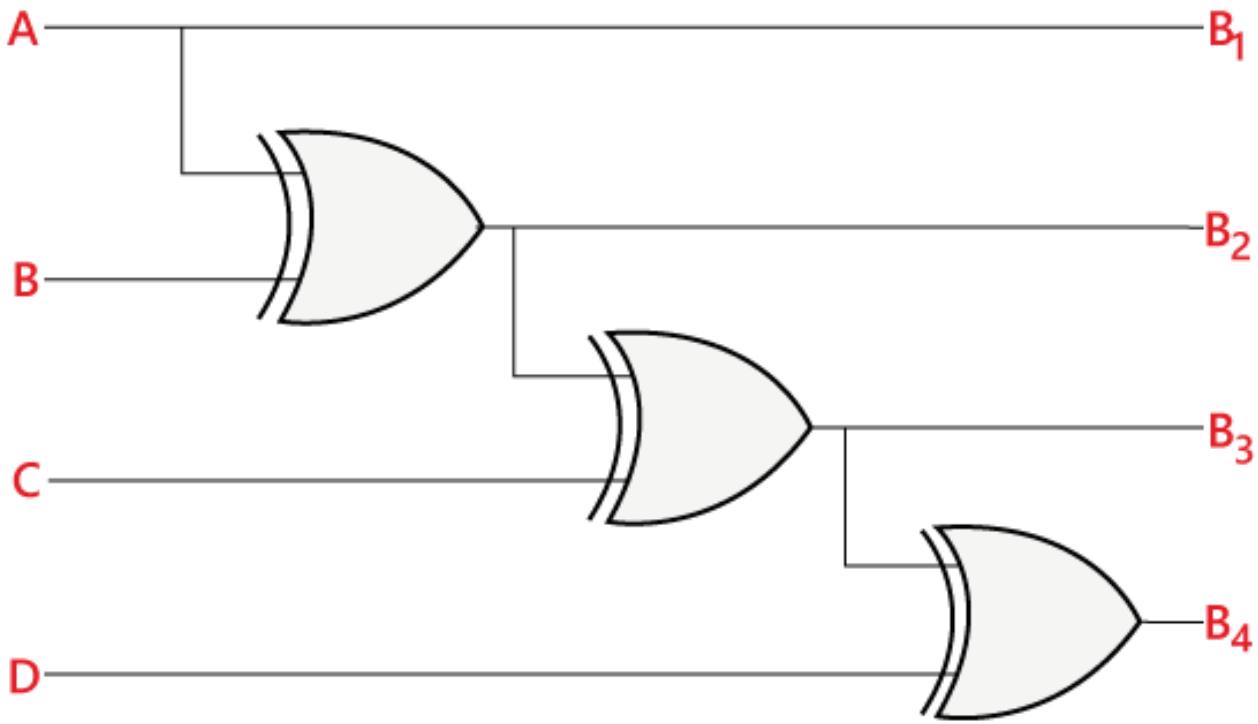
i.e

$$b(3) = g(3)$$

$$b(2) = b(3) \oplus g(2)$$

$$b(1) = b(2) \oplus g(1)$$

$$b(0) = b(1) \oplus g(0)$$



Logic Circuit for Gray to Binary Code Converter

ERROR DETECTING AND CORRECTING CODES

- **Error detection codes** – are used to detect the errors present in the received data bit stream.
- Examples: Parity code, Hamming code.
- **Error correction codes** – are used to correct the errors present in the received data bit stream.
- **Example** – Hamming code.

Parity Code:

- A ***parity bit(s)*** is ***an extra bit*** that is added with original message to detect error in the message during data transmission.
- This is a simplest method for error detection.

Even Parity

One bit is attached to the information so that the total number of 1 bit is an even number.

Message	Parity
1011001	0
1010010	1

Odd Parity

One bit is attached to the information so that the total number of 1 bit is an odd number.

Message	Parity
1011001	1
1010010	0

Binary Code	Even Parity bit	Even Parity Code
000	0	0000
001	1	0011
010	1	0101
011	0	0110
100	1	1001
101	0	1010
110	0	1100
111	1	1111

Even Parity Table

Binary Code	Odd Parity bit	Odd Parity Code
000	1	0001
001	0	0010
010	0	0100
011	1	0111
100	0	1000
101	1	1011
110	1	1101
111	0	1110

Odd Parity Table