Part – 5: Digital Circuits

5.1: Number Systems & Code Conversions

Characteristics of any number system are:

1. Base or radix is equal to the number of possible symbols in the system
2. The largest value of digit is one (1) less than the radix

Decimal to Binary Conversion:

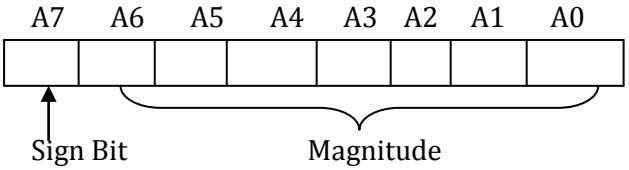
(a) Integer number: Divide the given decimal integer number repeatedly by 2 and collect the remainders. This must continue until the integer quotient becomes zero.

(b) Fractional Number: Multiply by 2 to give an integer and a fraction. The new fraction is multiplied by 2 to give a new integer and a new fraction. This process is continued until the fraction becomes 0 or until the numbers of digits have sufficient accuracy.

Note: To convert a decimal fraction to a number expressed in base r, a similar procedure is used. Multiplication is by r instead of 2 and the coefficients found from the integers any range in value from 0 to (r-1).

The conversion of decimal number with both integer and fraction parts separately and then combining the answers together.

- Don't care values or unused states in BCD code are 1010, 1011, 1100, 1101, 1110, 1111.
- Don't care values or unused state in excess – 3 codes are 0000, 0001, 0010, 1101, 1110, 1111.
- The binary equivalent of a given decimal number is not equivalent to its BCD value. Eg. Binary equivalent of 25_{10} is equal to 11001_2 while BCD equivalent is 00100101.
- In signed binary numbers,MSB is always sign bit and the remaining bits are used for magnitude.



- For positive and negative binary number, the sign is respectively '0' and '1'.
- Negative numbers can be represented in one of three possible ways.
  1. Signed – magnitude representation.
  2. Signed – 1's complement representation.
  3. Signed – 2's complement representation.

Example: +9 -9

Signed – magnitude 0 0001001 (a) 1 000 1001 signed – magnitude

(b) 1 111 0110 signed – 1's complement

(c) 1 111 0111 signed – 2's complement

- **Subtraction using 2's complement:** Represent the negative numbers in signed 2's complement form, add the two numbers, including their sign bit and discard any carry out of the most significant bit.
- Since negative numbers are represented in 2's complement form, negative results also obtained in signed 2's complement form.
- The range of binary integer number of n-bits using signed 1's complement form is given by $+(2^{n-1} - 1)$ to $-(2^{n-1} - 1)$,which includes both types of zero's i.e., +0 and -0.

* The range of integer binary numbers of n-bits length by using signed 2's complement representation is given by $+ (2^{n-1} - 1)$ to $- 2^{n-1}$ which includes only one type of zero i.e. + 0.
* In weighted codes, each position of the number has specific weight. The decimal value of a weighted code number is the algebraic sum of the weights of those positions in which 1's appears.
* Most frequently used weighted codes are 8421, 2421 code, 5211 code and 84 2'1' code.

5.2: Boolean Algebra & Karnaugh Maps

- Boolean properties:
  a) Properties of AND function

1. $X \cdot 0 = 0$
2. $0 \cdot X = 0$
3. $X \cdot 1 = X$
4. $1.X = X$

  b) Properties of OR function

5. $X + 0 = X$
6. $0 + X = X$
7. $X + 1 = 1$
8. $1 + X = 1$

  c) Combining a variable with itself or its complement

9. $X.X' = 0$
10. $X \cdot X = X$
11. $X + X = X$
12. $X + X' = 1$
13. $(X')' = X$

  d) Commutative laws: 14. $x.y = y.x$ 15. $x + y = y + x$
  e) Distributive laws: 16. $x(y+z) = x.y + x.z$ 17. $x + y.z = (x+y)(x+z)$
  f) Associative laws: 18. $x(y.z) = (x.y)z$ 19. $x + (y+z) = (x+y) + z$
  g) Absorption laws: 20. $x + xy = x$ 21. $x(x+y) = x$
  22. $x + x'y = x + y$ 23. $x(x'+y) = xy$
  h) Demorgan's laws: 24. $(x+y)' = x'.y'$ 25. $(x.y)' = x' + y'$

- Duality principle: It states that every algebraic expression deducible from theorems of Boolean algebra remains valid if the operators and identify elements are interchanged.
- To get dual of an algebraic function, we simply exchange AND with OR and exchange 1 with 0.
- The dual of the exclusive – OR is equal to its complement.
- To find the complement of a function is take the dual of the function and complement each literal.
- Maxterm is the compliment of its corresponding minterm and vice versa.
- Sum of all the minterms of a given Boolean function is equal to 1.
- Product of all the maxterms of a given Boolean function is equal to 0

Boolean Algebraic Theorems

Theorem No.	Theorem
1.	$(A + B).(A + \bar{B}) = A$
2.	$AB + \bar{A}C = (A + C)(\bar{A} + B)$
3.	$(A + B)(\bar{A} + C) = AC + \bar{A}B$
4.	$AB + \bar{A}C + BC = AB + \bar{A}C$
5.	$(A + B)(\bar{A} + C)(B + C) = (A + B)(\bar{A} + C)$
6.	$\overline{A.B.C.\ldots} = \bar{A} + \bar{B} + \bar{C} + \cdots$
7.	$\overline{A + B + C + \ldots} = \bar{A}.\bar{B}.\bar{C}\ldots$

Karnaugh Maps (K – maps)

- A map is a diagram made up of squares. Each square represents either a minterm or a maxterms.
- The number of squares in the karnaugh map is given by 2^n where n = number of variable.
- Gray code sequence is used in K – map so that any two adjacent cells will differ by only one bit.

Number of variables	No. of cells containing 1's grouped	No. of variables eliminated	No. of literals present in the resulting term
2	4	2	0
	2	1	1
	1	0	2
3	8	3	0
	4	2	1
	2	1	2
	1	0	3
4	16	4	0
	8	3	1
	4	2	2
	2	1	3
	1	0	4

5.3: Logic Gates

- OR, AND, NOT are basic gates
- NAND and NOR gates are called Universal gates because, by using only NAND gates or by using only NOR gates we can realize any gate or any circuit.
- EXOR, EXNOR are arithmetic gates.

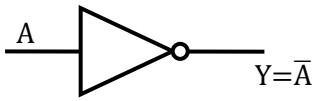
- **There are two types of logic systems**
  1) **Positive level logic system (PLLS)** : Out of the given two voltage levels, the more positive value is assumed as logic '1' and the other as logic '0'.
  2) **Negative level logic system (NLLS)**:out of the given two voltage levels, the more negative value is assumed as logic '1' and the other as logic '0'.

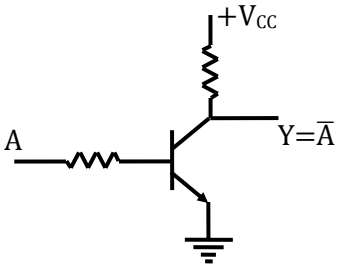
- **NOT gate:-**

Truth Table

A	Y
0	1
1	0

Symbol

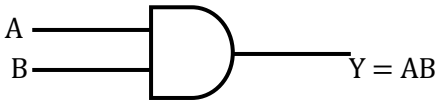


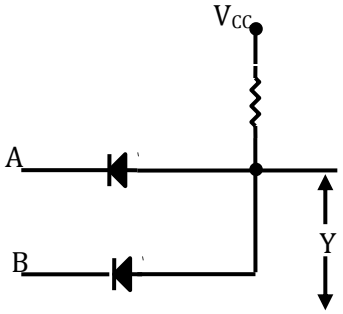


- **AND gate:**

Truth Table

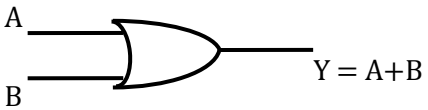
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

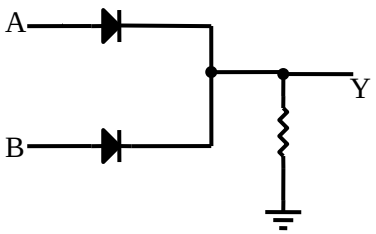




- **OR gate:**

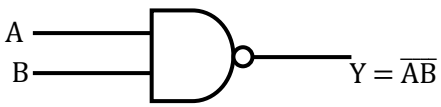
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1





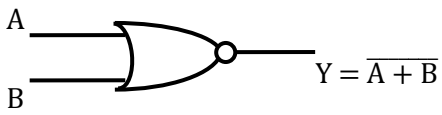
- **NAND gate:**

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



- **NOR gate:**

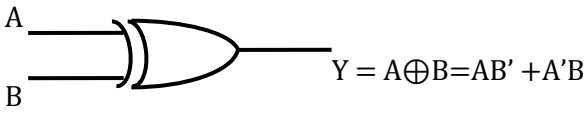
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



- The circuit, which is working as AND gate with positive level logic system, will work as OR gate with negative level logic system and vice-versa.
- The circuit which is behaving as NAND gate with positive level logic system will behave as NOR gate with negative level logic system and vice – versa.

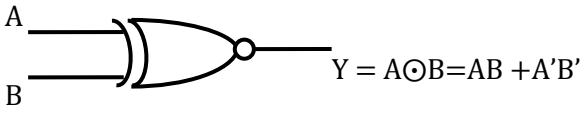
- **Exclusive OR gate (X– OR):** "The output of an X – OR gate is high for odd number of high inputs".

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



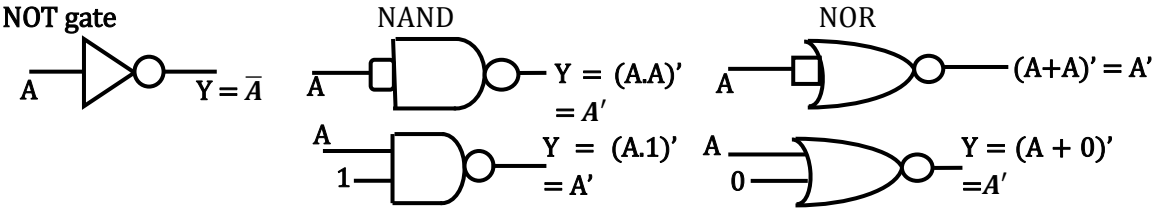
- **Exclusive NOR gate (X–NOR):** The output is high for odd number of low inputs". (OR) "The output is high for even number of high inputs".

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

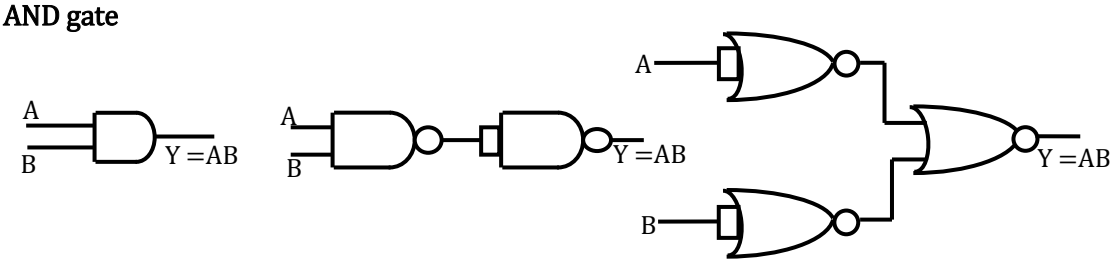


- **Realization of Basic gates using NAND and NOR gates:**

1. **NOT gate**



2. **AND gate**



3. **OR gate:**