UNIT-I : Fundamentals of Digital Electronics

1. 1. Introduction

Digital electronics is a type of electronics that deals with the digital systems which processes the data/information in the form of binary(0s and 1s) numbers, whereas analog electronics deals with the analog systems which processes the data/information in the form of continuous signals.

Continuous signals

A Continuous signal is function f(t), whose value is defined for all time 't'.
in other words
Continuous signal a varying quantity with respect to independent variable time.
Example: Figure 1.1(a) shows the continuous signal.

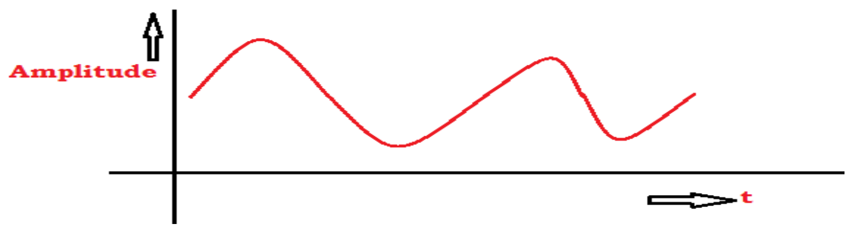


Figure 1.1(a): Continuous signals.

Digital signals

A digital signal is a quantized discrete time signals.
Example: Figure 1.1(b) shows the discrete and digital signals.

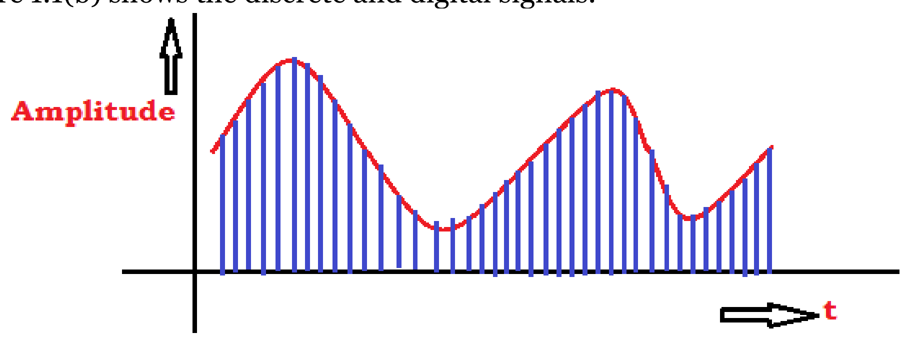


Figure 1.1(b1): Discrete signal.

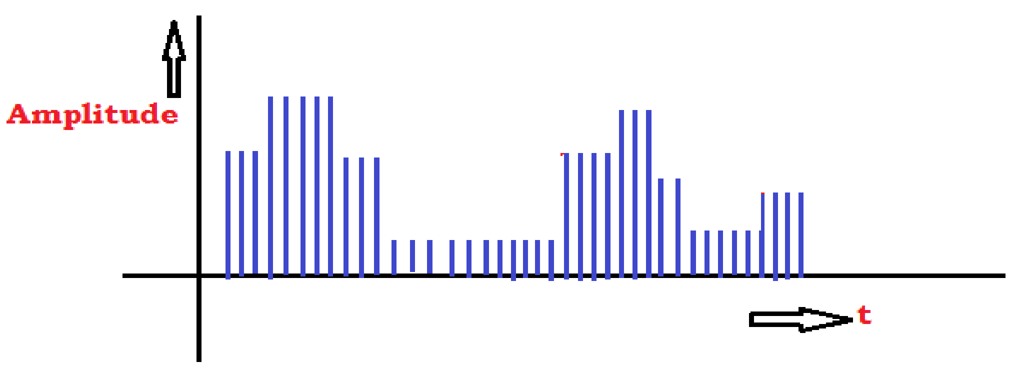


Figure 1.1(b2): Digital signal.

1.2. Boolean Algebra

Boolean algebra is a branch of Algebra (Mathematics) that deals with operations on logical values with Boolean variables, Boolean variables are represented as binary numbers which takes logic 1 and logic 0 values. Hence, the Boolean algebra is also called two-valued logic, Binary Algebra or Logical Algebra. The Boolean algebra was introduced by great mathematician George Boole in 1847. The Boolean algebra is a fundamental for the development of digital electronic systems, and is provided for in all programming languages. Set theory and statistics fields also uses Boolean algebra for the representation, simplification and analysis of mathematical quantities.

Logic levels are classified into two types

1. Positive logic

Logic 0 = False, 0V, Open Switch, OFF
Logic 1= True, +5V, Closed Switch, ON

2. Negative logic

Logic 0 = True, +5V, Closed Switch, ON
Logic 1= False, 0V, Open Switch, OFF

Boolean algebra differs from normal or elementary algebra. Latter deals with numerical operations such as, addition, subtraction, multiplication and division on decimal numbers. And former deals with the logical operations such as conjunction (OR), disjunction(AND) and negation(NOT).

In present context, positive logic has been used for the entire discussion, representation and simplification of Boolean variables.

1.2.1. Rules and properties of Boolean Algebra

1. Boolean variables takes only two values, logic 1 and logic 0, called binary numbers.
2. Basic operations of Boolean algebra are complement of a variable, ORing and ANDing of two or more variables.
3. Mathematical description of Boolean operations using variables is called Boolean expression.
4. Complement of variable is represented by an over-bar (-).
   Example: $Y = \bar{A}$, Y is the output variable
5. ORing of variables is represented by a plus symbol (+)
   Example:$Y = A + B$, Y is the output variable
6. ANDing of variables is represented by a dot symbol (.)
   Example:$Y = A.B$, Y is the output variable
7. Boolean operations are different from binary operations.
   Example : 1+1=10 in Binary Addition
   1+1=1 in Boolean algebra.

Table 1.1, shows the complement operation of a variable, table 1.2 summarized the OR operation and table 1.3, summarized the AND operation of two variables.

A	$\boldsymbol{Y = \bar{A}}$
0	1
1	0

Table 1.1: Complement of variable A

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Table 1.2: OR operation on A and B

A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Table 1.3: AND operation on A and B

The present chapter deals with the simplification of Boolean expressions and representation using sum of product form and product of sum forms.

1.2.2. Boolean Laws:

Law-1: Commutative law

The sequence of changing the variables does not effect on the result even after changing their sequence while performing OR/AND operations of Boolean expression.

$$\text{i.e.}, \mathrm{A.B = B.A \text{ and } A + B = B + A}$$

Law-2: Associative law

The order of operations on variables is independent.

$$\mathrm{A.(B.C) = (A.B).C \text{ and } A + (B + C) = (A + B) + C}$$

Law-3: Distributive Laws

$$A.(B + C) = A.B + A.C$$
$$A + BC = (A + B)(A + C)$$

Law-4: AND Laws

$$A.0 = 0$$
$$A.1 = A$$
$$A.A = A$$
$$A.\bar{A} = 0$$

Law-5: OR Laws

$$A + 0 = A$$
$$A + 1 = 1$$
$$A + A = A$$
$$A + \bar{A} = 1$$

Law-6: Inversion/Complement/NOT Laws

$$\bar{0} = 1$$

$$\bar{1} = 0$$
$$\bar{\bar{A}} = A$$

Law-7: Absorption Law

$$\boldsymbol{A(A + B) = A}$$
$$\boldsymbol{A + AB = A}$$
$$\boldsymbol{A + \bar{A}B = A + B}$$

Law-8: Demargon's Laws

De-Morgan's First Law

Statement: Sum of complement of two or more variables is equal to the product of the complement of their variables.

$$\text{i. e.,}\ \overline{\mathrm{A} + \mathrm{B} + \mathrm{C} + \cdots ..} = \bar{\mathrm{A}}.\bar{\mathrm{B}}.\overline{\mathrm{C} \ldots ..}$$

Proof:

consider three variables for the proof shown in figure 1.4

A	B	C	$\overline{A+B+C}$	$\bar{A}.\bar{B}.\bar{C}$
0	0	0	**1**	**1**
0	0	1	**0**	**0**
0	1	0	**0**	**0**
0	1	1	**0**	**0**
1	0	0	**0**	**0**
1	0	1	**0**	**0**
1	1	0	**0**	**0**
1	1	1	**0**	**0**

Table 1.4: De-Morgan's First Law

De-Morgan's Second Law

Statement: Product of complement of two or more variables is equal to the sum of the complement of their variables.

$$\text{i. e.,}\ \overline{\mathrm{ABC} + \cdots ..} = \cdots \bar{A} + \bar{B} + \bar{C} \ldots.$$

Proof:

consider three variables for the proof shown in figure 1.5

A	B	C	$\overline{A.B.C}$	$\bar{A} + \bar{B} + \bar{C}$
0	0	0	**1**	**1**
0	0	1	**1**	**1**
0	1	0	**1**	**1**
0	1	1	**1**	**1**
1	0	0	**1**	**1**
1	0	1	**1**	**1**
1	1	0	**1**	**1**
1	1	1	**0**	**0**

Table 1.5: De-Morgan's Second Law

Boolean expressions must be simplified and evaluated using the order of operator precedence shown in table 1.6

$F = (\overline{Y + X}) + (\overline{X} + Y)$
$F = \overline{Y}.\overline{X} + \overline{X} + Y$
$\mathbf{F = Y + \overline{X}}$

12. If $\mathbf{f(A, B) = A + B}$, then show that $\mathbf{f(f(X, YZ), \overline{X}Y) = X + Y}$
Solution:
$f(A, B) = f(X, YZ)$
$\therefore A = X \text{ and } B = YZ$
$f(X, YZ) = X + YZ$
$f(f(X, YZ), \overline{X}Y) = f(A + B)$
$A = f(X, YZ) \text{and } B = \overline{X}Y$
$f(f(X, YZ), \overline{X}Y) = (X + YZ) + \overline{X}Y$
$f(f(X, YZ), \overline{X}Y) = (X + \overline{X}Y) + YZ$
$f(f(X, YZ), \overline{X}Y) = (X + Y) + YZ$
$f(f(X, YZ), \overline{X}Y) = \big(X + Y(1 + Z)\big)$
$\mathbf{f(f(X, YZ), \overline{X}Y) = (X + Y)}$

13. Let $\mathbf{A * B = \overline{A} + B}$ and $\mathbf{C = A * B}$ then $\mathbf{C * A}$ is ________
Solution:
$C * A = \overline{C} + A$
$C * A = \overline{(A * B)} + A$
$C * A = \overline{(\overline{A} + B)} + A$
$C * A = A.\overline{B} + A$
$\mathbf{C * A = A}$

1.3. Realization of Boolean expressions using logic gates.

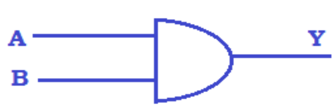
Logic gate is the basic building block of any digital circuits. The logic gates may have one or more inputs and only one output. The relationship between input and output is based on a certain logic, which is same as Boolean operations, such as AND, OR and NOT.

Based on the Boolean operations, the gates are named as AND gate, OR gate and NOT gate. These three gates are called basic gates, and some more gates can be derived by using the basic gates, they are named as NAND gate, NOR gate, EXOR gate and XNOR gate. NAND and NOR gates are called universal gates, because by using only the NAND gates /NOR gates we can realize all basic gates even all Boolean expression.

Logic gates, its truth table, expression and symbols are summarized in the table 1.7 as follows.

Sl. No. **Gate name and Logic Symbol** **Truth table and Logical Expression**

1 AND Gate



Inputs		**Output**
A	**B**	$\boldsymbol{Y = A.B}$
0	0	0
0	1	0
1	0	0
1	1	1

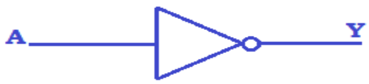
2 OR Gate

Inputs		**Output**
A	**B**	$\boldsymbol{Y = A + B}$

0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

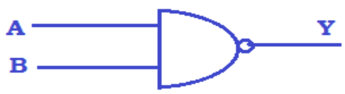
3



Inputs	**Output**
A	$Y = \bar{A}$
0	1
1	0

NAND Gate

4



Inputs		**Output**
A	**B**	**Y**
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate

5

Inputs		**Output**
A	**B**	**Y**
0	0	1
0	1	0
1	0	0
1	1	0

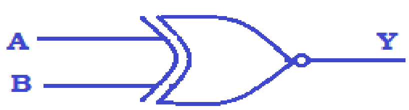
EX-OR Gate

6

Inputs		**Output**
A	**B**	**Y**
0	0	0
0	1	1
1	0	1
1	1	0

EX-NOR Gate

7



Inputs		**Output**
A	**B**	**Y**
0	0	1
0	1	0
1	0	0
1	1	1

1.3.1. Realize the following Boolean expression using basic gates.

Y = AB + BC + AC

Logic diagram

$$Y(A, B, C) = (M_0, M_3, M_6)$$

OR

$$Y(A, B, C) = \prod M(0,3,6)$$

Problem: Refer the truth table shown in table 1.9., write the Boolean expression in canonical POS form and minimal POS form. Also write the different ways of writing canonical SOP form.

A	B	C	Y
0	**0**	**0**	**0**
0	0	1	1
0	1	0	1
0	**1**	**1**	**0**
1	**0**	**0**	**0**
1	0	1	1
1	1	0	1
1	1	1	1

Standard or canonical form representation

$$Y = (A + B + C)(A + \overline{B} + \overline{C})(\overline{A} + B + C)$$

other representations

$$Y(A, B, C) = (M_0, M_3, M_4)$$

OR

$$Y(A, B, C) = \prod M(0,3,4)$$

Simplification of Boolean expression using Boolean algebra rules

$$Y = (A + B + C)(A + \overline{B} + \overline{C})(\overline{A} + B + C)$$
$$Y = (A + B + C)(\overline{A} + B + C)(A + \overline{B} + \overline{C})$$
$$Y = (A\overline{A} + B + C)(A + \overline{B} + \overline{C})$$
$$Y = (B + C)(A + \overline{B} + \overline{C}) --- \text{Minimal POS form}$$

1.5. K-Map simplification (Karnaugh Map simplification)

The simplification of Boolean expressions using Boolean algebraic rules is not unique and most of the cases, the resultant expression is not in minimal form. In order to get the uniqueness and final minimal form, K-map technique will be used. In the following section, the introduction to K-maps, grouping of variables and simplification procedures are discussed with examples.

NOTE:
Number of cells in K-map = number of possible cases
No. of possible cases=2^N
N is the number of input variables.

Example: Number of input variables=2, then the number of cells in K-map is 4.

NOTE: K-maps can take wither POS form or SOP form, in SOP form 1's are need to be grouped and in POS form 0's are need to be grouped.

NOTE: Only adjacent cells will be considered for grouping, diagonal cells should not be grouped.

NOTE: grouping can be done using 2 variables, 4 variables, 8 variables, 16 variables etc.., highest priority for grouping maximum variables in the above denomination. Variables are 0's for POS form and 1's for SOP form.

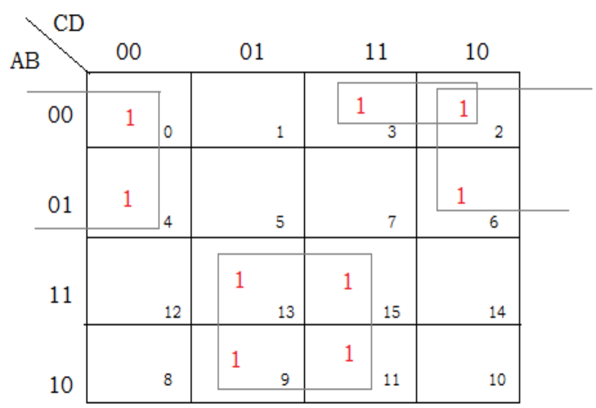
Procedure:

1. Select the number of cells according to the number of input variables.
2. Identify whether the given problem is SOP or POS form, minterms for SOP form and maxterms for POS form.
   NOTE: In SOP form, fill the cells by 1's at corresponding minterms and otherwise fill with 0's.
   NOTE: In POS form, fill the cells by 0's at corresponding maxterms and otherwise fill with 1's.
   NOTE: In POS form, take the complement of the output variable to get the resultant expression.
3. group the terms in the form of rectangular, the total number of terms is 2, 4, 8, etc.., try to cover as many elements as you can in one group.
4. from the groups, find the Product terms for SOP from and sum terms for POS form.

Example:

Simplify the following canonical SOP form of Boolean expression using K-map technique.

$$Y(A,B,C,D) = \sum m(0,2,3,4,6,9,11,13,15)$$



Simplified Boolean Expression $Y = \overline{A}\overline{D} + \overline{A}\overline{B}C + AD$

******