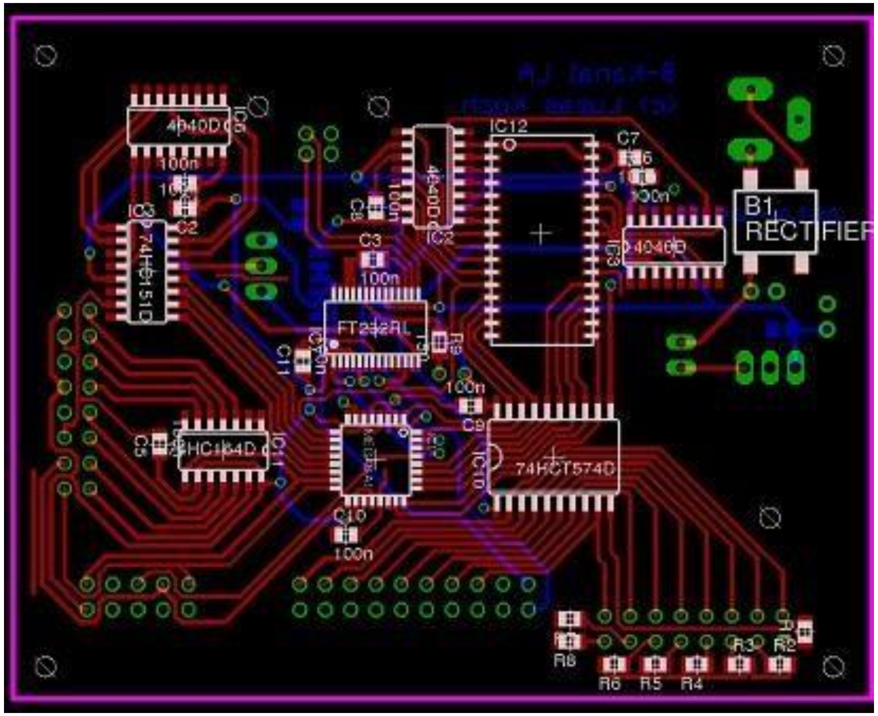


Digital Electronics

Course Title: Digital Electronics
Course No.: EEE2407

Boolean Expressions

- Boolean expressions is an expression written using logic circuit.
  - Boolean expressions are composed of
    - Literals – variables and their complements
    - Logical operations
  - Examples
    - $F = A.B'.C + A'.B.C' + A.B.C + A'.B'.C'$

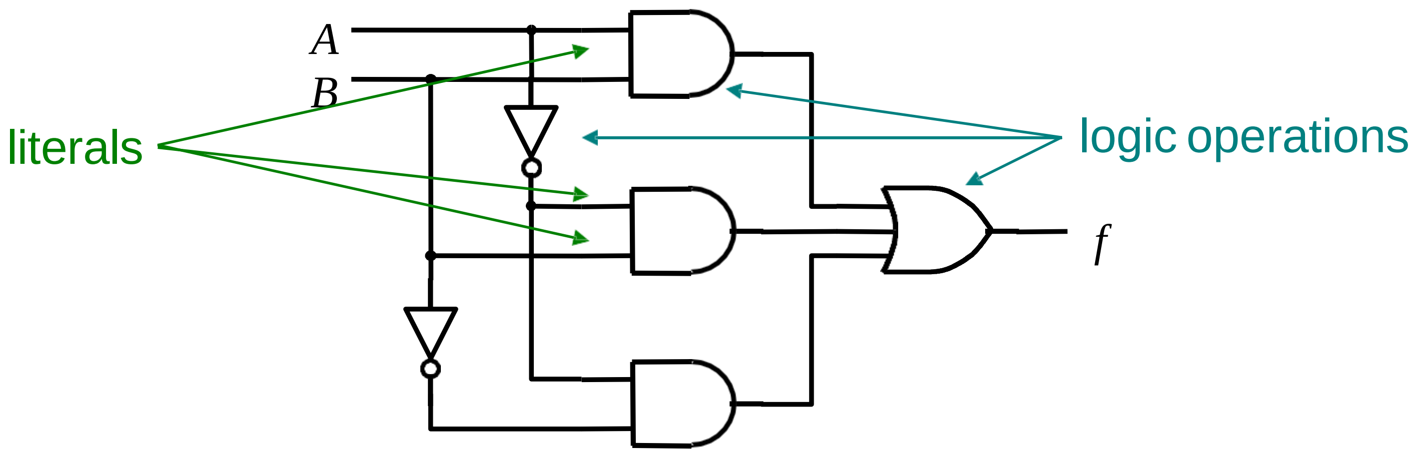
literals

logic operations

    - $F = (A+B+C').(A'+B'+C).(A+B+C)$
    - $F = A.B'.C' + A.(B.C' + B'.C)$

Boolean Expressions...

- Boolean expressions are realized using a network (or combination) of logic gates.
  - Each logic gate implements one of the logic operations in the Boolean expression
  - Each input to a logic gate represents one of the literals in the Boolean expression



Boolean Expressions…

- Boolean expressions are evaluated by
  - Substituting a 0 or 1 for each literal.
  - Calculating the logical value of the expression.
- A <u>Truth Table</u> specifies the value of the Boolean expression for every combination of the variables in the Boolean expression.
- For an n-variable Boolean expression, the truth table has 2^n rows (one for each combination).

Boolean Algebra

Basic Laws and Theorems

Commutative Law	$A+B=B+A$	$A.B=B.A$
Associative Law	$A+(B+C)=(A+B)+C$	$A.(B.C)=(A.B).C$
Distributive Law	$A.(B+C)=AB+AC$	$A+(B.C)=(A+B).(A+C).$
Null Elements	$A+1=1$	$0=0$
Identity	$A + 0 = A$	$A.1=A$
Idempotence	$A+A=A$	$A.A=A$
Complement	$A+A'=1$	$A.A'=0$
Involution	$A''=A$	
Absorption (Covering)	$A+AB=A$	$A.(A+B)=A$
Simplification	$A+A'B=A+B$	$A.(A'+B)=A.B$
DeMorgan's Rule	$(A+B)'=A'.B'$	$(A.B)'=A'+B'$
Logic Adjacency (Combining)	$AB+AB'=A$	$(A+B).(A+B')=A$
Consensus	$AB+BC+A'C=AB+A'C$	$(A+B).(B+C).(A'+C)=(A+B).(A'+C)$

DeMorgan's Laws

- Can be stated as follows:
    1. The complement of the product (AND) is the sum (OR) of the complements.

       $$(X.Y)' = X' + Y'$$

    2. The complement of the sum (OR) is the product (AND) of the complements.

       $$(X + Y)' = X' . Y'$$

- Easily generalized to n variables.
- Can be proven using a Truth table.

Proving DeMorgan's Law

(X . Y)' = X' + Y'

x	y	$x \cdot y$	$\overline{x \cdot y}$	$\overline{x}$	$\overline{y}$	$\overline{x} + \overline{y}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

LHS RHS

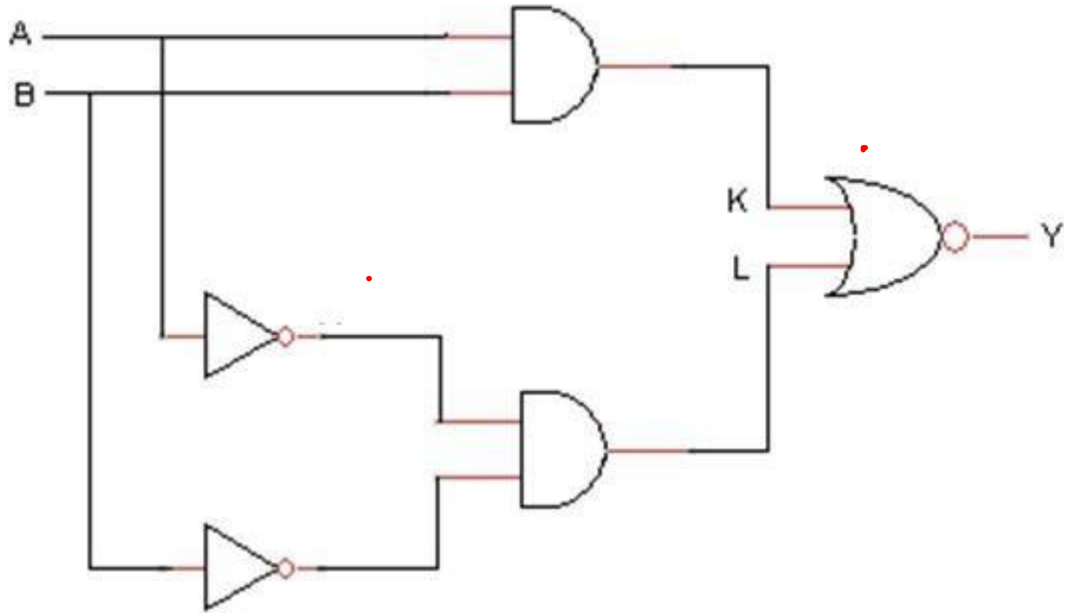
Self: (X + Y)' = X' .Y'

Importance of Boolean Algebra

- Boolean Algebra is used to simplify Boolean expressions.
  - Through application of the Laws and Theorems discussed
- Simpler expressions lead to simpler circuit realization, which, generally, reduces cost, area requirements, and power consumption.
- The objective of the digital circuit designer is to design and realize optimal digital circuits.

Examples

1. Consider the function ***F=xy'+x'z.*** Form the truth table and draw the circuit diagram.
2. For the circuit shown below write Boolean expression and truth table.



Canonical Forms for Boolean Expressions

Boolean functions are expressed as a Sum-of-Minterms or Product-of-Maxterms are said to be in *Canonical form*.

Algebraic Simplification

- Justification for simplifying Boolean expressions:
  - Reduces the cost associated with realizing the expression using logic gates.
  - Reduces the area (i.e. silicon) required to fabricate the switching function.
  - Reduces the power consumption of the circuit.
- In general, there is no easy way to determine when a Boolean expression has been simplified to a minimum number of terms or minimum number of literals.
  - No unique solution

Minterms

- A minterm, for a function of n variables, is a product term in which each of the n variables appears once.
- Each variable in the minterm may appear in its complemented or uncomplemented form.
- For a given row in the Truth table, the corresponding minterm is formed by
  - Including variable x_i, if $x_i = 1$
  - Including the complement of x_i, if $x_i = 0$

For all n variables in the function F

Minterms…

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2 x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1 x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1 x_2 x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2 x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1 x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1 x_2 x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

Maxterms

- A <u>Maxterm</u>, for a function of n variables, is a sum term in which each of the n variables appears once.
- Each variable in the Maxterm may appear in its complemented or uncomplemented form.
- For a given row in the Truth table, the corresponding Maxterm is formed by
  - Including the variable x_i, if $x_i = 0$
  - Including the complement of x_i, if $x_i = 1$

Maxterms...

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1x_2x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

Examples

Example 2-4

Express the Boolean function $F = A + B'C$ in a sum of minterms. The function has three variables, A, B, and C. The first term A is missing two variables; therefore:

$$A = A(B + B') = AB + AB'$$

This is still missing one variable:

$$\begin{aligned} A &= AB(C + C') + AB'(C + C') \\ &= ABC + ABC' + AB'C + AB'C' \end{aligned}$$

The second term $B'C$ is missing one variable:

$$B'C = B'C(A + A') = AB'C + A'B'C$$

Combining all terms, we have

$$\begin{aligned} F &= A + B'C \\ &= ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C \end{aligned}$$

But $AB'C$ appears twice, and according to theorem 1 ($x + x = x$), it is possible to remove one of them. Rearranging the minterms in ascending order, we finally obtain

$$\begin{aligned} F &= A'B'C + AB'C' + AB'C + ABC' + ABC \\ &= m_1 + m_4 + m_5 + m_6 + m_7 \end{aligned}$$

■

It is sometimes convenient to express the Boolean function, when in its sum of minterms, in the following short notation:

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

Examples…

Example 2-5

Express the Boolean function $F = xy + x'z$ in a product of maxterm form. First, convert the function into OR terms using the distributive law:

$$\begin{aligned} F = xy + x'z &= (xy + x')(xy + z) \\ &= (x + x')(y + x')(x + z)(y + z) \\ &= (x' + y)(x + z)(y + z) \end{aligned}$$

The function has three variables: x, y, and z. Each OR term is missing one variable; therefore:

$$x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$$
$$x + z = x + z + yy' = (x + y + z)(x + y' + z)$$
$$y + z = y + z + xx' = (x + y + z)(x' + y + z)$$

Combining all the terms and removing those that appear more than once, we finally obtain:

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') \\ &= M_0M_2M_4M_5 \end{aligned}$$

■

A convenient way to express this function is as follows:

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

The product symbol, Π, denotes the ANDing of maxterms; the numbers are the maxterms of the function.

Standard form for Boolean Expression

- Sum of product
- Product of sum

The *sum of products* is a Boolean expression containing AND terms, called *product terms*, of one or more literals each. The *sum* denotes the ORing of these terms. An example of a function expressed in sum of products is

$$F_1 = y' + xy + x'yz'$$

The expression has three product terms of one, two, and three literals each, respectively. Their sum is in effect an OR operation.

A *product of sums* is a Boolean expression containing OR terms, called *sum terms*. Each term may have any number of literals. The *product* denotes the ANDing of these terms. An example of a function expressed in product of sums is

$$F_2 = x(y' + z)(x' + y + z' + w)$$

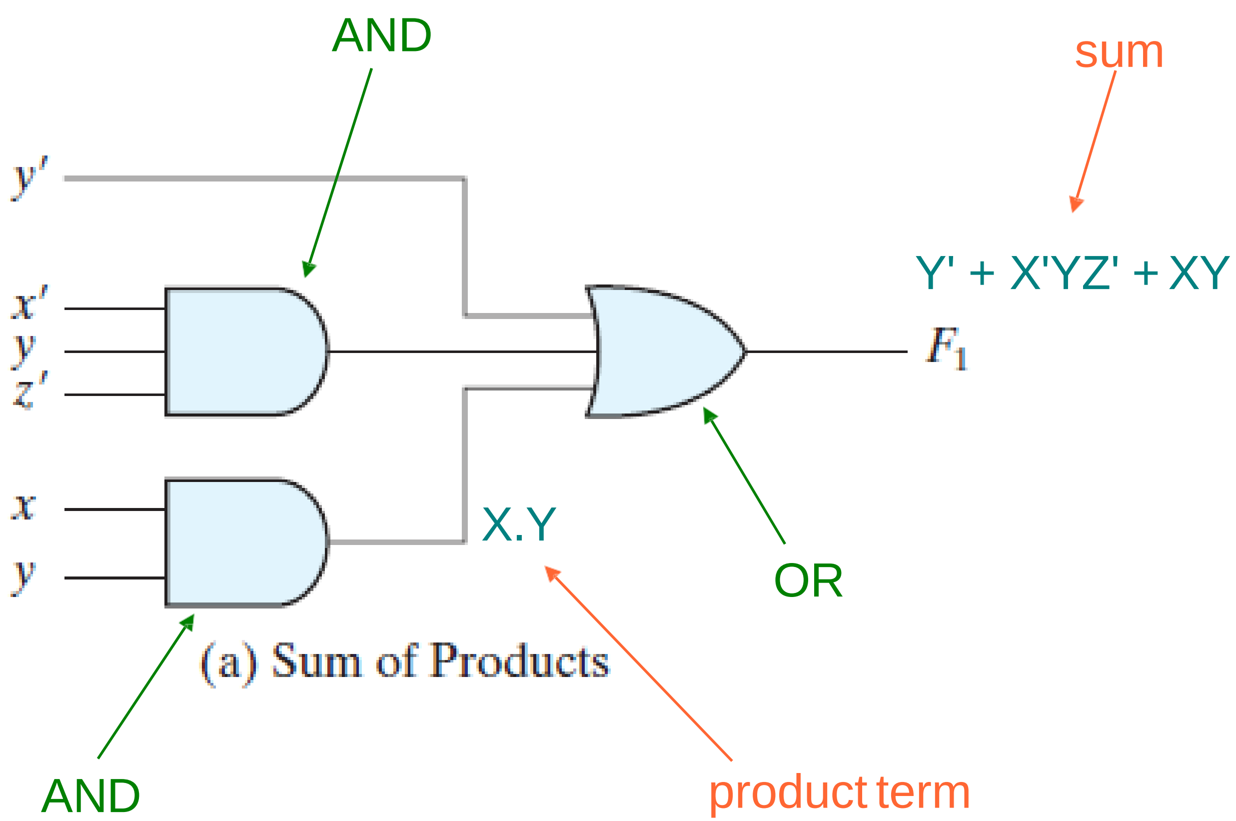
Sum-of-Products

- Any function F can be represented by a sum of minterms, where each minterm is ANDed with the corresponding value of the output for F.
  - $F = \sum (m_i . f_i)$
    - where m_i is a minterm
    - and f_i is the corresponding functional output

Denotes the logical sum operation

  - Only the minterms for which $f_i = 1$ appear in the expression for function F.
  - $F = \sum (m_i) = \sum m(i)$ shorthand notation

Sum-of-Products...



(a) Sum of Products

Product Term = Logical *AND*ing of literals
Sum = Logical *OR*ing of product terms

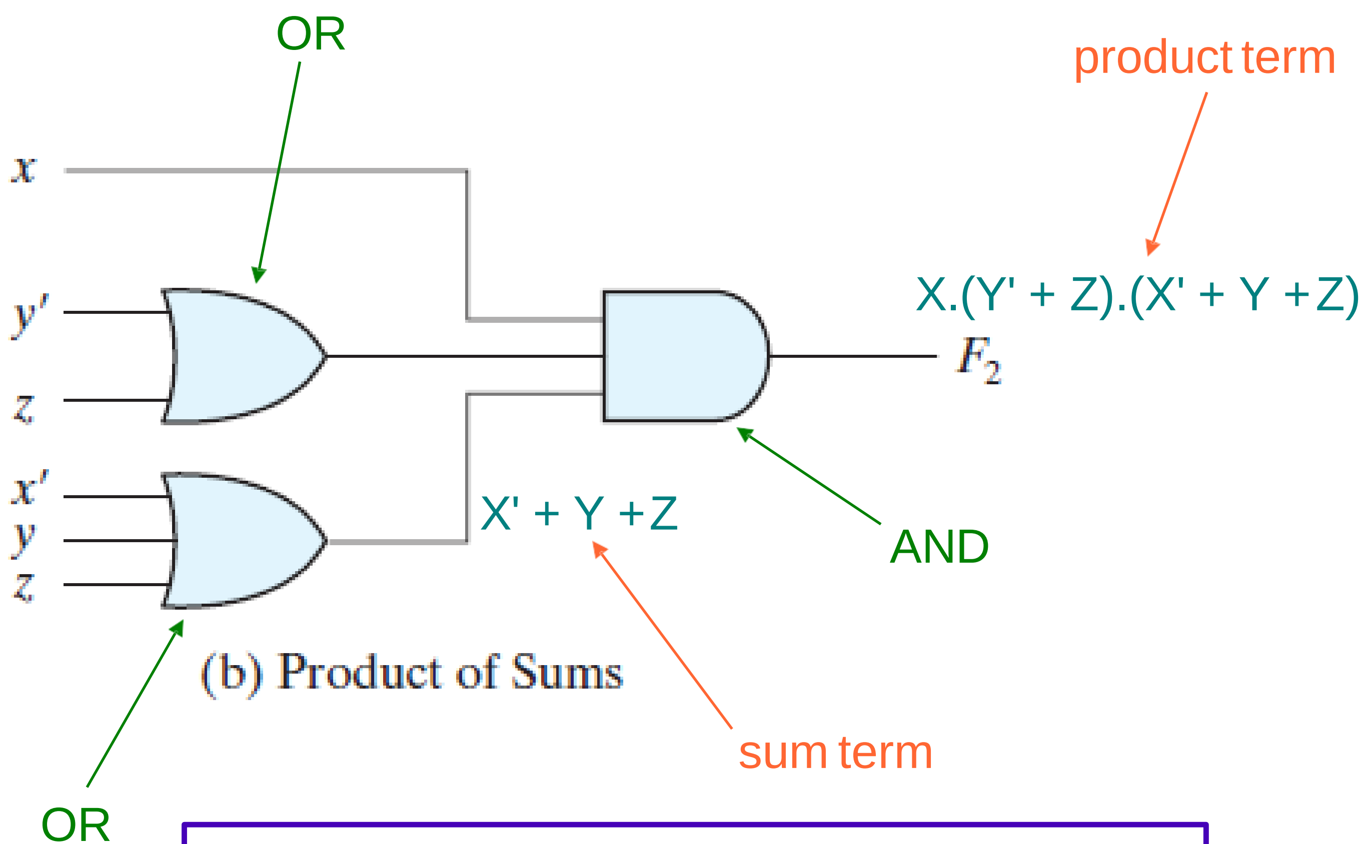
Product-of-Sums

- Any function F can be represented by a product of Maxterms, where each Maxterm is ANDed with the *complement* of the corresponding value of the output for F.
  - $F = \Pi\ (M_i \cdot f_i')$
    - where M_i is a Maxterm
    - and f_i' is the complement of the corresponding functional output

Denotes the logical product operation

  - Only the Maxterms for which $f_i = 0$ appear in the expression for function F.
  - $F = \Pi\ (M_i) = \Pi\ M(i)$ ← shorthand notation

Product-of-Sums...



Sum Term = Logical ORing of variables

Product = Logical ANDing of sum terms

Thank you