Boolean Functions

Lecture-7

By
Bhagyalaxmi Behera
Asst. Professor (Dept. of ECE)

Contents

Boolean function

- Boolean algebra is an algebra that deals with binary variables and logic operations.
- A Boolean function described by an algebraic expression consists of binary variables, the constants 0 and 1, and the logic operation symbols.
- For a given value of the binary variables, the function can be equal to either 1 or 0.
- A Boolean function expresses the logical relationship between binary variables and is evaluated by determining the binary value of the expression for all possible values of the variables.

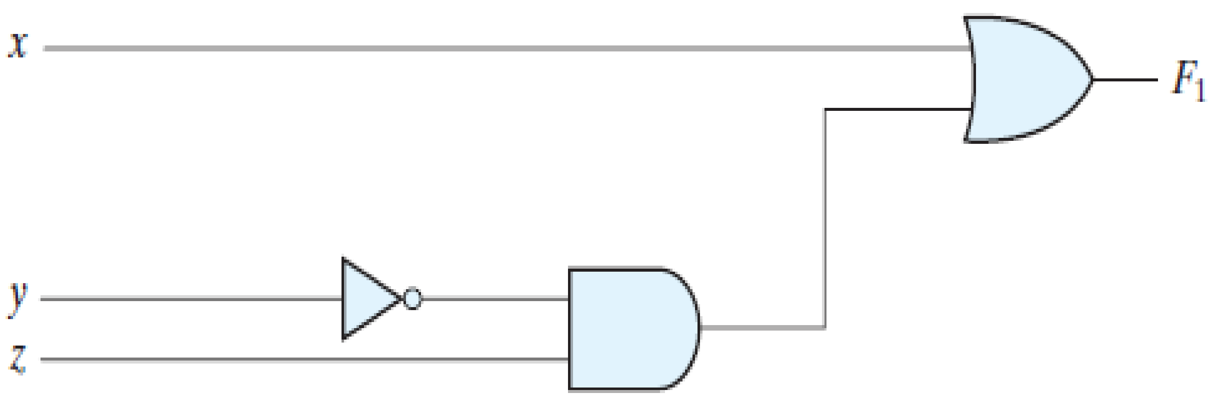
Boolean Functions

- Consider the following Boolean function:

  $F_1 = x + y'z$

- A Boolean function can be represented in a truth table. The binary combinations for the truth table are obtained by counting from 0 through 2^n-1 see 0 to 7.

x	y	z	F_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



Simplification of Boolean Functions

- There is only one way that a Boolean function can be represented in a truth table.

- In algebraic form, it can be expressed in a variety of ways and all of them have equivalent logic.

- A Boolean function can be transformed from an algebraic expression into a circuit diagram composed of logic gates connected in a particular structure.

- The particular expression used to represent the function will dictate the interconnection of gates in the logic-circuit diagram. Conversely, the interconnection of gates will dictate the logic expression.

- Here is a key fact that motivates our use of Boolean algebra: By manipulating a Boolean expression according to the rules of Boolean algebra, it is sometimes possible to obtain a simpler expression for the same function and thus reduce the number of gates in the circuit and the number of inputs to the gate.
- Designers are motivated to reduce the complexity and number of gates because their effort can significantly reduce the cost of a circuit.

Before simplification of Boolean function

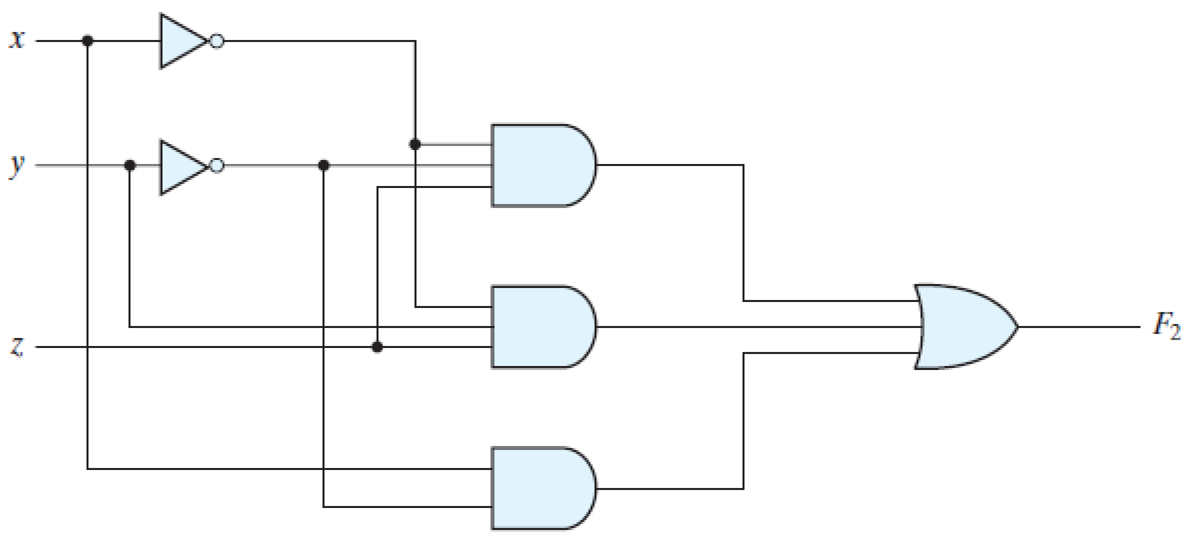
Consider the following Boolean function:

$$F_2 = x'y'z + x'yz + xy'$$

This function with logic gates is shown in Fig.

The function is equal to 1 when xyz = 001 or 011 or when xyz = 100,101 .

x	y	z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



After simplification of Boolean function

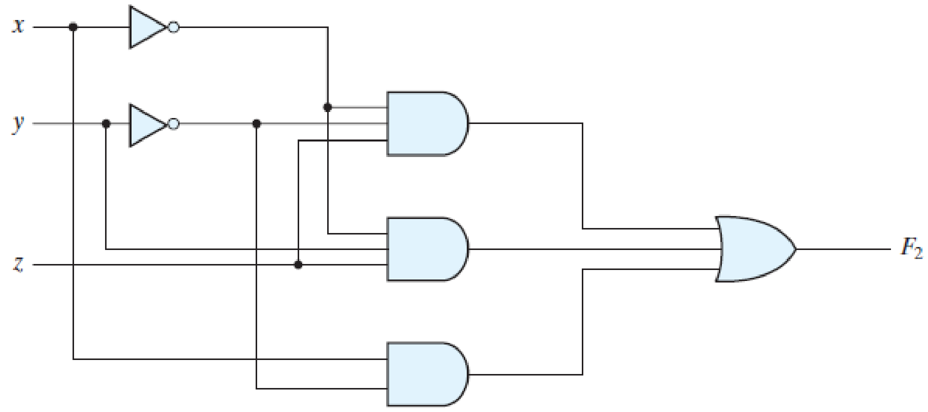
- Simplify the following Boolean function:

$$F_2 = x'y'z + x'yz + xy'$$

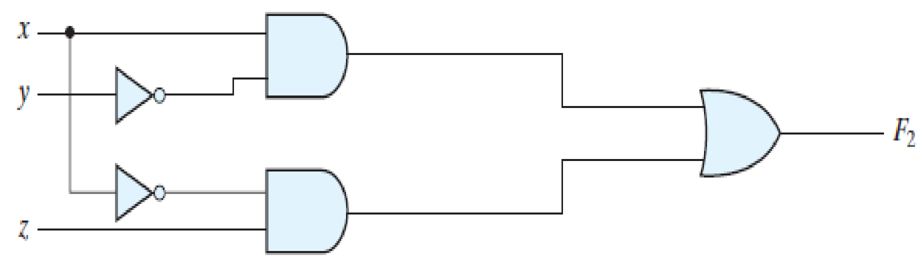
$$= x'z\,(y' + y) + xy'$$

$$= x'z + xy'$$

- The Reduced function would be preferable because it requires less wires and components.



simplified



Equivalent Expressions

$F_2 = x'y'z + x'yz + xy'$(primitive)

F_2=1 when xyz=001 or 011 or when xy=100,101

$F_2 = x'z + xy'$ (simplified)

F_2 =1 when xz=01 or when

xy=10

Since both expression produce the same truth table, they are said to be equivalent.

x	y	z	F_2
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

- Therefore, the two circuits have the same outputs for all possible binary combinations of inputs of the three variables.
- Each circuit implements the same identical function, but the one with fewer gates and fewer inputs to gates is preferable because it requires fewer wires and components.
- In general, there are many equivalent representations of a logic function. Finding the most economic representation of the logic is an important design task.

Algebraic Manipulation

- When a Boolean expression is implemented with logic gates, each term requires a gate and each variable within the term designates an input to the gate.
- We define a *literal to* be a single variable within a term, in complemented or un complemented form.

$F_2 = x'y'z + x'yz + xy'$ has three terms and eight literals

$$F2 = x'y'z + x'yz + xy'$$
$$= x'z(y' + y) + xy'$$
$$= x'z + xy'$$

$F_2 = x'z + xy'$ has two terms and four literals

- By reducing the number of terms, the number of literals, or both in a Boolean expression, it is often possible to obtain a simpler circuit.
- The manipulation of Boolean algebra consists mostly of reducing an expression for the purpose of obtaining a simpler circuit.

Simplification of Boolean functions

Simplify the following Boolean functions to a minimum number of literals.

1. $x(x' + y) = xx' + xy = 0 + xy = xy.$
2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y.$
3. $(x + y)(x + y') = x + xy + xy' + yy' = x(1 + y + y') = x.$
4. $$\begin{aligned} xy + x'z + yz &= xy + x'z + yz(x + x') \\ &= xy + x'z + xyz + x'yz \\ &= xy(1 + z) + x'z(1 + y) \\ &= xy + x'z. \end{aligned}$$
5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$, by duality from function 4.

Simplification of Boolean functions

- The fourth function illustrates the fact that an increase in the number of literals sometimes leads to a simpler final expression.
- Function 5 is not minimized directly, but can be derived from the dual of the steps used to derive function 4.
- Functions 4 and 5 are together known as the ***consensus theorem***.

Assignment

1 Demonstrate the validity of the following identities by means of truth tables:

(a) DeMorgan's theorem for three variables: $(x + y + z)' = x'y'z'$ and $(xyz)' = x' + y' + z'$

(b) The distributive law: $x + yz = (x + y)(x + z)$

(c) The distributive law: $x(y + z) = xy + xz$

(d) The associative law: $x + (y + z) = (x + y) + z$

(e) The associative law and $x(yz) = (xy)z$

2 Simplify the following Boolean expressions to a minimum number of literals:

(a)* $xy + xy'$

(b)* $(x + y)(x + y')$

(c)* $xyz + x'y + xyz'$

(d)* $(A + B)'(A' + B')'$

(e) $(a + b + c')(a'b' + c)$

(f) $a'bc + abc' + abc + a'bc'$

Assignment

3 Simplify the following Boolean expressions to a minimum number of literals:

(a)* $ABC + A'B + ABC'$ (b)* $x'yz + xz$

(c)* $(x + y)'(x' + y')$ (d)* $xy + x(wz + wz')$

(e)* $(BC' + A'D)(AB' + CD')$ (f) $(a' + c')(a + b' + c')$

4 Reduce the following Boolean expressions to the indicated number of literals:

(a)* $A'C' + ABC + AC'$ to three literals

(b)* $(x'y' + z)' + z + xy + wz$ to three literals

(c)* $A'B(D' + C'D) + B(A + A'CD)$ to one literal

(d)* $(A' + C)(A' + C')(A + B + C'D)$ to four literals

(e) $ABC'D + A'BD + ABCD$ to two literals