Section 4

Summer 2020

Chapter 2

TnF

CSE 231: Digital Logic Design

Boolean Algebra

- As binary logic is used in all of today's digital computers and devices, finding simpler and cheaper, but equivalent, realizations of a circuit can reap huge payoffs in reducing the overall cost of the design.
- Mathematical methods that simplify circuits rely. primarily on Boolean algebra.
- Boolean algebra, like any other deductive mathematical system, may be defined with a set of elements, a set of operators, and a number of unproved axioms or postulates.
- The postulates of a mathematical system form the basic assumptions from which it is possible to deduce the rules, theorems, and properties of the system. The most common postulates used to formulate various algebraic structures are as follows:

Postulates

- Closure: the set of natural numbers N = {1, 2, 3, 4, c} is closed with respect to the binary operator + by the rules of arithmetic addition, since, for any a, b ∈ N, there is a unique c ∈ N such that a + b = c.
- Associative law: A binary operator * on a set S is said to be associative whenever

$$(x * y) * z = x * (y * z) \text{ for all } x, y, z, \in S$$

- Commutative law: A binary operator * on a set S is said to be commutative whenever

$$x * y = y * x \text{ for all } x, y \in S$$

- Identity element: set S is said to have an identity element with respect to a binary operation * on S if there exists an element e ∈S with the property that

$$e * x = x * e = x \text{ for every } x \in S$$

- Inverse: A*A′=0, A+A′=1
- Distributive law: If * and . are two binary operators on a set S, * is said to be distributive over . whenever

$$x * (y + z) = (x * y) + (x * z)$$

List of Basic Boolean theorems & postulates

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$	**Identity Law**
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$	**Inverse Law**
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$	**Idempotent Law**
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$	**Null Law**
Theorem 3, involution		$(x')' = x$			
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$	
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$	
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$	
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$	
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$	

Proof of theorem: Absorption Law

$x + xy = x.$

Statement	Justification
$x + xy = x \cdot 1 + xy$	postulate 2(b)
$= x(1 + y)$	4(a)
$= x(y + 1)$	3(a)
$= x \cdot 1$	2(a)
$= x$	2(b)

Proof of theorem: DeMorgans Law

- Collect from class notes

Simplification of Boolean Functions using Boolean Theorems 1

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.

Boolean Function

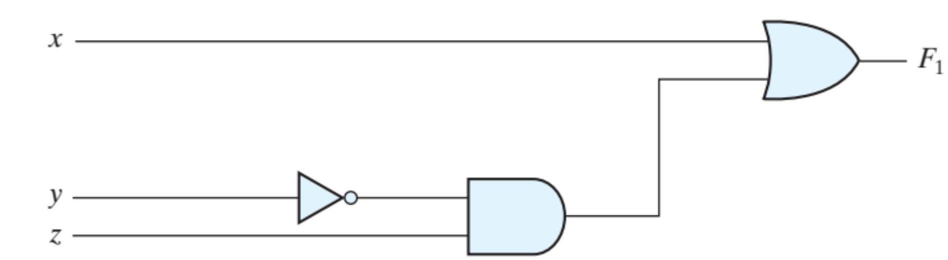
- Boolean algebra is an algebra that deals with binary variables and logic operations. A Boolean function described by an algebraic expression consists of binary variables, the constants 0 and 1, and the logic operation symbols.
- For a given value of the binary variables, the function can be equal to either 1 or 0. As an example, consider the Boolean function

  $$F_1 = x + y'z$$

  The function F_1 is equal to 1 if x is equal to 1 or if both y and z are equal to 1. F_1 is equal to 0 otherwise.
- A Boolean function can be represented in a truth table. The number of rows in the truth table is 2^n, where n is the number of variables in the function.
- The Table shows the truth table for the function F_1.
- A Boolean function can be transformed from an algebraic expression into a circuit diagram composed of logic gates connected in a particular structure.

Truth Tables for F_1

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



Gate implementation of $F_1 = x + y'z$

Simplification of Boolean Functions using Boolean Theorems 2

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

$AB + AB + AC + BB + BC$

$AB + AB + AC + B + BC$

$AB + AC + B + BC$

$AB + AC + B$

$B+AC$

Simplification of Boolean Functions using Boolean Theorems 3

$$C + \overline{BC}$$
$$C + (\overline{B} + \overline{C})$$
$$(C + \overline{C}) + \overline{B}$$
$$1 + \overline{B}$$
$$1$$

$$\overline{AB}\,(\overline{A} + B)\,(\overline{B} + B)$$
$$\overline{AB}\,(\overline{A} + B)$$
$$(\overline{A} + \overline{B})\,(\overline{A} + B)$$
$$\overline{A}\,\overline{A} + \overline{A}B + \overline{A}\,\overline{B} + \overline{B}B$$
$$\overline{A} + \overline{A}(B + \overline{B})$$
$$\overline{A} + \overline{A}$$
$$\overline{A}$$

Simplification of Boolean Functions using Boolean Theorems 4

$$\bar{A}(A+B)+(B+AA)(A+\bar{B})$$

$$\bar{A}A + \bar{A}B + (B+A)(A+\bar{B})$$

$$0 + \bar{A}B + AB + B\bar{B} + AA + A\bar{B}$$

$$\bar{A}B + AB + 0 + A + A\bar{B}$$

$$\bar{A}B + A(B+1+\bar{B})$$

$$\bar{A}B + A(1+1)$$

$$\bar{A}B + A$$

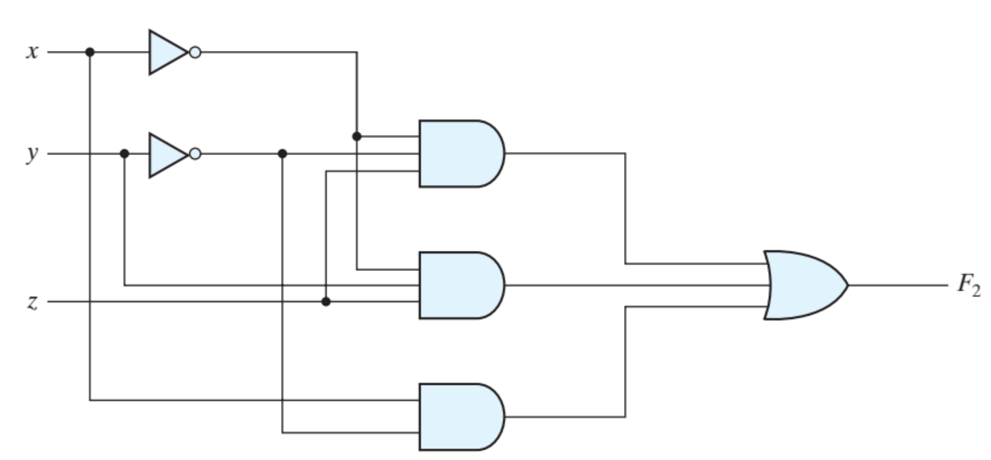
$$(\bar{A}+A)(A+B)$$

$$A+B$$

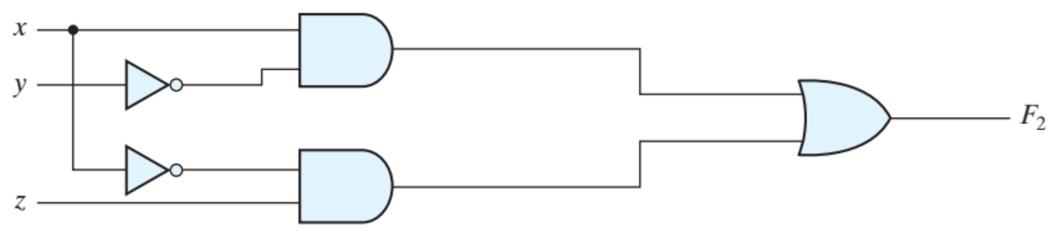
Circuit simplification

- The function is in algebraic form, it can be expressed in a variety of ways, all of which have equivalent logic.
- 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.

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



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



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

Complement of a Function

- The complement of a function F is Fand is obtained from an interchange of 0's for 1's and 1's for 0's in the value of F. The complement of a function may be derived algebraically through DeMorgan's theorems

$$
\begin{aligned}
(A + B + C)' &= (A + x)' && \text{let } B + C = x \\
&= A'x' && \text{by theorem 5(a) (DeMorgan)} \\
&= A'(B + C)' && \text{substitute } B + C = x \\
&= A'(B'C') && \text{by theorem 5(a) (DeMorgan)} \\
&= A'B'C' && \text{by theorem 4(b) (associative)}
\end{aligned}
$$

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

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

$$
\begin{aligned}
F_1' &= (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z') \\
F_2' &= [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')'(yz)' \\
&= x' + (y + z)(y' + z') \\
&= x' + yz' + y'z
\end{aligned}
$$

CANONICAL AND STANDARD FORMS

- Standard Forms: the terms that form the function may contain one, two, or any number of literals.

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

- Minterms and Maxterms:
- A binary variable may appear either in its normal form (x) or in its complement form (x). Now consider two binary variables x and y combined with an AND operation. Since each variable may appear in either form, there are four possible combinations: xy, xy, xy, and xy. Each of these four AND terms is called a minterm, or a standard product.
- In a similar fashion, n variables forming an OR term, with each variable being primed or unprimed, provide 2n possible combinations, called maxterms, or standard sums.
- A Boolean function can be expressed algebraically from a given truth table by forming a minterm for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

- Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form.

Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

SOP & POS

- Boolean functions expressed as a sum of minterms or product of maxterms are said to be in canonical form. The minterms whose sum defines the Boolean function are those which give the 1's of the function in a truth table.
- Consider the following table:

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- Sum of Minterm /Sum of Product (SOP):

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

- Product of Maxterm/Product of Sum (POS):

$$f_1 = (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z)$$
$$= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

$$f_2 = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z)$$
$$= M_0 M_1 M_2 M_4$$

Conversion between Canonical Forms

- The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function.

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

This function has a complement that can be expressed as

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

Now, if we take the complement of F' by DeMorgan's theorem, we obtain F in a different form:

$$F = (m_0 + m_2 + m_3)' = m_0' \cdot m_2' \cdot m_3' = M_0 M_2 M_3 = \Pi(0, 2, 3)$$

The last conversion follows from the definition of minterms and maxterms as shown in Table 2.3. From the table, it is clear that the following relation holds:

$$m_j' = M_j$$

That is, the **maxterm with subscript j is a complement of the minterm with the same subscript j and vice versa.**