CMPT-335 Discrete Structures

BOOLEAN ALGEBRA

Boolean Algebra

- Boolean algebra provides the operations and the rules for working with the set {0, 1}
- These are the rules that underlie electronic circuits, and the methods we will discuss are fundamental to VLSI design
- We are going to focus on three operations:
  - Boolean complementation (negation),
  - Boolean sum (OR, disjunction)
  - Boolean product (AND, conjunction)

Boolean Operations

- The complement (negation) is denoted by a bar It is defined by

$$\overline{1} = 0; \quad \overline{0} = 1$$

- The Boolean sum (disjunction), denoted by + or by OR, or by $\vee$ has the following values:

$$1 + 1 = 1, \quad 1 + 0 = 1, \quad 0 + 1 = 1, \quad 0 + 0 = 0$$

- The Boolean product (conjunction), denoted by $\cdot$ or by AND, or by & or by $\wedge$ has the following values:

$$1 \cdot 1 = 1, \quad 1 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 0 \cdot 0 = 0$$

Boolean Functions and Expressions

- Let $B = \{0, 1\}$. The variable x is called a Boolean variable if it assumes values only from B
- A function $f : B^n \rightarrow B$ from $B^n = \{(x_1, \ldots, x_n) \mid x_i \in B, i = 1, \ldots, n\}$ to B is called a Boolean function of degree n (A Boolean function of n variables)
- Boolean functions can be represented using expressions made up from Boolean variables and Boolean operations

Boolean Functions and Expressions

- The Boolean expressions in the variables $x_1, x_2, \ldots, x_n$ are defined recursively as follows:

➢ $0, 1, x_1, x_2, \ldots, x_n$ are Boolean expressions.

➢ If E_1 and E_2 are Boolean expressions, then $\overline{E_1}$, $(E_1 E_2)$, and $(E_1 + E_2)$ are Boolean expressions

- Each Boolean expression represents a Boolean function. The values of this function are obtained by substituting 0 and 1 for the variables in the expression

Boolean Functions and Expressions

- For example, we can create Boolean expression in the variables x, y, and z using the "building blocks" 0, 1, x, y, and z, and the construction rules:

➢ Since x and y are Boolean expressions, so is xy

➢ Since z is a Boolean expression, so is $\overline{z}$

➢ Since xy and $\overline{z}$ are Boolean expressions, so is $xy + \overline{z}$

- … and so on…

Boolean Functions and Expressions

- Example 1. Give a Boolean expression for the Boolean function ***F*(*x*, *y*)** as defined by the following table:

x	y	$F(x, y)$
0	0	0
0	1	1
1	0	0
1	1	0

Solution:

$$F(x, y) = \overline{x}y = \overline{x} \,\&\, y$$

Boolean Functions and Expressions

- Example 2. Give a Boolean expression for the Boolean function ***F*(*x*, *y*, *z*)** as defined by the following table:

x	*y*	*z*	*F*(*x*, *y*, *z*)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Solution :

$$F(x,y,z)=\overline{(xz+y)}=\overline{(x\,\&\,z)\vee y}$$

$$\ldots=\overline{(xz+y)}=\overline{(xz)}\,\overline{y}=\overline{(x\,\&\,z)}\,\&\,\overline{y}=$$

$$\ldots=(\overline{x}+\overline{z})\,\overline{y}=(\overline{x}\vee\overline{z})\,\&\,\overline{y}$$

Boolean Functions and Expressions

- Example 3. Give a Boolean expression for the Boolean function $F(x, y)$ as defined by the following table:

x	y	$F(x, y)$
0	0	0
0	1	1
1	0	1
1	1	0

Solution:

$$F(x, y) = \overline{x}y + x\overline{y} = \left(\overline{x} \,\&\, y\right) \vee \left(x \,\&\, \overline{y}\right)$$

$$\ldots = \left(x + y\right) \bmod 2 = \mathrm{XOR}(x, y)$$

Minterms and Disjunctive Normal Form
(A sum of Products Form)

- There is a simple method for deriving a Boolean expression for a Boolean function that is defined by a table. This method is based on minterms

- A literal is a Boolean variable or its complement. A **minterm** (an elementary conjunction) of the Boolean variables $x_1, x_2, \ldots, x_n$ is a Boolean product (conjunction) $y_1 y_2 \ldots y_n$, where $y_i = x_i$ or $y_i = \overline{x_i}$

- Hence, a minterm is a product of n literals, with one literal for each variable. For example,

$$x_1 x_2 \overline{x}_3; \quad x_1 x_2 x_3; \quad \overline{x}_1 x_2 \overline{x}_3; \quad x_1 \overline{x}_2 x_3$$

Minterms and Disjunctive Normal Form (A sum of Products Form)

- A Disjunctive Normal Form (DNF) (also referred to as a Sum of Products Form) is a representation of a Boolean function (a logical formula) through a disjunction of conjunctions of its literals (variables and their negations)

Minterms and Disjunctive Normal Form (A Full Sum of Products Form)

- A Full Disjunctive Normal Form (FDNF) (a Full Sum of Products Form-FSPF) is such a DNF-FSPF where each of its variables appears exactly once in every minterm either without negation or with negation, thus it is a disjunction of minterms
- Any FDNF-FSPF of n variables contains the same amount of minterms as the number of **1s** among the values of the corresponding function

Algorithm for finding FDNF (FSPF)

1) Distinguish in the table, which determines a Boolean function $f\left(x_1,\dots,x_n\right)$ those k rows where $f\left(x_1,\dots,x_n\right)=1$

2) Create literals $y_1^{(j)},\dots,y_n^{(j)},\ j=1,\dots,k$ corresponding to these rows using the following rule:
if $x_i^{(j)}=1$ then $y_i^{(j)}=x_i$ otherwise $y_i^{(j)}=\overline{x}_i$ (if $x_i^{(j)}=0$)
and create minterms $\left(y_1^{(j)}\cdot y_2^{(j)}\cdot\dots\cdot y_n^{(j)}\right),\ j=1,\dots,k$

3) Create a disjunction of these minterms, which is FDNF

$$\left(y_1^{(1)}\dots y_n^{(1)}\right)\vee\left(y_1^{(2)}\dots y_n^{(2)}\right)\vee\dots\vee\left(y_1^{(k)}\dots y_n^{(k)}\right)=f\left(x_1,\dots,x_n\right)$$

Finding FDNF (FSPF)

- Example 1. Create FDNF-FSPF for the Boolean function $f(x_1, x_2)$, which is defined by the following table:

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Solution:

$f(x_1, x_2) = 1$ if and only if

$x_1 = 0, x_2 = 1; \quad x_1 = 1, x_2 = 0$

$y_1^{(1)} = \overline{x}_1, y_2^{(1)} = x_2$

$y_1^{(2)} = x_1, y_2^{(2)} = \overline{x}_2$

$$f(x_1, x_2) = y_1^{(1)} y_2^{(1)} + y_1^{(2)} y_2^{(2)} = \overline{x}_1 x_2 + x_1 \overline{x}_2 = (\overline{x}_1 \,\&\, x_2) \vee (x_1 \,\&\, \overline{x}_2)$$

$$\ldots = (x_1 + x_2) \bmod 2 = \mathrm{XOR}(x_1, x_2)$$

Finding FDNF (FSPF)

- Example 2. Create FDNF-FSPF for the Boolean function ***F*(*x*, *y*, *z*)** as defined by the following table:

Solution :

x	y	z	$F(x, y, z)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$f\left(x, y, z\right) = 1$ if and only if

$x = 0, y = 0, z = 0;$

$x = 0, y = 0, z = 1;$

$x = 1, y = 0, z = 0$

$$y_1^{(1)} = \overline{x}, y_2^{(1)} = \overline{y}, y_3^{(1)} = \overline{z}$$

$$y_1^{(1)} = \overline{x}, y_2^{(1)} = \overline{y}, y_3^{(1)} = z$$

$$y_1^{(1)} = x, y_2^{(1)} = \overline{y}, y_3^{(1)} = \overline{z}$$

$$F\left(x, y, z\right) = y_1^{(1)} y_2^{(1)} y_3^{(1)} + y_1^{(2)} y_2^{(2)} y_3^{(2)} + y_1^{(3)} y_2^{(3)} y_3^{(3)} =$$
$$= \overline{x} \cdot \overline{y} \cdot \overline{z} + \overline{x} \cdot \overline{y} \cdot z + x \cdot \overline{y} \cdot \overline{z} =$$
$$= \left(\overline{x} \,\&\, \overline{y} \,\&\, \overline{z}\right) \vee \left(\overline{x} \,\&\, \overline{y} \,\&\, z\right) \vee \left(x \,\&\, \overline{y} \,\&\, \overline{z}\right)$$

Boolean Functions and Expressions

- The Boolean functions F and G of n variables are equal if and only if $F(b_1, b_2, \ldots, b_n) = G(b_1, b_2, \ldots, b_n)$ whenever $b_1, b_2, \ldots, b_n$ belong to B
- Two different Boolean expressions that represent the same function are called equivalent
- For example, the Boolean expressions xy, $xy + 0$, and $xy \cdot 1$ are equivalent

Boolean Functions and Expressions

- The complement of the Boolean function f is the function $\overline{f}$, where $\overline{f}(x_1,\ldots,x_n) = \overline{f(x_1,\ldots,x_n)}$
- Let f and g be Boolean functions of degree n. The Boolean sum $f + g$ and Boolean product fg are then defined by

$$f + g = f(x_1,\ldots,x_n) + g(x_1,\ldots,x_n)$$
$$f \cdot g = f(x_1,\ldots,x_n) \cdot g(x_1,\ldots,x_n)$$

Boolean Functions and Expressions

- Question: How many different Boolean functions of degree 1 are there?
- Solution: There are four of them, F_0, F_1, F_2, and F_3:

x	F_0	F_1	F_2	F_3
0	0	0	1	1
1	0	1	0	1

Boolean Functions and Expressions

- **Question:** How many different Boolean functions of degree 2 are there?
- **Solution:** There are 16 of them, $F_1, F_2, F_3, \ldots, F_{16}$:

x_1	x_2	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Boolean Functions and Expressions

- Question: How many different Boolean functions of degree n are there?
- Solution:
- There are 2^n different n-tuples of 0s and 1s.
- A Boolean function is an assignment of 0 or 1 to each of these 2^n different n-tuples
- Therefore, there are $\mathbf{2^{2^n}}$ different Boolean functions

Identities

- There are useful identities of Boolean expressions that can help us to transform an expression A into an equivalent expression B, e.g.:

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0+x = x$
Null (or Dominance) Law	$0x = 0$	$1+x = 1$
Idempotent Law	$xx = x$	$x+x = x$
Inverse Law	$x\overline{x} = 0$	$x+\overline{x} = 1$
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x+(y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$
Absorption Law	$x(x+y) = x$	$x+xy = x$
DeMorgan's Law	$(\overline{xy}) = \overline{x}+\overline{y}$	$(\overline{x+y}) = \overline{x}\overline{y}$
Double Complement Law	$\overline{\overline{x}} = x$	

Definition of a Boolean Algebra

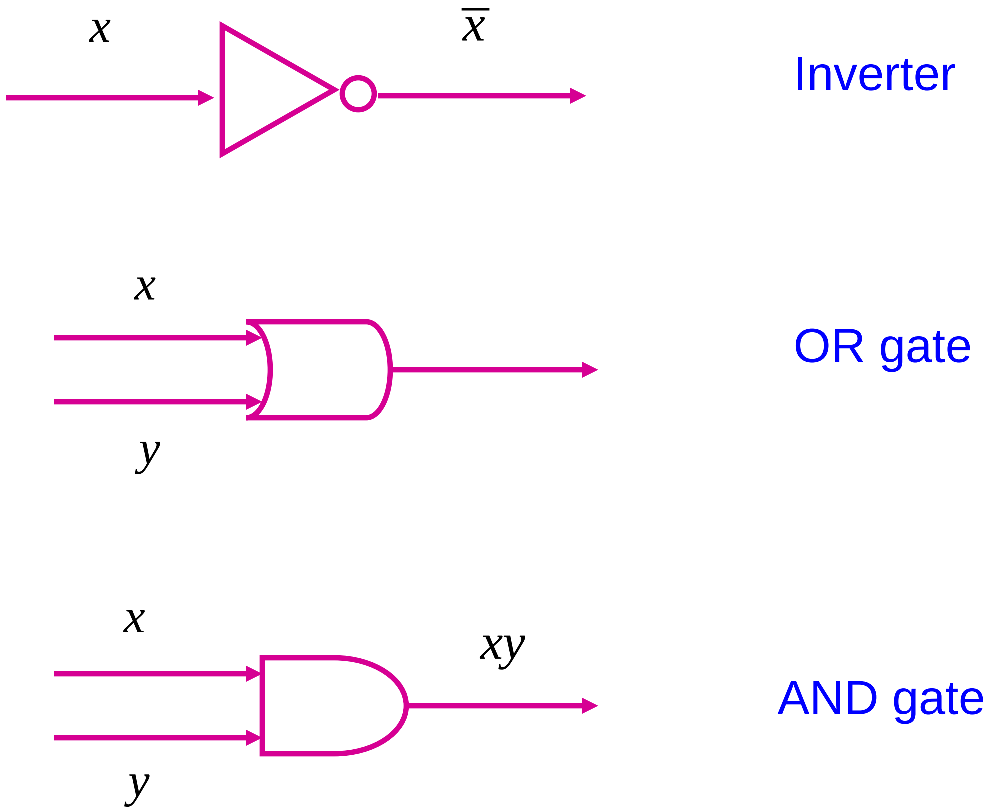
- All the properties of Boolean functions and expressions that we have discovered also apply to other mathematical structures such as propositions and sets and the operations defined on them
- If we can show that a particular structure is a Boolean algebra, then we know that all results established about Boolean algebras apply to this structure
- For this purpose, we need an abstract definition of a Boolean algebra

Definition of a Boolean Algebra

- Definition: A Boolean algebra is a set B with two binary operations $\vee$ and $\wedge$, elements 0 and 1, and a unary operation – such that the following properties hold for all x, y, and z in B :
- $x \vee 0 = x$ and $x \wedge 1 = x$ (identity laws)
- $x \vee (-x) = 1$ and $x \wedge (-x) = 0$ (domination laws)
- $(x \vee y) \vee z = x \vee (y \vee z)$ and
  $(x \wedge y) \wedge z = x \wedge (y \wedge z)$ and (associative laws)
- $x \vee y = y \vee x$ and $x \wedge y = y \wedge x$ (commutative laws)
- $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ and
  $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ (distributive laws)

Logic Gates

- Electronic circuits consist of so-called gates. There are three basic types of gates:



Logic Gates

- **Example:** How can we build a circuit that computes the function $xy + \overline{x}y$?