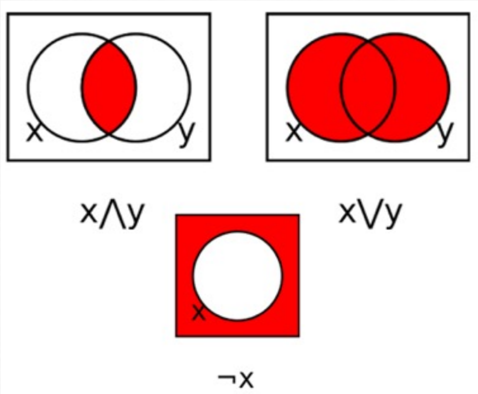
COMP1003 Computer Organization

Lecture 3 Boolean Algebra: From Bits to Logic



United International College

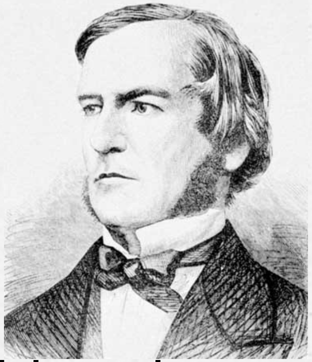
From Bits to Boolean Logic

- Computers represent information by bit (Binary digit)
- A bit has two possible values, namely 0 and 1.
- A bit can be used to represent a truth value, true and false.
- Therefore <u>bits operations</u> correspond to the <u>logical operations</u> in Boolean Algebra.

Algebra

- Algebra is the study of mathematical symbols and the rules for manipulating these symbols.
- The word comes from the Arabic language from the title of the book *Ilm al-jabr wa'l-muḳābala* by *al-Khwarizmi*.
- The original meaning of algebra (al-jabr) is “reunion of broken parts”

George Boole (1815-1864)



- Logic was not considered as part of mathematics
- In 1854 Boole published An investigation into the Laws of Thought, on Which are founded the Mathematical Theories of Logic and Probabilities.
- He pointed out the analogy between algebraic symbols and those that represent logical forms.
- It began the algebra of logic called Boolean algebra

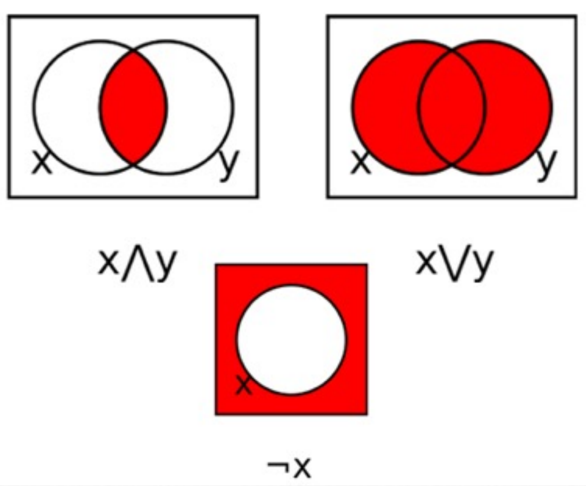
Boolean Algebra

- Algebra is a branch of mathematics that substitutes letters for numbers.
  - x+3=y

- Boolean algebra, sometimes referred to as the algebra of logic, is a two-valued system of algebra that represents logical relationships and operations.

Boolean Variables & Operators

- Boolean variables are variables that can take only two values: true/false; or 1/0
  - A,B,C = {0, 1}
- Boolean Operators
  - AND (A AND B, AB, A ∧ B)
  - OR (A OR B, A+B, A v B)
  - NOT   NOT A, A’, $\neg A$



Venn diagrams

Truth Table of Boolean Operators

AND		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
A	$\overline{A}$
0	1
1	0

Boolean Functions

- A function is a relation that uniquely associates members of one set with members of another set
- A Boolean function has
  - At least one Boolean variable
  - At least one Boolean operator
  - At least one input from the set {0,1}
- It produces an output that is also a member of the set {0,1}
- $F = X+YZ'$

Precedence of Boolean Operators

- There might be many Boolean operators in one Boolean function.
- Which operator to apply first?
- The rules of precedence
  - NOT top priority,
  - followed by AND
  - then OR

$F = X+YZ'$

Truth Table of a Boolean Function

F = X+YZ'

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Exercise

- Write the truth table for

$$F = A' + B'$$

- Write the truth table for

$$Q = A.B.\overline{C} + A.\overline{B}.C + \overline{A}.B.C$$

Simplify a Boolean Function

- Digital computers contain circuits that implement Boolean functions
- The simpler that we can make a Boolean function, the smaller the circuit that will result
- Simpler circuits are cheaper to build, consume less power, and run faster than complex circuits.
- With this in mind, we always want to reduce our Boolean functions to their simplest form.
- There are a number of Boolean identities that help us to do this.

Boolean Identities

1) $X \cdot 0 = 0$

2) $X \cdot 1 = X$

3) $X \cdot X = X$

4) $X \cdot \overline{X} = 0$

5) $X + 0 = X$

6) $X + 1 = 1$

7) $X + X = X$

8) $X + \overline{X} = 1$

9) $\overline{\overline{X}} = X$

Commutative Law

10A) $X \cdot Y = Y \cdot X$

10B) $X + Y = Y + X$

Associative Law

11A) $X(YZ) = (XY)Z$

11B) $X + (Y + Z) = (X + Y) + Z$

Distributive Law

12A) $X(Y + Z) = XY + XZ$

12B) $(X + Y)(W + Z) = XW + XZ + YW + YZ$

Consensus Theorem

13A) $X + \overline{X}Y = X + Y$

13B) $\overline{X} + XY = \overline{X} + Y$

13C) $X + \overline{X}\,\overline{Y} = X + \overline{Y}$

13D) $\overline{X} + X\overline{Y} = \overline{X} + \overline{Y}$

DeMorgan's

14A) $\overline{XY} = \overline{X} + \overline{Y}$

14B) $\overline{X + Y} = \overline{X}\;\overline{Y}$

How to Prove These Boolean Identities

- All of the above identities can be proved using truth tables
- To do this, you use truth tables to show all of the possible values of both sides of the equation
- If they are identical, then the identity is true

Proof of $(X + Y)' = X'Y'$

X	Y	$X + Y$	$(X + Y)'$	X'	Y'	$X'Y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

- the values in the columns under $(X + Y)'$ and $X'Y'$ are identical

Simplifying Boolean Expression

- Which identity is used at each step?

$$
\begin{aligned}
X'Y' + XYZ + X'Y &= X'Y' + X'Y + XYZ \\
&= X'(Y' + Y) + XYZ \\
&= X'(1) + XYZ \\
&= X' + XYZ \\
&= X' + YZ
\end{aligned}
$$

$(A+\overline{B}+\overline{C})(A+\overline{B}+C)(A+B+\overline{C})$
$A(A+\overline{B}+C)(A+B+\overline{C}) + \overline{B}(A+\overline{B}+C)(A+B+\overline{C}) + \overline{C}(A+\overline{B}+C)(A+B+\overline{C})$
$(AA+A\overline{B}+AC)(AA+AB+A\overline{C}) + (A\overline{B}+\overline{B}\,\overline{B}+\overline{B}\,\overline{C})(A\overline{B}+B\overline{B}+\overline{B}\,\overline{C}) + (A\overline{C}+\overline{B}\,\overline{C}+C\overline{C})(A\overline{C}+B\overline{C}+\overline{C}\,\overline{C})$
$AA = A$ (Idempotent Law)
$B\overline{B} = C\overline{C} = 0$ (Complement Law)
$(A+A\overline{B}+AC)(A+AB+A\overline{C}) + (A\overline{B}+\overline{B}+\overline{B}\,\overline{C})(A\overline{B}+0+\overline{B}\,\overline{C}) + (A\overline{C}+\overline{B}\,\overline{C}+0)(A\overline{C}+B\overline{C}+\overline{C})$
$A + AB = A$ (Absorption Law)
$A + AC = A$ (Absorption Law)
$A + 0 = A$ (Identity Law)
$(A+AC)(A+AB+A\overline{C}) + (A\overline{B}+\overline{B}+\overline{B}\,\overline{C})(A\overline{B}+\overline{B}\,\overline{C}) + (A\overline{C}+\overline{B}\,\overline{C})(A\overline{C}+B\overline{C}+\overline{C})$
$A(A+AB+A\overline{C}) + (\overline{B}+\overline{B}\,\overline{C})(A\overline{B}+\overline{B}\,\overline{C}) + (A\overline{C}+\overline{B}\,\overline{C})(B\overline{C}+\overline{C})$
$A(A+A\overline{C}) + \overline{B}(A\overline{B}+\overline{B}\,\overline{C}) + \overline{C}(A\overline{C}+\overline{B}\,\overline{C})$
$AA + (A\overline{B}+\overline{B}\,\overline{B}\,\overline{C}) + (A\overline{C}\,\overline{C}+\overline{B}\,\overline{C}\,\overline{C})$
$A + (A\overline{B}+\overline{B}\,\overline{C}) + (A\overline{C}+\overline{B}\,\overline{C})$
$A + (A\overline{B}) + (\overline{B}\,\overline{C}) + (A\overline{C})$
$A + (\overline{B}\,\overline{C}) + (A\overline{C})$
$A + (\overline{B}\,\overline{C})$

Complements

- Sometimes it is more economical to build a circuit using <u>the complement of a function</u>

- DeMorgan's law provides an easy way of finding the complement of a Boolean function

$$\overline{(XY)} = \overline{X} + \overline{Y}, \quad \overline{(X+Y)} = \overline{X} \cdot \overline{Y}$$

- F = X'+YZ', what is F'?

$$\overline{\overline{X} + Y\overline{Z}} \rightarrow X(\overline{Y} + Z)$$

Canonical Forms

- There are many ways of representing the same Boolean expression
- Logically equivalent expressions have identical truth tables. For example, (X+Y)' = X'Y’
- In order to eliminate confusion, designers express Boolean functions in standardized or canonical form
- There are two canonical forms
  - Sum-of-products
  - Product-of-sums

Sum-of-Products

- different "product" terms from inputs are "summed" together
- Also called Disjunctive Normal Form (DNF)

Inputs			Output	Product
C	B	A	Q	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$A.B.\overline{C}$
1	0	0	0	
1	0	1	1	$A.\overline{B}.C$
1	1	0	1	$\overline{A}.B.C$
1	1	1	0	

$$Q = A.B.\overline{C} + A.\overline{B}.C + \overline{A}.B.C$$

Example

- Write the sum-of-products expression

Inputs			Output
C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Product-of-Sums

- different "sum" terms from inputs are "product-ed" together
- Also called Conjunctive Normal Form (CNF)

Inputs			Output	Product
C	B	A	Q	
0	0	0	0	$A+B+C$
0	0	1	1	
0	1	0	0	$A+\overline{B}+C$
0	1	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	0	$A+\overline{B}+\overline{C}$
1	1	1	1	

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

Example

- Write the product-of-sums expression

Inputs			Output
C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Binary Addition

- Binary arithmetic operations are different from Boolean operations
- How to express binary addition in Boolean algebra?
  1. Draw the truth table
  2. Write out the expression in standard form
  3. Simplify it
- S = ?
- C = ?

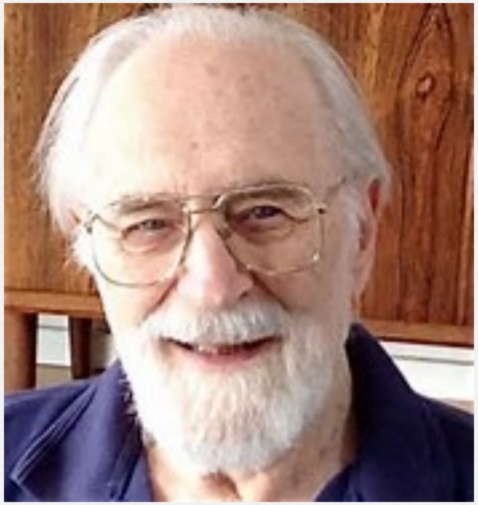
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

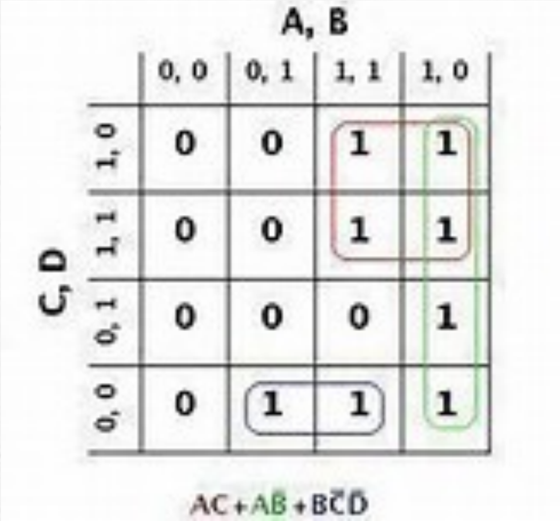
Why must the Boolean expressions be simplified?

- Both SOP and POS may not be the simplest form; only guarantee a standard form
- The expression must eventually be converted to its simplest form
  - A one-to-one correspondence exists between a Boolean expression and its implementation using electrical circuits
  - Unnecessary product terms in the expression lead to unnecessary components in the physical circuit, which in turn yield a more expensive circuit.

Karnaugh map

- The Karnaugh map (KM or K-map 卡诺图) is a method of simplifying Boolean algebra expressions introduced by Maurice Karnaugh (1924~) in 1953.

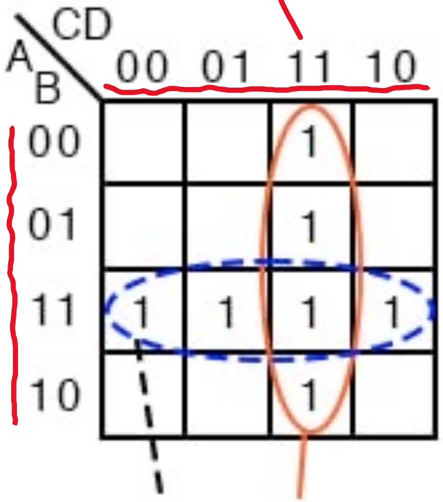




Be careful of the order!

Example

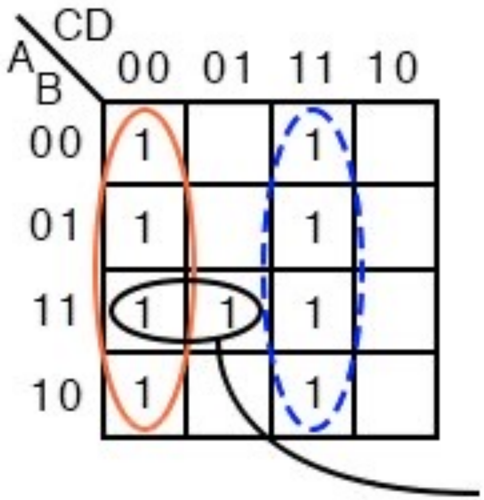
$$Out = \overline{A}\,\overline{B}CD + \overline{A}BCD + ABCD + A\overline{B}CD + AB\overline{C}\,\overline{D} + AB\overline{C}D + ABC\overline{D}$$



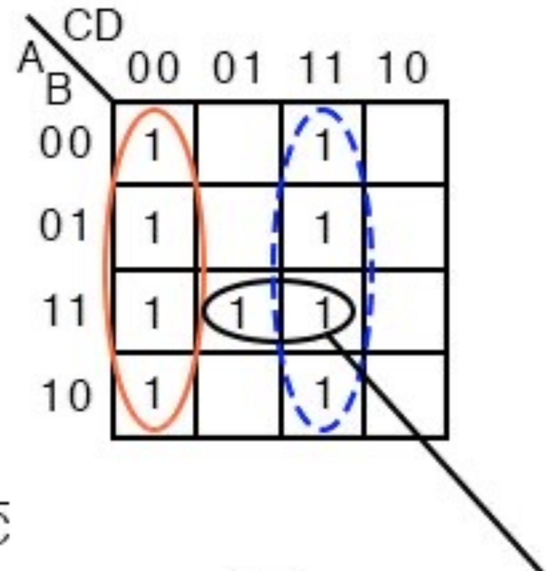
$$Out = AB + CD$$

Example

$$Out = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}\,\overline{D} + \overline{A}BCD + AB\overline{C}\,\overline{D}$$
$$+ AB\overline{C}D + ABCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}CD$$



$$Out = \overline{C}\,\overline{D} + CD + AB\overline{C}$$



$$Out = \overline{C}\,\overline{D} + CD + ABD$$

What's Next

- How to physically build the logic device to compute Boolean functions?