Switching Circuits & Logic Design

Jie-Hong Roland Jiang
江介宏

Department of Electrical Engineering
National Taiwan University

Fall 2013

§3 Boolean Algebra (Continued)

Photo: http://supercolossal.ch/2008/09/page/2/

Outline

- Multiplying out and factoring expressions
- Exclusive-OR and equivalence operations
- The consensus theorem
- Algebraic simplification of switching expressions
- Proving validity of an equation

Multiplying Out and Factoring Expressions

- Besides the distributive laws

  $X(Y+Z) = XY+XZ$ and $(X+Y)(X+Z) = X+YZ$,

  a useful theorem:

  $$(X+Y)(X'+Z) = XZ+X'Y$$

  - YZ ($=XYZ+X'YZ$) can be removed as

    $XYZ+XZ = XZ(Y+1) = XZ$ and

    $X'YZ+X'Y = X'Y(Z+1) = X'Y$

    (c.f. the consensus theorem)

Ex1. $(AB+A'C) = (A+C)(A'+B)$

Ex2. $(Q+AB')(C'D+Q') = QC'D+Q'AB'$

Multiplying Out and Factoring Expressions

- Example (multiplying out)

$$(\underline{A+B}+C')(\underline{A+B}+D)(A+B+E)(A+\underline{D'+E})(A'+C)$$

$$= (\underline{A+B}+C'D)(\underline{A+B}+E)[AC+A'(D'+E)]$$

$$= (\underline{A+B}+C'DE)(AC+A'D'+A'E)$$

$$= AC+\cancel{ABC}+A'BD'+A'BE+A'C'DE$$

Why?

Without simplification, there are 162 terms after multiplying out!

Multiplying Out and Factoring Expressions

- Example (factoring)

$$AC+A'BD'+A'BE+A'C'DE$$

$$= \underbrace{AC}_{XZ}+\underbrace{A'}_{X'}(\underbrace{BD'+BE+C'DE}_{Y})$$

$$= (A+BD'+BE+C'DE)(A'+C)$$

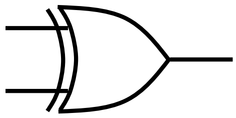
$$= [\underbrace{A+C'DE}_{X}+\underbrace{B}_{Y}(\underbrace{D'+E}_{Z})](A'+C)$$

$$= (A+B+C'DE)(A+C'DE+D'+E)(A'+C)$$

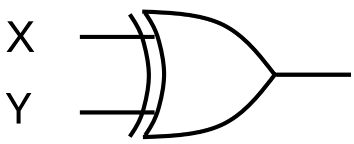
$$= (A+B+C')\ (A+B+D)\ (A+B+E)\ (A+D'+E)(A'+C)$$

Exclusive-OR and Equivalence Operations

- XOR (exclusive-OR)
  - Notation: "⊕", "≢"
  - Logic gate symbol:



$$\begin{cases} 0 \oplus 0 = 0 \\ 0 \oplus 1 = 1 \\ 1 \oplus 0 = 1 \\ 1 \oplus 1 = 0 \end{cases}$$



$Z = X \oplus Y$

XOR-gate

XY	$Z = X \oplus Y$
00	0
01	1
10	1
11	0

Exclusive-OR and Equivalence Operations

- $X \oplus Y = X'Y + XY' = (X+Y)(X'+Y')$

Properties:

- $X \oplus 0 = X$
- $X \oplus 1 = X'$
- $X \oplus X = 0$
- $X \oplus X' = 1$
- $X \oplus Y = Y \oplus X$ (commutative law)
- $(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z) = X \oplus Y \oplus Z$ (associative law)
- $X(Y \oplus Z) = XY \oplus XZ$ (distributive law)
- $(X \oplus Y)' = X \oplus Y' = X' \oplus Y = XY + X'Y'$

  Proof by truth table or by the equalities $X \oplus Y = X'Y + XY' = (X+Y)(X'+Y')$

Exclusive-OR and Equivalence Operations

- Exercise
  - $(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z)$ (associative law)
    - LHS = $(X \oplus Y)Z' + (X \oplus Y)'Z = (X'Y + XY')Z' + (XY + X'Y')Z = X'(YZ' + Y'Z) + X(YZ + Y'Z') = X'(Y \oplus Z) + X(Y \oplus Z)'$ = RHS

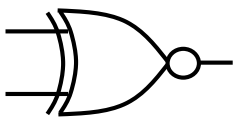
    F = $(X \oplus Y \oplus Z)$ is a parity function (i.e., F=1 iff the truth assignments on (X,Y,Z) have odd number of 1's)

  - $X(Y \oplus Z) = XY \oplus XZ$ (distributive law)
    - RHS = $(XY)(XZ)' + (XY)'(XZ) = (XY)(X' + Z') + (X' + Y')(XZ) = XYZ' + XY'Z = X(YZ' + Y'Z)$ = LHS

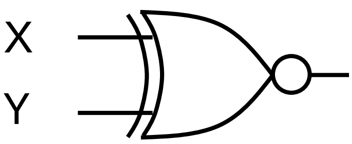
    Note that $X \oplus (YZ) \neq (X \oplus Y)(X \oplus Z)$

Exclusive-OR and Equivalence Operations

- XNOR (exclusive-NOR, equivalence)
  - Notation: "$\equiv$", "$\overline{\oplus}$"
  - Logic gate symbol:



$$\begin{cases} 0 \equiv 0 = 1 \\ 0 \equiv 1 = 0 \\ 1 \equiv 0 = 0 \\ 1 \equiv 1 = 1 \end{cases}$$



$Z = X \equiv Y$

XNOR-gate

XY	$Z = X \equiv Y$
00	1
01	0
10	0
11	1

Exclusive-OR and Equivalence Operations

- $X \equiv Y = XY + X'Y' = (X'+Y)(X+Y') = (X \oplus Y)'$

Simplify $F = (A'B \equiv C) + (B \oplus AC')$

$$
\begin{aligned}
F &= (A'B)C + (A'B)'C' + B'(AC') + B(AC')' \\
&= A'BC + (A+B')C' + AB'C' + B(A'+C) \\
&= B(A'C+A'+C) + C'(A+B'+AB') \\
&= B(A'+C) + C'(A+B')
\end{aligned}
$$

Exclusive-OR and Equivalence Operations

- Useful equality $(X'Y+XY')' = XY+X'Y'$

Simplify $F = A' \oplus B \oplus C$

$$
\begin{aligned}
F &= [A'B'+(A')'B] \oplus C \\
&= (A'B'+AB)C' + (A'B'+AB)'C \\
&= (A'B'+AB)C' + (A'B+AB')C \\
&= A'B'C'+ABC'+A'BC+AB'C
\end{aligned}
$$

Consensus Theorem

- The consensus theorem:

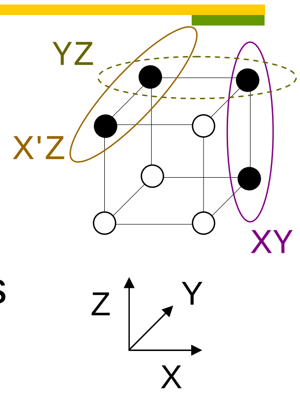
  XY+X'Z+YZ = XY+X'Z

  Proof.

  YZ (=XYZ+X'YZ) can be removed as

  XYZ+XY = XY(Z+1) = XY and

  X'YZ+X'Z = X'Z(Y+1) = X'Z



  - YZ is called the **consensus** of XY and X'Z
  - Removing (redundant) consensus terms can simplify Boolean expressions

Consensus Theorem

Example

a'b' + ac + bc' + b'c + ab

= a'b' + ac + bc'

Given a Boolean expression, e.g., F = a'bc+acd'+bcd'e,

1. search a pair of product terms p_1 (a'bc) and p_2 (acd') with complementary literals of the same variable x (a)
2. build their consensus (bcd') by ANDing p_1 (a'bc) and p_2 (acd') with their literals of variable x (a) removed
3. remove the terms (bcd'e) of F that are **covered** (in the sense of solution space) by the consensus (bcd')

   (since bcd'+ bcd'e = bcd'(1+e) = bcd')

Consensus Theorem

- The dual of the consensus theorem:

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

  Proof.

  $Y+Z$ $(=(X+Y+Z)(X'+Y+Z))$ can be removed as

  $(X+Y+Z)(X+Y) = (X+Y)(Z+1) = (X+Y)$ and

  $(X'+Y+Z)(X'+Z) = (X'+Z)(Y+1) = (X'+Z)$

  - $(Y+Z)$ is called the **consensus** (more accurately, **resolvent**) of $(X+Y)$ and $(X'+Z)$
  - Removing the (redundant) consensus terms can simplify Boolean expressions

Consensus Theorem

Example

$(a+b+c')(a+b+d')(b+c+d')$
$= (a+b+c')(b+c+d')$

$(a+b+c')\cancel{(a+b+d'+e)}(b+c+d')$
$= (a+b+c')(b+c+d')$

The clause $(a+b+d'+e)$ can be removed since it **covers** (in the sense of solution space) the consensus of $(a+b+c')$ and $(b+c+d')$

$(a+b+d')\ (a+b+d'+e) = (a+b+d')\ (1+e) = (a+b+d')$

Consensus Theorem

- Simplification by the consensus theorem may depend on the order in which terms are eliminated

a'b' + ac + a'c' + b'c + ab
= a'c' + b'c + ab

a'b' + ac + a'c' + b'c + ab
= a'b' + ac + a'c' + ab

Consensus Theorem

- Sometimes adding a consensus term may further reduce a Boolean expression

F = ABCD+B'CDE+A'B'+BCE'

F = ABCD+B'CDE+A'B'+BCE'+ACDE

Algebraic Simplification of Switching Expressions

- Simplifying an expression reduces the cost of realizing the expression using gates
  - Simplification methods:
    - Multiplying out and factoring
    - Algebraic methods
      1. Combining terms
      2. Eliminating terms
      3. Eliminating literals
      4. Adding redundant terms
    - Graphical methods (Unit 5: Karnaugh maps)

Algebraic Simplification Combining Terms

- Algebraic Method 1:

  Combining terms by XY+XY'=X

  E.g.,

  $ab'c+abc+a'bc = \underline{ab'c+abc}+\underline{abc+a'bc} = ac+bc$

  $(a+bc)(d+e')+a'(b'+c')(d+e') = d+e'$

Algebraic Simplification
Eliminating Terms

- Algebraic Method 2:
  Eliminating terms by $X+XY=X$ and by the consensus theorem $XY+X'Z+YZ=XY+X'Z$

  E.g.,

  $a'b+a'bc = a'b$

  $a'bc'+bcd+a'bd = a'bc'+bcd$

Algebraic Simplification
Eliminating Literals

- Algebraic Method 3:
  Eliminating literals by $X+X'Y=X+Y$

  E.g.,

  $A'B+A'B'C'D'+ABCD'$
  $= A'(B+B'C'D')+ABCD'$
  $= A'(B+C'D')+ABCD'$
  $= B(A'+ACD')+A'C'D'$
  $= B(A'+CD')+A'C'D'$
  $= A'B+BCD'+A'C'D'$

Algebraic Simplification
Adding Redundant Terms

- Algebraic Method 4:

  Adding redundant terms, e.g., adding xx', multiplying (x+x'), adding yz to xy+x'z, adding xy to x.

  E.g.,

  WX+XY+X'Z'+WY'Z' (add WZ' by consensus thm)

  = WX+XY+X'Z'+WY'Z'+WZ' (eliminate WY'Z')

  = WX+XY+X'Z'+WZ' (eliminate WZ')

  = WX+XY+X'Z'

Algebraic Simplification of Switching Expressions

Exercise (p.73)

$\underline{A'B'C'D'+A'BC'D'}+A'BD+\cancel{A'BC'D}+ABCD+ACD'+B'CD'$

$= A'C'D'+BD\ \underline{(A'+AC)}\ +ACD'+B'CD'$

$= A'C'D'+A'BD+\underline{BCD+ACD'}+B'CD'$

$= A'C'D'+A'BD+\cancel{BCD}+\cancel{ACD}'+B'CD'+ABC$

$= A'C'D'+A'BD+B'CD'+ABC$

Algebraic Simplification of Switching Expressions

- To simplify POS expressions, the duals of the previous four algebraic methods can be applied

Exercise (p.74)

$$\underline{(A'+B'+C')(A'+B'+C)}(B'+C)(A+C)\cancel{(A+B+C)}$$
$$= (A'+B')\cancel{(B'+C)}(A+C)$$
$$= (A'+B')(A+C)$$

Algebraic Simplification of Switching Expressions

- No easy way of determinizing when a Boolean expression has a minimum number of terms or a minimum number of literals

- Systematic methods for finding minimum SOP and POS expressions will be discussed in Units 5 and 6

Proving Validity of an Equation

- A Boolean expression is **valid** (**satisfiable**) if it is true under every (some) truth assignment of the variables
  - Validity/satisfiability checking is one of the central problems in computer science
- The equation F = G is valid if and only if (iff) the expression ($F \equiv G$) is valid
- To prove equation F = G is not valid, it is sufficient to find a truth assignment of the variables that makes F and G produce different values

  E.g., $X \oplus (YZ) \neq (X \oplus Y)(X \oplus Z)$ under (X,Y,Z)=(1,0,1)

Proving Validity of an Equation

- Given an equation F = G, its validity can be determined by the following methods:
  1. Prove by the truth table
  2. Rewrite one side of the equation by applying various theorems until it is identical with the other side
  3. Rewrite both sides to the same expression
     a) Rewrite every side independently
     b) Perform the same **reversible** operation on both sides

     E.g.,

     complement both sides (reversible)

     multiply both sides with the same expression (irreversible)

     add the same term to both sides (irreversible)

     If F=G, then aF=aG and b+F=b+G for arbitrary a, b
     The converse is not true

     Why?

Proving Validity of an Equation

- When methods 2 and 3 above are used, the following steps can be useful
  1. First reduce both sides to SOP
  2. Compare the difference between both sides
  3. Add terms to one side of the equation that are present on the other side
  4. Finally eliminate terms from one side that are not present on the other side

Proving Validity of an Equation

Example

Show that A'BD'+BCD+ABC'+AB'D = BC'D'+AD+A'BC

A'BD'+BCD+ABC'+AB'D

= A'BD'+BCD+ABC'+AB'D+BC'D'+A'BC+ABD

= AD+ ~~A'BD'~~+~~BCD~~+ ~~ABC'~~+BC'D'+A'BC

= BC'D'+AD+A'BC

Proving Validity of an Equation

Example

Show that A'BC'D+(A'+BC)(A+C'D')+BC'D+A'BC' =
ABCD+A'C'D'+ABD+ABCD'+BC'D

LHS
A'BC'D+(A'+BC)(A+C'D')+BC'D+A'BC'
= (A'+BC)(A+C'D')+BC'D+A'BC'
= ABC+A'C'D'+BC'D+A'BC'
= ABC+A'C'D'+BC'D
RHS
ABCD+A'C'D'+ABD+ABCD'+BC'D
= ABC+A'C'D'+ABD+BC'D
= ABC+A'C'D'+BC'D

What rules are used?

Boolean Algebra vs. Ordinary Algebra

- Some theorems of Boolean algebra (BA) are not true for ordinary algebra (OA), and vice versa
  - E.g.,
    Cancellation law for OA (not for BA):
    If x+y=x+z, then y=z
    (counterexample for BA: x=1,y=0,z=1)

    If xy=xz for x≠0, then y=z
    (counterexample for BA: x=0,y=0,z=1)

Boolean Algebra vs. Ordinary Algebra

- The converse is true for BA:

  If $y=z$, then $x+y=x+z$

  If $y=z$, then $xy=xz$

  Why?

Proving Validity of an Equation

- More exercises in p.77-82