Lecture (05)
Boolean Algebra 3

By:

Dr. Ahmed ElShafee

Multiplying Out and Factoring Expressions (POS& SOP)

- **Multiplying out**
- $X(Y+Z) = XY + XZ$
- $(X+Y)(X+Z) = X + YZ$

$$(X+Y)(X+Z) = XX + XZ + YX + YZ$$
$$= X + XZ + XY + YZ$$
$$= X + XY + YZ = X + YZ$$

Operations with 0 and 1:
1. $X + 0 = X$ — 1D. $X \cdot 1 = X$
2. $X + 1 = 1$ — 2D. $X \cdot 0 = 0$

Idempotent laws:
3. $X + X = X$ — 3D. $X \cdot X = X$

Involution law:
4. $(X')' = X$

Laws of complementarity:
5. $X + X' = 1$ — 5D. $X \cdot X' = 0$

Commutative laws:
6. $X + Y = Y + X$ — 6D. $XY = YX$

Associative laws:
7. $(X + Y) + Z = X + (Y + Z) = X + Y + Z$ — 7D. $(XY)Z = X(YZ) = XYZ$

Distributive laws:
8. $X(Y + Z) = XY + XZ$ — 8D. $X + YZ = (X + Y)(X + Z)$

DeMorgan's laws:
9. $(X + Y)' = X'Y'$ — 9D. $(XY)' = X' + Y'$

, absorption

$x + xy = x$ — (b) $x(x + y) = x$

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

$$= XX' + XZ + YX' + YZ$$
$$= 0 + XZ + X'Y + YZ$$
$$= XZ + X'Y + (X + X')YZ$$
$$= XZ + X'Y + XYZ + X'YZ$$
$$= XZ(1+Y) + X'Y(1+Z)$$
$$= XZ + X'Y$$

Operations with 0 and 1:
1. $X + 0 = X$ — 1D. $X \cdot 1 = X$
2. $X + 1 = 1$ — 2D. $X \cdot 0 = 0$

Idempotent laws:
3. $X + X = X$ — 3D. $X \cdot X = X$

Involution law:
4. $(X')' = X$

Laws of complementarity:
5. $X + X' = 1$ — 5D. $X \cdot X' = 0$

Commutative laws:
6. $X + Y = Y + X$ — 6D. $XY = YX$

Associative laws:
7. $(X + Y) + Z = X + (Y + Z) = X + Y + Z$ — 7D. $(XY)Z = X(YZ) = XYZ$

Distributive laws:
8. $X(Y + Z) = XY + XZ$ — 8D. $X + YZ = (X + Y)(X + Z)$

DeMorgan's laws:
9. $(X + Y)' = X'Y'$ — 9D. $(XY)' = X' + Y'$

, absorption

$x + xy = x$ (b) $x(x + y) = x$

- **Factoring (rules from multiplying out)**
- $XY + XZ = X(Y+Z)$
- $X + YZ = (X+Y)(X+Z)$
- $XZ + X'Y = (X+Y)(X'+Z)$

Example 01

Multiplying out (POS => SOP)

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

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

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

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

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

A+B → X, C'→Y, E→Z; 1st law
A→X, D'+E→Y, C→Z; 1st law

$$=(A+B+C'D)(A+B+E)[AC+A'(D'+E)]$$

A+B → X, C'D→Y, E→Y; 1st law
Distributive law

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

Distributive law

= AAC + AA'D'+AA'E+ABC+A'BD'+A'BE+ACC'DE+A'C'DD'E+A'C'DEE

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

Example 02

Factoring (SOP => POS)

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

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

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

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

A' common factor

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

A→X, (BD'+BE+C'DE)→Y, C→Z;2nd law

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

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

A+C'DE→X, (B)→Y, (D'+E)→Z;1st law

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

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

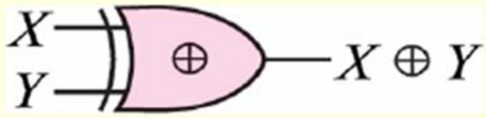
$$= (A + B + C')(A + B + D)(A + B + E)(A + D' + E)(A' + C)$$

Exclusive-OR

- Exclusive-OR: $0 \oplus 0 = 0 \quad 0 \oplus 1 = 1$
  $1 \oplus 0 = 1 \quad 1 \oplus 1 = 0$

Truth table and gate for $X \oplus Y$

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0



$X \oplus Y = 1$ if and only if X=1 or Y=1 and X and Y are not both 1.

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

- $X \oplus Y = X'Y + XY'$
- Theorems apply to *exclusive OR:*

$$X \oplus 0 = X$$

$$X \oplus 1 = X'$$

$$X \oplus X = 0$$

$$X \oplus X' = 1$$

$$X \oplus Y = Y \oplus X \text{ (commutative law)}$$

$$(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z) = X \oplus Y \oplus Z \text{ (associative law)}$$

$$X(Y \oplus Z) = XY \oplus XZ \text{ (distributive law)}$$

$$(X \oplus Y)' = X \oplus Y' = X' \oplus Y = XY + X'Y'$$ D’emorgans

- Proof of the distributive law

$$X(Y \oplus Z) = XY \oplus XZ$$

Proof:

$$XY \oplus XZ = XY(XZ)' + (XY)'XZ = XY(X'+Z') + (X'+Y')XZ$$

$$= XYZ' + XY'Z$$

$$= X(YZ' + Y'Z) = X(Y \oplus Z)$$

Equivalence

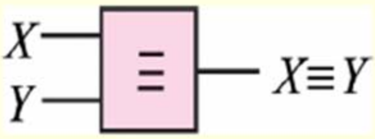
- The equivalence operation($\equiv$) is defined by

$(0 \equiv 0) = 1 \qquad (0 \equiv 1) = 0$

$(1 \equiv 0) = 0 \qquad (1 \equiv 1) = 1$

- The truth table for $X \equiv Y$ is

X	Y	$X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1



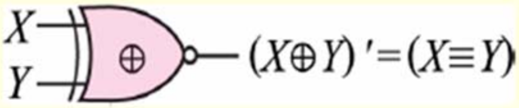
- ◆ $(X \equiv Y) = 1$ if and only if $X = Y$.

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

  Equivalence is the **complement** of *exclusive-OR*:

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

- Alternate symbol for the equivalence gate



The equivalence gate is also called an ***exclusive-NOR*** gate.

Example 03

- Find SOP $A' \oplus B \oplus C$

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

Xor complement - .equiv

Conversion of Sentences to Boolean Equation

- The *three main steps* in designing a **single output combinational switching circuit** are:
- **1**. Find a switching function which specifies the desired behavior of the circuit.
- **2**. Find a simplified algebraic expression for the function.
- **3**. Realize the simplified function using available logic eldments.

Example 08

- Translate sentences into Boolean equation
- "*Mary watches TV* if *it is Monday night* and *she has finished her homework*."

- "*Mary watches TV* if *it is Monday night* and *she has finished her homework*."

- Define two-valued variables:
- **F**=1 if "*Mary watches TV* " is true; otherwise, **F**=0.
- **A**=1 if "*it is Monday night* " is true; otherwise **A**=0.
- **B**=1 if "*she has finished her homework*" is true; otherwise **B**=0.
- **F=A.B, F is true if A and B are both true.**

Example 09

An *alarm circuit* is to be designed which operates as follows:

"*The alarm will ring* if and only if *the alarm switch is on* and *the door is not closed* or *it is after 6 P.M.* and *the window is not closed*."

- An *alarm circuit* is to be designed which operates as follows:

  "*The alarm will ring* if and only if *the alarm*

  Z A

  switch is on and *the door is not closed* or *it is*

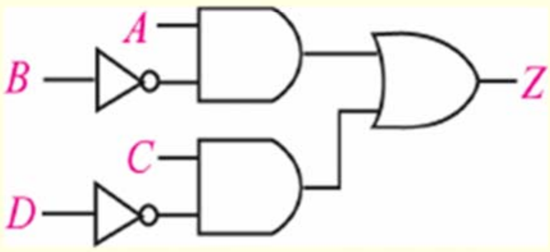
  B'

  after 6 P.M. and *the window is not closed*."

  C D'

$$Z = AB' + CD'$$

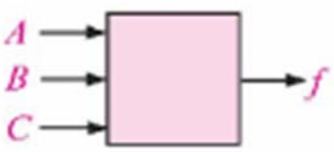
$$Z = A(B' + CD')$$



Example 10

Design a switching circuit with three inputs *A, B,* and *C* and one output *f*. The input *A,B,* and *C* represent the *first*, *second*, and *third* bits, respectively, for a binary number *N*. *f=1* if $N \geq 011_2$ and *f=0* if $N < 011_2$.

- *N*. *f=1* if $N \geq 011_2$ and *f=0* if $N < 011_2$.



(a)

A	B	C	f	f'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

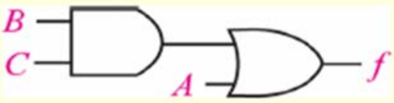
(b)

A	B	C	f	f'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Derive an algebraic expression for f from the truth table by using the combinations of values of A, B, and C for which $f=1$.

$$f = A'BC + AB'C' + AB'C + ABC' + ABC$$
$$= A'BC + AB' + AB = A'BC + A = A + BC$$

The circuit is



First write f' as a sum of products, and then complement the result. f' is 1 for input combinations ABC=000, 001, 010, so

$$f' = A'B'C' + A'B'C + A'BC'$$

$$f = (f')' = (A'B'C' + A'B'C + A'BC')'$$
$$= (A+B+C)(A+B+C')(A+B'+C)$$
(Three 3-input OR gates and one 3-input AND gate)
$$= (A+B)(A+B'+C) \quad \text{[Two OR gates and one AND gate]}$$
$$= A + BC$$

Thanks,..

See you next week (ISA),…