# Truth Tables

***Algebra:*** **variables, values, operations**

**In Boolean algebra, the values are the symbols 0 and 1**
**If a logic statement is false, it has value 0**
**If a logic statement is true, it has value 1**

**Operations: AND, OR, NOT**

| X | Y | X AND Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| X | NOT X |
|---|---|
| 0 | 1 |
| 1 | 0 |

| X | Y | X OR Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# Boolean Equations

***Boolean Algebra***

**values: 0, 1**
**variables: A, B, C, . . ., X, Y, Z**
**operations: NOT, AND, OR, . . .**

**NOT X is written as $\overline{X}$**
**X AND Y is written as X & Y, or sometimes X Y**
**X OR Y is written as X + Y**

***Deriving Boolean equations from truth tables:***

| A | B | Sum | Carry |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Sum = $\overline{A}\,B + A\,\overline{B}$**

**OR'd together *product* terms for each truth table row where the function is 1**

**if input variable is 0, it appears in complemented form; if 1, it appears uncomplemented**

**Carry = A B**

# Boolean Algebra

***Another example:***

| A | B | Cin | Sum | Cout |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 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 |

$$\text{Sum} = \overline{A}\,\overline{B}\,\text{Cin} + \overline{A}\,B\,\overline{\text{Cin}} + A\,\overline{B}\,\overline{\text{Cin}} + A\,B\,\text{Cin}$$

$$\text{Cout} = \overline{A}\,B\,\text{Cin} + A\,\overline{B}\,\text{Cin} + A\,B\,\overline{\text{Cin}} + A\,B\,\text{Cin}$$

# Boolean Algebra

***Reducing the complexity of Boolean equations***

**Laws of Boolean algebra can be applied to full adder's carry out function to derive the following simplified expression:**

**Cout = A Cin + B Cin + A B**

| | A | B | Cin | Cout |
|---|---|---|---|---|
| B Cin | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 0 |
| | 0 | 1 | 0 | 0 |
| | 0 | 1 | 1 | 1 |
| A Cin | 1 | 0 | 0 | 0 |
| | 1 | 0 | 1 | 1 |
| | 1 | 1 | 0 | 1 |
| A B | 1 | 1 | 1 | 1 |

***Verify equivalence with the original Carry Out truth table:***

**place a 1 in each truth table row where the product term is true**

**each product term in the above equation covers exactly two rows in the truth table; several rows are "covered" by more than one term**

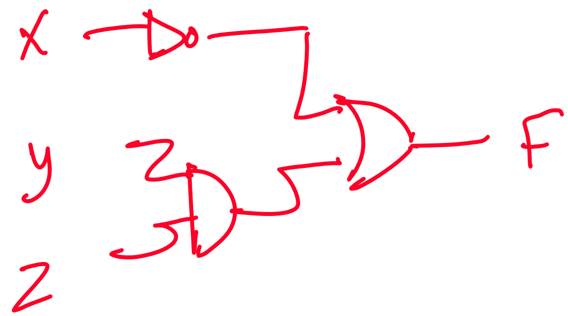
# Representations of Boolean Functions

Boolean Function: $F = \overline{X} + YZ$

Truth Table:

| X | Y | Z | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Circuit Diagram:




$f = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + xyz$

# Why Boolean Algebra/Logic Minimization?

Logic Minimization: reduce complexity of the gate level implementation

- reduce number of literals (gate inputs)
- reduce number of gates
- reduce number of levels of gates

fewer inputs implies faster gates in some technologies

fan-ins (number of gate inputs) are limited in some technologies

fewer levels of gates implies reduced signal propagation delays

number of gates (or gate packages) influences manufacturing costs

## Basic Boolean Identities:

Page 57

$X + 0 =$ X

$X * 1 =$ X

$X + 1 =$ 1

$X * 0 =$ 0

$X + X =$ x

$X * X =$ X

$X + \overline{X} =$ 1

$X * \overline{X} =$ 0

$\overline{\overline{X}} =$ X

## Basic Laws

Commutative Law:

| | |
|---|---|
| $X + Y = Y + X$ | $XY = YX$ |

Associative Law:

| | |
|---|---|
| $X+(Y+Z) = (X+Y)+Z$ | $X(YZ)=(XY)Z$ |

Distributive Law:

| | |
|---|---|
| $X(Y+Z) = XY + XZ$ | $X+YZ = (X+Y)(X+Z)$ |

## Boolean Manipulations

Boolean Function: $F = XYZ + \overline{X}Y + XY\overline{Z}$

Truth Table:

| X | Y | Z | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Reduce Function:

$F = XYZ + \overline{X}Y + XY\overline{Z}$

$= Y(XZ + \overline{X} + X\overline{Z})$

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

$= Y(X(1) + \overline{X})$

$= Y(X + \overline{X})$

$= Y(1)$

$= \boxed{Y}$

# Advanced Laws

- $X+XY =$ $X(1+Y) = X(1) = X$
- $XY + X\overline{Y} =$ $X(Y+\overline{Y}) = X(1) = X$
- $X+\overline{X}Y =$ $X+Y$
- $X(X+Y) =$ $XX + XY = X + XY = X$
- $(X+Y)(X+\overline{Y}) =$ $X(X+Y) + \overline{Y}(X+Y) = (\overset{\to X}{XX} + XY) + X\overline{Y} + \overline{Y}Y =$

  $= X + X\overline{Y} + \overline{Y}Y$

  $= X + X\overline{Y}$

  $= X(1+\overline{Y})$

  $= X$
- $X(\overline{X}+Y) =$

  $X\overline{X} + XY$

  $= 0 + XY$

  $= XY$

## Boolean Manipulations (cont.)

Boolean Function: $F = \overline{X}YZ + XZ$

Truth Table:

| X | Y | Z | F |
|---|---|---|---|
| 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 | 0 |
| 1 | 1 | 1 | 1 |

Reduce Function:

$$F = z(\bar{x}y + x)$$

$$= z(x+y)$$

## Boolean Manipulations (cont.)

Boolean Function: $F = (X+\bar{Y}+X\bar{Y})(XY+\bar{X}Z+YZ)$

Truth Table:

| X | Y | Z | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Reduce Function:

$F = XXY + X\bar{X}Z + XYZ + XY\bar{Y}$

$+ \bar{X}\bar{Y}Z + Y\bar{Y}Z$

$+ X\bar{Y}XY + X\bar{Y}\bar{X}Z + XYYZ$

$= XXY + XYZ + \bar{X}\bar{Y}Z$

$= XY + XYZ + \bar{X}\bar{Y}Z$

$= XY(1+Z) + \bar{X}\bar{Y}Z$

$= XY + \bar{X}\bar{Y}Z$

# DeMorgan’s Law

$$\overline{(X + Y)} = \overline{X} * \overline{Y}$$

| X | Y | $\overline{X}$ | $\overline{Y}$ | $\overline{X+Y}$ | $\overline{X} \bullet \overline{Y}$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |

$$\overline{(X * Y)} = \overline{X} + \overline{Y}$$

| X | Y | $\overline{X}$ | $\overline{Y}$ | $\overline{X \bullet Y}$ | $\overline{X} + \overline{Y}$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |

**DeMorgan's Law can be used to convert AND/OR expressions to OR/AND expressions**

**Example:**

$$Z = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + A\,\overline{B}\,C + A\,B\,\overline{C}$$

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