CALIFORNIA STATE UNIVERSITY FULLERTON

EGEC 180 – Digital Logic and Computer Structures

Spring 2024

Lecture 4: Boolean Algebra(2.2-2.3)

Rakesh Mahto, Ph.D.
Office: *E 314, California State University, Fullerton*
Office Hour: *Monday and Wednesday 2:00 - 3:30 pm*
Or by appointment
Office Hour Zoom Meeting ID: *891 2907 5346*
Email: *ramahto@fullerton.edu*
Phone No: *657-278-7274*

Boolean Functions/Expressions and Boolean Algebra

Boolean Functions/Expressions

Boolean functions are used to specify logic or digital circuits (yes-no, true-false, 0-1, salt-pepper, High-Low, …). Specify a “state”.

Different ways to express Boolean Functions.

1. Boolean Algebra is a mathematical form used to represent Boolean functions.
2. Gates are a graphical form to represent Boolean functions.
3. VHDL (Very High Speed Integrated Circuit Hardware Description Language) is a textual form to represent Boolean functions.

(Three more: K-MAPS, Truth Tables, Timing Diagrams)

One more? 7 total ways

Examples of Digital Circuits: ???

Boolean Algebra

Basic mathematics needed for the study of the logic design of Digital Systems. Has many other applications including Set Theory and Mathematical Logic. Two-valued Boolean Algebra is also referred to as Switching Algebra, all variables assume only two values.

We will use a Boolean variable such as X or Y to represent the input or output of a switching element. Each variable can only take two different values. The symbols "1" or "0" are used to represent these two values.

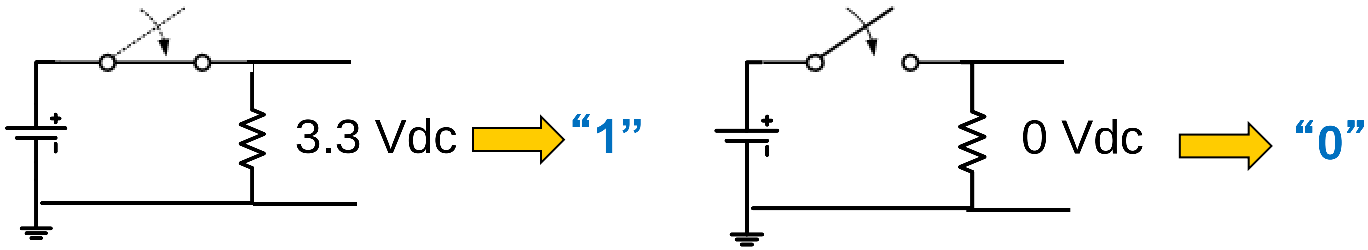
Also known as Binary Logic: deals with variables that take on two discrete values, and with operations that assume logical meaning.

Boolean Algebra

The symbols "0" and "1" in Boolean Algebra do not have a numeric value; instead they represent two different states in a logical circuit and are the two values of a switching variable.

"High/Low", "True/False","On/Off

In a logic gate, 0 (usually) represents a range of low voltages, and 1 represents a range of high voltages. In a switch, 0 (usually) represents an open switch, and 1 represents a closed circuit.



Logic gates are blocks of hardware (digital circuits) that produce a logic-1 or a logic-0 output if input logic is valid.

Digital circuit = switching circuit = logic circuit = gate.

Well suited for the analysis and design of Digital Systems (collection of digital circuits).

Boolean Algebra

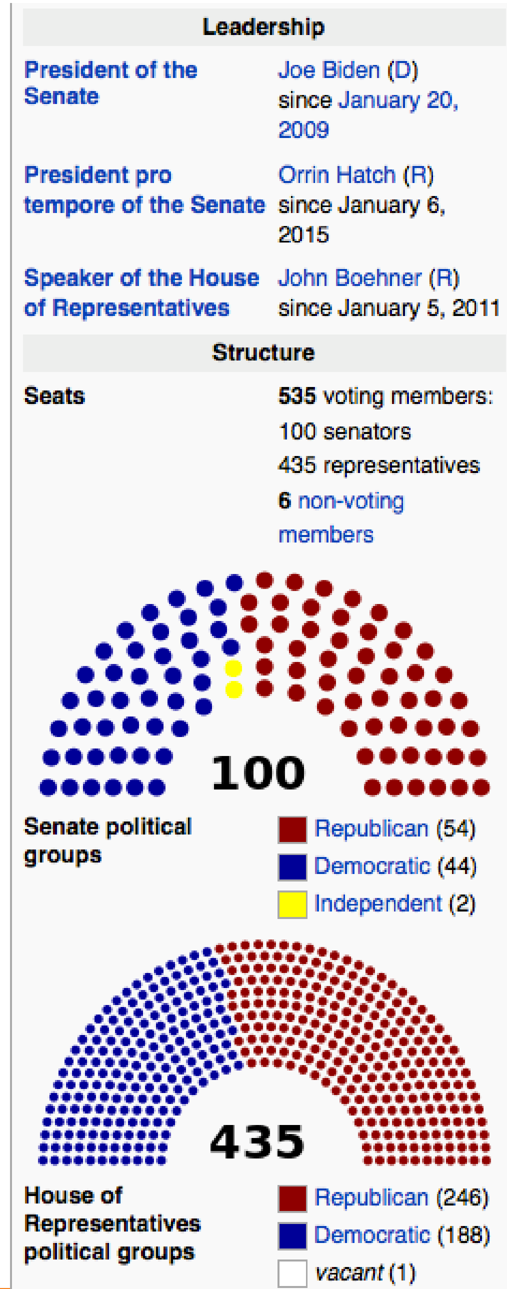
Elements, Operators and Postulates (Laws – unproven axioms)

Boolean Algebra

Algebra is defined on a *set of elements* {0,1}, a *set of operators* (AND, OR, NOT), and a number of *postulates (laws)* (unproven axioms).

Postulates are the building blocks from which it is possible to deduce the

- rules,
- theorems, and
- properties of the algebraic system.

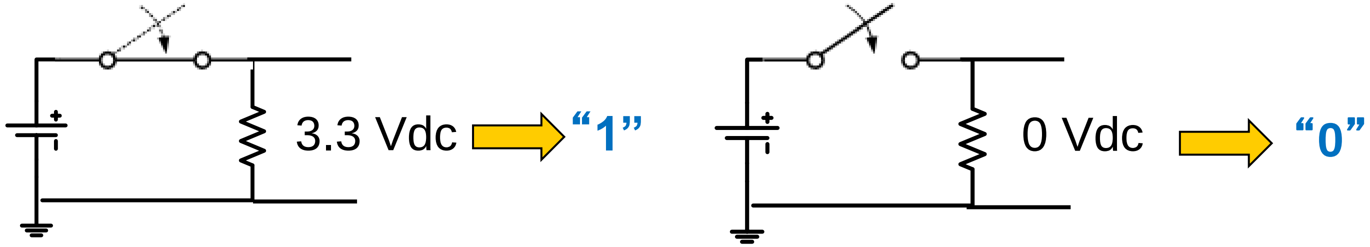


Boolean Algebra

The symbols "0" and "1" in Boolean Algebra do not have a numeric value; instead they represent two different states in a logical circuit and are the two values of a switching variable.

"High/Low", "True/False","On/Off

In a logic gate, 0 (usually) represents a range of low voltages, and 1 represents a range of high voltages. In a switch, 0 (usually) represents an open switch, and 1 represents a closed circuit.



Logic gates are blocks of hardware (digital circuits) that produce a logic-1 or a logic-0 output if input logic is valid.

Digital circuit = switching circuit = logic circuit = gate.

Well suited for the analysis and design of Digital Systems (collection of digital circuits).

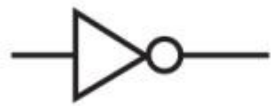
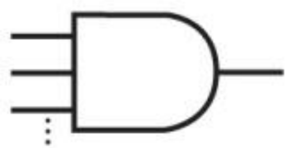
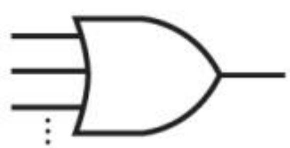
Boolean Algebra Postulates

Boolean Algebra Postulates: List of Huntington's first set of postulates

Each postulate or rule has a dual: interchange the identity elements (0,1) and the binary operators (AND, OR)

Postulates for Boolean algebra with two distinct values (1 or 0) for each variable		
Variable dominant rule	P1a: $X \cdot 1 = X$ P1b: $X + 0 = X$	Identities
Commutative rule	P2a: $X \cdot Y = Y \cdot X$ P2b: $X + Y = Y + X$	
Distributive rule	P3a: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$ P3b: $X + Y \cdot Z = (X + Y) \cdot (X + Z)$	
Complement rule	P4a: $X \cdot \overline{X} = 0$ P4b: $X + \overline{X} = 1$	Fallacy Tautology

Basic Logic Symbols

Boolean algebra expression	Name of expression	Distinctive-shape logic symbol	Logic name
$\overline{A}$	NOT or Complement operation	A  $\overline{A}$	Inverter or NOT Gate
$A \cdot B \cdot C \cdot \ldots$	AND operation	A, B, C, $\vdots$  $A \cdot B \cdot C \cdot \ldots$	AND Gate
$A + B + C + \ldots$	OR operation	A, B, C, $\vdots$  $A + B + C + \ldots$	OR Gate

Truth Tables for the NOT, AND, and OR operator definitions

Definitions

NOT operation			AND operation			OR operation	
X	$\overline{X}$		$X\ Y$	$X \cdot Y$		$X\ Y$	$X + Y$
0	1		0 0	0		0 0	0
1	0		0 1	0		0 1	1
			1 0	0		1 0	1
			1 1	1		1 1	1

Definition of binary operators NOT, AND, and OR

Definition of NOT operator (for 1 identity element)	**Definition of AND operator (for 2 identity elements)**	**Definition of AND operator (for 3 identity elements)**	**Definition of AND operator (for 4 identity elements)**	**Definition of OR operator (for 2 identity elements)**	**Definition of OR operator (for 3 identity elements)**	**Definition of OR operator (for 4 identity elements)**
$\overline{1} = 0$	$0 \cdot 0 = 0$	$0 \cdot 0 \cdot 0 = 0$	$0 \cdot 0 \cdot 0 \cdot 0 = 0$	$0 + 0 = 0$	$0 + 0 + 0 = 0$	$0 + 0 + 0 + 0 = 0$
$\overline{0} = 1$	$0 \cdot 1 = 0$	$0 \cdot 0 \cdot 1 = 0$	$0 \cdot 0 \cdot 0 \cdot 1 = 0$	$0 + 1 = 1$	$0 + 0 + 1 = 1$	$0 + 0 + 1 + 0 = 1$
	$1 \cdot 0 = 0$	$0 \cdot 1 \cdot 0 = 0$	$0 \cdot 0 \cdot 1 \cdot 0 = 0$	$1 + 0 = 1$	$0 + 1 + 0 = 1$	$0 + 0 + 1 + 0 = 1$
	$1 \cdot 1 = 1$	$0 \cdot 1 \cdot 1 = 0$	$\cdot$	$1 + 1 = 1$	$0 + 1 + 1 = 1$	$\cdot$
		$1 \cdot 0 \cdot 0 = 0$	$\cdot$		$1 + 0 + 0 = 1$	$\cdot$
		$1 \cdot 0 \cdot 1 = 0$	$\cdot$		$1 + 0 + 1 = 1$	$\cdot$
		$1 \cdot 1 \cdot 0 = 0$	$1 \cdot 1 \cdot 1 \cdot 0 = 0$		$1 + 1 + 0 = 1$	$1 + 1 + 1 + 0 = 1$
		$1 \cdot 1 \cdot 1 = 1$	$1 \cdot 1 \cdot 1 \cdot 1 = 1$		$1 + 1 + 1 = 1$	$1 + 1 + 1 + 1 = 1$

Boolean Algebra Theorems

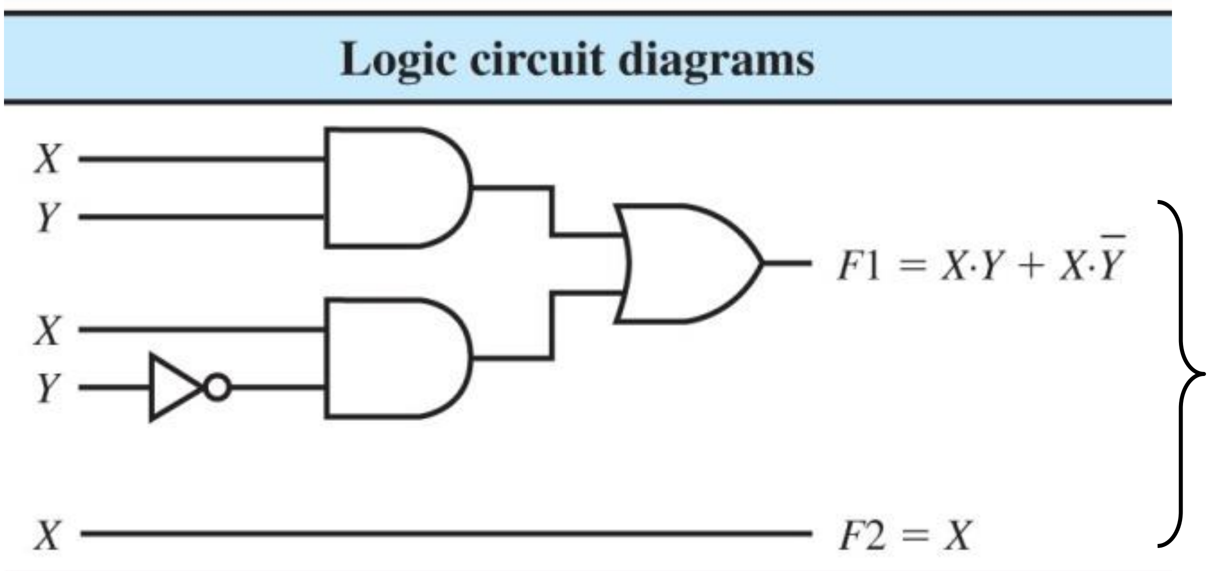
Absorption Theorem	T1a:	$X \cdot (X + Y) = X$
	T1b:	$X + X \cdot Y = X$
Adjacency Theorem	T2a:	$X \cdot Y + X \cdot \overline{Y} = X$
	T2b:	$(X + Y) \cdot (X + \overline{Y}) = X$
Associative Theorem	T3a:	$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$
	T3b:	$X + (Y + Z) = (X + Y) + Z$
Consensus Theorem	T4a:	$X \cdot Y + \overline{X} \cdot Z + Y \cdot Z = X \cdot Y + \overline{X} \cdot Z$
	T4b:	$(X + Y) \cdot (\overline{X} + Z) \cdot (Y + Z) = (X + Y) \cdot (\overline{X} + Z)$

Double Complementation or Double Negation Theorem	T7:	$\overline{\overline{X}} = X$
Idempotency Theorem	T8a:	$X \cdot X = X$
	T8b:	$X + X = X$
Identity Element Theorem	T9a:	$X \cdot 0 = 0$
	T9b:	$X + 1 = 1$
Simplification Theorem	T10a:	$X \cdot (\overline{X} + Y) = X \cdot Y$
	T10b:	$X + \overline{X} \cdot Y = X + Y$

DeMorgan's Theorem (with two variables)	T5a:	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$
	T5b:	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$
DeMorgan's Theorem (with multiple variables)	T6a:	$\overline{X \cdot Y \cdot Z \ldots} = \overline{X} + \overline{Y} + \overline{Z} + \ldots$
	T6b:	$\overline{X + Y + Z + \ldots} = \overline{X} \cdot \overline{Y} \cdot \overline{Z} \cdot \ldots$

Comparing circuit complexity for the Boolean functions F1 and F2

In general, it is better to use fewer logic gates to minimize complexity, cost, and power requirements.



Application of T2a

Adjacency Theorem

T2a: $X \cdot Y + X \cdot \overline{Y} = X$

T2b: $(X + Y) \cdot (X + \overline{Y}) = X$

Proving Boolean Algebra Theorems

Simplification Theorem T10b: $X + \overline{X} \cdot Y = X + Y$

Proof by perfect induction method

Step 1: Make the truth table

X	Y	$\overline{X}$	$\overline{X} \cdot Y$	$X + \overline{X} \cdot Y$	$X + Y$
0	0				
0	1				
1	0				
1	1				

Step 2: Fill in each column in the table using the operator definitions

X	Y	$\overline{X}$	$\overline{X} \cdot Y$	$X + \overline{X} \cdot Y$	$X + Y$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1
				EOL	= EOR

Step 3: Verify that EOL = EOR

Proof by mathematical manipulation method (proving that EOR = EOL)

Substitute postulates and/or theorems to show that EOR = EOL, which can be tricky because it involves trial and error

	Using the following postulates and/or theorems:	
$X + Y$		
$X + Y \cdot 1$	Variable dominate rule	P1a: $X \cdot 1 = X$
$X + Y \cdot (X + \overline{X})$	Complement rule	P4b: $X + \overline{X} = 1$
$X + Y \cdot X + Y \cdot \overline{X}$	Distributive rule	P3a: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
$X \cdot 1 + Y \cdot X + Y \cdot \overline{X}$	Variable dominate rule	P1a: $X \cdot 1 = X$
$X \cdot 1 + X \cdot Y + \overline{X} \cdot Y$	Commutative rule	P2a: $X \cdot Y = Y \cdot X$
$X \cdot (1 + Y) + \overline{X} \cdot Y$	Distributive rule	P3a: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
$X \cdot 1 + \overline{X} \cdot Y$	Identity element theorem	T9b: $X + 1 = 1$
$X + \overline{X} \cdot Y$	Variable dominate rule	P1a: $X \cdot 1 = X$

Note that EOR = EOL

Proof of Adjacency Theorem

Adjacency Theorem T2a: $X \cdot Y + X \cdot \overline{Y} = X$

Proof by perfect induction method

X	Y	$\overline{Y}$	$X \cdot Y$	$X \cdot \overline{Y}$	$X \cdot Y + X \cdot \overline{Y}$
0	0	1	0	0	0
0	1	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
EOR			=		EOL

Proof by mathematical manipulation method (proving that EOL = EOR)

	Using the following postulates and/or theorems:	
$X \cdot Y + X \cdot \overline{Y}$		
$X \cdot (Y + \overline{Y})$	Distributive rule	P3a: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
$X \cdot 1$	Complement rule	P4b: $X + \overline{X} = 1$
X	Variable dominate rule	P1a: $X \cdot 1 = X$
Note that EOL = EOR		

Boolean Functions and Truth Tables

Practice Problem #1

Simplify $((Z'+Y)+(X.Z')).(Y'.Z')$ if possible

Step #1 – Check for obvious identities (direct applications of postulates)

Step #2 – Expand the function Multiply terms out if necessary

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

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

Step #3 – Check for obvious identities (direct applications of postulates)

$$((Z'+Y)+(X.Z')).(Y'.Z') = Y'\underbrace{.Z'.Z'}_{Z'} + \underbrace{Y.Y'}_{0}.Z' + X.Y'\underbrace{.Z'.Z'}_{Z'}$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = Y'.Z' + \underbrace{0.Z'}_{0} + X.Y'.Z'$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'+0}_{Y'.Z'} + X.Y'.Z'$$

Practice Problem #1 (Continued)

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'+0}_{Y'.Z'} + X.Y'.Z'$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'}_{} + X.\underbrace{Y'.Z'}_{}$$

Factor

$$((Z'+Y)+(X.Z')).(Y'.Z') = Y'.Z'.\underbrace{(1+X)}_{1}$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'.(1)}_{Y'.Z'}$$

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

Practice Problem #1

Simplify $((Z'+Y)+(X.Z')).(Y'.Z')$ if possible

Step #1 – Check for obvious identities (direct applications of postulates)

Step #2 – Expand the function Multiply terms out if necessary

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

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

Step #3 – Check for obvious identities (direct applications of postulates)

$$((Z'+Y)+(X.Z')).(Y'.Z') = Y'.\underbrace{Z'.Z'}_{Z'}+\underbrace{Y.Y'}_{0}.Z' +X.Y'.\underbrace{Z'.Z'}_{Z'}$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = Y'.Z'+\underbrace{0.Z'}_{0} +X.Y'.Z'$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'+0}_{Y'.Z'} +X.Y'.Z'$$

Practice Problem #2 (Continued)

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'+0}_{Y'.Z'} + X.Y'.Z'$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'} + X.\underbrace{Y'.Z'}$$

Factor

$$((Z'+Y)+(X.Z')).(Y'.Z') = Y'.Z'.(\underbrace{1 + X}_{1})$$

$$((Z'+Y)+(X.Z')).(Y'.Z') = \underbrace{Y'.Z'.(1)}_{Y'.Z'}$$

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

Practice Problem #2

Simplify ((X'+X).(Y.X'))+(Z.X') if possible

Step #1 – Check for obvious identities (direct applications of postulates)

$$((X'+X).(Y.X'))+(Z.X') = (\underbrace{(X'+X)}_{1}.(Y.X'))+(Z.X')$$

$$((X'+X).(Y.X'))+(Z.X') = (\underbrace{1.(Y.X')}_{Y.X'})+(Z.X')$$

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

Step #2 – Expand the function Multiply terms out if necessary

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

Step #3 – Check for obvious identities (direct applications of postulates)

$$((X'+X).(Y.X'))+(Z.X') = \underbrace{Y.X'+Z.X'}_{\text{Factor}}$$

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

Practice Problem #3

Simplify $((X'+Z').(Y'+Y))+(X'+Z')$ if possible

Step #1 – Check for obvious identities (direct applications of postulates)

$$((X'+Z').(Y'+Y))+(X'+Z') = ((X'+Z').(\underbrace{Y'+Y}_{1}))+(X'+Z')$$

$$((X'+Z').(Y'+Y))+(X'+Z') = (\underbrace{(X'+Z').1}_{(X'+Z')})+(X'+Z')$$

$$((X'+Z').(Y'+Y))+(X'+Z') = \underbrace{(X'+Z')+(X'+Z')}_{X'+Z'}$$

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

Practice Problem #4

Simplify $((Z+Y')+(Y.Z)).(Y'.Z)$ if possible

Step #1 – Check for obvious identities (direct applications of postulates)

Step #2 – Expand the function Multiply terms out if necessary

$$((Z+Y')+(Y.Z)).(Y'.Z) = (Z+Y').(Y'.Z) + \underbrace{(Y.Z).(Y'.Z)}_{\text{Expand}}$$

Step #3 – Check for obvious identities (direct applications of postulates)

$$((Z+Y')+(Y.Z)).(Y'.Z) = (Z+Y').(Y'.Z) + \underbrace{Y.Y'}_{0}.\underbrace{Z.Z}_{Z}$$

$$((Z+Y')+(Y.Z)).(Y'.Z) = (Z+Y').(Y'.Z) + \underbrace{0.Z}_{0}$$

Practice Problem #4 (Concluded)

$$((Z+Y')+(Y.Z)).(Y'.Z) = \underbrace{(Z+Y').(Y'.Z) + 0}_{(Z+Y').(Y'.Z)}$$

$$((Z+Y')+(Y.Z)).(Y'.Z) = \underbrace{(Z+Y').(Y'.Z)}_{\text{Expand}}$$

$$((Z+Y')+(Y.Z)).(Y'.Z) = Y'\underbrace{.Z.Z}_{Z} + \underbrace{Y'.Y'}_{Y'}.Z$$

$$((Z+Y')+(Y.Z)).(Y'.Z) = \underbrace{Y'.Z + Y'.Z}_{Y'.Z}$$

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

Truth Table

We can use a Truth Table to specify the output values of a circuit of logic gates in terms of the values of the input variables.

If an expression has n variables and each variable can have the value 0 or 1, the number of different combinations of values of the variables is 2^n.

Thus, a Truth Table for an n-variable expression will have n columns and 2^n rows.

Two expressions are equal if they have the same value for every possible combination of the variables.

Truth Table for 3 variables

Since the expression (A + C)(B’ + C) has the same value as AB’ + C for all eight combinations of values of the variables A, B, and C, we conclude that:

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

A B C	B’	AB’	AB’+C		A+C	B’+C	(A+C)(B’+C)
0 0 0	1	0.1=0	0+0=0		0+0=0	1+0=1	0.1=0
0 0 1	1	0.1=0	0+1=1		0+1=1	1+1=1	1.1=1
0 1 0	0	0.0=0	0+0=0		0+0=0	0+0=0	0.0=0
0 1 1	0	0.0=0	0+1=1		0+1=1	0+1=1	1.1=1
1 0 0	1	1.1=1	0+1=1		1+0=1	0+1=1	1.1=1
1 0 1	1	1.1=1	1+1=1		1+1=1	1+1=1	1.1=1
1 1 0	0	1.0=0	0+0=0		1+0=1	0+0=0	1.0=0
1 1 1	0	1.0=0	1+0=1		1+1=1	0+1=1	1.1=1

Q&A