

# Fundamentals of Computer Systems

## Boolean Logic

Stephen A. Edwards

Columbia University

Fall 2012

# Boolean Logic

AN INVESTIGATION  
OF  
THE LAWS OF THOUGHT,  
ON WHICH ARE FOUNDED  
THE MATHEMATICAL THEORIES OF LOGIC  
AND PROBABILITIES.

BY  
GEORGE BOOLE, LL.D.

PROFESSOR OF MATHEMATICS IN QUEEN'S COLLEGE, COBURG.

LONDON:  
WALTON AND MABERLY,  
UPPER GOWER-STREET, AND IVY-LANE, PATERNOSTER-ROW.  
CAMBRIDGE: MACMILLAN AND CO.

1854.



George Boole  
1815–1864

# Boole's Intuition Behind Boolean Logic

Variables  $X, Y, \dots$  represent classes of things

No imprecision: A thing either is or is not in a class

If  $X$  is "sheep"  
and  $Y$  is "white  
things,"  $XY$  are  
all white sheep,

$$XY = YX$$

and

$$XX = X.$$

If  $X$  is "men"  
and  $Y$  is  
"women,"  $X + Y$   
is "both men  
and women,"

$$X + Y = Y + X$$

and

$$X + X = X.$$

If  $X$  is "men,"  $Y$   
is "women," and  
 $Z$  is "European,"  
 $Z(X + Y)$  is  
"European men  
and women"  
and

$$Z(X + Y) = ZX + ZY.$$

# The Axioms of (Any) Boolean Algebra

A Boolean Algebra consists of

A set of values  $A$

An “and” operator “ $\cdot$ ”

An “or” operator “ $+$ ”

A “not” operator  $\bar{X}$

A “false” value  $0 \in A$

A “true” value  $1 \in A$

# The Axioms of (Any) Boolean Algebra

A Boolean Algebra consists of

A set of values  $A$

An “and” operator “ $\cdot$ ”

An “or” operator “ $+$ ”

A “not” operator  $\bar{X}$

A “false” value  $0 \in A$

A “true” value  $1 \in A$

---

## Axioms

---

$$X + Y = Y + X$$

$$X \cdot Y = Y \cdot X$$

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

$$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

$$X + (X \cdot Y) = X$$

$$X \cdot (X + Y) = X$$

$$X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$$

$$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

$$X + \bar{X} = 1$$

$$X \cdot \bar{X} = 0$$

---

# The Axioms of (Any) Boolean Algebra

A Boolean Algebra consists of

A set of values  $A$

An “and” operator “ $\cdot$ ”

An “or” operator “ $+$ ”

A “not” operator  $\bar{X}$

A “false” value  $0 \in A$

A “true” value  $1 \in A$

---

## Axioms

---

$$X + Y = Y + X$$

$$X \cdot Y = Y \cdot X$$

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

$$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

$$X + (X \cdot Y) = X$$

$$X \cdot (X + Y) = X$$

$$X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$$

$$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

$$X + \bar{X} = 1$$

$$X \cdot \bar{X} = 0$$

---

We will use the first non-trivial Boolean Algebra:

$A = \{0, 1\}$ . This adds the law of excluded middle: if

$X \neq 0$  then  $X = 1$  and if  $X \neq 1$  then  $X = 0$ .

# Simplifying a Boolean Expression

“You are a New Yorker if you were born in New York or were not born in New York and lived here ten years.”

$$X + (\bar{X} \cdot Y)$$

---

## Axioms

---

$$X + Y = Y + X$$

$$X \cdot Y = Y \cdot X$$

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

$$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

$$X + (X \cdot Y) = X$$

$$X \cdot (X + Y) = X$$

$$X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$$

$$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

$$X + \bar{X} = 1$$

$$X \cdot \bar{X} = 0$$

---

Lemma:

$$\begin{aligned} X \cdot 1 &= X \cdot (X + \bar{X}) \\ &= X \cdot (X + Y) \text{ if } Y = \bar{X} \\ &= X \end{aligned}$$

# Simplifying a Boolean Expression

“You are a New Yorker if you were born in New York or were not born in New York and lived here ten years.”

$$\begin{aligned} X + (\bar{X} \cdot Y) \\ = (X + \bar{X}) \cdot (X + Y) \end{aligned}$$

---

## Axioms

---

$$X + Y = Y + X$$

$$X \cdot Y = Y \cdot X$$

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

$$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

$$X + (X \cdot Y) = X$$

$$X \cdot (X + Y) = X$$

$$X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$$

$$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$$

$$X + \bar{X} = 1$$

$$X \cdot \bar{X} = 0$$

---

Lemma:

$$\begin{aligned} X \cdot 1 &= X \cdot (X + \bar{X}) \\ &= X \cdot (X + Y) \text{ if } Y = \bar{X} \\ &= X \end{aligned}$$

# Simplifying a Boolean Expression

“You are a New Yorker if you were born in New York or were not born in New York and lived here ten years.”

$$\begin{aligned} X + (\bar{X} \cdot Y) \\ &= (X + \bar{X}) \cdot (X + Y) \\ &= 1 \cdot (X + Y) \end{aligned}$$

---

## Axioms

---

$$\begin{aligned} X + Y &= Y + X \\ X \cdot Y &= Y \cdot X \\ X + (Y + Z) &= (X + Y) + Z \\ X \cdot (Y \cdot Z) &= (X \cdot Y) \cdot Z \\ X + (X \cdot Y) &= X \\ X \cdot (X + Y) &= X \\ X \cdot (Y + Z) &= (X \cdot Y) + (X \cdot Z) \\ X + (Y \cdot Z) &= (X + Y) \cdot (X + Z) \\ X + \bar{X} &= 1 \\ X \cdot \bar{X} &= 0 \end{aligned}$$

---

Lemma:

$$\begin{aligned} X \cdot 1 &= X \cdot (X + \bar{X}) \\ &= X \cdot (X + Y) \text{ if } Y = \bar{X} \\ &= X \end{aligned}$$

# Simplifying a Boolean Expression

“You are a New Yorker if you were born in New York or were not born in New York and lived here ten years.”

$$\begin{aligned} X + (\bar{X} \cdot Y) \\ &= (X + \bar{X}) \cdot (X + Y) \\ &= 1 \cdot (X + Y) \\ &= X + Y \end{aligned}$$

---

## Axioms

---

$$\begin{aligned} X + Y &= Y + X \\ X \cdot Y &= Y \cdot X \\ X + (Y + Z) &= (X + Y) + Z \\ X \cdot (Y \cdot Z) &= (X \cdot Y) \cdot Z \\ X + (X \cdot Y) &= X \\ X \cdot (X + Y) &= X \\ X \cdot (Y + Z) &= (X \cdot Y) + (X \cdot Z) \\ X + (Y \cdot Z) &= (X + Y) \cdot (X + Z) \\ X + \bar{X} &= 1 \\ X \cdot \bar{X} &= 0 \end{aligned}$$

---

Lemma:

$$\begin{aligned} X \cdot 1 &= X \cdot (X + \bar{X}) \\ &= X \cdot (X + Y) \text{ if } Y = \bar{X} \\ &= X \end{aligned}$$

## More properties

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

$$1 + 1 + \dots + 1 = 1$$

$$X + 0 = X$$

$$X + 1 = 1$$

$$X + X = X$$

$$X + XY = X$$

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

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

$$1 \cdot 1 \cdot \dots \cdot 1 = 1$$

$$X \cdot 0 = 0$$

$$X \cdot 1 = X$$

$$X \cdot X = X$$

$$X \cdot (X + Y) = X$$

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

## More Examples

$$\begin{aligned}XY + YZ(Y + Z) &= XY + YZY + YZZ \\ &= XY + YZ \\ &= Y(X + Z)\end{aligned}$$

$$\begin{aligned}X + Y(X + Z) + XZ &= X + YX + YZ + XZ \\ &= X + YZ + XZ \\ &= X + YZ\end{aligned}$$

## More Examples

$$\begin{aligned}XYZ + X(\bar{Y} + \bar{Z}) &= XYZ + X\bar{Y} + X\bar{Z} && \text{Expand} \\ &= X(YZ + \bar{Y} + \bar{Z}) && \text{Factor w.r.t. } X \\ &= X(YZ + \bar{Y} + \bar{Z} + Y\bar{Z}) && \bar{Z} \rightarrow Y\bar{Z} \\ &= X(YZ + Y\bar{Z} + \bar{Y} + \bar{Z}) && \text{Reorder} \\ &= X(Y(Z + \bar{Z}) + \bar{Y} + \bar{Z}) && \text{Factor w.r.t. } Y \\ &= X(Y + \bar{Y} + \bar{Z}) && Y + \bar{Y} = 1 \\ &= X(1 + \bar{Z}) && 1 + \bar{Z} = 1 \\ &= X && X1 = X\end{aligned}$$

$$\begin{aligned}(X + \bar{Y} + \bar{Z})(X + \bar{Y}Z) &= XX + X\bar{Y}Z + \bar{Y}X + \bar{Y}\bar{Y}Z + \bar{Z}X + \bar{Z}\bar{Y}Z \\ &= X + X\bar{Y}Z + X\bar{Y} + \bar{Y}Z + X\bar{Z} \\ &= X + \bar{Y}Z\end{aligned}$$

## Sum-of-products form

Can always reduce a complex Boolean expression to a sum of product terms:

$$\begin{aligned}XY + \bar{X}(X + Y(Z + X\bar{Y}) + \bar{Z}) &= XY + \bar{X}(X + YZ + YX\bar{Y} + \bar{Z}) \\ &= XY + \bar{X}X + \bar{X}YZ + \bar{X}YX\bar{Y} + \bar{X}\bar{Z} \\ &= XY + \bar{X}YZ + \bar{X}\bar{Z} \\ &\quad \text{(can do better)} \\ &= Y(X + \bar{X}Z) + \bar{X}\bar{Z} \\ &= Y(X + Z) + \bar{X}\bar{Z} \\ &= \overline{Y\bar{X}\bar{Z}} + \bar{X}\bar{Z} \\ &= Y + \bar{X}\bar{Z}\end{aligned}$$

# What Does This Have To Do With Logic Circuits?

A SYMBOLIC ANALYSIS  
OF  
RELAY AND SWITCHING CIRCUITS

by

Claude Elwood Shannon  
B.S., University of Michigan  
1936

Submitted in Partial Fulfillment of the  
Requirements for the Degree of  
MASTER OF SCIENCE  
from the  
Massachusetts Institute of Technology  
1940

Signature of Author \_\_\_\_\_  
Department of Electrical Engineering, August 10, 1937

Signature of Professor  
in Charge of Research \_\_\_\_\_

Signature of Chairman of Department  
Committee on Graduate Students \_\_\_\_\_



Claude Shannon  
1916–2001

# Shannon's MS Thesis

“We shall limit our treatment to circuits containing only relay contacts and switches, and therefore at any given time the circuit between any two terminals must be either open (infinite impedance) or closed (zero impedance).”



# Shannon's MS Thesis

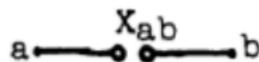


Fig. 1

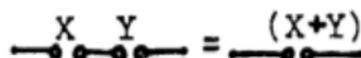


Fig. 2

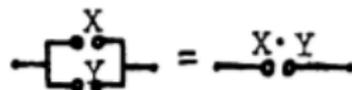


Fig. 3

"It is evident that with the above definitions the following postulates hold.

$0 \cdot 0 = 0$       A closed circuit in parallel with a closed circuit is a closed circuit.

$1 + 1 = 1$       An open circuit in series with an open circuit is an open circuit.

$1 + 0 = 0 + 1 = 1$       An open circuit in series with a closed circuit in either order is an open circuit.

$0 \cdot 1 = 1 \cdot 0 = 0$       A closed circuit in parallel with an open circuit in either order is a closed circuit.

$0 + 0 = 0$       A closed circuit in series with a closed circuit is a closed circuit.

$1 \cdot 1 = 1$       An open circuit in parallel with an open circuit is an open circuit.

At any give time either  $X = 0$  or  $X = 1$

# Alternate Notations for Boolean Logic

Operator	Math	Engineer	Schematic
Copy	$x$	$X$	$x$ — or $x$ —  — $x$
Complement	$\neg x$	$\bar{X}$	$x$ —  — $\bar{x}$
AND	$x \wedge y$	$XY$ or $X \cdot Y$	$X$ —  — $XY$ $Y$ —
OR	$x \vee y$	$X + Y$	$X$ —  — $X + Y$ $Y$ —

## Definitions

*Literal:* a Boolean variable or its complement

E.g.,  $X$   $\bar{X}$   $Y$   $\bar{Y}$

*Implicant:* A product of literals

E.g.,  $X$   $XY$   $X\bar{Y}Z$

*Minterm:* An implicant with each variable once

E.g.,  $X\bar{Y}Z$   $XYZ$   $\bar{X}\bar{Y}Z$

*Maxterm:* A sum of literals with each variable once

E.g.,  $X + \bar{Y} + Z$   $X + Y + Z$   $\bar{X} + \bar{Y} + Z$

## Be Careful with Bars

$$\overline{XY} \neq \overline{X}Y$$

## Be Careful with Bars

$$\overline{X\overline{Y}} \neq \overline{X\overline{Y}}$$

Let's check all the combinations of  $X$  and  $Y$ :

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

# Truth Tables

A *truth table* is a canonical representation of a Boolean function

$X$	$Y$	<b>Minterm</b>	<b>Maxterm</b>	$\bar{X}$	$XY$	$\overline{XY}$	$X+Y$	$\overline{X+Y}$
0	0	$\bar{X}\bar{Y}$	$X+Y$	1	0	1	0	1
0	1	$\bar{X}Y$	$X+\bar{Y}$	1	0	1	1	0
1	0	$X\bar{Y}$	$\bar{X}+Y$	0	0	1	1	0
1	1	$XY$	$\bar{X}+\bar{Y}$	0	1	0	1	0

Each row has a unique minterm and maxterm

The minterm is 1  
maxterm is 0 for only its row

## Sum-of-minterms and Product-of-maxterms

Two mechanical ways to translate a function's truth table into an expression:

$X$	$Y$	<b>Minterm</b>	<b>Maxterm</b>	$F$
0	0	$\bar{X}\bar{Y}$	$X+Y$	0
0	1	$\bar{X}Y$	$X+\bar{Y}$	1
1	0	$X\bar{Y}$	$\bar{X}+Y$	1
1	1	$XY$	$\bar{X}+\bar{Y}$	0

The sum of the minterms where the function is 1:

$$F = \bar{X}Y + X\bar{Y}$$

The product of the maxterms where the function is 0:

$$F = (X+Y)(\bar{X}+\bar{Y})$$

# Expressions to Schematics

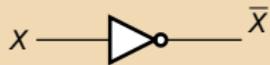
$$F = \bar{X}Y + X\bar{Y}$$

x

y

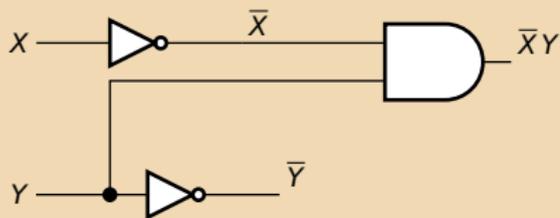
# Expressions to Schematics

$$F = \bar{X}Y + X\bar{Y}$$



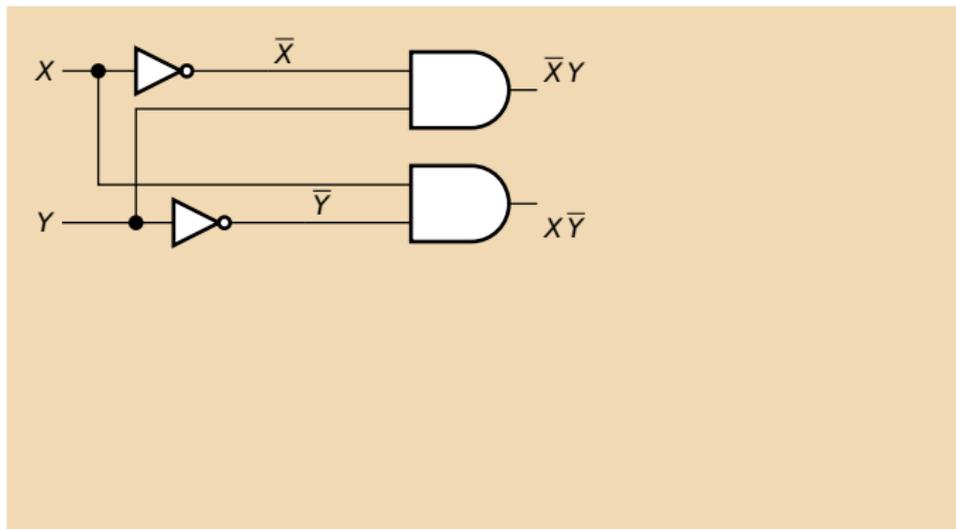
# Expressions to Schematics

$$F = \bar{X}Y + X\bar{Y}$$



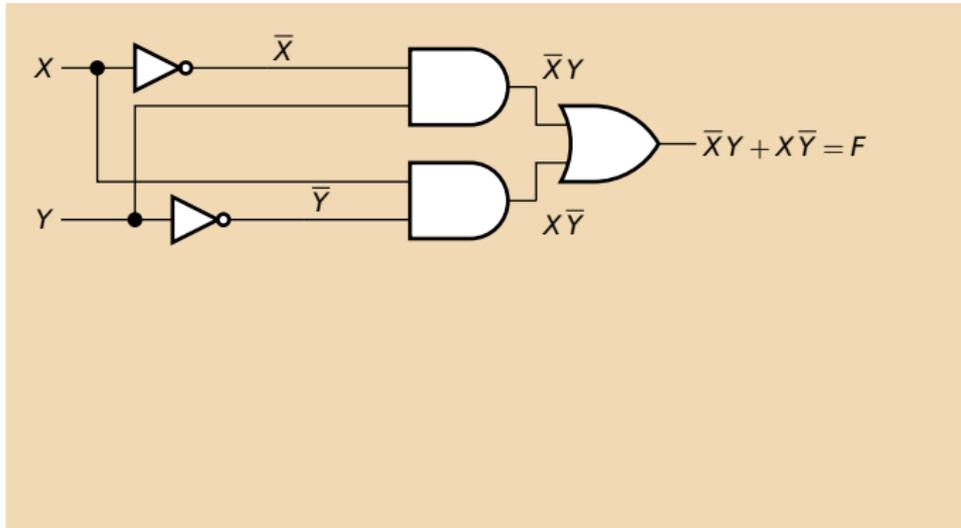
# Expressions to Schematics

$$F = \bar{X}Y + X\bar{Y}$$



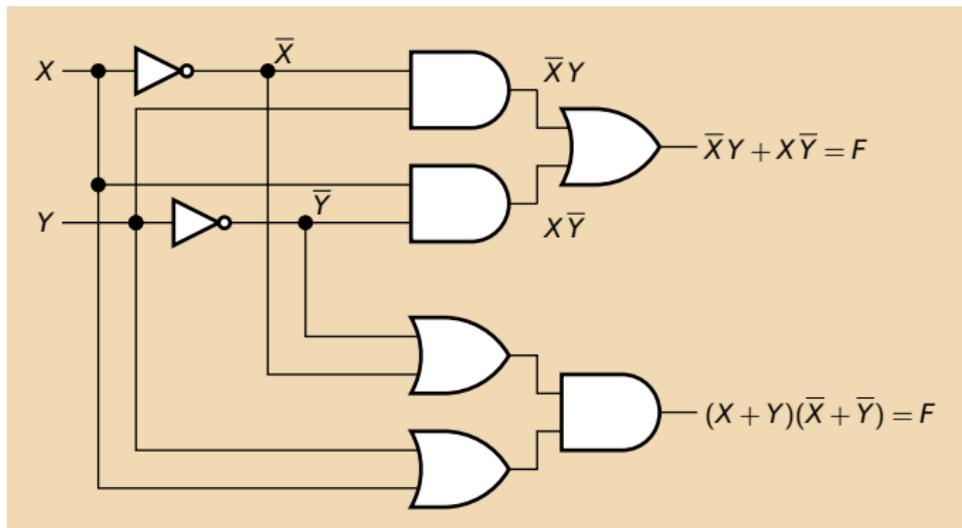
# Expressions to Schematics

$$F = \bar{X}Y + X\bar{Y}$$



# Expressions to Schematics

$$F = \bar{X}Y + X\bar{Y} = (X + Y)(\bar{X} + \bar{Y})$$



## Minterms and Maxterms: Another Example

The minterm and maxterm representation of functions may look very different:

$X$	$Y$	<b>Minterm</b>	<b>Maxterm</b>	$F$
0	0	$\bar{X}\bar{Y}$	$X+Y$	0
0	1	$\bar{X}Y$	$X+\bar{Y}$	1
1	0	$X\bar{Y}$	$\bar{X}+Y$	1
1	1	$XY$	$\bar{X}+\bar{Y}$	1

The sum of the minterms where the function is 1:

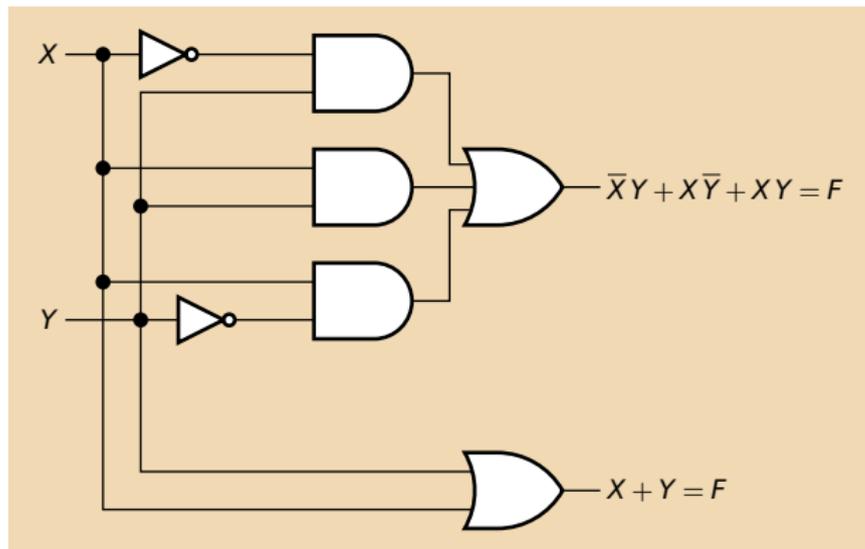
$$F = \bar{X}Y + X\bar{Y} + XY$$

The product of the maxterms where the function is 0:

$$F = X + Y$$

## Expressions to Schematics 2

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



# The Menagerie of Gates



# The Menagerie of Gates

Buffer



0		0
1		1

Inverter



0		1
1		0

AND



.		0	1
0		0	0
1		0	1

NAND



.		0	1
0		1	1
1		1	0

OR



+		0	1
0		0	1
1		1	1

NOR



$\bar{+}$		0	1
0		1	0
1		0	0

XOR



$\oplus$		0	1
0		0	1
1		1	0

XNOR



$\bar{\oplus}$		0	1
0		1	0
1		0	1

# De Morgan's Theorem

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

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

Proof by Truth Table:

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

# De Morgan's Theorem in Gates

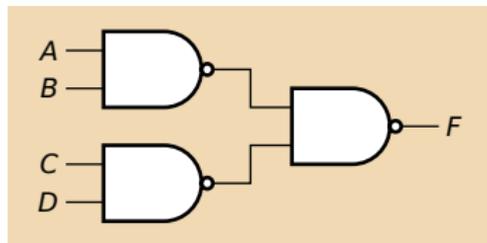
$$\overline{AB} = \overline{A} + \overline{B}$$



$$\overline{A + B} = \overline{A} \cdot \overline{B}$$



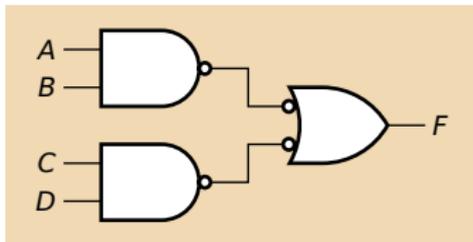
# Bubble Pushing



Apply De Morgan's Theorem:

Transform NAND into OR with inverted inputs

# Bubble Pushing

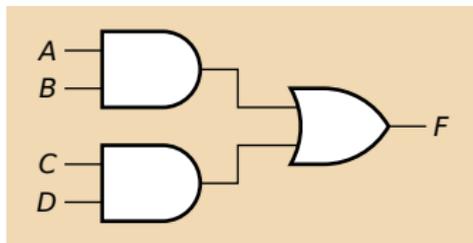


Apply De Morgan's Theorem:

Transform NAND into OR with inverted inputs

Two bubbles on a wire cancel

# Bubble Pushing

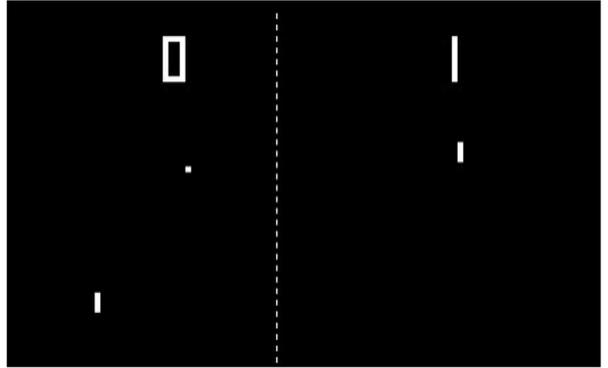


Apply De Morgan's Theorem:

Transform NAND into OR with inverted inputs

Two bubbles on a wire cancel

# PONG



*PONG*, Atari 1973

Built from TTL logic gates; no computer, no software

Launched the video arcade game revolution

## Horizontal Ball Control in PONG

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0
1	1	0	1	1
1	1	1	X	X

The ball moves either left or right.

Part of the control circuit has three inputs: *M* (“move”), *L* (“left”), and *R* (“right”).

It produces two outputs *A* and *B*.

Here, “X” means “I don’t care what the output is; I never expect this input combination to occur.”

## Horizontal Ball Control in PONG

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

E.g., assume all the X's are 0's and use Minterms:

$$A = M\bar{L}R + ML\bar{R}$$

$$B = \bar{M}\bar{L}R + \bar{M}L\bar{R} + ML\bar{R}$$

3 inv + 4 AND3 + 1 OR2 + 1 OR3

## Horizontal Ball Control in PONG

$M$	$L$	$R$	$A$	$B$
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

Assume all the X's are 1's and use Maxterms:

$$A = (M + L + \bar{R})(M + \bar{L} + R)$$

$$B = \bar{M} + L + \bar{R}$$

3 inv + 3 OR3 + 1 AND2

# Horizontal Ball Control in PONG

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0

Choosing better values for the X's  
and being much more clever:

$$A = M$$

$$B = \overline{MR}$$

1 NAND2 (!)

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0
1	1	0	1	1
1	1	1	X	X

The *M*'s are already arranged nicely

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0
		1	1	0
		1	1	1

Let's rearrange the *L*'s by permuting two pairs of rows

1	1
X	X

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0

---

Let's rearrange the *L*'s by permuting two pairs of rows

1	1	0	1	1
1	1	1	X	X

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0

---

Let's rearrange the *L*'s by permuting two pairs of rows

1	1	0	1	1
1	1	1	X	X

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0

Let's rearrange the *L*'s by permuting two pairs of rows

1	1	0	1	1
1	1	1	X	X

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	0	0	X	X
1	0	1	1	0

Let's rearrange the *L*'s by permuting two pairs of rows

1	1	0	1	1
1	1	1	X	X

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>	
0	0	0	X	X	
0	0	1	0	1	
0	1	0	0	1	
0	1	1	X	X	
			1	1	0
			1	1	1
1	0	0	X	X	
1	0	1	1	0	

Let's rearrange the *L*'s by permuting two pairs of rows

1	1
X	X

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>		
0	0	0	X	X		
0	0	1	0	1		
0	1	0	0	1		
0	1	1	X	X		
		1	1	0	1	1
		1	1	1	X	X
1	0	0	X	X		
1	0	1	1	0		

Let's rearrange the *L*'s by permuting two pairs of rows

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	1	0	1	1
1	1	1	X	X
1	0	0	X	X
1	0	1	1	0

Let's rearrange the *L*'s by permuting two pairs of rows

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
0	0	0	X	X
0	0	1	0	1
0	1	0	0	1
0	1	1	X	X
1	1	0	1	1
1	1	1	X	X
1	0	0	X	X
1	0	1	1	0

The *R*'s are really crazy; let's use the second dimension

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

$M$	$L$	$R$	$A$	$B$
$0_0$	$0_0$	$0_1$	$X_0$	$X_1$
$0_0$	$1_1$	$0_1$	$0_X$	$1_X$
$1_1$	$1_1$	$0_1$	$1_X$	$1_X$
$1_1$	$0_0$	$0_1$	$X_1$	$X_0$

The  $R$ 's are really crazy; let's use the second dimension

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
00	00	01	X0	X1
00	11	01	0X	1X
11	11	01	1X	1X
11	00	01	X1	X0

The *R*'s are really crazy; let's use the second dimension

# Karnaugh Maps

Basic trick: put “similar” variable values near each other so simple functions are obvious

<i>M</i>	<i>L</i>	<i>R</i>	<i>A</i>	<i>B</i>
00	00	01	X0	X1
00	11	01	0X	1X
11	11	01	1X	1X
11	00	01	X1	X0

*MR*

*M*

# Maurice Karnaugh's Maps

## The Map Method for Synthesis of Combinational Logic Circuits

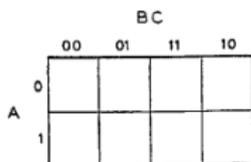
M. KARNAUGH

NONMEMBER AIEE

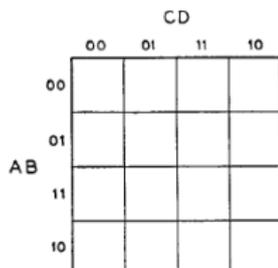
**T**HE SEARCH for simple abstract techniques to be applied to the design of switching systems is still, despite some recent advances, in its early stages. The problem in this area which has been attacked most energetically is that of the synthesis of efficient combinational that is, nonsequential, logic circuits.

be convenient to describe other methods in terms of Boolean algebra. Whenever the term "algebra" is used in this paper, it will refer to Boolean algebra, where addition corresponds to the logical connective "or," while multiplication corresponds to "and."

The minimizing chart,<sup>2</sup> developed at



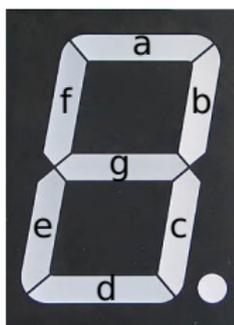
(A)



(B)

Fig. 2. Graphical representations of the input conditions for three and for four variables

# The Seven-Segment Decoder Example



<i>W</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	0	0	0	0	0	0	0

## Karnaugh Map for Seg. a

W	X	Y	Z	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	0

		Z				
		1	0	1	1	
X	{	0	1	1	1	
		X	X	0	X	W
		1	1	X	X	
				Y		

### The Karnaugh Map Sum-of-Products Challenge

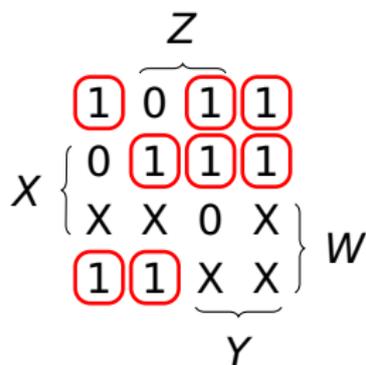
Cover all the 1's and none of the 0's using **as few literals** (gate inputs) as possible.

Few, large rectangles are good.

Covering X's is optional.

# Karnaugh Map for Seg. a

W	X	Y	Z	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	0



The minterm solution: cover each 1 with a single implicant.

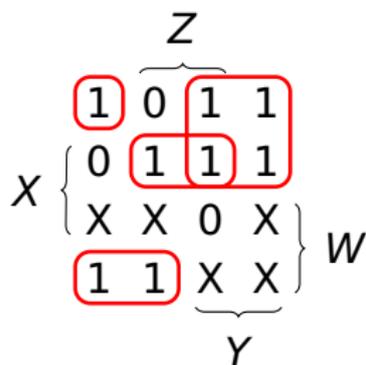
$$\begin{aligned}
 a = & \overline{W}\overline{X}\overline{Y}\overline{Z} + \overline{W}\overline{X}YZ + \overline{W}\overline{X}Y\overline{Z} + \\
 & \overline{W}X\overline{Y}Z + \overline{W}XYZ + \overline{W}XY\overline{Z} + \\
 & W\overline{X}\overline{Y}\overline{Z} + W\overline{X}\overline{Y}Z
 \end{aligned}$$

$8 \times 4 = 32$  literals

4 inv + 8 AND4 + 1 OR8

# Karnaugh Map for Seg. a

W	X	Y	Z	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	0



Merging implicants helps

Recall the distributive law:

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

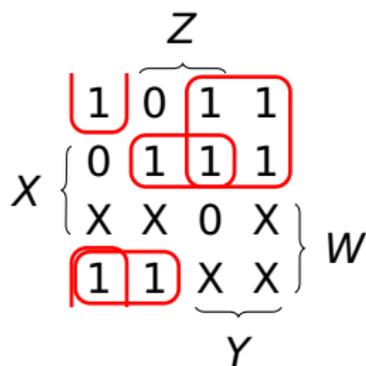
$$a = \overline{W}\overline{X}\overline{Y}\overline{Z} + \overline{W}Y + \overline{W}XZ + W\overline{X}\overline{Y}$$

$$4 + 2 + 3 + 3 = 12 \text{ literals}$$

$$4 \text{ inv} + 1 \text{ AND}_4 + 2 \text{ AND}_3 + 1 \text{ AND}_2 + 1 \text{ OR}_4$$

# Karnaugh Map for Seg. a

W	X	Y	Z	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	0



Missed one: Remember this is actually a torus.

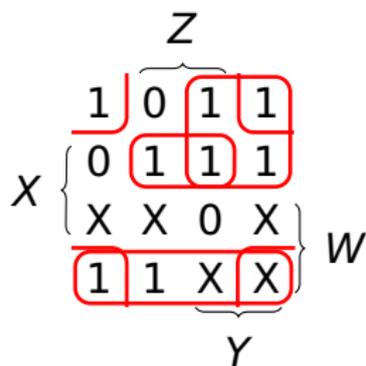
$$a = \overline{X}\overline{Y}\overline{Z} + \overline{W}Y + \overline{W}XZ + W\overline{X}\overline{Y}$$

3 + 2 + 3 + 3 = 11 literals

4 inv + 3 AND3 + 1 AND2 + 1 OR4

## Karnaugh Map for Seg. a

W	X	Y	Z	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	0



Taking don't-cares into account, we can enlarge two implicants:

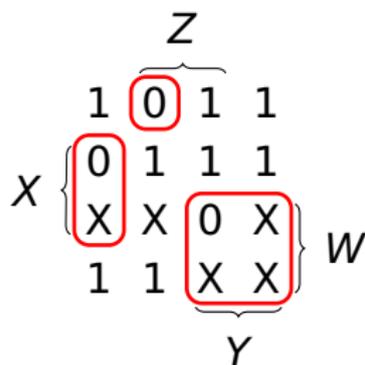
$$a = \overline{X}\overline{Z} + \overline{W}Y + \overline{W}XZ + W\overline{X}$$

2 + 2 + 3 + 2 = 9 literals

3 inv + 1 AND3 + 3 AND2 + 1 OR4

## Karnaugh Map for Seg. a

W	X	Y	Z	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	0



Can also compute the complement of the function and invert the result.

Covering the 0's instead of the 1's:

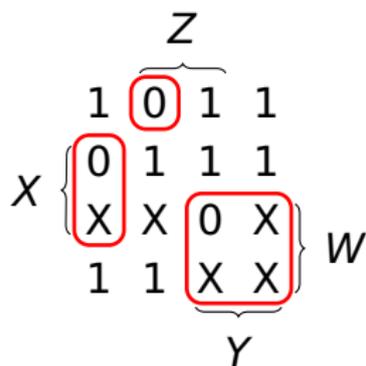
$$\bar{a} = \bar{W}\bar{X}\bar{Y}Z + X\bar{Y}\bar{Z} + WY$$

4 + 3 + 2 = 9 literals

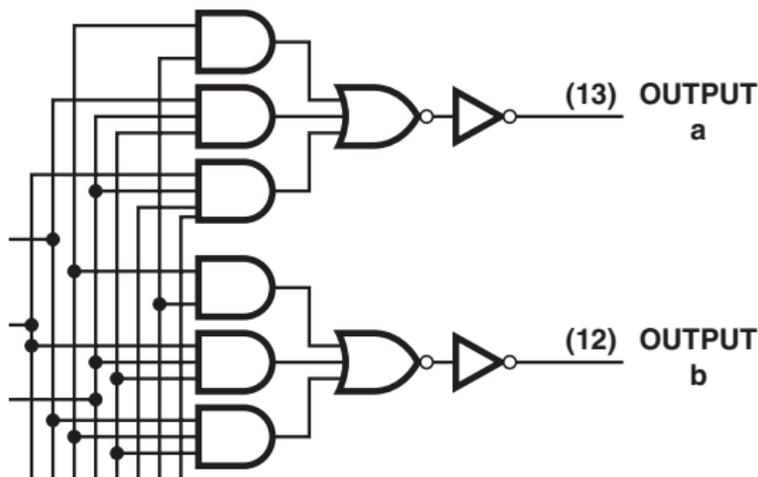
5 inv + 1 AND4 + 1 AND3 + 1 AND2  
+ 1 OR3

# Karnaugh Map for Seg. a

W	X	Y	Z	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	0



To display the score, PONG used a TTL chip with this solution in it:



# Boolean Laws and Karnaugh Maps

		W			
		0	0		
Y	{	0	0	1	1
		0	0	1	1
		0	0	1	1
		0	0	1	1
		X		Z	

$$\begin{aligned} &WX\bar{Y}\bar{Z} + \bar{W}X\bar{Y}\bar{Z} + \\ &WXY\bar{Z} + \bar{W}XY\bar{Z} + \\ &WXYZ + \bar{W}XYZ + \\ &WX\bar{Y}Z + \bar{W}X\bar{Y}Z \end{aligned}$$

Factor out the  $W$ 's

# Boolean Laws and Karnaugh Maps

		W			
Y	{	0	0	1	1
		0	0	1	1
		0	0	1	1
		0	0	1	1
				Z	
		X			

$$\begin{aligned} &(W + \bar{W})X\bar{Y}\bar{Z} + \\ &(W + \bar{W})XY\bar{Z} + \\ &(W + \bar{W})XYZ + \\ &(W + \bar{W})X\bar{Y}Z \end{aligned}$$

Use the identities

$$W + \bar{W} = 1$$

and

$$1X = X.$$

# Boolean Laws and Karnaugh Maps

	W					
	0	0	1	1		
Y	{	0	0	1	1	}
		0	0	1	1	
		0	0	1	1	
		0	0	1	1	
			X			

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

Factor out the Y's

## Boolean Laws and Karnaugh Maps

	W						
	0	0	1	1			
Y	{	0	0	1	1	}	Z
	0	0	1	1			
	0	0	1	1			
	0	0	1	1			
			X				

$$(\bar{Y} + Y)X\bar{Z} +$$
$$(\bar{Y} + Y)XZ$$

Apply the identities again

# Boolean Laws and Karnaugh Maps

	W				
	0	0	1	1	
Y	0	0	1	1	
	0	0	1	1	Z
	0	0	1	1	
	0	0	1	1	
	X				

$$X\bar{Z}+$$

$$XZ$$

Factor out Z

# Boolean Laws and Karnaugh Maps

	W					
	0	0	1	1		
Y	{	0	0	1	1	
	{	0	0	1	1	
	{	0	0	1	1	Z
	{	0	0	1	1	
			X			

$$X(\bar{Z} + Z)$$

Simplify

# Boolean Laws and Karnaugh Maps

	W					
	0	0	1	1		
Y	{	0	0	1	1	}
		0	0	1	1	
		0	0	1	1	
		0	0	1	1	
			X			

X

Done