Chapter #2: Two-Level Combinational Logic

Contemporary Logic Design

Randy H. Katz
University of California, Berkeley

May 1993

Logic Functions: Boolean Algebra

Algebraic structure consisting of:

set of elements B

binary operations {+, •}

unary operation {'}

such that the following axioms hold:

1. B contains at least two elements, a, b, such that $a \neq b$

2. *Closure* a,b in B,
  (i) $a + b$ in B
  (ii) $a \bullet b$ in B

3. *Commutative Laws*: a,b in B,
  (i) $a + b = b + a$
  (ii) $a \bullet b = b \bullet a$

4. *Identities*: 0, 1 in B
  (i) $a + 0 = a$
  (ii) $a \bullet 1 = a$

5. *Distributive Laws*:
  (i) $a + (b \bullet c) = (a + b) \bullet (a + c)$
  (ii) $a \bullet (b + c) = a \bullet b + a \bullet c$

6. *Complement*:
  (i) $a + a' = 1$
  (ii) $a \bullet a' = 0$

Logic Functions: Boolean Algebra

B **= {0,1}, + = OR, • = AND, ' = NOT is a Boolean Algebra**

must verify that the axioms hold:

E.g., Commutative Law:

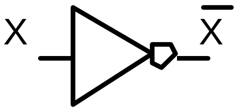
0 + 1 = 1 + 0?
1 = 1

0 • 1 = 1 • 0?
0 = 0

Theorem: any Boolean function that can be expressed as a truth table can be written as an expression in Boolean Algebra using ', +, •

Description
If X = 0 then X' = 1
If X = 1 then X ' = 0

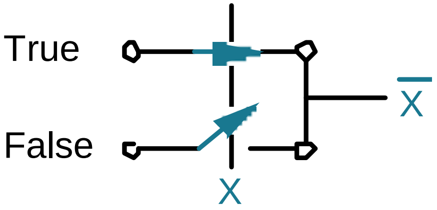
Gates



Truth Table

X	$\overline{X}$
0	1
1	0

Switches

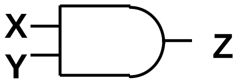


NOT

Review from Chapter 1

Description
Z = 1 if X and Y are both 1

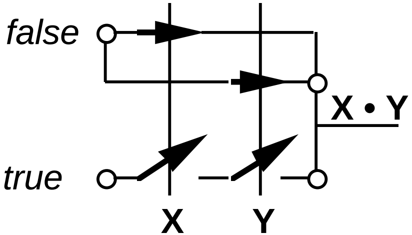
Gates



Truth Table

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

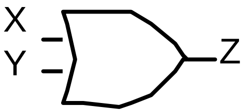
Switches



AND

Description
Z = 1 if X or Y (or both) are 1

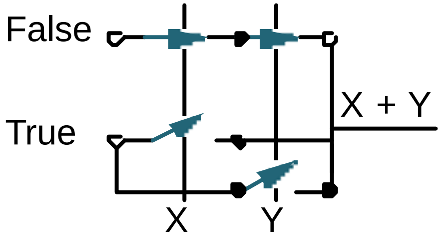
Gates



Truth Table

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

Switches

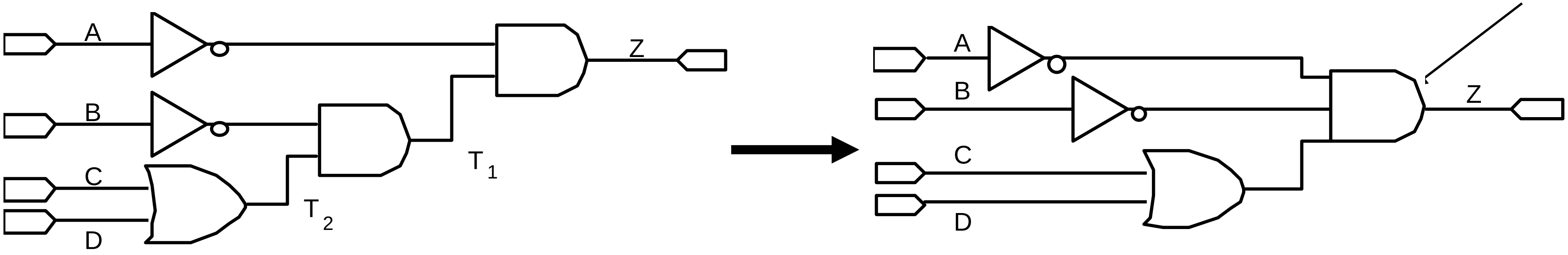


OR

Logic Functions: From Expressions to Gates

More than one way to map an expression to gates

E.g., $$Z = A' \cdot B' \cdot (C + D) = (A' \cdot \underbrace{(B' \cdot \overbrace{(C + D)}^{T2})}_{T1})$$



use of 3-input gate

***Literal*: each appearance of a variable or its complement in an expression**

E.g., $Z = A B' C + A' B + A' B C' + B' C$

3 variables, 10 literals

Logic Functions: NAND, NOR, XOR, XNOR

16 functions of two variables:

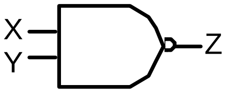
X	Y	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		0	X • Y		X		Y		X + Y			$\overline{Y}$		$\overline{X}$			1

X, X', Y, Y', X•Y, X+Y, 0, 1 only half of the possible functions

NAND

Description
Z = 1 if X is 0 or Y is 0

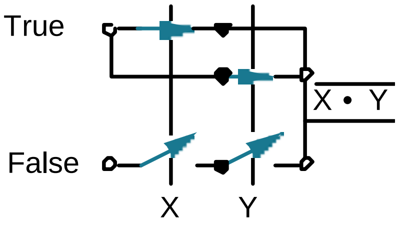
Gates



Truth Table

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

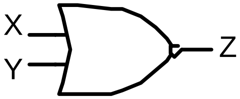
Switches



NOR

Description
Z = 1 if both X and Y are 0

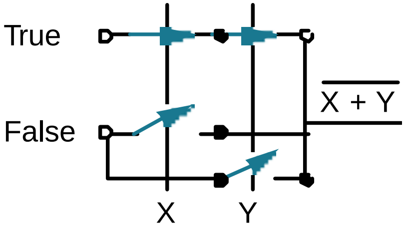
Gates



Truth Table

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

Switches



Logic Functions: NAND, NOR Implementation

NAND, NOR gates far outnumber AND, OR in typical designs

easier to construct in the underlying transistor technologies

Any Boolean expression can be implemented by NAND, NOR, NOT gates

In fact, NOT is superfluous
(NOT = NAND or NOR with both inputs tied together)

X	Y	X NOR Y
0	0	1
1	1	0

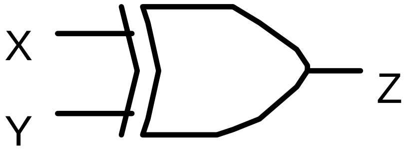
X	Y	X NAND Y
0	0	1
1	1	0

Logic Functions: XOR, XNOR

XOR: X or Y but not both ("inequality", "difference")
XNOR: X and Y are the same ("equality", "coincidence")

Description
Z = 1 if X has a different
value than Y

Gates



Truth Table

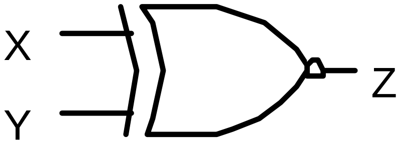
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

(a) XOR

$X \text{ ล } Y = X Y' + X' Y$

Description
Z = 1 if X has the same
value as Y

Gates



Truth Table

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

(b) XNOR

$\overline{X \text{ ล } Y} = X Y + X' Y'$

Gate Logic: Laws of Boolean Algebra

Duality: a dual of a Boolean expression is derived by replacing AND operations by ORs, OR operations by ANDs, constant 0s by 1s, and 1s by 0s (literals are left unchanged).

Any statement that is true for an expression is also true for its dual!

Useful Laws/Theorems of Boolean Algebra:

Operations with 0 and 1:

1. $X + 0 = X$ 1D. $X \bullet 1 = X$

2. $X + 1 = 1$ 2D. $X \bullet 0 = 0$

Idempotent Law:

3. $X + X = X$ 3D. $X \bullet X = X$

Involution Law:

4. $(X')' = X$

Laws of Complementarity:

5. $X + X' = 1$ 5D. $X \bullet X' = 0$

Commutative Law:

6. $X + Y = Y + X$ 6D. $X \bullet Y = Y \bullet X$

Gate Logic: Laws of Boolean Algebra (cont)

Associative Laws:

7. (X + Y) + Z = X + (Y + Z)
= X + Y + Z

7D. (X • Y) • Z = X • (Y • Z)
= X • Y • Z

Distributive Laws:

8. X • (Y+ Z) = (X • Y) + (X • Z)

8D. X + (Y• Z) = (X + Y) • (X + Z)

Simplification Theorems:

9. X • Y + X • Y' = X

10. X + X • Y = X

11. (X + Y') • Y = X • Y

9D. (X + Y) • (X + Y') = X

10D. X • (X + Y) = X

11D. (X • Y') + Y = X + Y

DeMorgan's Law:

12. (X + Y + Z + ...)' = X' • Y' • Z' • ...

12D. (X • Y • Z • ...) ' = X' + Y' + Z' + ...

13. {F(X1,X2,...,Xn,0,1,+,•)}' = {F(X1',X2',...,Xn',1,0,•,+)}

Duality:

14. $(X + Y + Z + ...)^D$ = X • Y • Z • ...

14D. $(X \bullet FY \bullet Z \bullet ...)^D$ = X + Y + Z + ...

15. $\{F(X1,X2,...,Xn,0,1,+,\bullet)\}^D$ = {F(X1,X2,...,Xn,1,0,•,+)}

Theorems for Multiplying and Factoring:

16. (X + Y) • (X' + Z) = X • Z + X' • Y

16D. X • Y + X' • Z = (X + Z) • (X' + Y)

Consensus Theorem:

17. (X • Y) + (Y • Z) + (X' • Z) =
X • Y + X' • Z

17D. (X + Y) • (Y + Z) • (X' + Z) =
(X + Y) • (X' + Z)

Gate Logic: Laws of Boolean Algebra

Proving theorems via axioms of Boolean Algebra:

E.g., prove the theorem: $X \bullet Y + X \bullet Y' = X$

E.g., prove the theorem: $X + X \bullet Y = X$

Gate Logic: Laws of Boolean Algebra

Proving theorems via axioms of Boolean Algebra:

E.g., prove the theorem: $X \bullet Y + X \bullet Y' = X$

distributive law (8) $X \bullet Y + X \bullet Y' = X \bullet (Y + Y')$

complementary law (5) $X \bullet (Y + Y') = X \bullet (1)$

identity (1D) $X \bullet (1) = X$

E.g., prove the theorem: $X + X \bullet Y = X$

identity (1D) $X + X \bullet Y = X \bullet 1 + X \bullet Y$

distributive law (8) $X \bullet 1 + X \bullet Y = X \bullet (1 + Y)$

identity (2) $X \bullet (1 + Y) = X \bullet (1)$

identity (1) $X \bullet (1) = X$

Gate Logic: Laws of Boolean Algebra

DeMorgan's Law

(X + Y)' = X' • Y'

NOR is equivalent to AND with inputs complemented

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

(X • Y)' = X' + Y'

NAND is equivalent to OR with inputs complemented

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 = A' B' C + A' B C + A B' C + A B C'

Z' = (A + B + C') • (A + B' + C') • (A' + B + C') • (A' + B' + C)

Gate Logic: Laws of Boolean Algebra

Apply the laws and theorems to simplify Boolean equations

Example: full adder's carry out function

Cout = A' B Cin + A B' Cin + A B Cin' + A B Cin

Gate Logic: Laws of Boolean Algebra

Apply the laws and theorems to simplify Boolean equations

Example: full adder's carry out function

identity

Cout = A' B Cin + A B' Cin + A B Cin' + A B Cin

= A' B Cin + A B' Cin + A B Cin' + $\boxed{\text{A B Cin + A B Cin}}$

= A' B Cin + A B Cin + A B' Cin + A B Cin' + A B Cin

= (A' + A) B Cin + A B' Cin + A B Cin' + A B Cin

= (1) B Cin + A B' Cin + A B Cin' + A B Cin

= B Cin + A B' Cin + A B Cin' + $\boxed{\text{A B Cin + A B Cin}}$

= B Cin + A B' Cin + A B Cin + A B Cin' + A B Cin

associative

= B Cin + A (B' + B) Cin + A B Cin' + A B Cin

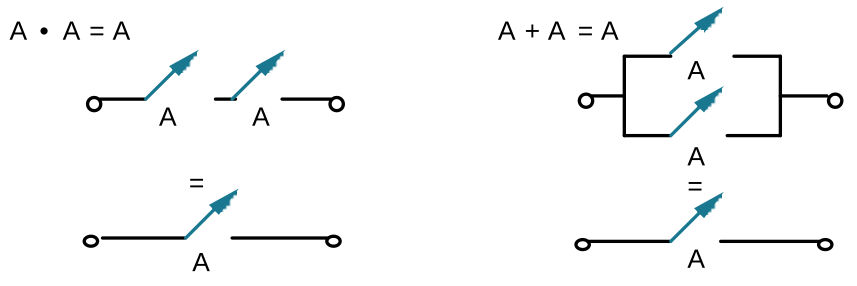
= B Cin + A (1) Cin + A B Cin' + A B Cin

= B Cin + A Cin + A B (Cin' + Cin)

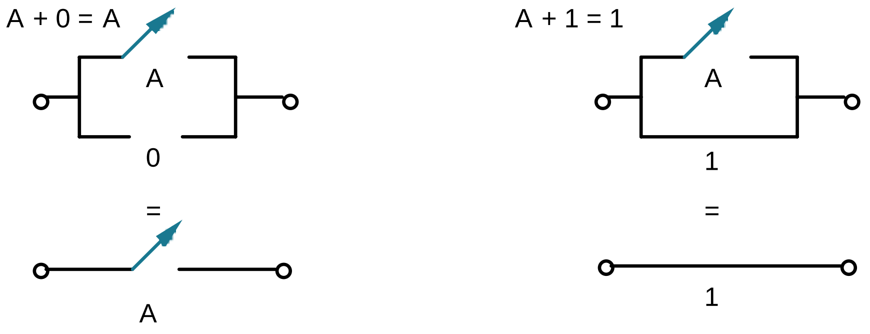
= B Cin + A Cin + A B (1)

= B Cin + A Cin + A B

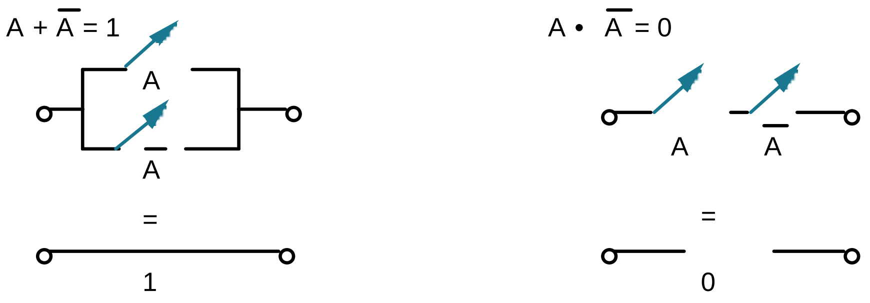
Gate Logic: Switching Equivalents



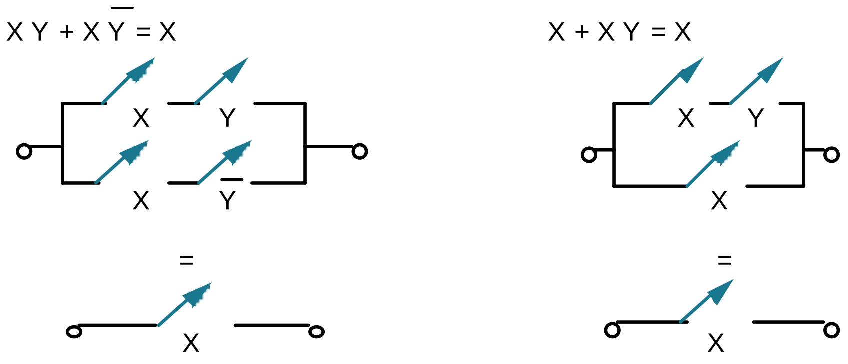
Idempotent Laws



Identity Laws



Complementarity Laws



Simplification Theorems

Gate Logic: 2-Level Canonical Forms

Truth table is the unique signature of a Boolean function

Many alternative expressions (and gate realizations) may have the same truth table

Canonical form: standard form for a Boolean expression provides a unique algebraic signature

Sum of Products Form

also known as disjunctive normal form, minterm expansion

A	B	C	F	$\overline{F}$
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

$$F = \overset{011}{A'BC} + \overset{100}{AB'C'} + \overset{101}{AB'C} + \overset{110}{ABC'} + \overset{111}{ABC}$$

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