

Boolean Algebra & Logic Design

Ref: Appendix B

1

Boolean Algebra

- Developed by George Boole in the 1850s
- Mathematical theory of logic.

- Shannon was the first to use Boolean Algebra to solve problems in electronic circuit design. (1938)

2

Variables & Operations

- All variables have the values 1 or 0
 - sometimes we call the values TRUE / FALSE
- Three operators:
 - OR written as +, as in $A + B$
 - AND written as \cdot , as in $A \cdot B$
 - NOT written as an overline, as in \overline{A}

3

Operators: OR

- The result of the OR operator is 1 if either of the operands is a 1.
- The only time the result of an OR is 0 is when both operands are 0s.
- OR is like our old pal *addition*, but operates only on binary values.

4

Operators: AND

- The result of an AND is a 1 only when both operands are 1s.
- If either operand is a 0, the result is 0.
- AND is like our old nemesis *multiplication*, but operates on binary values.

5

Operators: NOT

- NOT is a *unary* operator – it operates on only one operand.
- NOT *negates* its operand.
- If the operand is a 1, the result of the NOT is a 0.
- If the operand is a 0, the result of the NOT is a 17.678.
just kidding – it's a 1 (*wake up*)!

6

Equations

Boolean algebra uses equations to express relationships. For example:

$$X = A \cdot (\bar{B} + C)$$

This equation expresses a relationship between the value of X and the values of A , B and C .

7

Quiz (already?)

What is the value of each X:

$$X_1 = 1 \cdot (0 + 1)$$

$$X_2 = A + \bar{A}$$

$$X_3 = A \cdot \bar{A}$$

$$X_4 = X_4 + 1 \quad \leftarrow \text{huh?}$$

8

Laws of Boolean Algebra

Just like in *good old algebra*, Boolean Algebra has postulates and identities.

We can often use these laws to reduce expressions or put expressions in to a more desirable form.

9

Basic Postulates of Boolean Algebra

- Using just the basic postulates – everything else can be derived.

Commutative laws

Distributive laws

Identity

Inverse

10

Identity Laws

$$A + 0 = A$$

$$A \cdot 1 = A$$

11

Inverse Laws

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

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

12

Commutative Laws

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

13

Distributive Laws

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$

14

Other Identities

Can be derived from the basic postulates.

Laws of Ones and Zeros

Associative Laws

DeMorgan's Theorems

15

Zero and One Laws

$$A + 1 = 1 \quad \text{Law of Ones}$$

$$A \cdot 0 = 0 \quad \text{Law of Zeros}$$

16

Associative Laws

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

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

17

DeMorgan's Theorems

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

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

18

Other Operators

- Boolean Algebra is defined over the 3 operators AND, OR and NOT.
 - this is a *functionally complete set*.
- There are other useful operators:
 - NOR: is a 0 if either operand is a 1
 - NAND: is a 0 only if both operands are 1
 - XOR: is a 1 if the operands are different.
- NOTE: NOR is (by itself) a functionally complete set!

19

Boolean Functions

- Boolean functions are functions that operate on a number of Boolean variables.
- The result of a Boolean function is itself either a 0 or a 1.
- Example: $f(a,b) = a+b$

20

Question

- How many Boolean functions of 1 variable are there?
- We can answer this by listing them all!

$$f_1(x) = x$$

$$f_2(x) = \bar{x}$$

$$f_3(x) = 0$$

$$f_4(x) = 1$$

21

Tougher Question

- How many Boolean functions of 2 variables are there?
- It's much harder to list them all, but it is still possible...

22

Alternative Representation

- We can define a Boolean function by describing it with algebraic operations.
- We can also define a Boolean function by listing the value of the function for all possible inputs.

23

OR as a Boolean Function

$$f_{or}(a,b) = a + b$$

This is called a "truth table"

<i>a</i>	<i>b</i>	$f_{or}(a,b)$
0	0	0
0	1	1
1	0	1
1	1	1

24

Truth Tables

<i>a</i>	<i>b</i>	OR	AND	NOR	NAND	XOR
0	0	0	0	1	1	0
0	1	1	0	0	1	1
1	0	1	0	0	1	1
1	1	1	1	0	0	0

25

Truth Table for $(X+Y) \cdot Z$

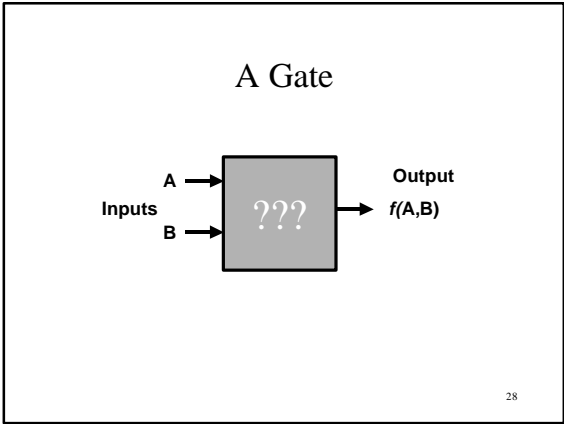
X	Y	Z	$(X+Y) \cdot Z$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

26

Gates

- Digital logic circuits are electronic circuits that are implementations of some Boolean function(s).
- A circuit is built up of *gates*, each *gate* implements some simple logic function.
- The term *gates* is named for Bill Gates, in much the same way as the term *gore* is named for Al Gore – the inventor of the Internet.

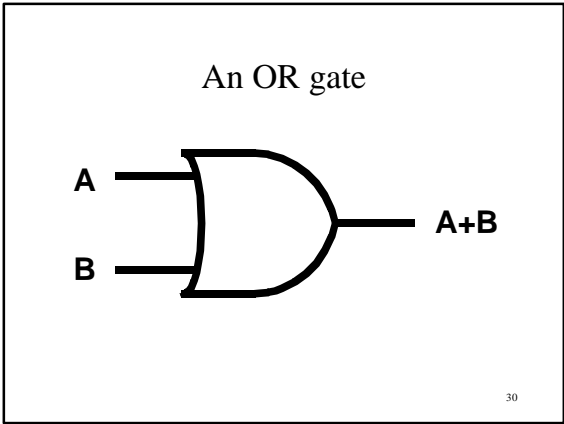
27



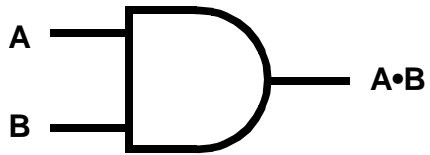
Gates compute something!

- The output depends on the inputs.
- If the input changes, the output might change.
- If the inputs don't change – the output does not change.

29

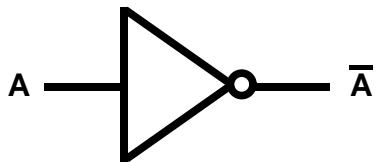


An AND gate



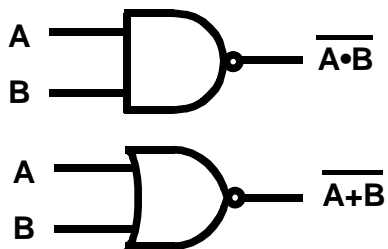
31

A NOT gate



32

NAND and NOR gates



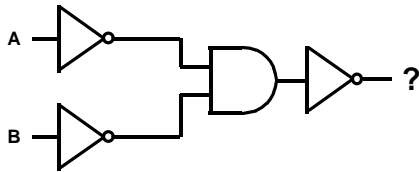
33

Combinational Circuits

- We can put gates together into circuits
 - output from some gates are inputs to others.
- We can design a circuit that represents any Boolean function!

34

A Simple Circuit



35

Truth Table for our circuit

a	b	\bar{a}	\bar{b}	$\bar{a} \cdot \bar{b}$	$\overline{\bar{a} \cdot \bar{b}}$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

36

Alternative Representations

- Any of these can express a Boolean function. :

Boolean Equation
Circuit (Logic Diagram)
Truth Table

37

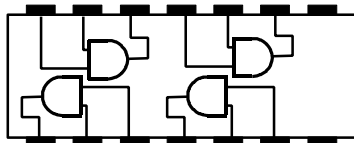
Implementation

- A logic diagram is used to design an *implementation* of a function.
- The implementation is the specific gates and the way they are connected.
- We can buy a bunch of gates, put them together (along with a power source) and build a machine.

38

Integrated Circuits

- You can buy an AND gate *chip*:



39

Function Implementation

- Given a Boolean function expressed as a truth table or Boolean Equation, there are many possible implementations.
- The actual implementation depends on what kind of gates are available.
- In general we want to minimize the number of gates.

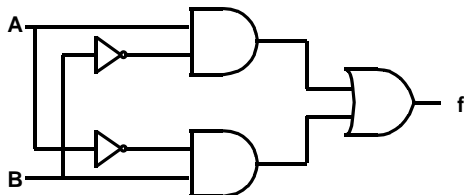
40

Example: $f = A \bullet \bar{B} + \bar{A} \bullet B$

A	B	$A \bullet \bar{B}$	$\bar{A} \bullet B$	f
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

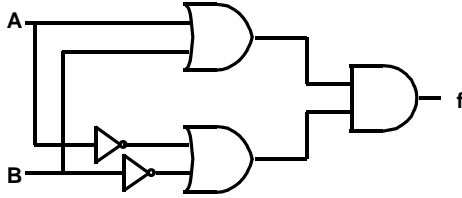
41

One Implementation $f = A \bullet \bar{B} + \bar{A} \bullet B$



42

Another Implementation



$$f = A \cdot \bar{B} + \bar{A} \cdot B = (A + B) \cdot (\bar{A} + \bar{B})$$

43

Proof it's the same function

$$\begin{aligned}
 &A \cdot \bar{B} + \bar{A} \cdot B = \\
 &\overline{(\overline{A \cdot \bar{B}}) \cdot (\overline{\bar{A} \cdot B})} = && \text{DeMorgan's Law} \\
 &\overline{(A + B) \cdot (A + \bar{B})} = && \text{DeMorgan's Laws} \\
 &\overline{((A + B) \cdot A) + ((A + B) \cdot \bar{B})} = && \text{Distributive} \\
 &\overline{(\bar{A} \cdot A + B \cdot A) + (\bar{A} \cdot \bar{B} + B \cdot \bar{B})} = && \text{Distributive} \\
 &\overline{(B \cdot A) + (\bar{A} \cdot \bar{B})} = && \text{Inverse, Identity} \\
 &\overline{(B \cdot A) \cdot (\bar{A} \cdot \bar{B})} = && \text{DeMorgan's Law} \\
 &\overline{(\bar{B} + \bar{A}) \cdot (A + B)} = && \text{DeMorgan's Laws} \quad 44
 \end{aligned}$$

Possible Test Questions

- Prove that NOR is a *functionally complete set*.
 - Show how you can express the functions AND, OR and NOT in terms of NOR.
- Prove that two expressions are (or are not) really the same boolean function.
 - Use identities/postulates to transform one expression in to another
 - Compare truth tables.
- Know DeMorgan's Theorems (prove them!).

45
