DIGITAL LOGIC

Chapter 2:
Introduction to logic circuits

Ru Han

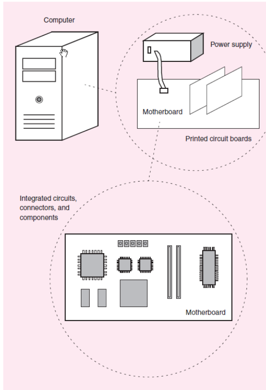


Figure 1.4 A digital hardware system (Part a).

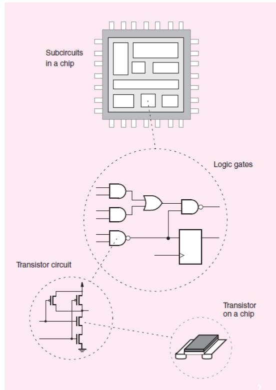


Figure 1.4 A digital hardware system (Part b).

OUTLINE

Chapter 2:

Introduction to logic circuits

- Logic function and circuits
- Boolean algebra for dealing with logic function
- Logic gates and synthesis of simple circuits
- CAD tools and the Verilog hardware description language
- Minimization of functions and Karnaugh maps

2.1 VARIABLES AND FUNCTIONS

<u>Switch</u>: Basic element of every digital (computer) system.

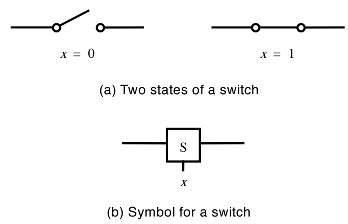
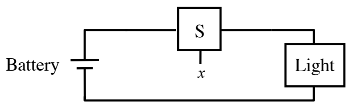


Figure 2.1. A binary switch.

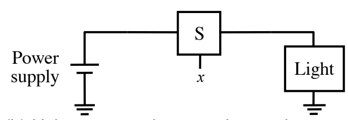
<u>Switch application</u>



(a) Simple connection to a battery

Light on: L=1
Light off: L=0

x=1 L=1
x=0 L=0

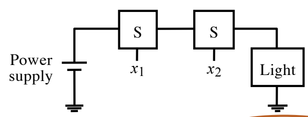


(b) Using a ground connection as the return path

<u>A function of the input variable x:</u>
$L(x)=x$

Figure 2.2. A light controlled by a switch.

<u>Logic Operation: AND</u>



(a) The logical AND function (series connection)

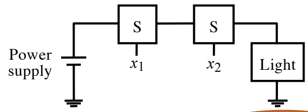
Figure 2.3. Two basic functions.

Logic Abstraction:

- Switch x1, x2:
  0 = Open, 1 = Closed
- Bulb L:
  0 = Off, 1 = On

x1	x2	L
0	0	
0	1	
1	0	
1	1	

Logic Operation: AND



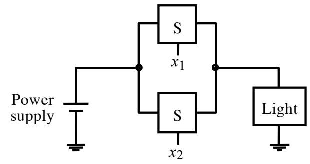
(a) The logical AND function (series connection)

Figure 2.3. Two basic functions.

AND: L(x1,x2)=x1·x2

Design a 2-switch light circuit, such that the light is on only when both switches are closed.

Logic Operation: OR



(b) The logical OR function (parallel connection)

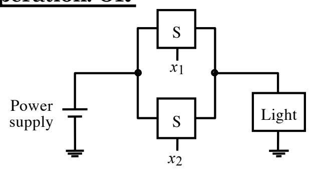
Figure 2.3. Two basic functions.

Logic Abstraction:

- Switch x1, x2:
  0 = Open, 1 = Closed
- Bulb L:
  0 = Off, 1 = On

x1	x2	L
0	0	
0	1	
1	0	
1	1	

Logic Operation: OR



(b) The logical OR function (parallel connection)

Figure 2.3. Two basic functions.

OR: L(x1,x2)=x1+x2

Design a 2-switch light circuit, such that the light is on if either or both switches are closed.

Example: The referee's circuit.

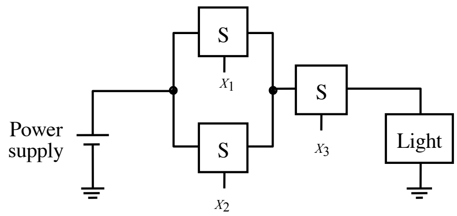


Figure 2.4. A series-parallel connection.

2.2 Inversion

Logic Operation: NOT

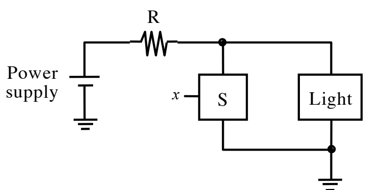


Figure 2.5. An inverting circuit.

Logic Abstraction:

- Switch x:
  0 = Open, 1 = Closed
- Bulb L:
  0 = Off, 1 = On

x	L
0	
1	

Inversion

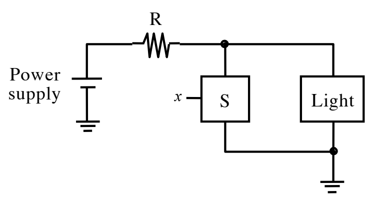


Figure 2.5. An inverting circuit.

NOT: L(x)=x'=!x=~x

Design a 1-switch light circuit, such that the light is on only when the switch is open.

2.3 Truth Tables

x_1	x_2	$x_1 \cdot x_2$	$x_1 + x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1
		AND	OR

Figure 2.6. A truth table for the AND and OR operations.

x_1	x_2	x_3	$x_1 \cdot x_2 \cdot x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 2.7. Three-input AND and OR operations.

The AND and OR operations can be extended to n variables.

An AND function of variables $x_1, x_2, \ldots, x_n$ has the value 1 only if all n variables are equal to 1.

An OR function of variables $x_1, x_2, \ldots, x_n$ has the value 1 if one or more of the variables is equal to 1.

2.4 Logic Gates and Networks

Logic Gates

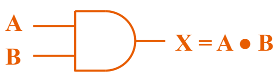
- Switch Model ↔ Logic Gates ↔ Physical Devices
- Symbolic Representations
- Functional Descriptions
- Logic Networks
- Logic Design
- Venn Diagrams

Physical Devices

- Constructed using transistors
- Take one or more electrical input signals, and produce one output signal
- High Voltage (4~5V) = LOGIC 1
  Low Voltage (0~1V) = LOGIC 0
  (Sometime this conversion is reversed)

Symbolic Representations

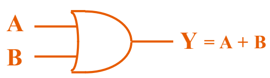
(1) AND Gate



(a) { A, B, X } ∈ { 0, 1 }
(b) may have > 2 inputs
(c) **X = 1 if and only if <u>ALL</u> inputs = 1**

Symbolic Representations

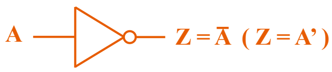
(2) OR Gate



(a) { A, B, X } ∈ { 0, 1 }
(b) may have > 2 inputs
(c) **Y = 0 if and only if <u>ALL</u> inputs = 0**

Symbolic Representations

(3) NOT Gate (Inverter)



(a) { A, Z } ∈ { 0, 1 }

(b) 1 input, 1 output

(c) **Z = 1 if and only if A = 0**

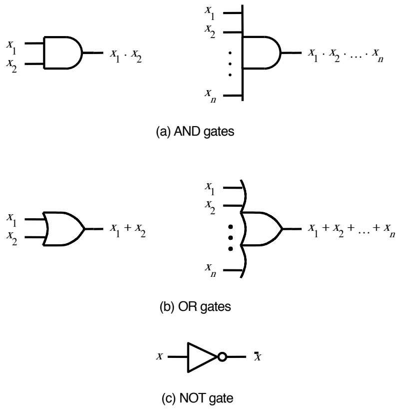


Figure 2.8. The basic gates.

Symbolic Representations

- Example: 3-input network

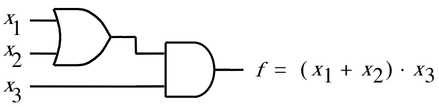


Figure 2.9. The function from Figure 2.4.

Functional Descriptions

Design problem:

- *Analysis* process:

 For an existing logic network, it must be possible to determine the function performed by the network.

- *Synthesis* process:

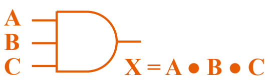
 The reverse task of designing a new network that implements a desired functional behavior.

Functional Descriptions

(1) Logic Function

(2) Truth Table

- Table that lists all possible combinations of inputs, and the corresponding output.
- Example, 3-input AND gate.



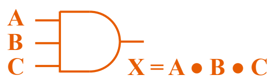
A	B	C	X

Functional Descriptions

(1) Logic Function

(2) Truth Table

- Table that lists all possible combinations of inputs, and the corresponding output.
- Example, 3-input AND gate.



A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Functional Descriptions

(3) Timing Diagram

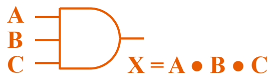
- Draw waveforms to show inputs and outputs as functions of time.
- Create waveforms so that all possible input combinations are shown, then determine output waveform.
- Gives all information in truth table PLUS implementation details such as propagation delay.

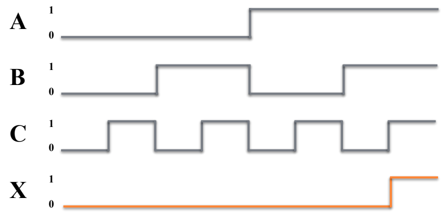
25

Functional Descriptions

(3) Timing Diagram

- Example, 3-input AND gate





26

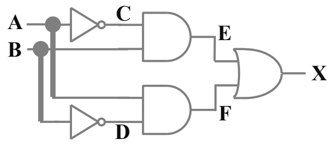
Logic Network

- Any logic function with any number of input signals can be realized as a network of AND, OR, and NOT gates.
- We can use truth tables and timing diagrams to analyze network.
  - Provide columns in truth table, or waveforms in timing diagram for all inputs, outputs and intermediate signals.
  - Note: **We Will Use Logic Network Or Logic Circuit Interchangeably**

27

Logic Network

- Example: 2-input network



(a) Algebraic Evaluation

$C = \overline{A}$

$D = \overline{B}$

$E = C \bullet B = \overline{A} \bullet B$

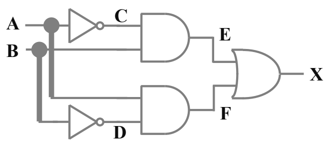
$F = A \bullet D = A \bullet \overline{B}$

$X = E + F = (\overline{A} \bullet B) + (A \bullet \overline{B})$

28

Logic Network

- Example: 2-input network



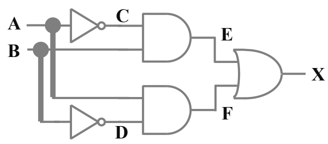
(B) Truth Table

A	B	C	D	E	F	X
0	0					
0	1					
1	0					
1	1					

29

Logic Network

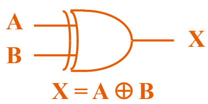
- Example: 2-input network



(B) Truth Table

A	B	C	D	E	F	X
0	0					0
0	1					1
1	0					1
1	1					0

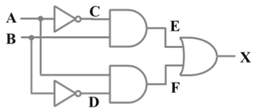
Note: This type of network is known as Exclusive-OR (XOR)

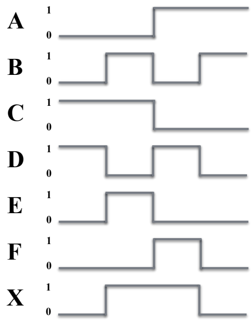


30

Logic Network

(C) Timing Diagram



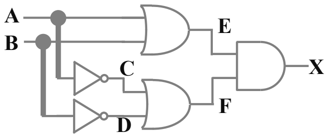


Functional Equivalent Network

- For any logic function, there are ***many different networks*** that could be used to implement the function.
- If two networks have the same output column in their truth tables, then they are ***functional equivalent***.

Functional Equivalent Network

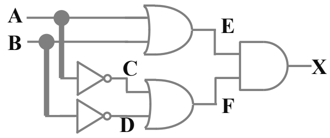
- Example: 2-input network



(a) Algebraic Evaluation

Functional Equivalent Network

- Example: 2-input network



(a) Algebraic Evaluation

$C = \overline{A}$

$D = \overline{B}$

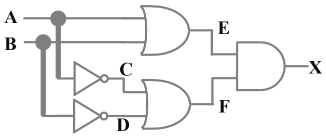
$E = A + B$

$F = C + D = \overline{A} + \overline{B}$

$X = E \bullet F = (A + B) \bullet (\overline{A} + \overline{B})$

Functional Equivalent Network

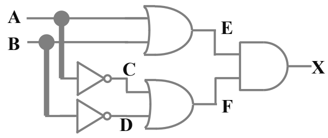
- Example: 2-input network



(B) Truth Table

Functional Equivalent Network

- Example: 2-input network

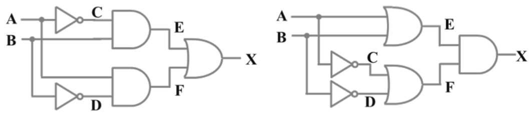


(B) Truth Table

A	B	C	D	E	F	X
0	0	1	1	0	1	0
0	1	1	0	1	1	1
1	0	0	1	1	1	1
1	1	0	0	1	0	0

It is also an XOR network

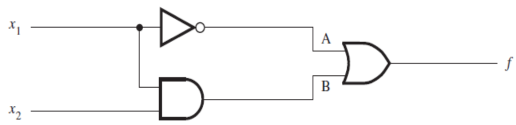
Functional Equivalent Network



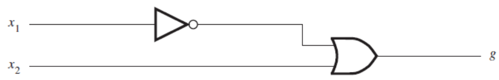
- The two logic networks are used to implement the same function – XOR gate.
- They are functional equivalent networks.

37

Functional Equivalent Network



(a) Network that implements



(d) Network that implements

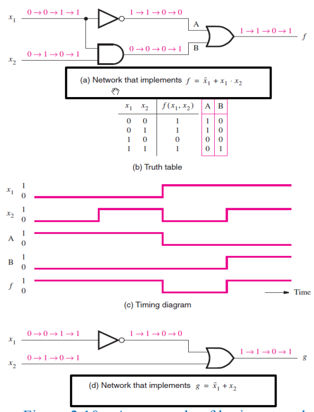


Figure 2.10. An example of logic networks.

39

SO:

- More than one logic functions based on One Truth Table
- They are functional equivalent networks
- Is there one network which is the best for cost among them ?

- **Boolean Algebra Will Give You The Answer**

40

Conclusion

- Logic variable and functions
- Understand the Truth Table
- Truth table/timing diagram/ Logic function
- XOR

- Text Reading: Chapter: 2.1 ~ 2.4, page22~33
- Assignment: Page 111: 2.8

41

2.5 Boolean Algebra

Boolean Algebra

- Huntington's Postulates
- Proofs
- Basic Theorems
- Operator Precedence

42

Boolean Algebra

In 1845, George Boole introduced a systematic treatment of logic now called **Boolean algebra**. Logical statements are built up from:

- **Variables**: a, x, etc. Variables can be used to represent propositions (statements that are either true or false) or signals in digital circuits (voltages that are either high or low, representing 0 or 1).
- **Operators**: The two operators (+, •) are used to combine variables to produce more complex statements or logic functions. These operators must satisfy certain properties.

The most well-known form of Boolean algebra is a two-valued system, in which all variables take values on the set {0, 1} and the operators (+,•) correspond to (OR, AND) respectively.

43

Huntington's Postulates: In 1904, E. V. Huntington defined Boolean algebra by providing 6 postulates that must be satisfied.

- **Closure** with respect to the operators:
  any logical operation yields a value in the set {0, 1}.
- **Identity** elements with respect to the operators:
  $x + 0 = x$ $\quad$ $x \bullet 1 = x$
- **Commutativity** with respect to the operators:
  $x + y = y + x$ $\quad$ $x \bullet y = y \bullet x$
- **Distributivity**:
  $x \bullet (y + z) = (x \bullet y) + (x \bullet z)$ $\quad$ $x + (y \bullet z) = (x + y) \bullet (x + z)$
- **Complements** exist for all the elements:
  $x + x' = 1$ $\quad$ $x \bullet x' = 0$
- **Distinct elements**:
  $0 \neq 1$

44

- In 1938, Claude Shannon showed that a two-valued Boolean Algebra, called **switching algebra**, could be used to describe *digital circuits.*
  - Two elements: {0,1}
  - Two binary operators: + (OR), • (AND)
  - One unary operator: the complement ' ($\bar{\ }$)

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

x	y	$x \bullet y$
0	0	0
0	1	0
1	0	0
1	1	1

x	x'
0	1
1	0

45

Is this really a Boolean algebra? Does it satisfy Huntington's Postulates?

- **Closure** with respect to +, •:
  Yes, both operations produce 0 or 1 for all input combinations.
- **Identity** elements with respect to +, •:
  Yes, from truth tables, $x + 0 = x$, $x \bullet 1 = x$.
- **Commutativity** with respect to +, •:
  Yes, if x and y columns are swapped, results are unchanged.

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

x	y	$x \bullet y$
0	0	0
0	1	0
1	0	0
1	1	1

x	x'
0	1
1	0

46

Is this really a Boolean algebra? Does it satisfy Huntington's Postulates?

- **Distributivity**:
  Yes (proof to follow).
- **Complements** exist for all the elements:
  Yes, 0' = 1, 1' = 0.
- **Distinct Elements**:
  Yes, by definition: $0 \neq 1$.

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

x	y	$x \bullet y$
0	0	0
0	1	0
1	0	0
1	1	1

x	x'
0	1
1	0

47

Is this really a Boolean algebra? Does it satisfy Huntington's Postulates?

- **Closure** with respect to +, •:
  Yes, both operations produce 0 or 1 for all input combinations.
- **Identity** elements with respect to +, •:
  Yes, from truth tables, $x + 0 = x$, $x \bullet 1 = x$.
- **Commutativity** with respect to +, •:
  Yes, if x and y columns are swapped, results are unchanged.
- **Distributivity**:
  Yes (proof to follow).
- **Complements** exist for all the elements:
  Yes, 0' = 1, 1' = 0.
- **Distinct Elements**:
  Yes, by definition: $0 \neq 1$.

Note: In this book, it is called Axioms On pages:33

48

Proofs: What are they?

Proofs are a tool for establishing new results or **theorems** in Boolean algebra. The kinds of things we need to prove usually fall into one of several categories.

- Prove that two logical expressions are equivalent.
- Prove that two logic networks are functionally equivalent.
- Prove that one logical expression is the complement of another.

49

OK, so prove it! Hmmm, how do we do that?

Here are four ways to prove that two expressions are equivalent.

- **Perfect Induction**: show that the two expressions have identical truth tables.

If there are n input variables, this requires 2^n input combinations. This is a bad idea if there are more than 3 or 4 variables.

- **Axiomatic Proof**: apply Huntington's postulates (or other theorems that have already been proven) to the expressions, until the two expressions are identical.
- **Duality Principle**: every theorem in Boolean algebra remains valid if we interchange all AND's and OR's, and interchange all 0's and 1's.
- **Proof by contradiction**: assume that the hypothesis is ***false*** and then prove that the resulting equation can never be true.

50

Duality

- Given a logic expression, its dual can be obtained by
1. Replace all + with . and vice versa
2. Replace all 0 with 1 and vice versa

Pages: 34

- Understand the importance of the duality

How to Understand Duality Principle

The duality principle ensures that "if we exchange ***every*** symbol by its dual in a formula, we get the ***dual*** result".

Examples:

- **0 • 1 = 0** is a true statement asserting that "false and true evaluates to false"

  1 + 0 = 1 is its dual: it is a true statement asserting that "true or false evaluates true."
- **1 • 1 = 1** is a true statement asserting that "true and true evaluates to true".

  0 + 0 = 0 is its dual : it is a true statement asserting, correctly, that "false or false evaluates to false".

52

Proof of Distributivity of • over +, by Perfect Induction.

$x \bullet (y + z) = (x \bullet y) + (x \bullet z)$

x	y	z	$y + z$	$x \bullet (y + z)$	$x \bullet y$	$x \bullet z$	$(x \bullet y) + (x \bullet z)$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

53

Proof of Distributivity of • over +, by Perfect Induction.

$x \bullet (y + z) = (x \bullet y) + (x \bullet z)$

x	y	z	$y + z$	$x \bullet (y + z)$	$x \bullet y$	$x \bullet z$	$(x \bullet y) + (x \bullet z)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

54

Proof of Distributivity of + over •, by Perfect Induction.

$x + (y \bullet z) = (x + y) \bullet (x + z)$

x	y	z	$y \bullet z$	$x + (y \bullet z)$	$x + y$	$x + z$	$(x + y) \bullet (x + z)$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Proof of Distributivity of + over •, by Perfect Induction.

$x + (y \bullet z) = (x + y) \bullet (x + z)$

x	y	z	$y \bullet z$	$x + (y \bullet z)$	$x + y$	$x + z$	$(x + y) \bullet (x + z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

OK, where are we so far?

- So far, we have only shown (through 1 rigorous proof and several less formal arguments) that Huntington's postulates are satisfied. Therefore, the two valued system with the operations (•, +) as defined above is a Boolean algebra.

Why do we care?

- Armed with these 6 postulates, we can now go on to establish other theorems that will help us analyze logic circuits.

Basic Theorems

These theorems can be verified by examining the truth tables for {+, •, '}.

- ***Theorem 1: Idempotency***
  - a) $x + x = x$
  - b) $x \bullet x = x$

x	$x + x$	$x \bullet x$
0	0	0
1	1	1

- ***Theorem 2: Tautology & Contradiction***
  - a) $x + 1 = 1$
  - b) $x \bullet 0 = 0$

x	$x + 1$	$x \bullet 0$
0	1	0
1	1	0

- ***Theorem 3: Involution***
  - $(x')' = x$

x	x'	$(x')'$
0	1	0
1	0	1

Basic Theorems

- ***Theorem 4: Associativity***
  - a) $(x + y) + z = x + (y + z)$

x	y	z	$(x + y)$	$(y + z)$	$(x + y) + z$	$x + (y + z)$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Basic Theorems

- ***Theorem 4: Associativity***
  - a) $(x + y) + z = x + (y + z)$

x	y	z	$(x + y)$	$(y + z)$	$(x + y) + z$	$x + (y + z)$
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Basic Theorems

- ***Theorem 4:*** ***Associativity***
  a) $(x + y) + z = x + (y + z)$
  b) $(x \bullet y) \bullet z = x \bullet (y \bullet z)$

x	y	z	$(x \bullet y)$	$(y \bullet z)$	$(x \bullet y) \bullet z$	$x \bullet (y \bullet z)$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

61

Basic Theorems

- ***Theorem 4:*** ***Associativity***
  a) $(x + y) + z = x + (y + z)$
  b) $(x \bullet y) \bullet z = x \bullet (y \bullet z)$

x	y	z	$(x \bullet y)$	$(y \bullet z)$	$(x \bullet y) \bullet z$	$x \bullet (y \bullet z)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

62

Theorem 5: DeMorgan's Law

(a) $(x + y)' = x' \bullet y'$

Proof: by Perfect Induction.

x	y	$(x + y)$	$(x + y)'$	x'	y'	$x' \bullet y'$
0	0					
0	1					
1	0					
1	1					

63

Theorem 5: DeMorgan's Law

(a) $(x + y)' = x' \bullet y'$

Proof: by Perfect Induction.

x	y	$(x + y)$	$(x + y)'$	x'	y'	$x' \bullet y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

(b) $(x \bullet y)' = x' + y'$

Proof: apply the duality principle to the expression in part (a). If we interchange all • and + operators, we obtain the expression in part (b).

64

Theorem 6: Absorption

(a) $x + (x \bullet y) = x$

Axiomatic Proof:

$x + (x \bullet y) = (x \bullet 1) + (x \bullet y)$	by $x \bullet 1 = x$
$= x \bullet (1 + y)$	by distributivity
$= x \bullet (y + 1)$	by commutativity
$= x \bullet 1$	by $x + 1 = \mathbf{1}$
$= x$	by $x \bullet 1 = x$

(b) $x \bullet (x + y) = x$

Proof: apply the duality principle to the expression in part (a). If we interchange all • and + operators, we obtain the expression in part (b).

65

Two More Theorems

- ***Theorem 7:*** ***Common Identities***
  a) $x + (x' \bullet y) = x + y$
  b) $x \bullet (x' + y) = x \bullet y$

- ***Theorem 8:*** ***Consensus***
  a) $(x \bullet y) + (y \bullet z) + (x' \bullet z) = (x \bullet y) + (x' \bullet z)$
  b) $(x + y) \bullet (y + z) \bullet (x' + z) = (x + y) \bullet (x' + z)$

- Try to prove these theorems on your own.
- Note that each pair of equations are related through the duality principle.

66

So now we have all these theorems. What good are they?

Example 1: Find the complement of the logical function *f*.

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

Solution: by repeated application of DeMorgan's Law.

$$\bar{f} = \overline{\bar{x} \bullet (\bar{y} + \bar{z}) \bullet (x + y + \bar{z})}$$
$$= \overline{\bar{x} \bullet (\bar{y} + \bar{z})} + \overline{(x + y + \bar{z})}$$
$$= \bar{\bar{x}} + \overline{(\bar{y} + \bar{z})} + \overline{(x + y + \bar{z})}$$
$$= x + (y \bullet z) + (\bar{x} \bullet \bar{y} \bullet z)$$

67

So now we have all these theorems. What good are they?

Example 2: Simplify the previous expression.

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

Solution: by algebraic manipulation.
Apply the theorems of Boolean algebra.

$\bar{f} = x + (\bar{x} \bullet \bar{y} \bullet z) + (y \bullet z)$	by Commutativity
$= x + (\bar{y} \bullet z) + (y \bullet z)$	by Theorem 7a
$= x + (\bar{y} + y) \bullet z$	by Distributivity
$= x + 1 \bullet z$	by Postulate 5a
$= x + z$	by Postulate 2b

68

More Examples

Equivalent Expression	Justification
$\bar{x}y\bar{z} + \bar{x}yz + x\bar{y}\bar{z} + \mathbf{xy\bar{z}} + xyz$	Original Expression
$\boxed{(\mathbf{\bar{x}y\bar{z}} + \mathbf{\bar{x}yz}) + (\mathbf{x\bar{y}\bar{z}} + \mathbf{xy\bar{z}}) + (\mathbf{xyz} + \mathbf{xy\bar{z}})}$	Idempotency
$\bar{x}y(\mathbf{z} + \mathbf{\bar{z}}) + x\bar{z}(\mathbf{y} + \mathbf{\bar{y}}) + xy(\mathbf{z} + \mathbf{\bar{z}})$	**Distributivity**
$\boxed{\bar{x}y(1) + x\bar{z}(1) + xy(1)}$	Complement
$\bar{x}y + x\bar{z} + xy$	**Identity for AND**
$\boxed{y(\mathbf{x} + \mathbf{\bar{x}}) + x\bar{z}}$	Distributivity
$\boxed{y(1) + x\bar{z}}$	Complement
$y + x\bar{z}$	Identity for AND

69

Example on pages:36

- Example 2.3
- Example 2.4

Precedence of Boolean Operators

- To avoid confusion or incorrect evaluation, the operators in Boolean expressions are applied according to the following order of precedence:
  1. Expressions in parentheses: ()
  2. NOT: '
  3. AND: •
  4. OR: +
- Also, it is conventional to omit the • symbol for AND where convenient. As a result, many Boolean expressions can be written in a compact form, often eliminating extraneous parentheses.

$$\overline{(x \bullet y)} \Leftrightarrow \overline{xy}$$
$$x \bullet (y + z) \Leftrightarrow x(y + z)$$
$$(\bar{x} \bullet z) + (\bar{y} \bullet \bar{z}) \Leftrightarrow \bar{x}z + \bar{y}\bar{z}$$

71

Reference Sheet: Postulates and Theorems

- Postulate 1: Closure
- Postulate 2: Identity
  a) $x + 0 = x$
  b) $x \bullet 1 = x$
- Postulate 3: Commutativity
  a) $x + y = y + x$
  b) $xy = yx$
- Postulate 4: Distributivity
  a) $x(y + z) = xy + xz$
  b) $x + yz = (x + y)(x + z)$
- Postulate 5: Complement
  a) $x + x' = 1$
  b) $x \bullet x' = 0$
- Theorem 1: Idempotency
  a) $x + x = x$
  b) $x \bullet x = x$
- Theorem 2: Tautology and Contradiction
  a) $x + 1 = 1$
  b) $x \bullet 0 = 0$
- Theorem 3: Involution
  $(x')' = x$
- Theorem 4: Associativity
  a) $x + (y + z) = (x + y) + z$
  b) $x(yz) = (xy)z$
- Theorem 5: DeMorgan's Law
  a) $(x + y)' = x'y'$
  b) $(xy)' = x' + y'$
- Theorem 6: Absorption
  a) $x + xy = x$
  b) $x(x + y) = x$
- Theorem 7: Common Identities
  a) $x + x'y = x + y$
  b) $x(x' + y) = xy$
- Theorem 8: Consensus
  a) $xy + yz + x'z = xy + x'z$
  b) $(x + y)(y + z)(x' + z)$
  $= (x + y)(x' + z)$

Notice: 1、 Precedence : “ ( ) ” > “ ’ ” > “ · ” > “+ ”
2、$A \cdot B$ can be written as AB.

72

Venn Diagrams

- A graphical technique to illustrate logic functions
- ***Large rectangle*** to represent all possible input combinations
- ***Overlapping bubbles*** to represent each input variable
- Represent logical function by ***sharing regions*** where the function is equal to 1

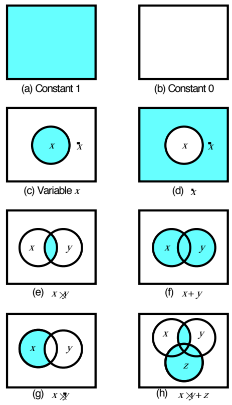
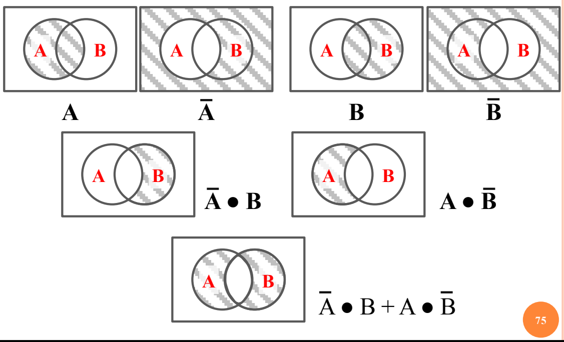


Figure 2.14. The Venn diagram representation.

Venn Diagrams

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



Conclusion

- Axioms of Boolean algebra
- Duality
- Properties (7 kinds and includes 16 identities)
- Can use the Properties freely to simplify or prove or explain logic expressions

- Textbook Reading: Chapter 2.5, Pages:33-43
- Assignment: Page 111: 2.2 , 2.3 and 2.7