+

Boolean Algebra

- Mathematical discipline used to design and analyze the behavior of the digital circuitry in digital computers and other digital systems

- Named after George Boole
  - English mathematician
  - Proposed basic principles of the algebra in 1854

- Claude Shannon suggested Boolean algebra could be used to solve problems in relay-switching circuit design

- Is a convenient tool:
  - Analysis
    - It is an economical way of describing the function of digital circuitry
  - Design
    - Given a desired function, Boolean algebra can be applied to develop a simplified implementation of that function

+ Boolean Variables and Operations

- Makes use of variables and operations
  - Are logical
  - A variable may take on the value 1 (TRUE) or 0 (FALSE)
  - Basic logical operations are AND, OR, and NOT
- AND
  - Yields true (binary value 1) if and only if both of its operands are true
  - In the absence of parentheses the AND operation takes precedence over the OR operation
  - When no ambiguity will occur the AND operation is represented by simple concatenation instead of the dot operator
- OR
  - Yields true if either or both of its operands are true
- NOT
  - Inverts the value of its operand

Table 11.1
Boolean Operators

P	Q	NOT P ($\overline{P}$)	P AND Q ($P \bullet Q$)	P OR Q ($P + Q$)	P NAND Q ($\overline{P \bullet Q}$)	P NOR Q ($\overline{P + Q}$)	P XOR Q ($P \oplus Q$)
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0

(a) Boolean Operators of Two Input Variables

Operation	Expression	Output = 1 if
AND	$A \bullet B \bullet \ldots$	All of the set {A, B, …} are 1.
OR	$A + B + \ldots$	Any of the set {A, B, …} are 1.
NAND	$\overline{A \bullet B \bullet \ldots}$	Any of the set {A, B, …} are 0.
NOR	$\overline{A + B + \ldots}$	All of the set {A, B, …} are 0.
XOR	$A \oplus B \oplus \ldots$	The set {A, B, …} contains an odd number of ones.

(b) Boolean Operators Extended to More than Two Inputs (A, B, . . .)

Table 11.2
Basic Identities of Boolean Algebra

Basic Postulates		
$A \bullet B = B \bullet A$	$A + B = B + A$	Commutative Laws
$A \bullet (B + C) = (A \bullet B) + (A \bullet C)$	$A + (B \bullet C) = (A + B) \bullet (A + C)$	Distributive Laws
$1 \bullet A = A$	$0 + A = A$	Identity Elements
$A \bullet \overline{A} = 0$	$A + \overline{A} = 1$	Inverse Elements
Other Identities		
$0 \bullet A = 0$	$1 + A = 1$	
$A \bullet A = A$	$A + A = A$	
$A \bullet (B \bullet C) = (A \bullet B) \bullet C$	$A + (B + C) = (A + B) + C$	Associative Laws
$\overline{A \bullet B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \bullet \overline{B}$	DeMorgan's Theorem

Table 11.2 Basic Identities of Boolean Algebra

Basic Logic Gates

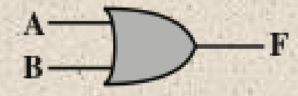
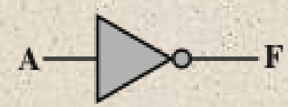
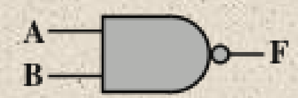
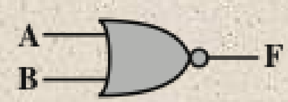
Name	Graphical Symbol	Algebraic Function	Truth Table
AND		$F = A \bullet B$ or $F = AB$	A B | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = A + B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT		$F = \overline{A}$ or $F = A'$	A | F 0 | 1 1 | 0
NAND		$F = \overline{AB}$	A B | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = \overline{A + B}$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
XOR		$F = A \oplus B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0

Figure 11.1 Basic Logic Gates

+

Uses of NAND Gates

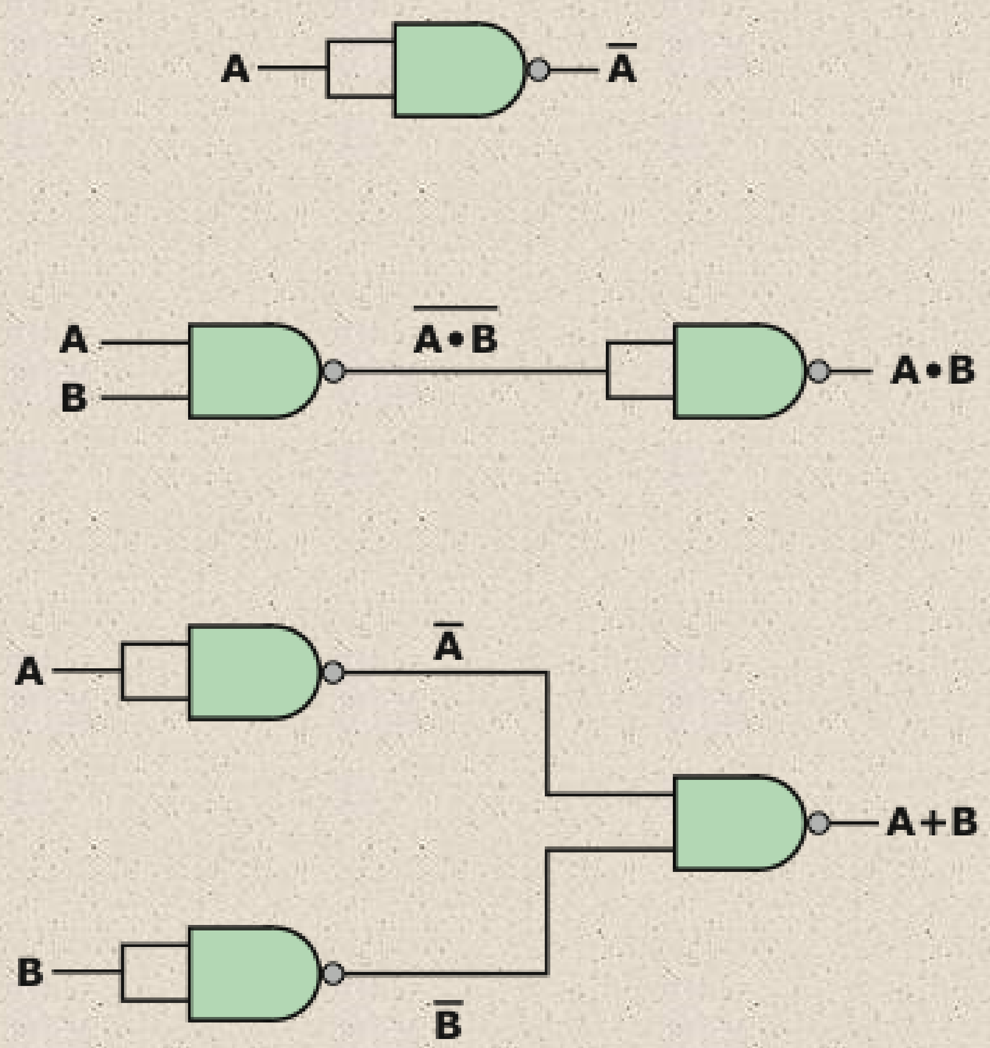


Figure 11.2 Some Uses of NAND Gates

Uses of NOR Gates

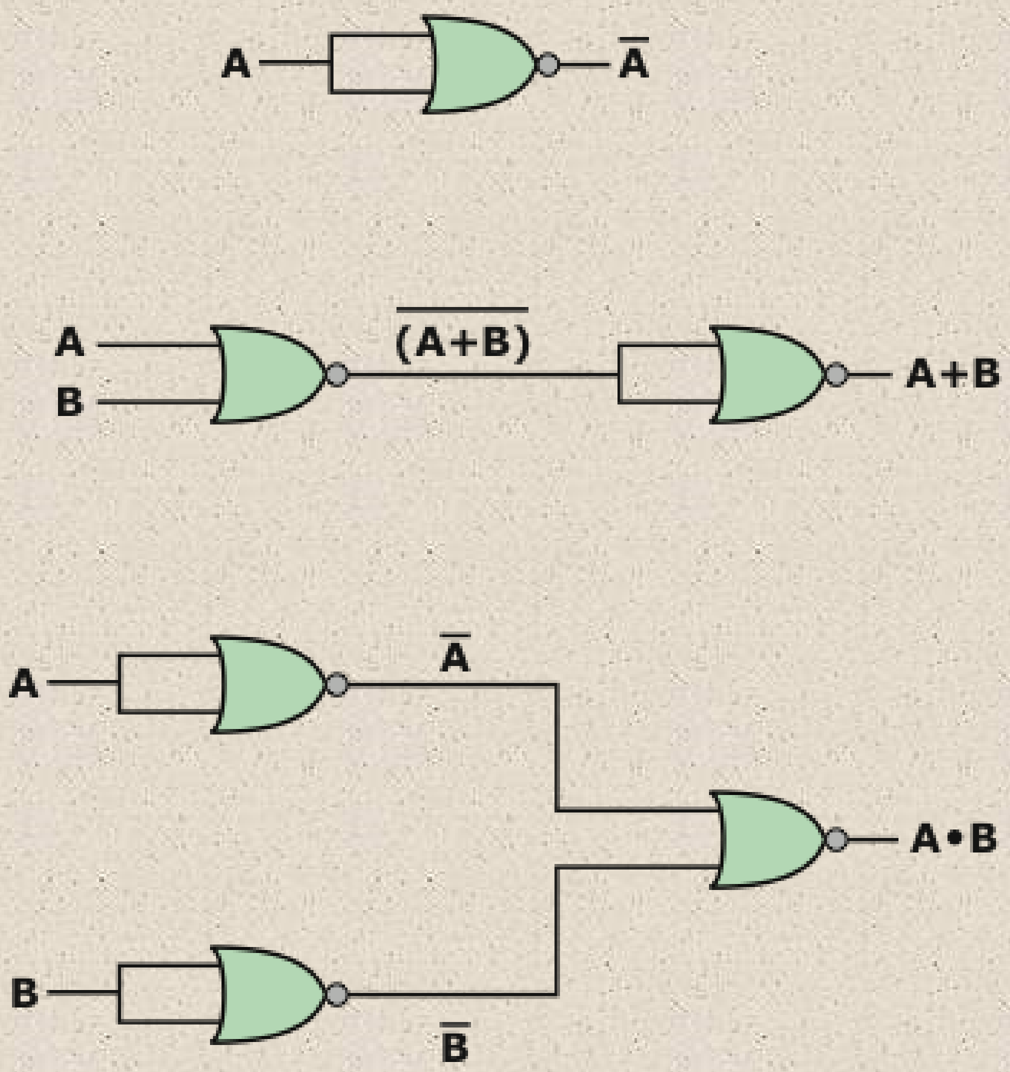
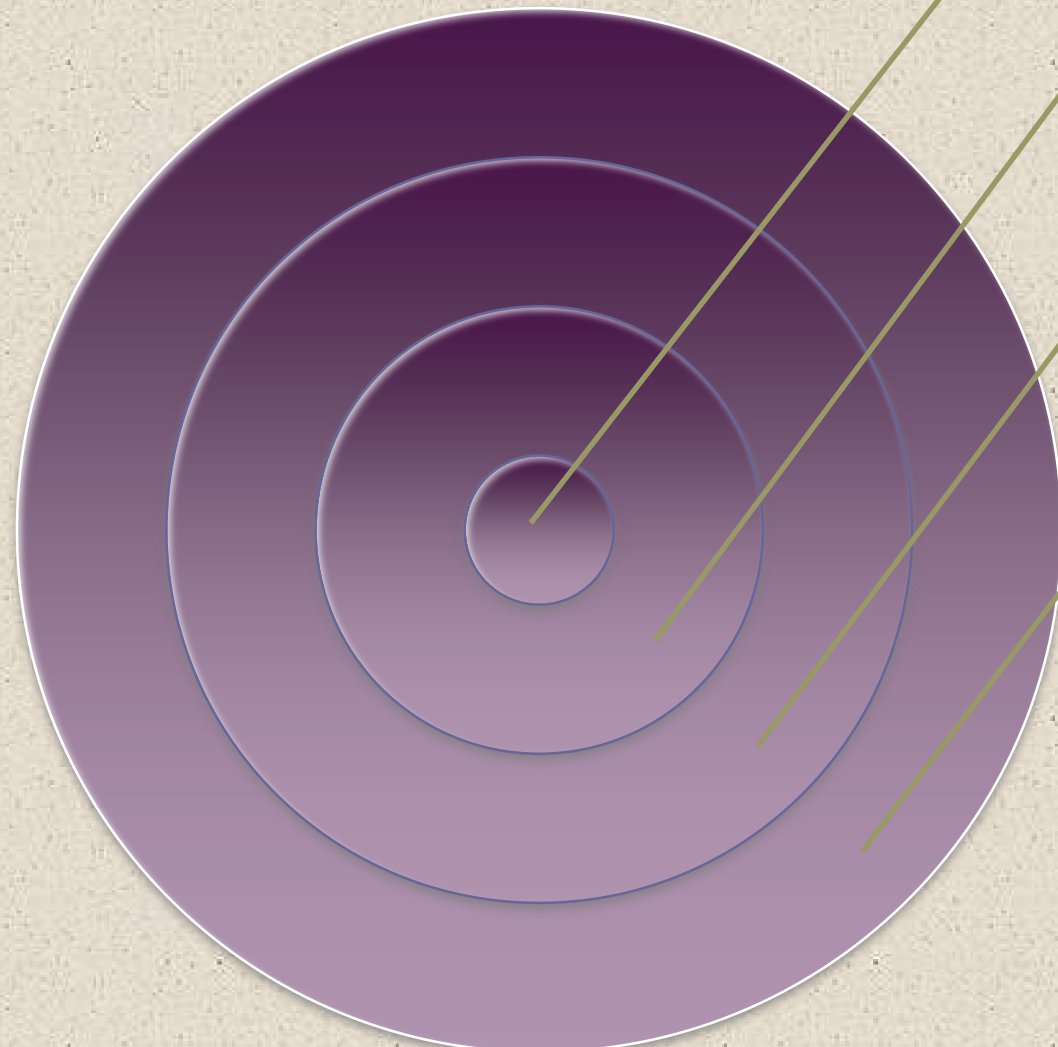


Figure 11.3 Some Uses of NOR Gates

Combinational Circuit



An interconnected set of gates whose output at any time is a function only of the input at that time

The appearance of the input is followed almost immediately by the appearance of the output, with only gate delays

Consists of *n* binary inputs and *m* binary outputs

Can be defined in three ways:

- Truth table
  - For each of the 2^n possible combinations of input signals, the binary value of each of the *m* output signals is listed
- Graphical symbols
  - The interconnected layout of gates is depicted
- Boolean equations
  - Each output signal is expressed as a Boolean function of its input signals

+

Boolean Function of Three Variables

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Table 11.3 A Boolean Function of Three Variables

Product-of-Sums Implementation of Table 11.3

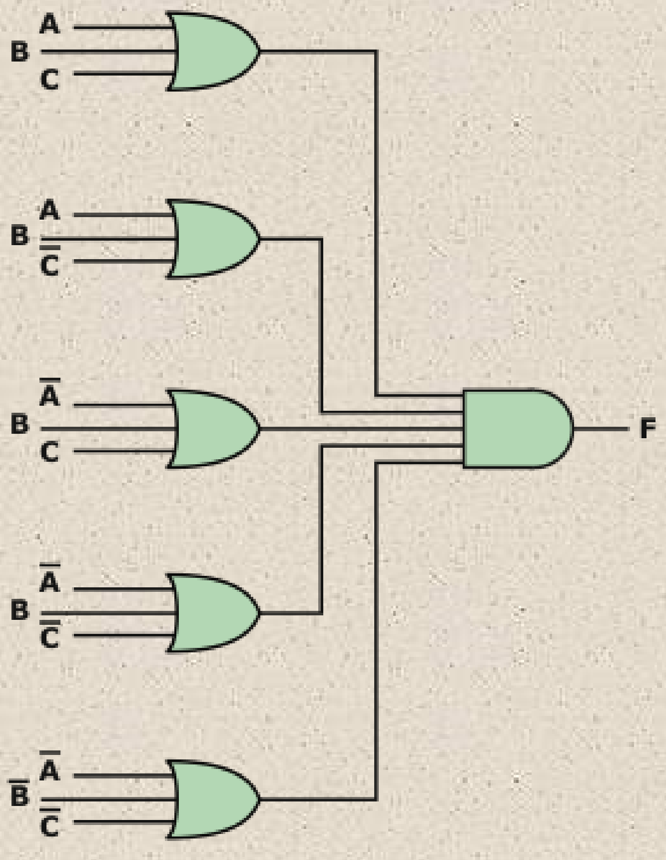


Figure 11.5 Product-of-Sums Implementation of Table 11.3

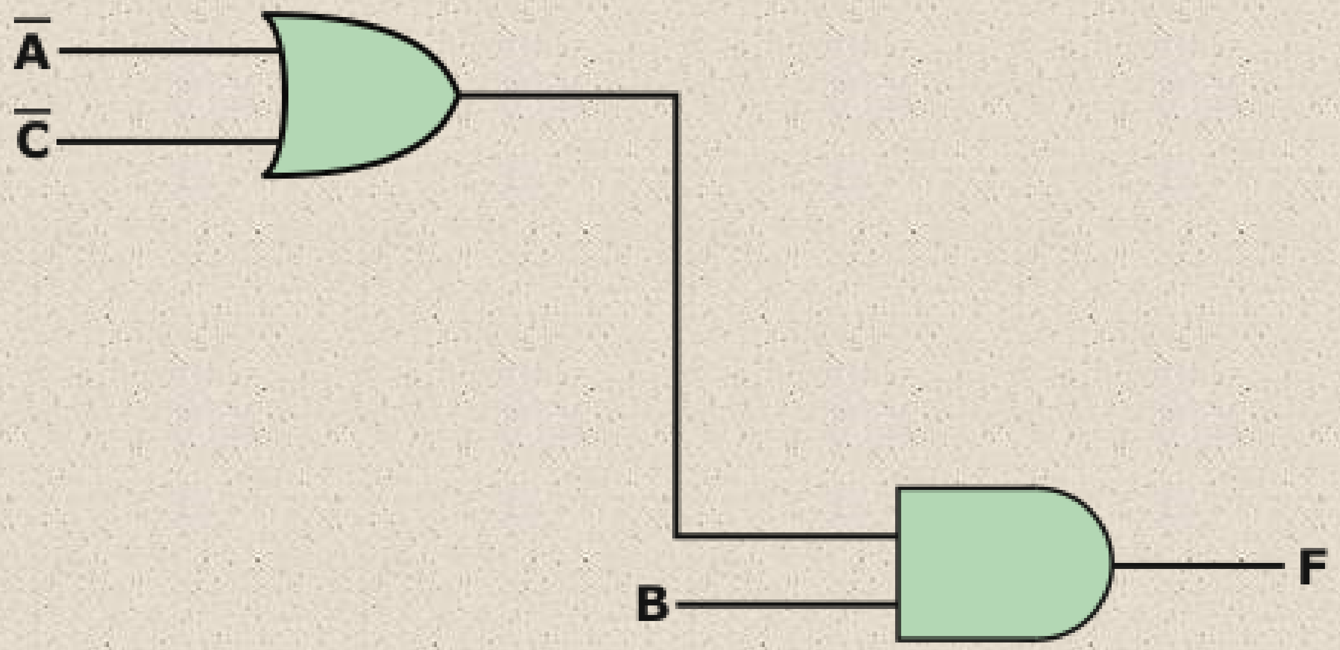


Figure 11.6 Simplified Implementation of Table 11.3

+

Algebraic Simplification

- Involves the application of the identities of Table 11.2 to reduce the Boolean expression to one with fewer elements

+

NAND and NOR Implementations

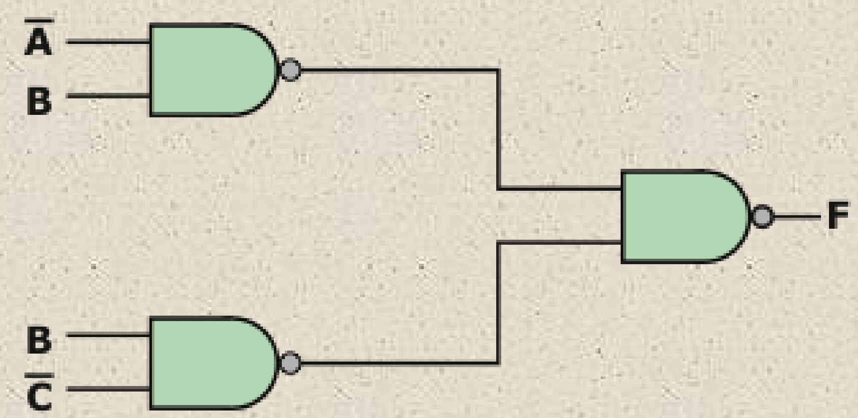


Figure 11.11 NAND Implementation of Table 11.3

Multiplexers

- connect multiple inputs to a single output

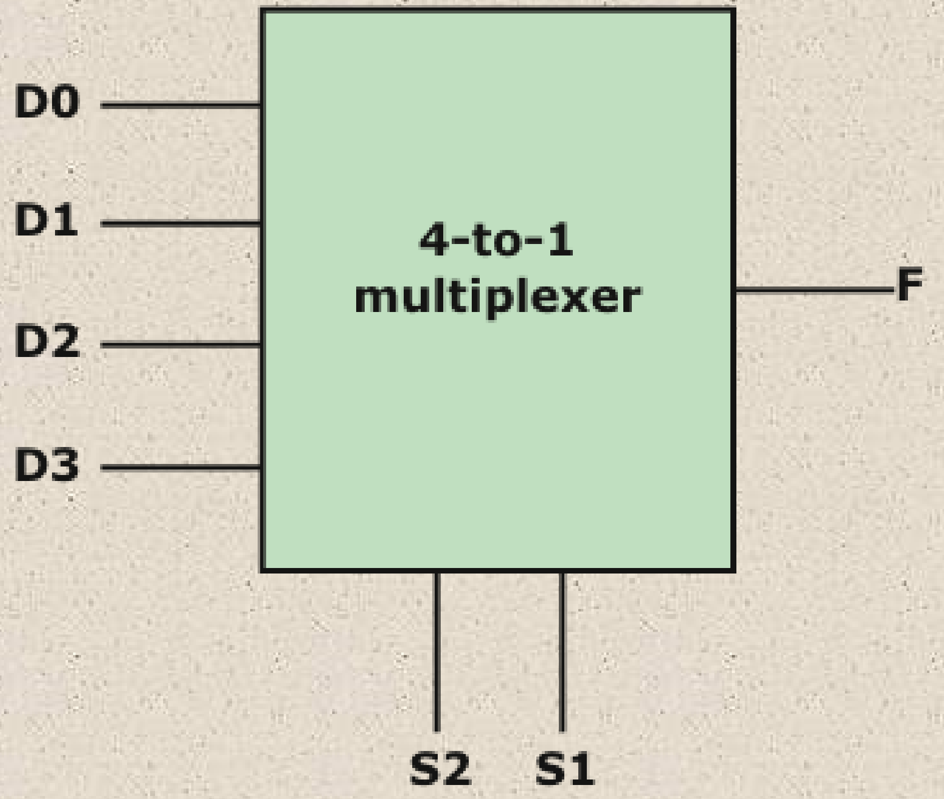


Figure 11.12 4-to-1 Multiplexer Representation

+

4-to-1 Multiplexer Truth Table

S2	S1	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

Table 11.7 4-to-1 Multiplexer Truth Table

Read-Only Memory (ROM)

- Memory that is implemented with combinational circuits
  - Combinational circuits are often referred to as “memoryless” circuits because their output depends only on their current input and no history of prior inputs is retained
- Memory unit that performs only the read operation
  - Binary information stored in a ROM is permanent and is created during the fabrication process
  - A given input to the ROM (address lines) always produces the same output (data lines)
  - Because the outputs are a function only of the present inputs, ROM is a combinational circuit

Input				Output			
X_1	X_2	X_3	X_4	Z_1	Z_2	Z_3	Z_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Table

11.8

+

Truth Table for a ROM

Binary Addition Truth Tables

(a) Single-Bit Addition

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) Addition with Carry Input

C_{in}	A	B	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 11.9 Binary Addition Truth Tables

4-Bit Adder

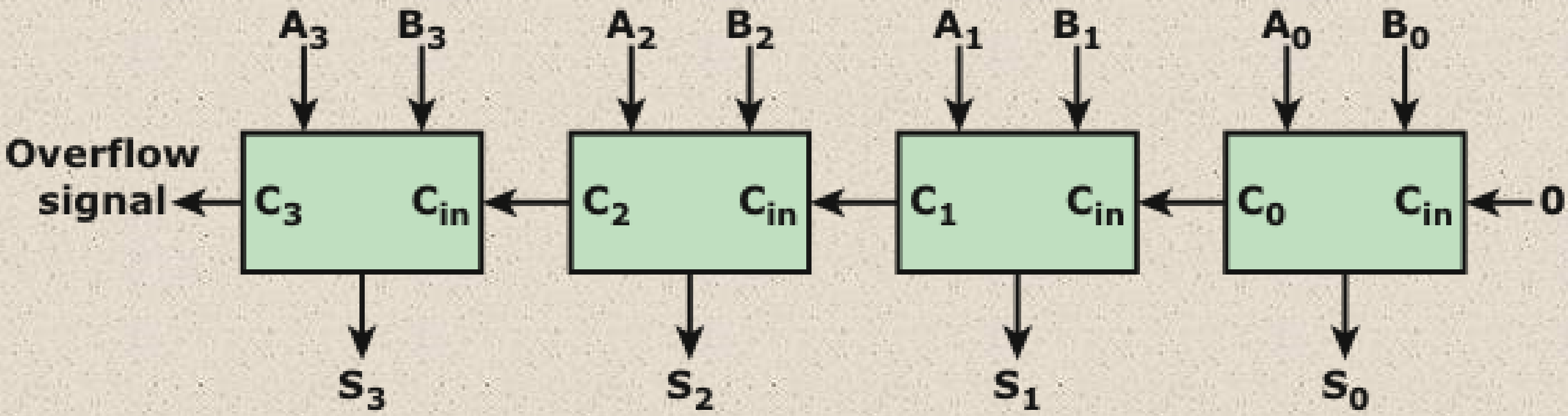


Figure 11.19 4-Bit Adder

+

Problems

11.1 Construct a truth table for the following Boolean expressions:

a. $(A + \overline{B} + C)(\overline{A} + B + \overline{C})$
b. $(A + B + \overline{C})(A + B + C)(\overline{A} + B + \overline{C})$
c. $\overline{A} + ((B + \overline{C})(\overline{B} + \overline{C}))$
d. $(A\,B) + (\overline{A}\,\overline{B}) + (A\overline{C})$

11.2 Simplify the following expressions according to the commutative law:

a. $(\overline{A} + D)(D + \overline{A})(B + E + \overline{A})(\overline{A} + E + B)(A + B + C)$
b. $(X + Y)(Y + Z)(Y + X)$
c. $(A + B + C)(D + E + F)(X + Y)(C + A + B)$
d. $(ABC) + (\overline{A}\,\overline{B}\,\overline{C}) + (\overline{B}\,\overline{C}) + A(B + C) + (CBA) + (\overline{C}\,\overline{B})$