Lecture – 3

Logic Gates

Lesson Outcomes

After completing this lecture, students will be able to

- *describe the operation of the inverter, the AND gate, and the OR gate*
- *describe the operation of the NAND gate and the NOR gate*
- *express the operation of NOT, AND, OR, NAND, and NOR gates with Boolean algebra*
- *describe the operation of the exclusive-OR and exclusive-NOR gates*
- *use logic gates in simple applications*

Key Terms

- **AND gate** A logic gate that produces a HIGH output only when all of the inputs are HIGH.
- **OR gate** A logic gate that produces a HIGH output when one or more inputs are HIGH.
- **NOT gate** A logic gate that inverts or complements its input.
- **NAND gate** A logic gate that produces a LOW output only when all the inputs are HIGH.
- **NOR gate** A logic gate in which the output is LOW when one or more of the inputs are HIGH.
- **Exclusive-NOR (XNOR) gate** A logic gate that produces a LOW only when the two inputs are at opposite levels.
- **Exclusive-OR (XOR) gate** A logic gate that produces a HIGH output only when its two inputs are at opposite levels.
- **Boolean algebra** The mathematics of logic circuits.
- **Truth table** A table showing the inputs and corresponding output(s) of a logic circuit.
- **Inverter** A logic circuit that inverts or complements its input.

Key Terms (continue)

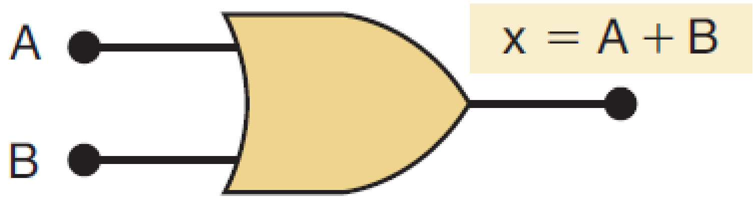
- **CMOS** Complementary metal-oxide semiconductor; a class of integrated logic circuits that is implemented with a type of field-effect transistor.
- **EEPROM** A type of nonvolatile PLD reprogrammable link based on electrically erasable programmable read-only memory cells and can be turned on or off repeatedly by programming.
- **EPROM** A type of PLD nonvolatile programmable link based on electrically programmable read-only memory cells and can be turned either on or off once with programming.
- **SRAM** A type of PLD volatile reprogrammable link based on static random-access memory cells and can be turned on or off repeatedly with programming.
- **Propagation delay time** The time interval between the occurrence of an input transition and the occurrence of the corresponding output transition in a logic circuit.
- **VHDL** A standard hardware description language that describes a function with an entity/architecture structure.

Common logic term

TABLE 3-1 Common logic terms.

Logic 0	Logic 1
False	True
Off	On
LOW	HIGH
No	Yes
Open switch	Closed switch

Operation of OR gate

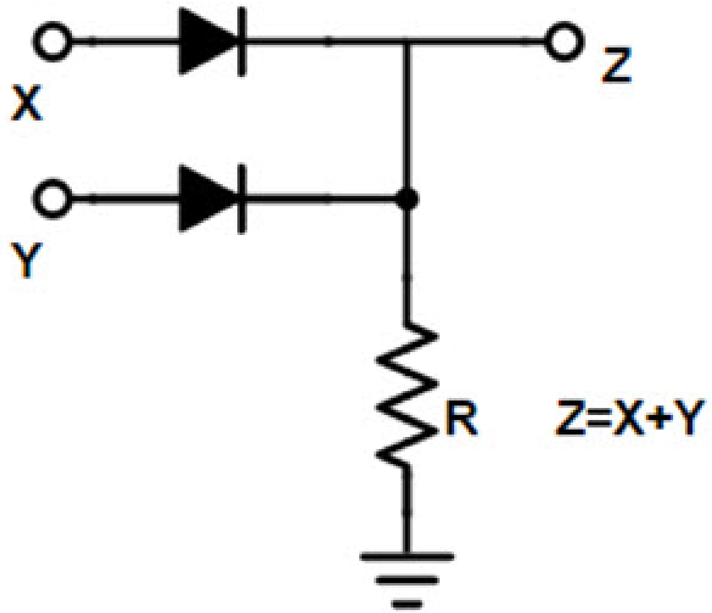


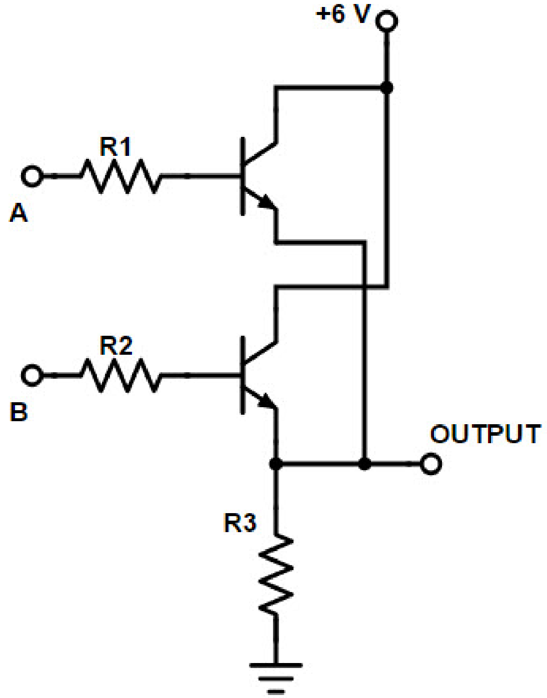
Symbol of TWO input OR gate

Truth Table

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

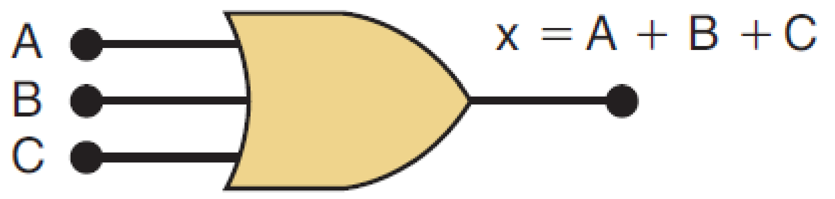
Internal structure





Operation of OR gate

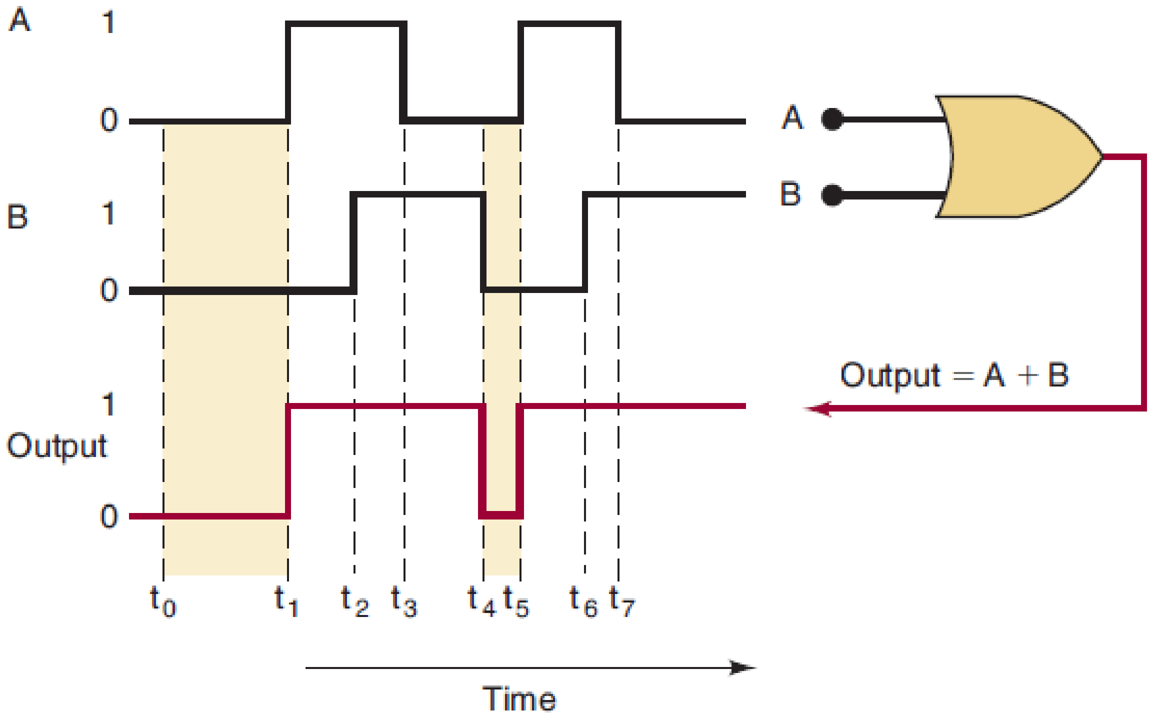
Boolean expression: $x = A + B + C$



Symbol of THREE input OR gate

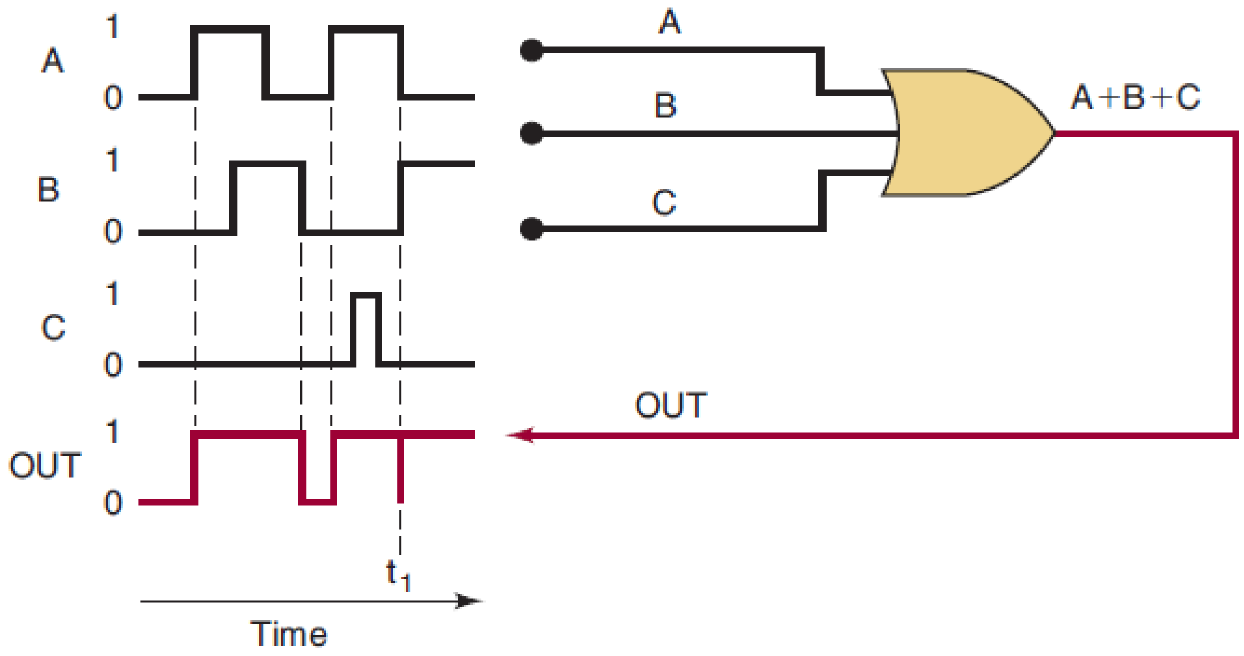
Truth Table

A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

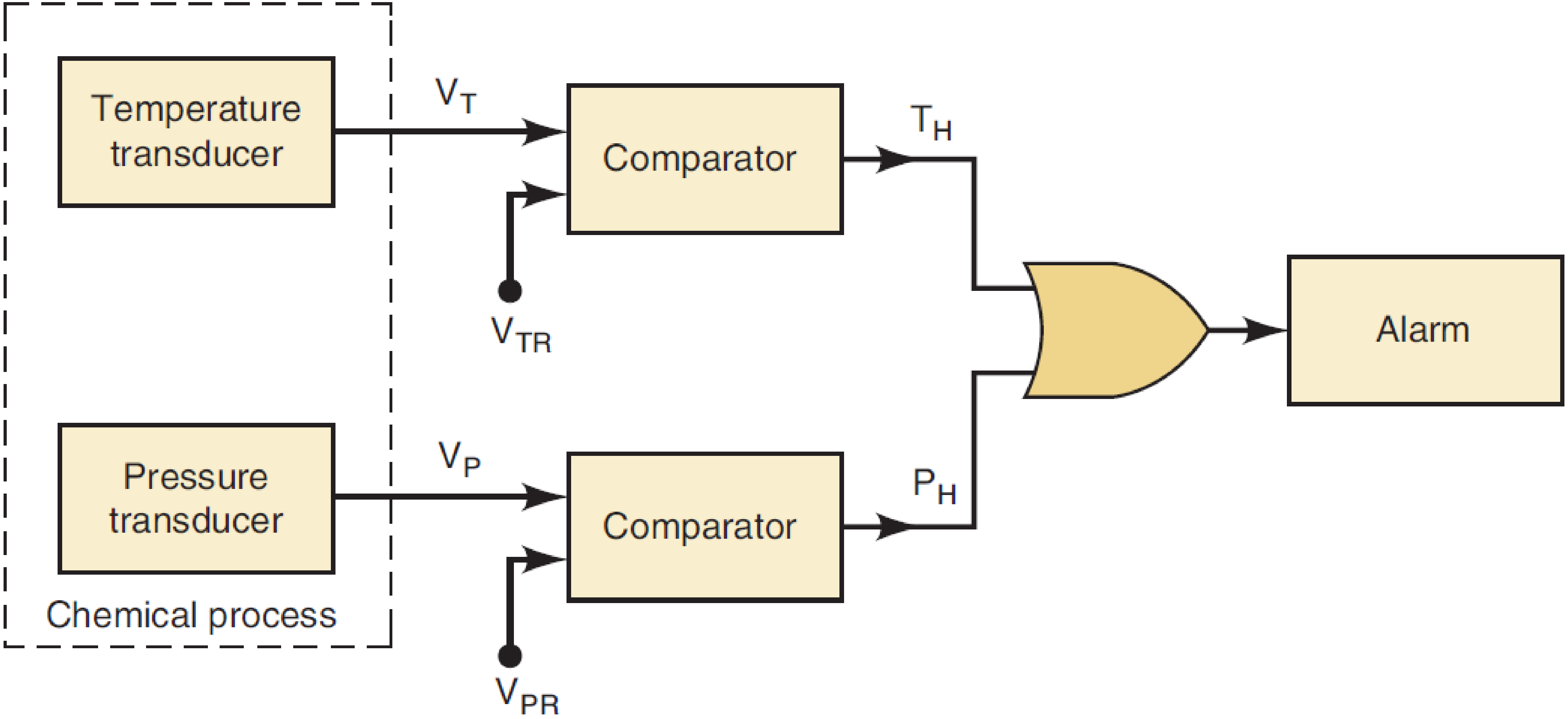


TWO input OR operation

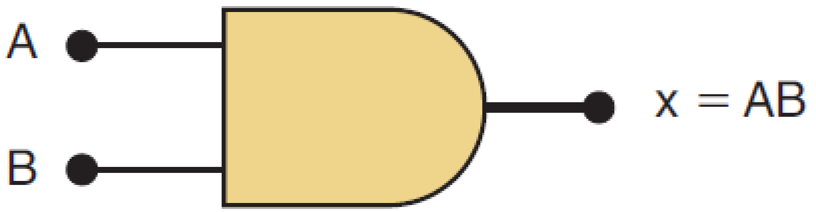
THREE input OR operation



OR GATE application – alarms system in chemical process



Operation of AND gate



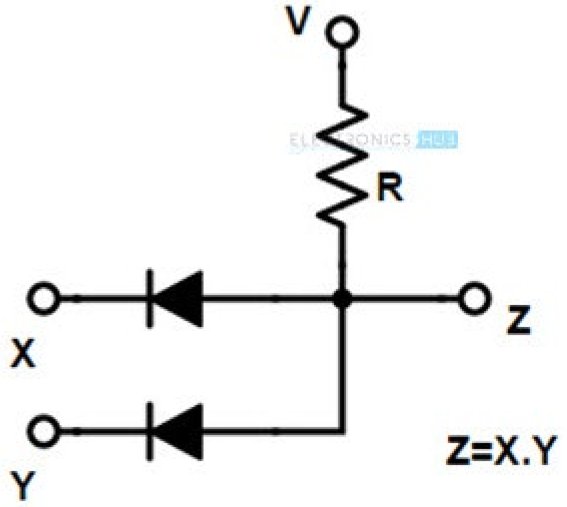
Symbol of TWO input **AND** gate

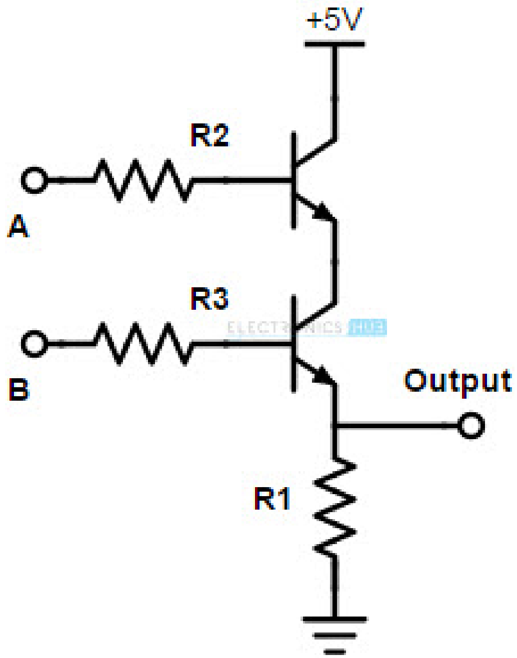
Truth Table

A	B	x = A·B
0	0	0
0	1	0
1	0	0
1	1	1

Boolean expression: $x = A \bullet B$

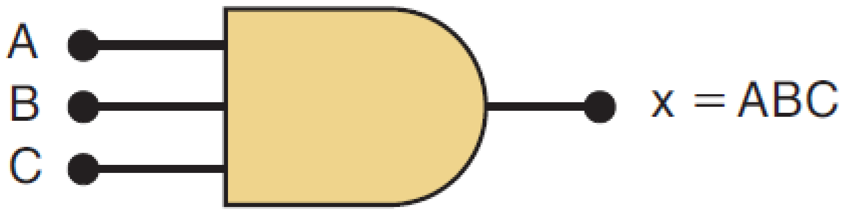
Internal structure





Operation of AND gate

Boolean expression: $x = A \bullet B \bullet C$

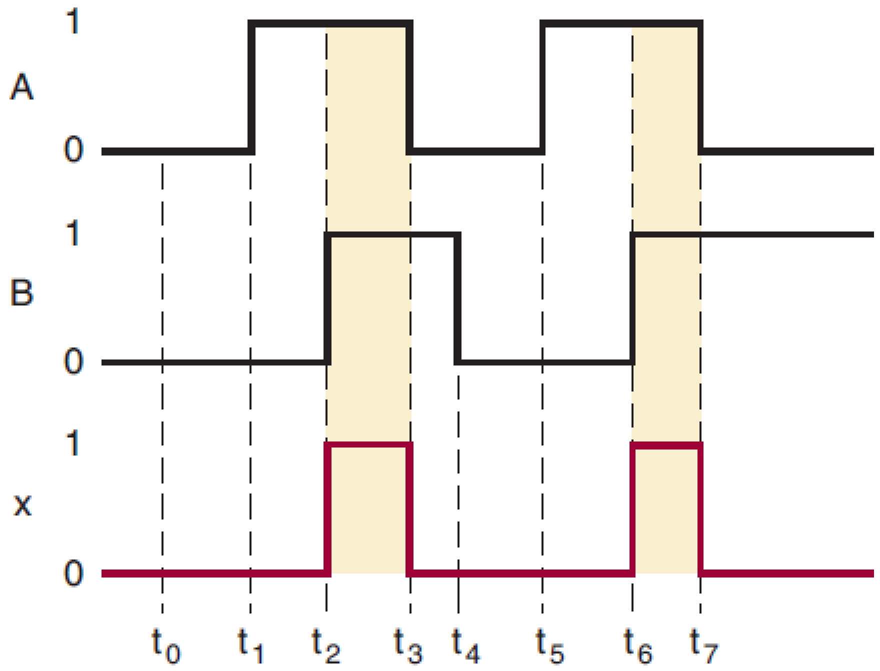


Symbol of THREE input AND gate

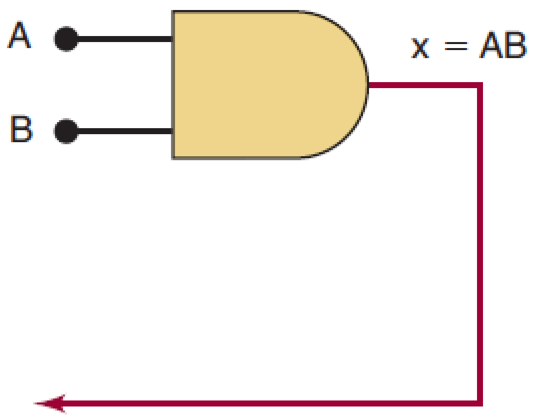
Truth Table

A	B	C	x = ABC
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

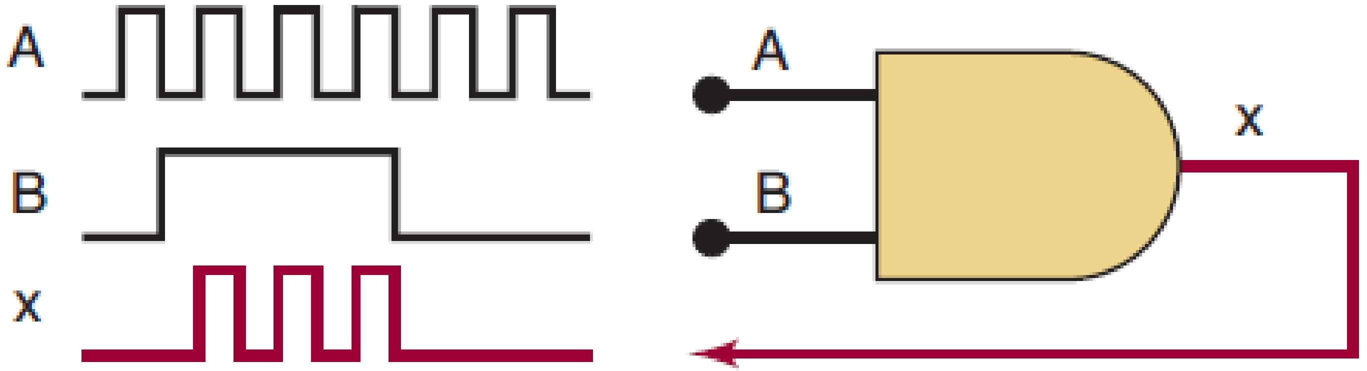
Operation of AND gate



TWO input AND operation



TWO input AND operation

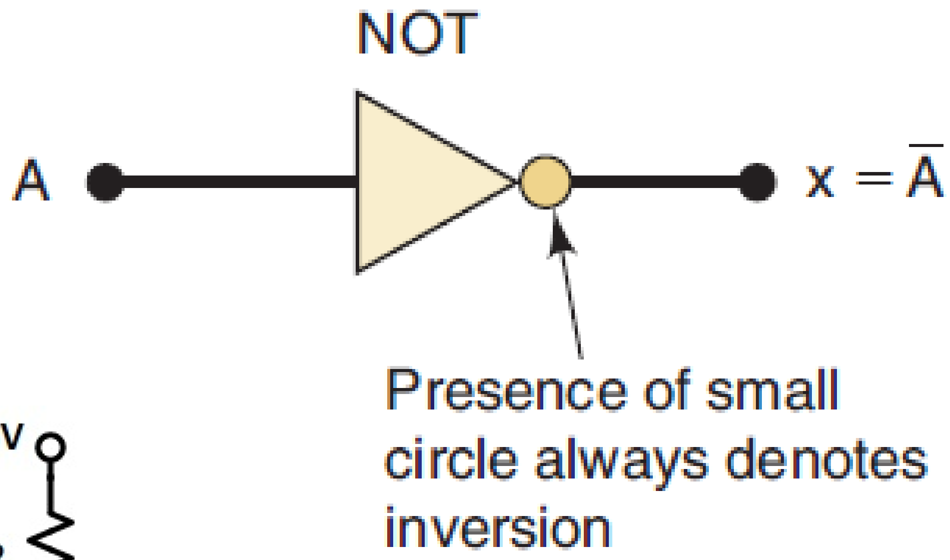


Operation of NOT gate

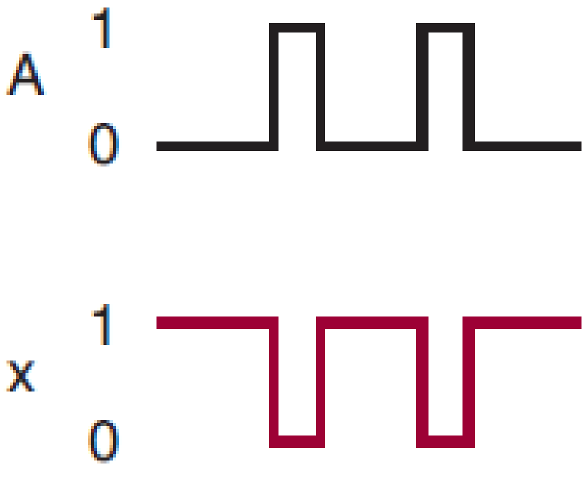
NOT

A	$x = \bar{A}$
0	1
1	0

(a)

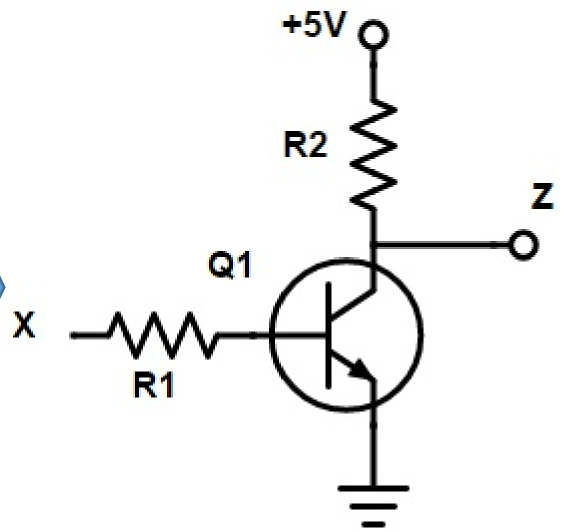


(b)

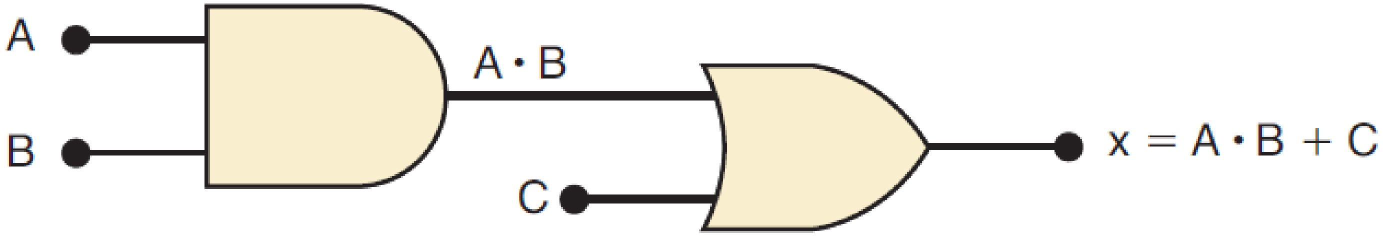


(c)

Internal structure

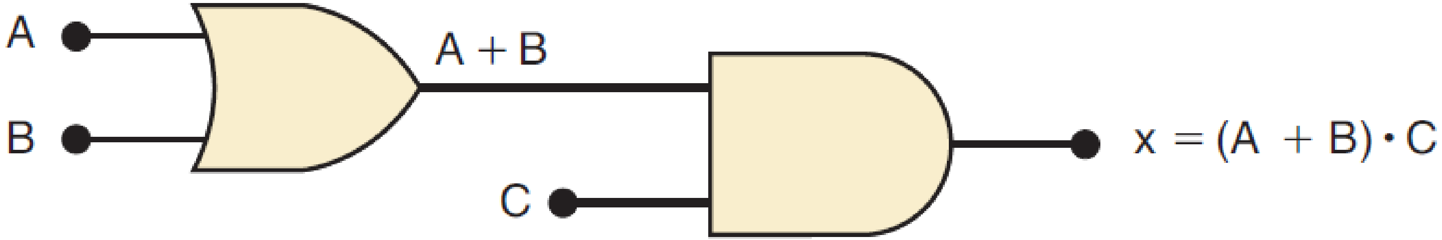


Boolean expression of NOT operation: $x = \bar{A}$



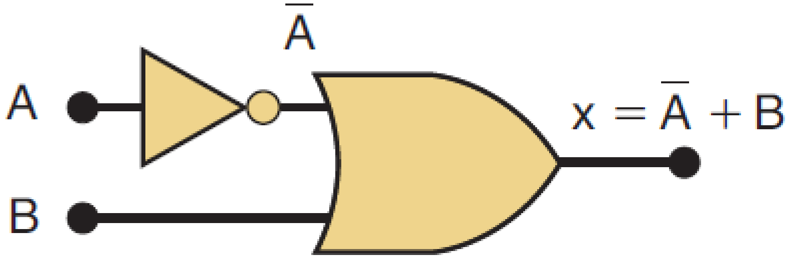
A
B
A • B
C
x = A • B + C

(a)



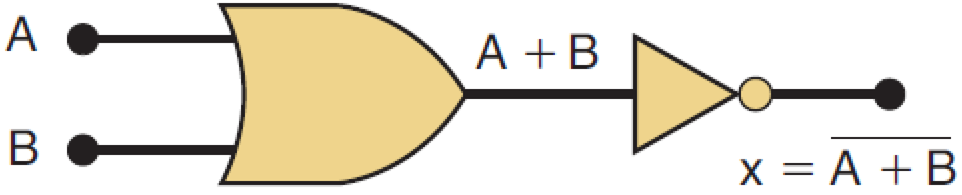
A
B
A + B
C
x = (A + B) • C

(b)



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

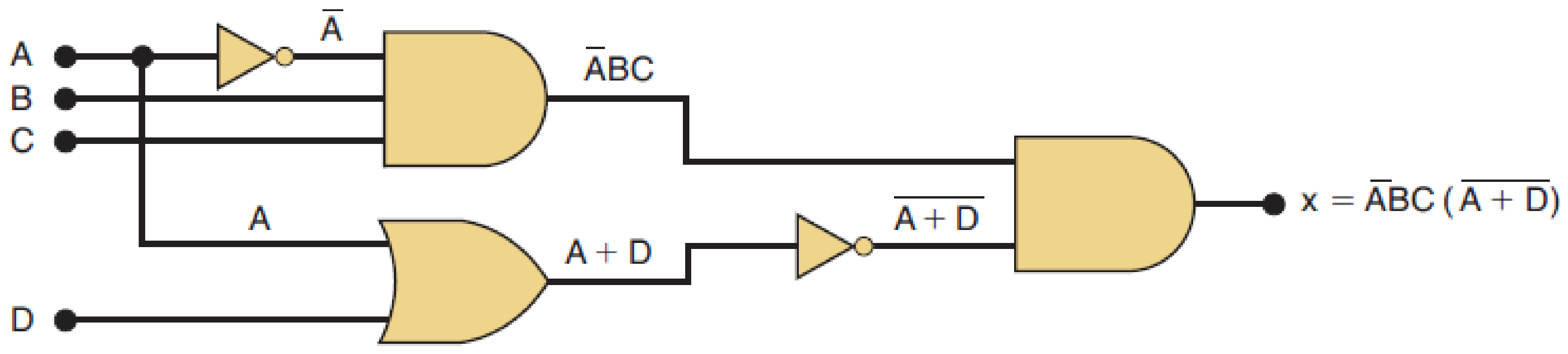
(a)



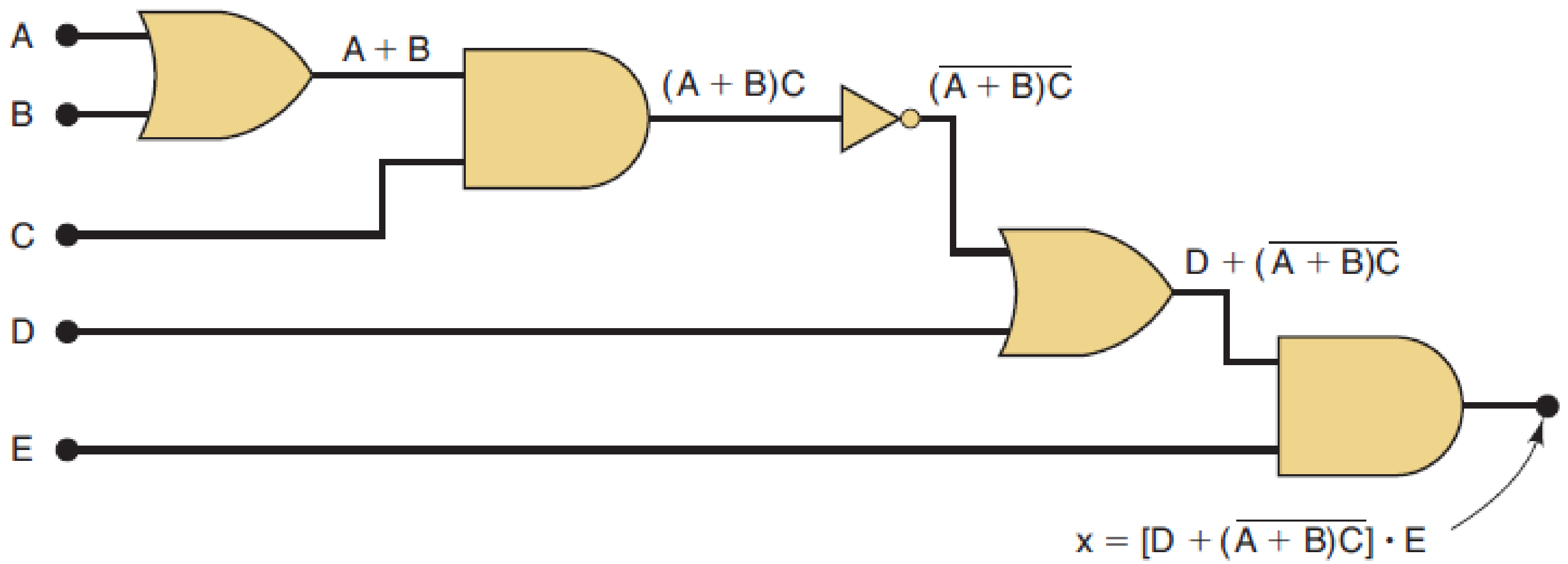
A
B
A + B
$x = \overline{A + B}$

(b)

Logic circuits with Boolean expression (contd..)

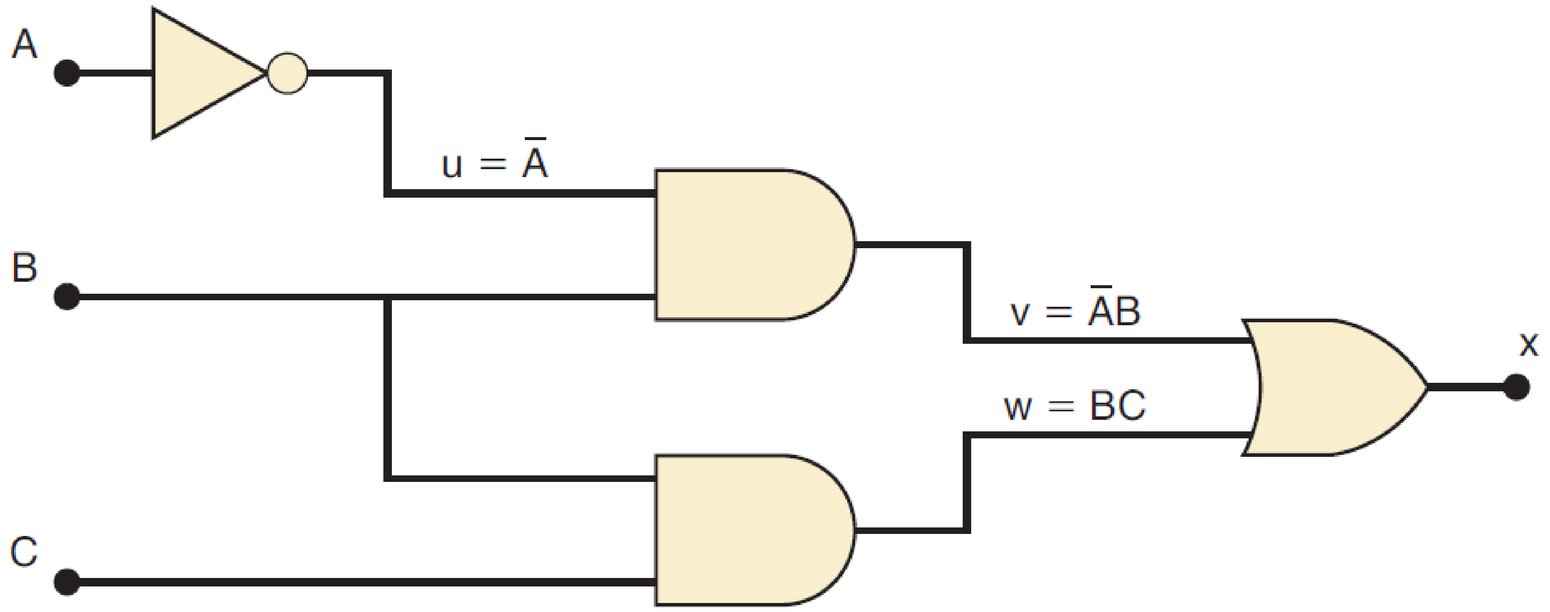


(a)



(b)

Logic circuits with Boolean expression (contd..)

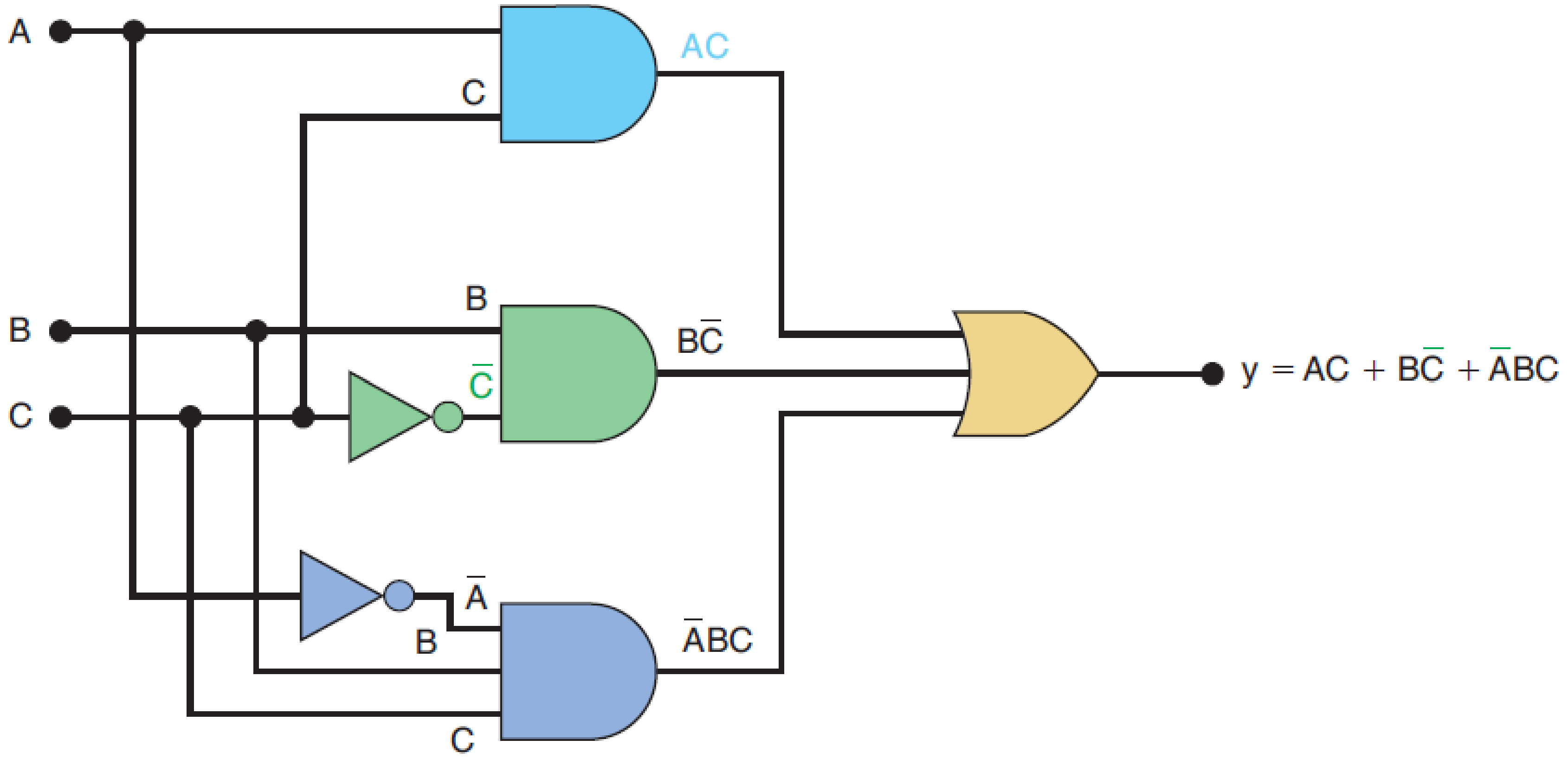


A	B	C	$u = \overline{A}$	$v = \overline{A}B$	$w = BC$	$x = v+w$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Constructing logic circuit from Boolean expression

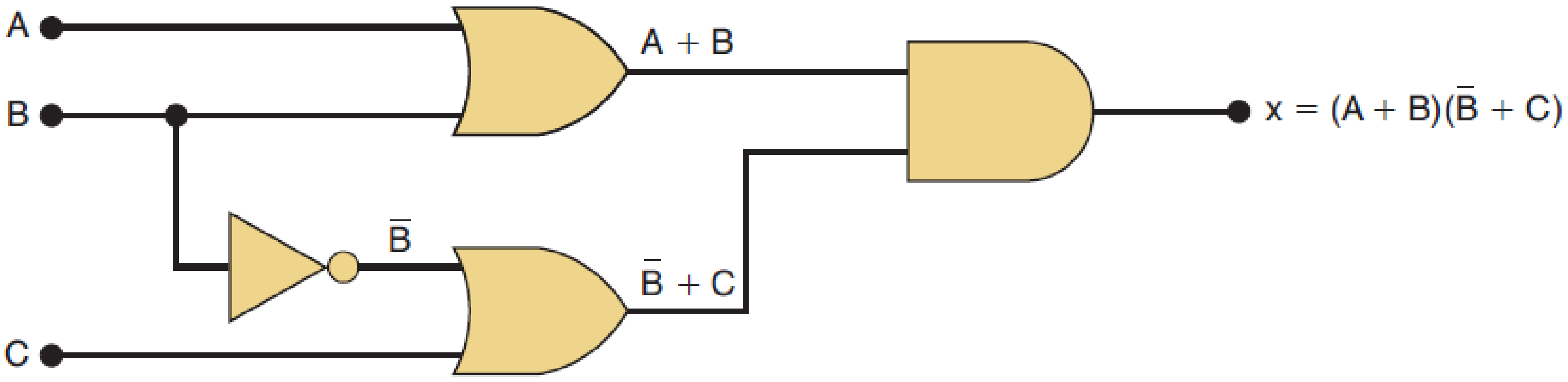
PROBLEM Construct a logic circuit from Boolean expression: $y = AC + B\overline{C} + \overline{A}BC$

SOLUTION

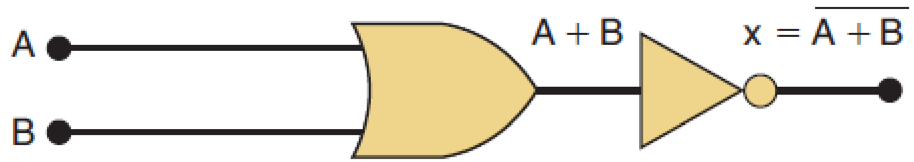


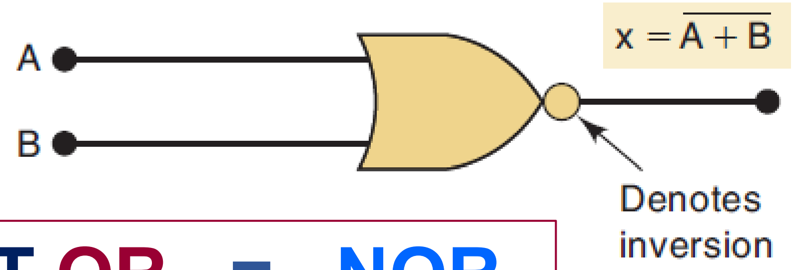
PROBLEM Construct a logic circuit from Boolean expression: $x = (A + B)(\overline{B} + C)$

SOLUTION



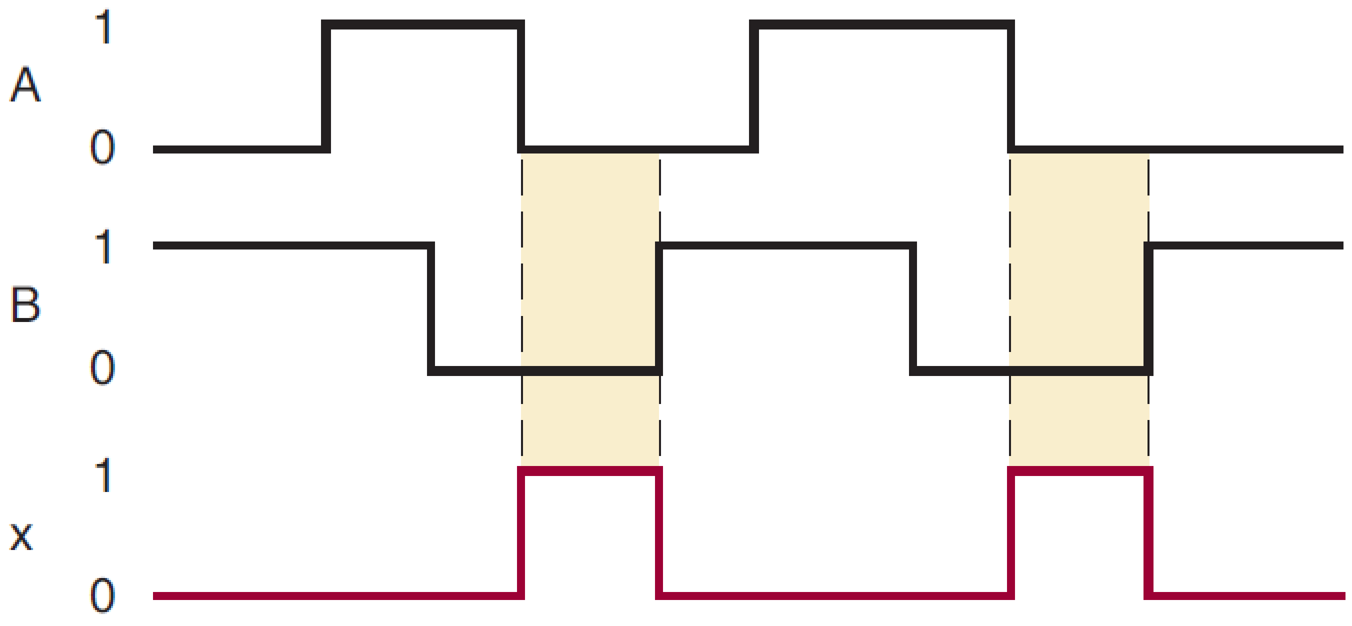
NOR operation

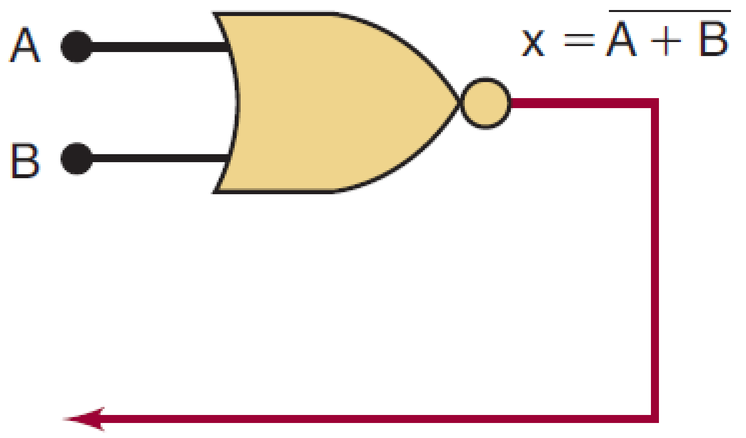




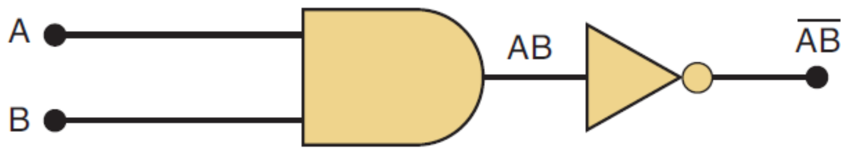
OR + NOT = NOT OR = NOR

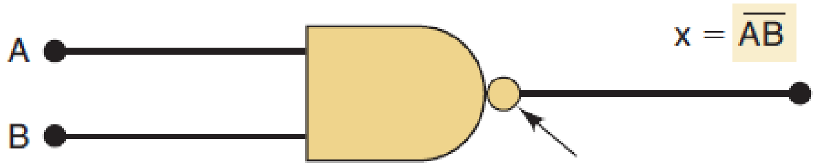
		OR	NOR
A	B	$A+B$	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0





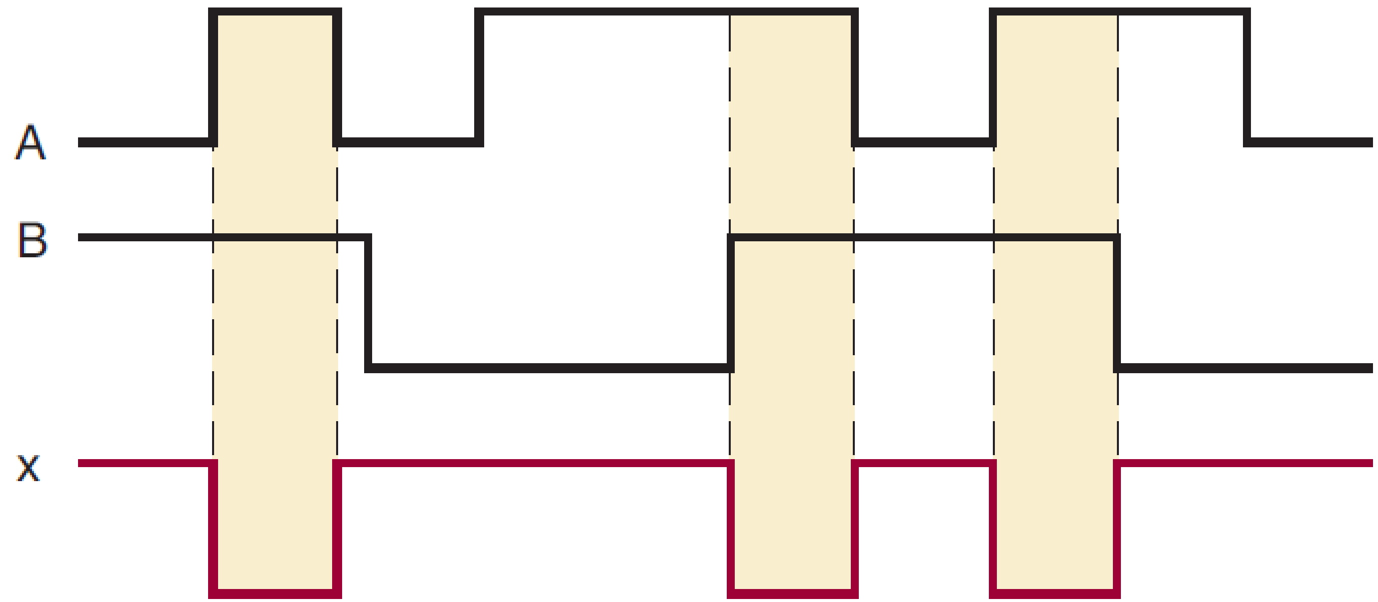
NAND operation

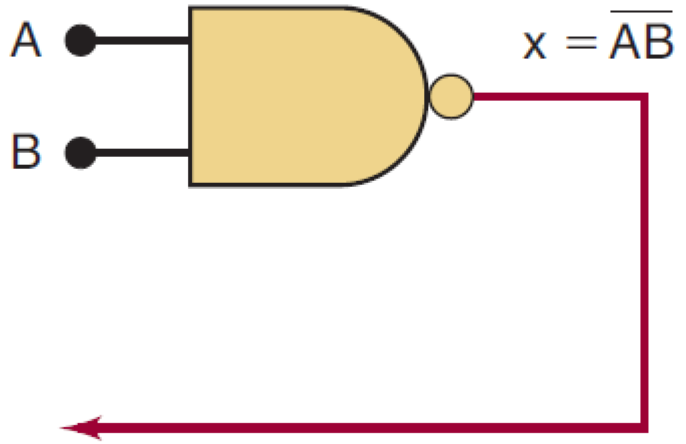




A	B	AND: AB	NAND: $\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

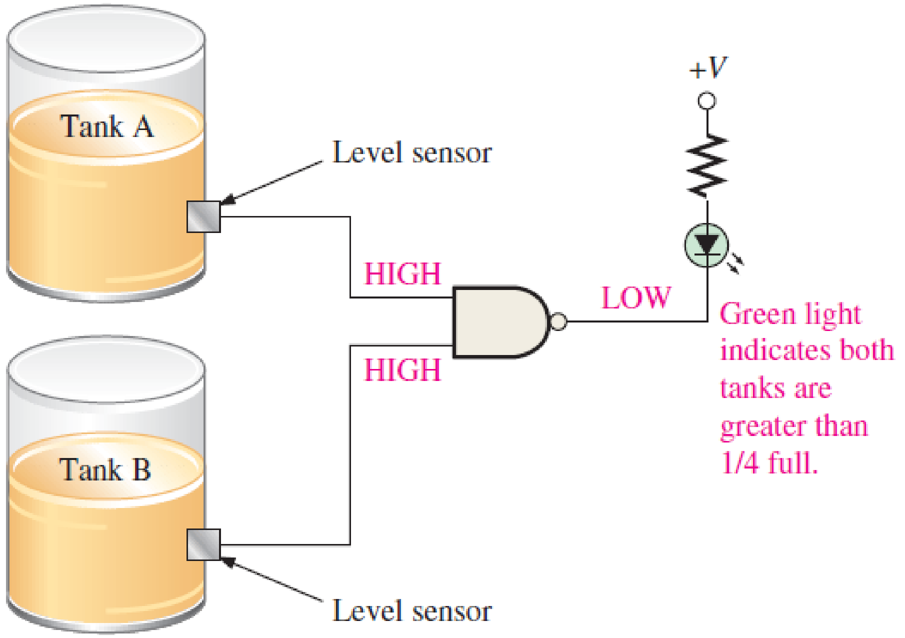
AND + NOT = NOT AND = NAND

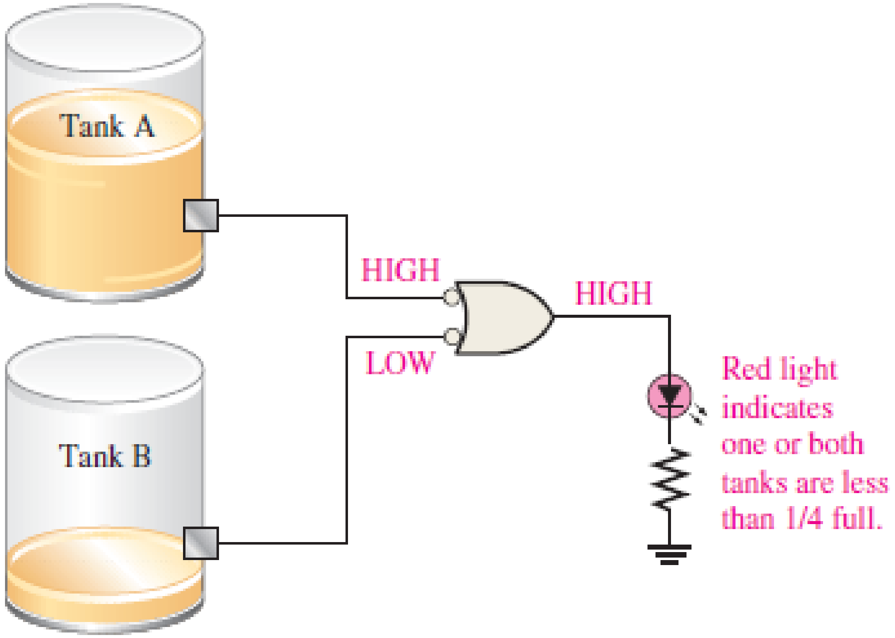




NAND application

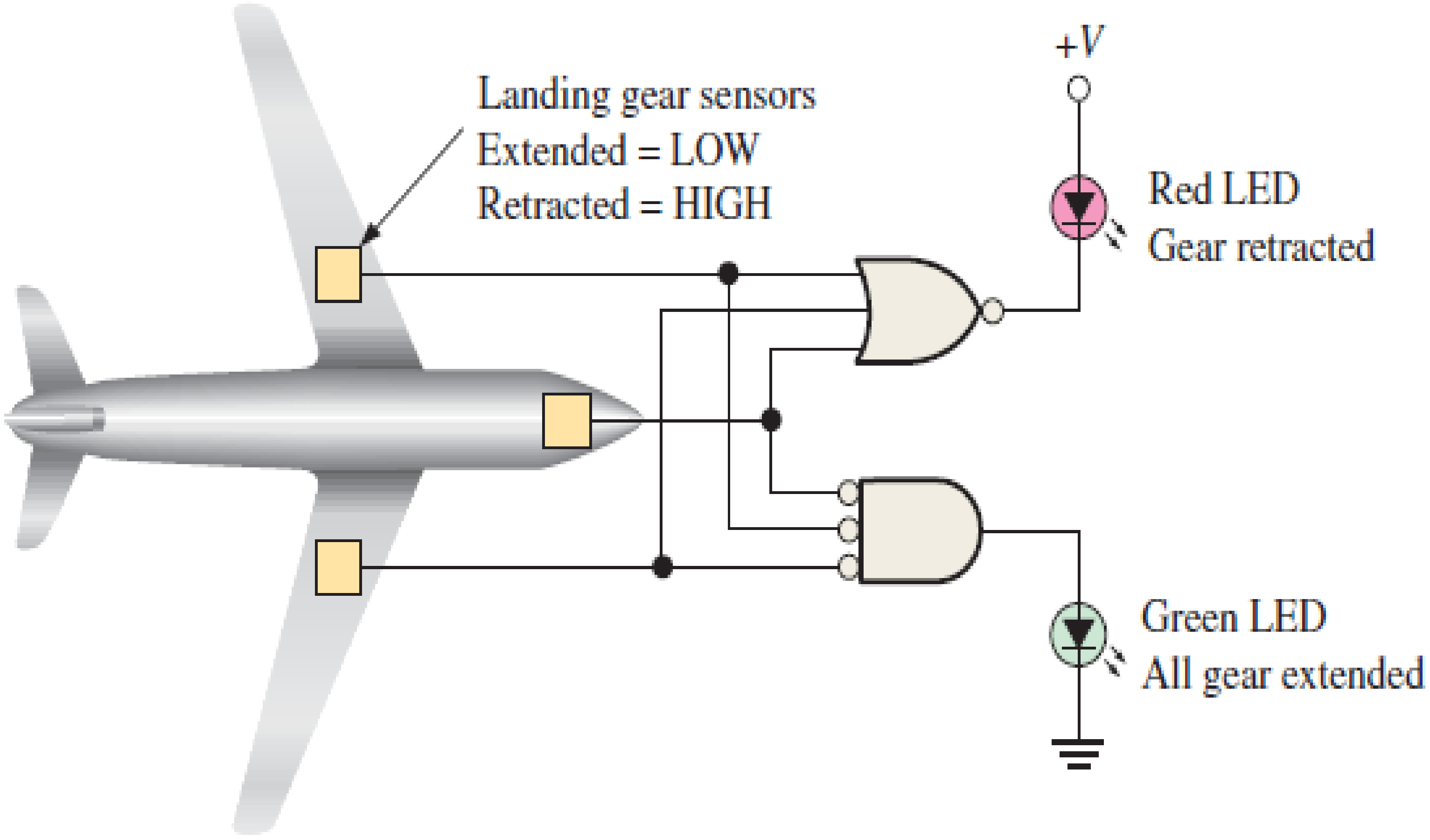
PROBLEM. Two tanks store certain liquid chemicals that are required in a manufacturing process. Each tank has a sensor that detects when the chemical level drops to 25% of full. It is required that a green LED on an indicator panel show when both tanks are more than one-quarter full. A red LED light will TURN ON when at least one of the tanks falls to the quarter-full level. Show how these requirements can be implemented.





NAND application

PROBLEM. As part of an aircraft's functional monitoring system, a circuit is required to indicate the status of the landing gears prior to landing. A **green** LED display turns on if all three gears are properly extended when the "gear down" switch has been activated in preparation for landing. A **red** LED display turns on if any of the gears fail to extend properly prior to landing. When a landing gear is extended, its sensor produces a LOW voltage. When a landing gear is retracted, its sensor produces a HIGH voltage. Implement a circuit to meet this requirement.



Boolean theorems

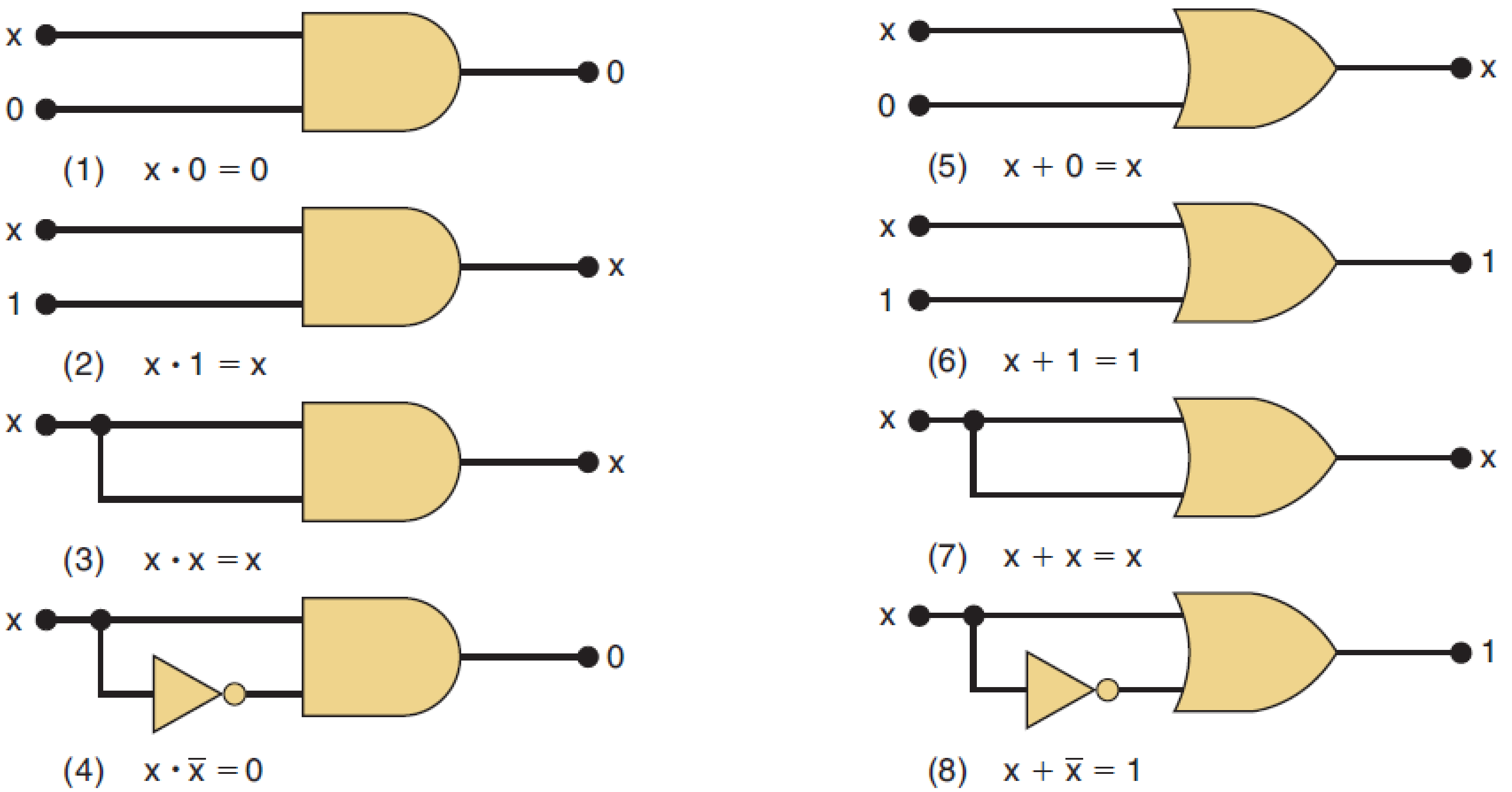


FIGURE 3-25 Single-variable theorems.

Multivariable Theorems

The theorems presented below involve more than one variable:

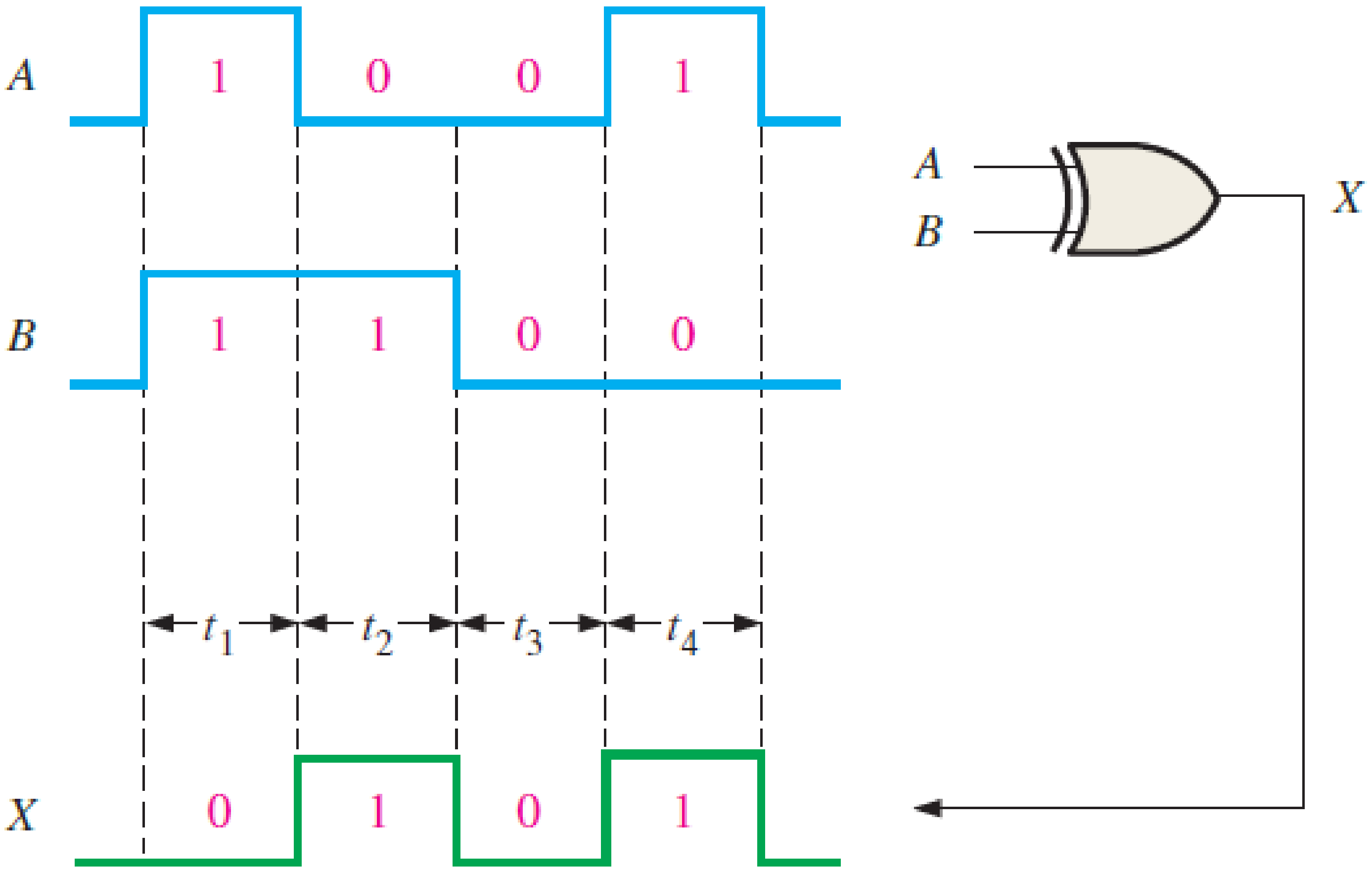
(9)	$x + y = y + x$
(10)	$x \cdot y = y \cdot x$
(11)	$x + (y + z) = (x + y) + z = x + y + z$
(12)	$x(yz) = (xy)z = xyz$
(13*a*)	$x(y + z) = xy + xz$
(13*b*)	$(w + x)(y + z) = wy + xy + wz + xz$
(14)	$x + xy = x$
(15*a*)	$x + \overline{x}y = x + y$
(15*b*)	$\overline{x} + xy = \overline{x} + y$

Exclusive OR gate

TABLE 3–11

Truth table for an exclusive-OR gate.

Inputs		Output
A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	0



Exclusive NOR

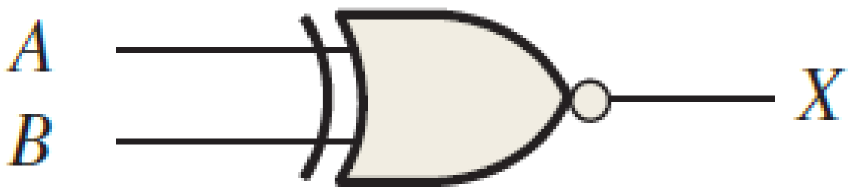
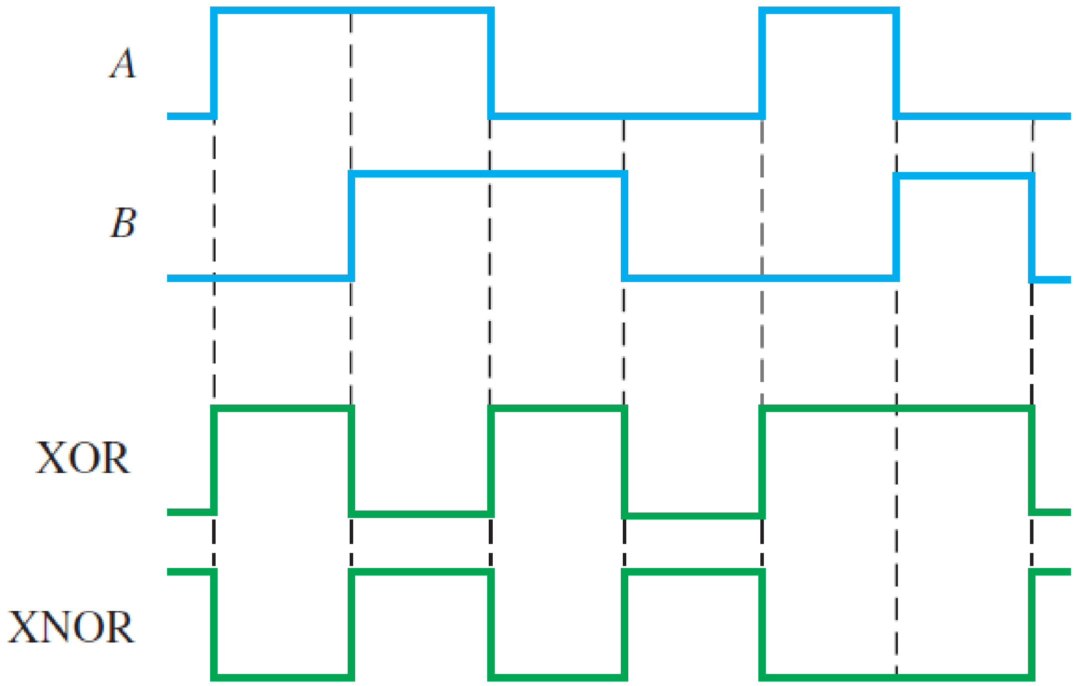
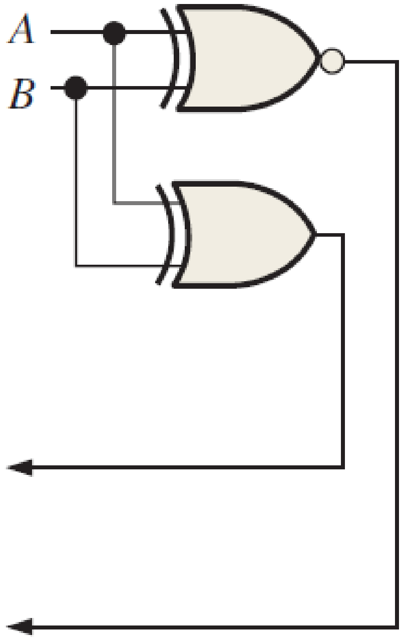


TABLE 3–12

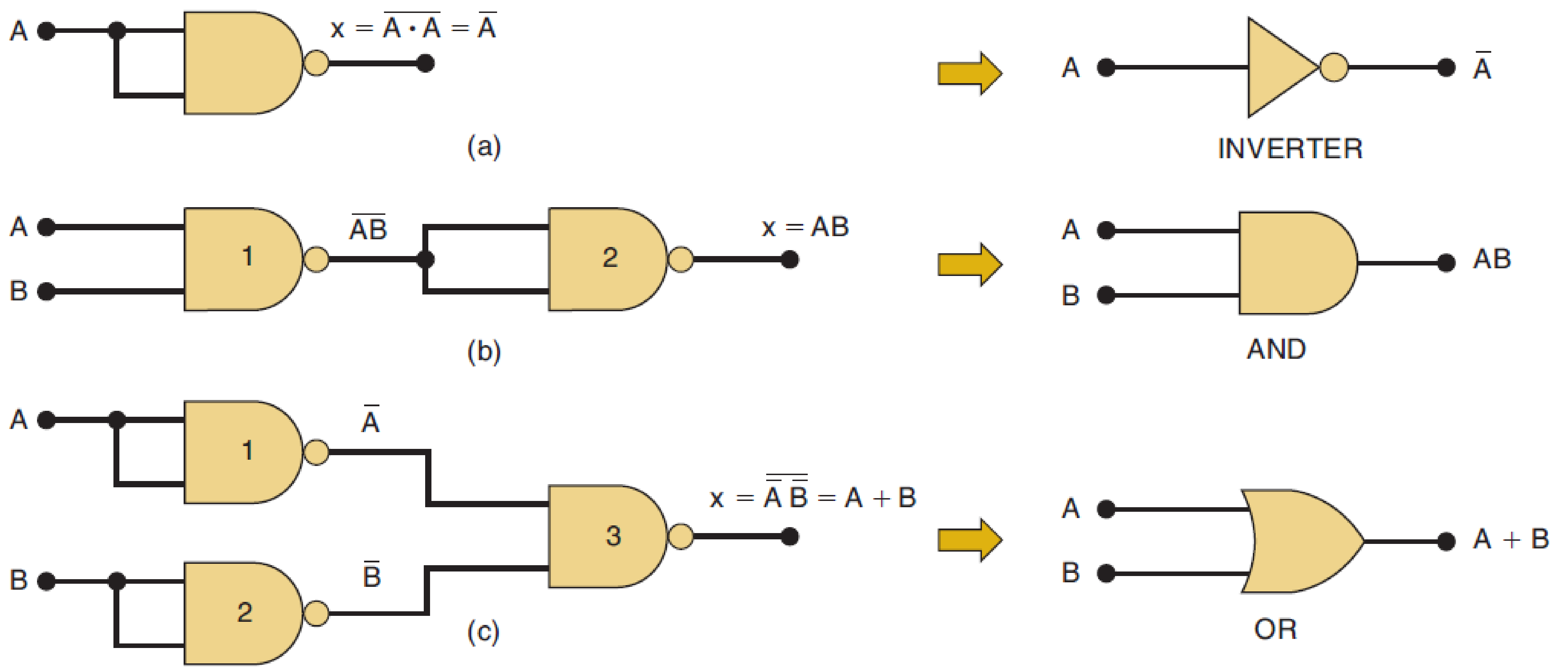
Truth table for an exclusive-NOR gate.

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	1

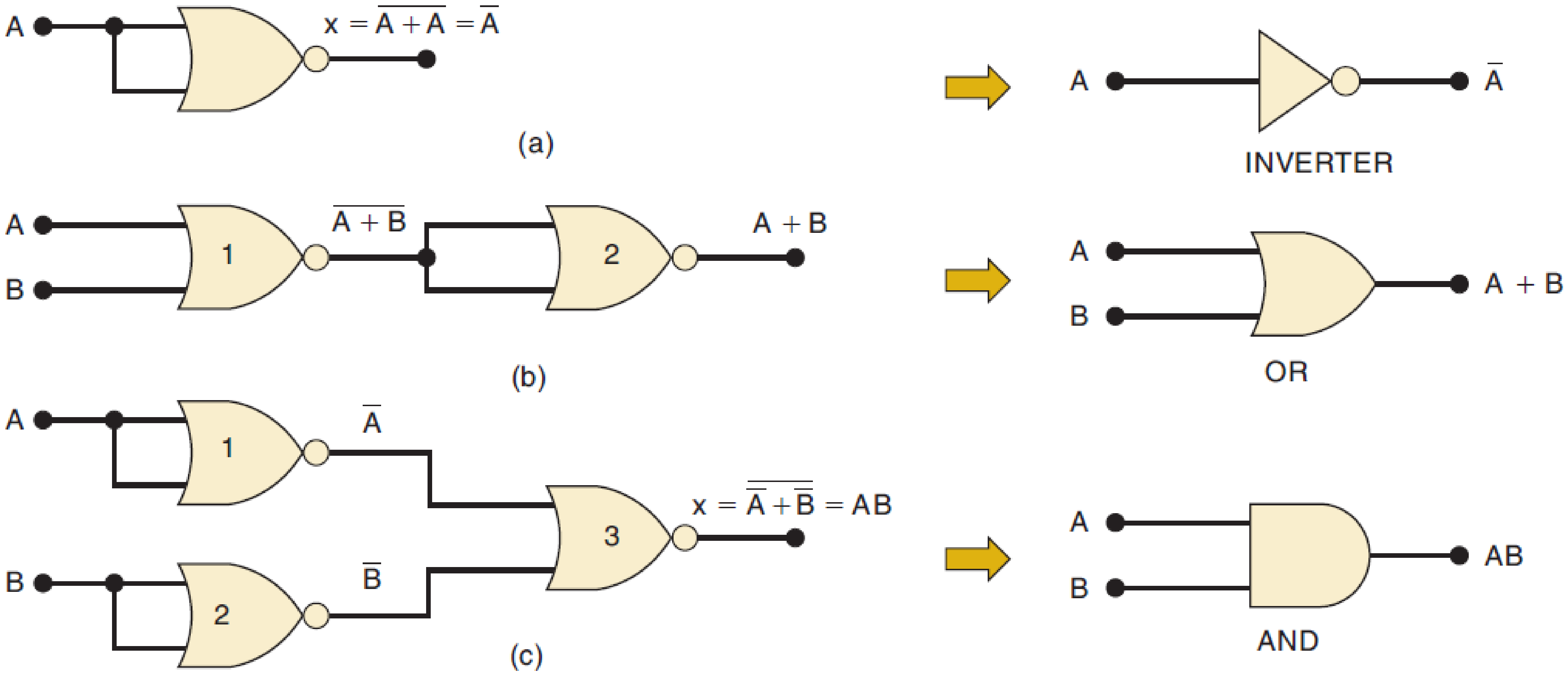




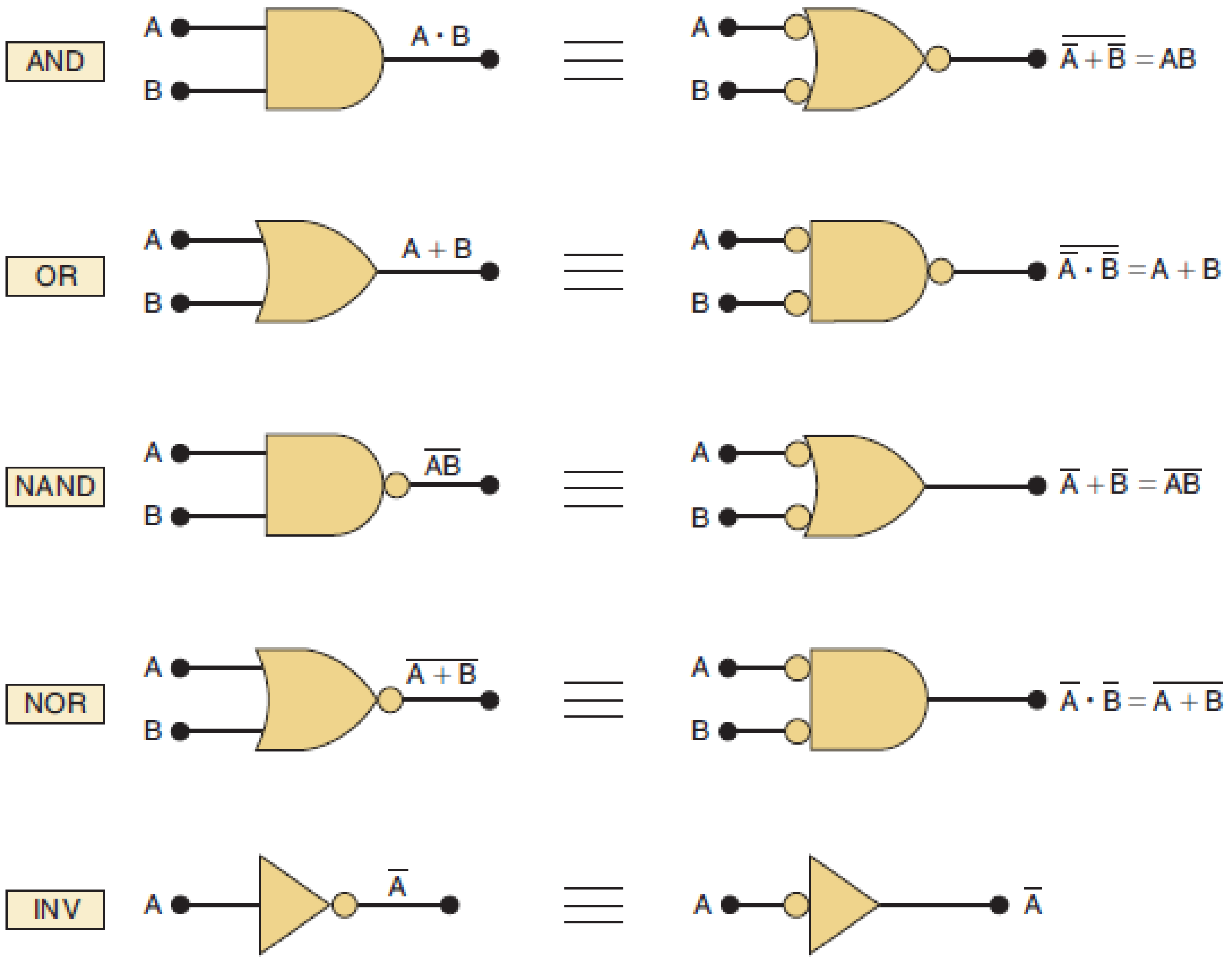
NAND as universal gate



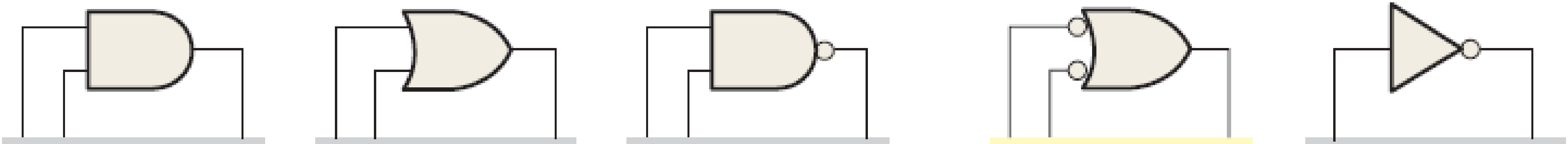
NOR as universal gate



Standard alternate symbols



Logic gates summary



AND

0	0	0
0	1	0
1	0	0
1	1	1

OR

0	0	0
0	1	1
1	0	1
1	1	1

NAND

0	0	1
0	1	1
1	0	1
1	1	0

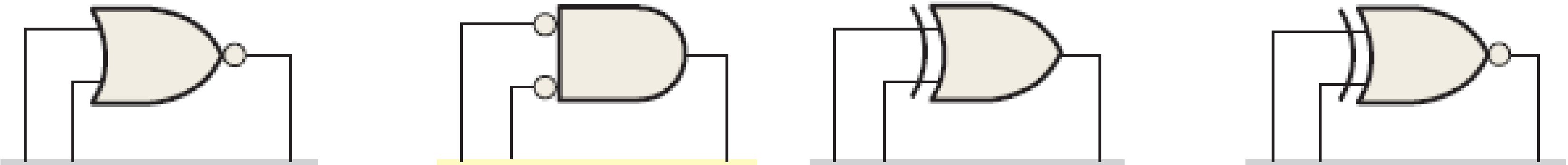
≡

Negative-OR

0	0	1
0	1	1
1	0	1
1	1	0

Inverter

0	1
1	0



NOR

0	0	1
0	1	0
1	0	0
1	1	0

≡

Negative-AND

0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR

0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-NOR

0	0	1
0	1	0
1	0	0
1	1	1

References

1. ***Digital Fundamentals*** by Thomas Floyd, Pearson International Edition, 11th Edition, Chapter 3, Page 125-170.
2. ***Digital Systems: Principles and Applications*** by Ronald Tocci, Neal Widmer and Greg Moss, Pearson International Edition, 12th Edition, Chapter 3, Page 68-90.

Next class

Boolean Algebra
And
Logic Simplification