XMUT 101
Engineering Technology

A/Prof. Pawel Dmochowski

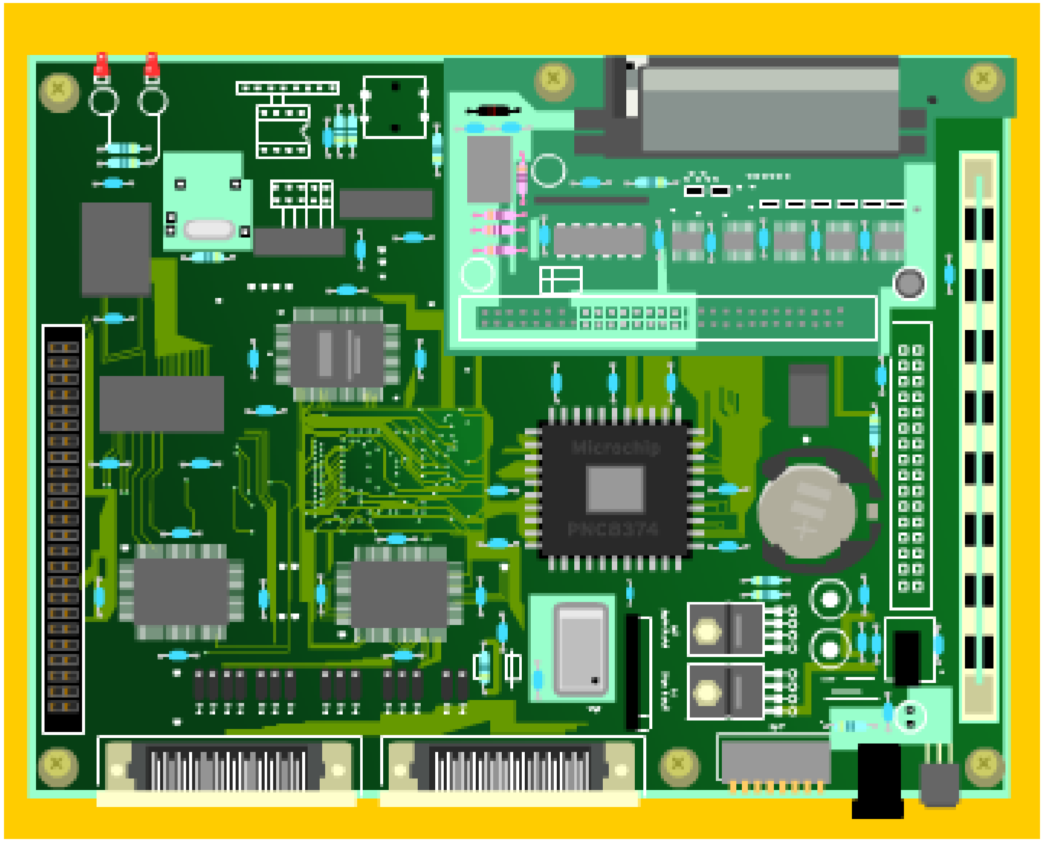
School of Engineering and Computer Science
Victoria University of Wellington

Victoria
UNIVERSITY OF WELLINGTON
Te Whare Wānanga
o te Ūpoko o te Ika a Māui

CAPITAL CITY UNIVERSITY

What are Logic Gates?

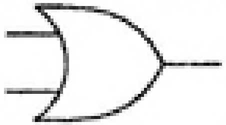
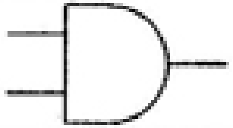
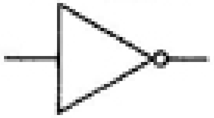
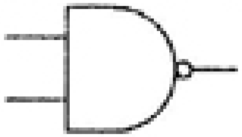
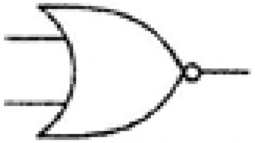
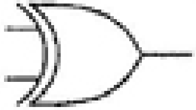
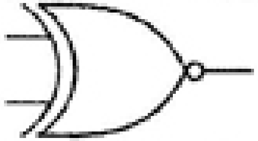
- Basic building blocks of a digital circuit



What are Logic Gates?

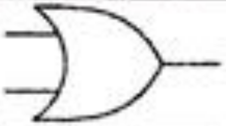
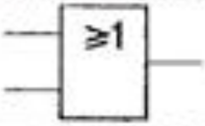
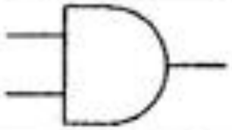
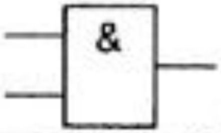
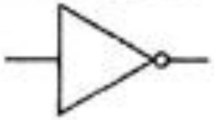
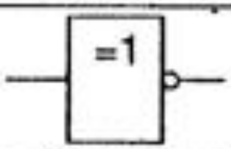
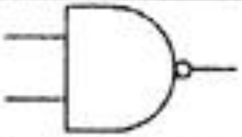
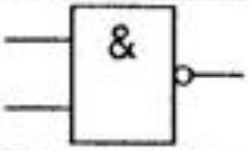
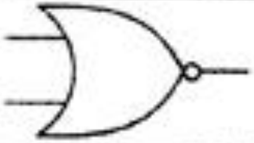
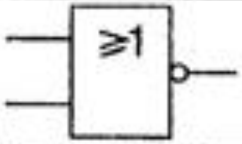
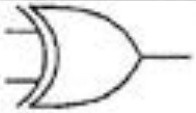
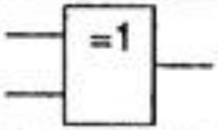
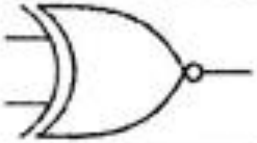
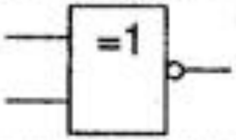
- Basic building blocks of a digital circuit
- Data processing on the circuit is controlled using transistors
- Output depends on the logic gate and the input
- Input is one of two states – high (1) or low (0)
- Output is one of two states – high (1) or low (0)
- There are seven types of logic gates:
  - 3 basic types: AND, OR, NOT
  - NAND, NOR, XOR, XNOR

Logic Gates Symbols

Gate	Symbol
OR	
AND	
NOT	
NAND	
NOR	
EX-OR or X-OR	
EX-NOR or X-NOR	

Logic Gates Symbols

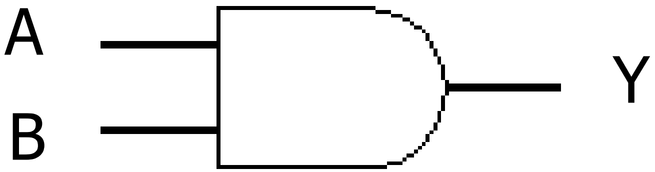
Another symbol

Gate	Symbol	ICE
OR		
AND		
NOT		
NAND		
NOR		
EX-OR or X-OR		
EX-NOR or X-NOR		

Three Basic Logic Gates

1. AND gate
2. OR gate
3. NOT gate (also known as Inverter)

The AND Gate



- If both of the 2 input signals are High, the output will also be High.

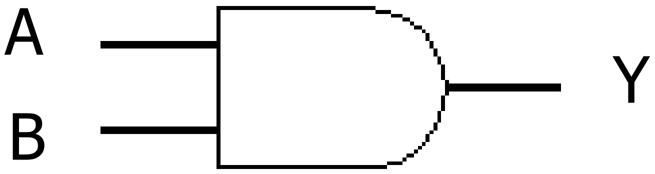
- If 1 or 2 of the input signals are Low, the output will also be Low.

Truth Tables

- A truth table describes (or shows) the relationship between the input(s) and output of a logic circuit.

- The number of entries corresponds to the number of inputs.
  - A 2-inputs table would have $2^2 = 4$ entries.
  - A 3-inputs table would have $2^3 = 8$ entries.
  - A 4-inputs table would have $2^4 = 16$ entries.

The AND Gate



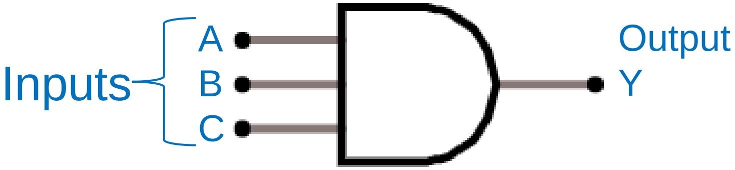
- If the 2 input signals are High, the output will also be High.
- If 1 or 2 of the input signals are Low, the output will also be Low.

Truth Table

Inputs

A	B	Y (Output)
0	0	0
0	1	0
1	0	0
1	1	1

The AND Gate – Exercise



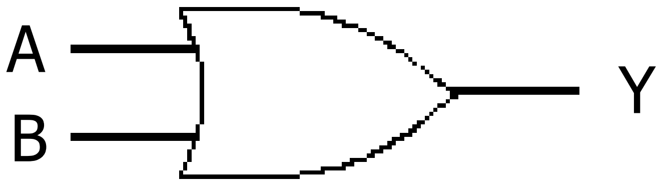
- If the 3 input signals are High, the output will also be High.
- If 1 input signal is Low, the output will also be Low.

Truth Table

Inputs

A	B	C	Y (Output)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

The OR Gate



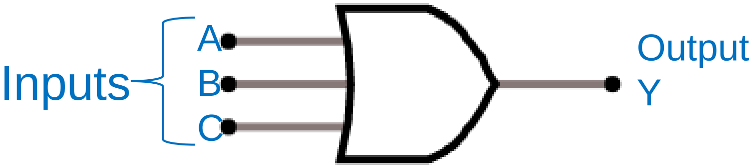
- If either of the two input signals are High (= 1), or both of them are High, the output will be High.

Truth Table

Inputs

A	B	Y
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

The OR Gate – Exercise



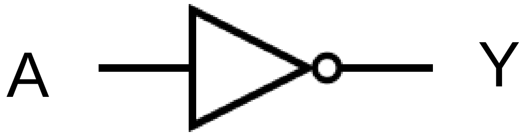
- If at least one of the inputs is High, the output will be High.

Truth Table

Inputs

A	B	C	Y (Output)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

The NOT Gate (also known as the Inverter)



- The output is the opposite of the input signal.

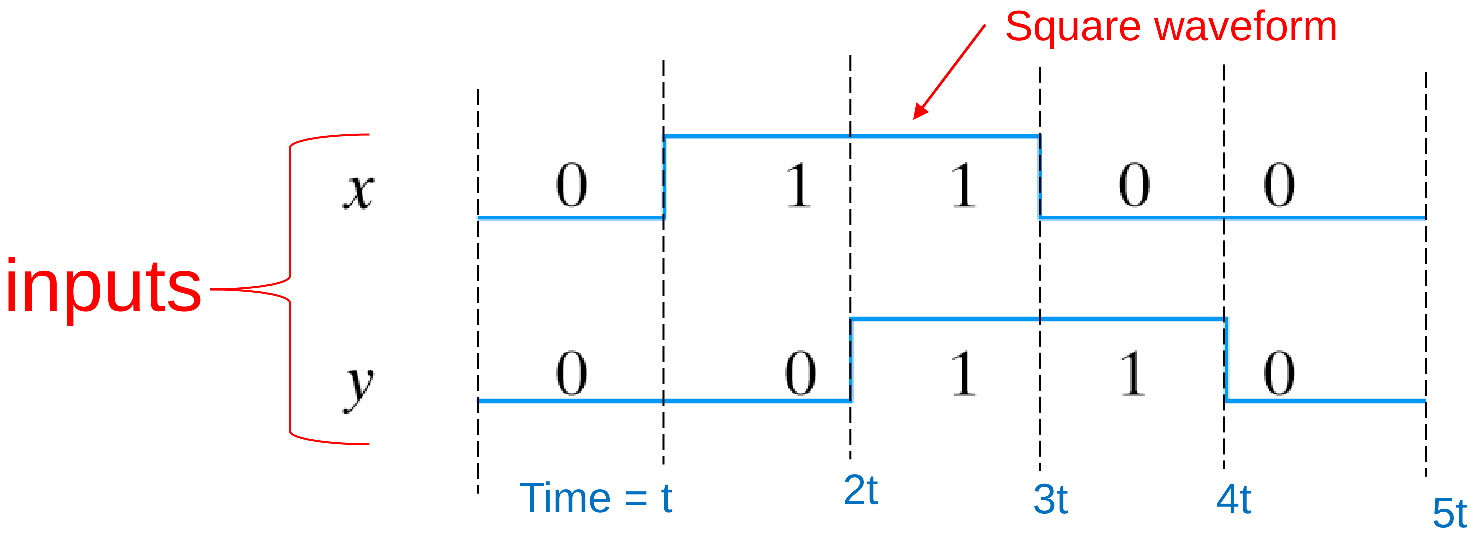
Truth Table

A	Y
0	**1**
1	**0**

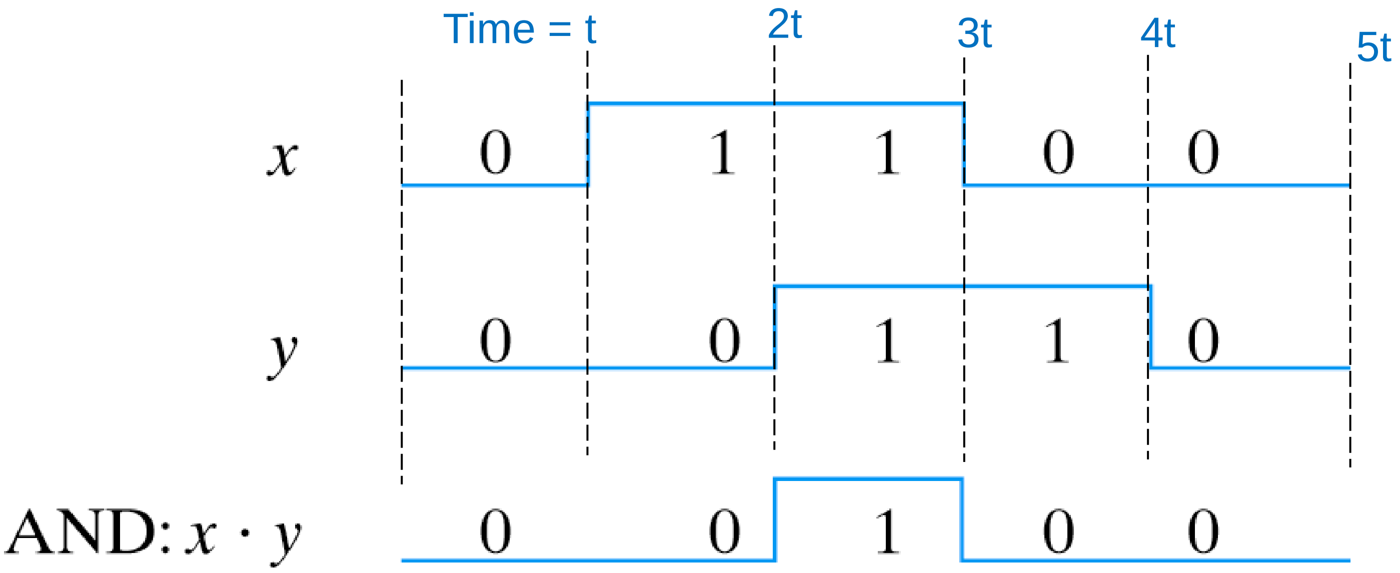
Describing Circuit Functionality: Waveforms

- Waveforms provide another approach for representing functionality.
- Values are either high (logic 1) or low (logic 0).

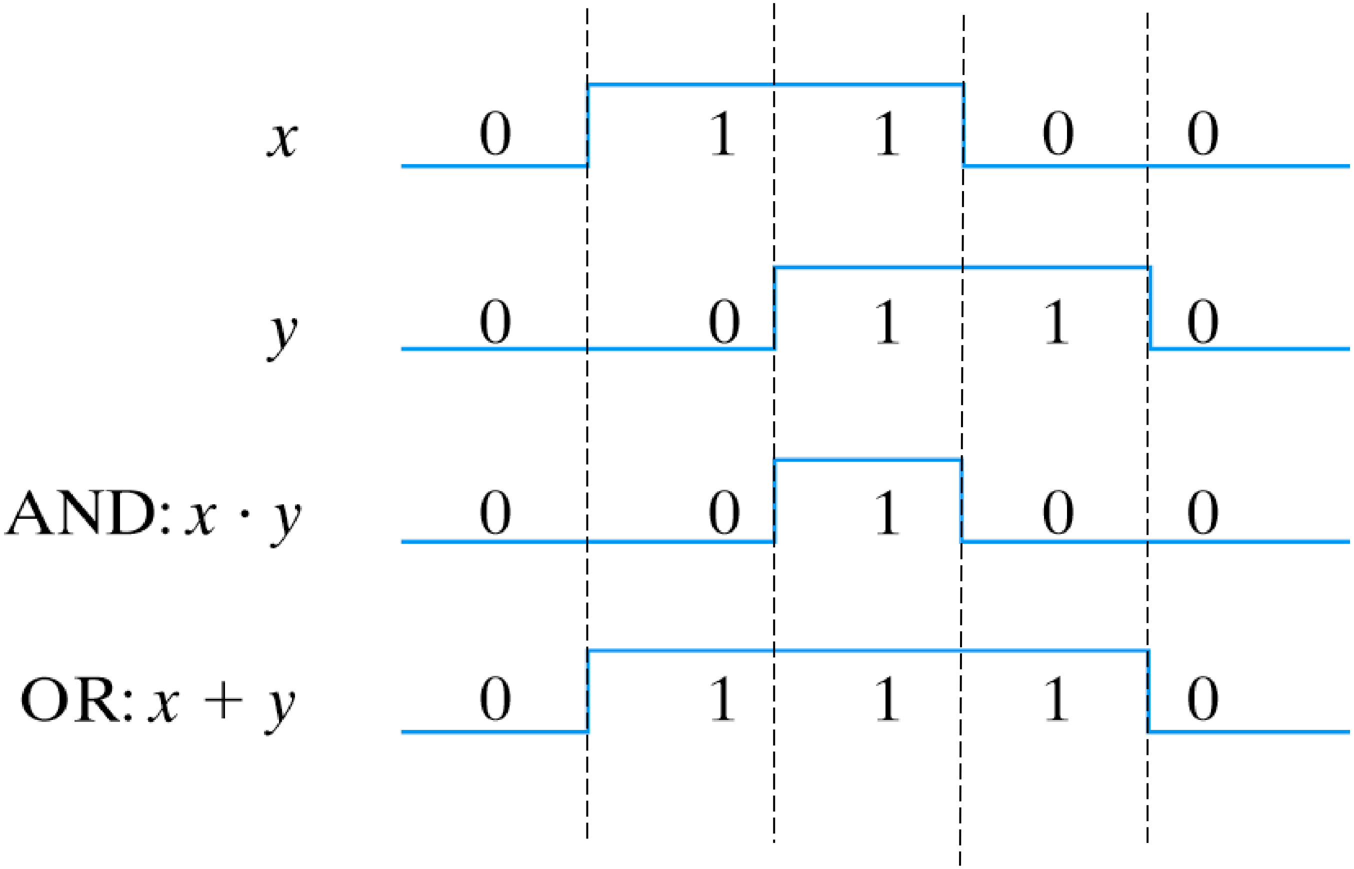
Describing Circuit Functionality: Waveforms



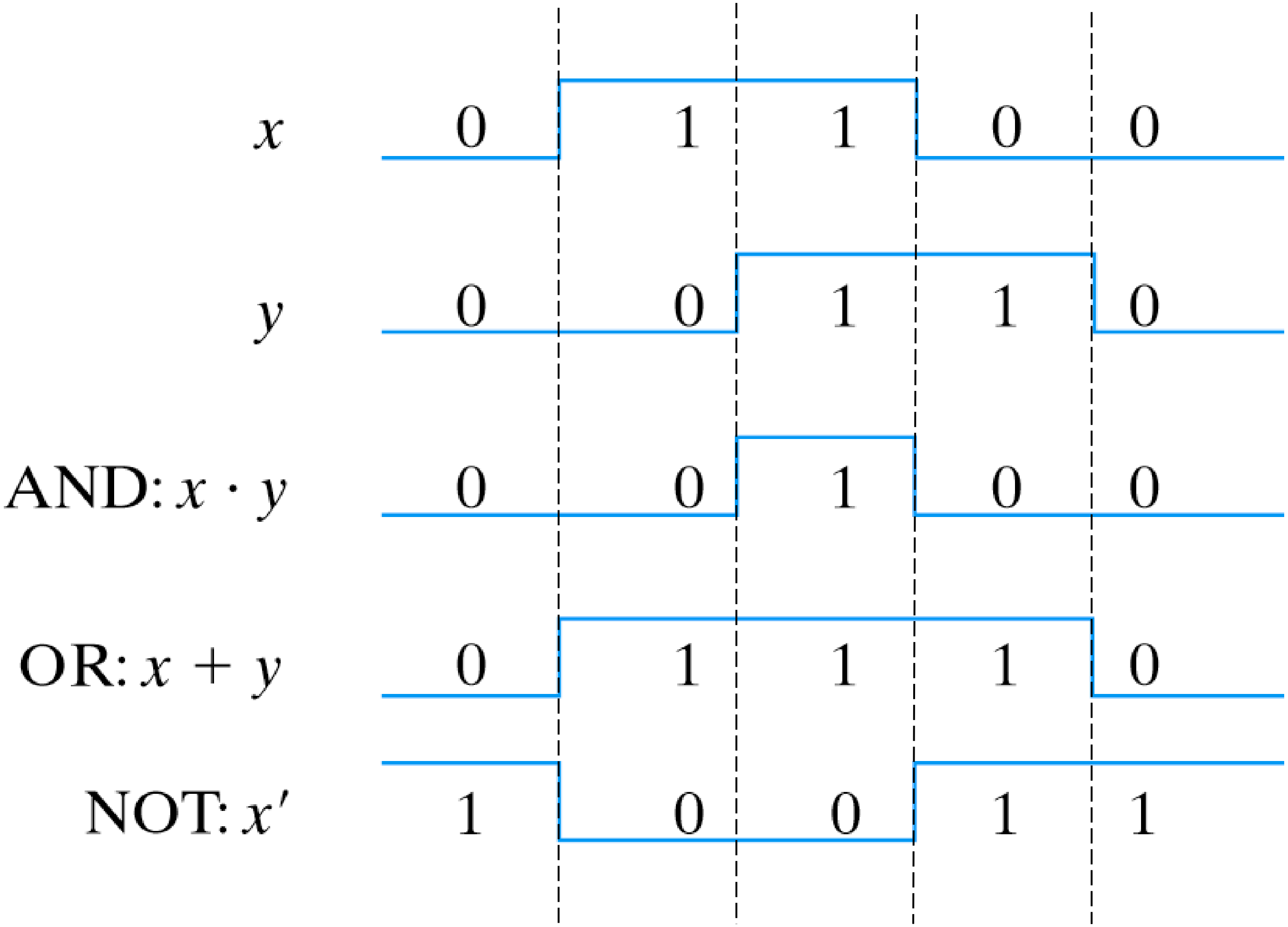
Describing Circuit Functionality: Waveforms



Describing Circuit Functionality: Waveforms

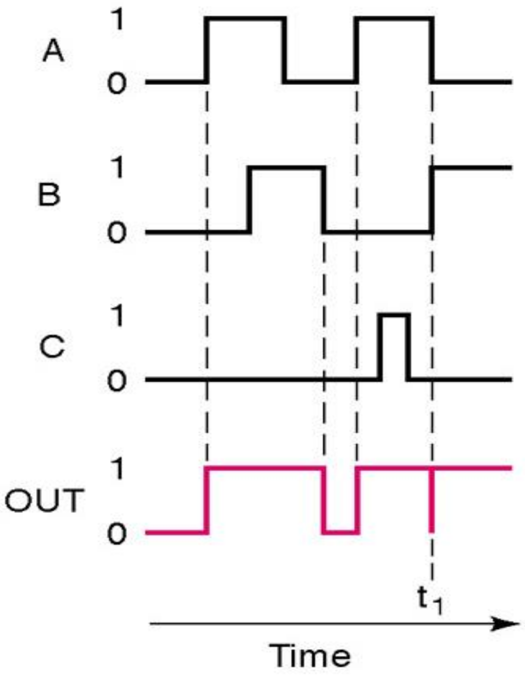


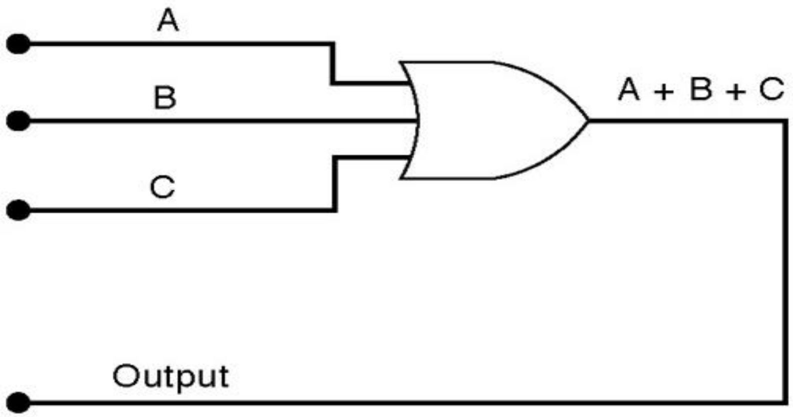
Describing Circuit Functionality: Waveforms



Consider three-input gates

3 Inputs OR Gate

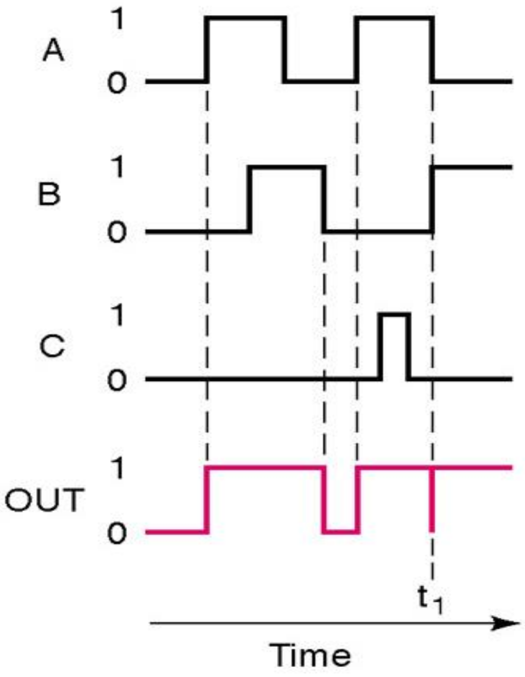


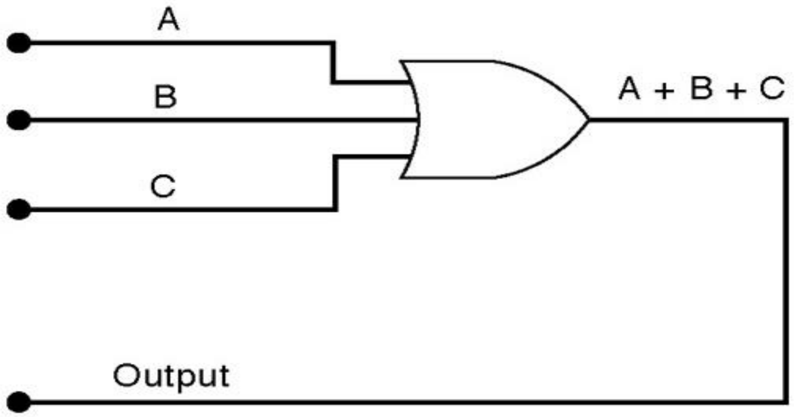


Waveform diagram

Logic symbol

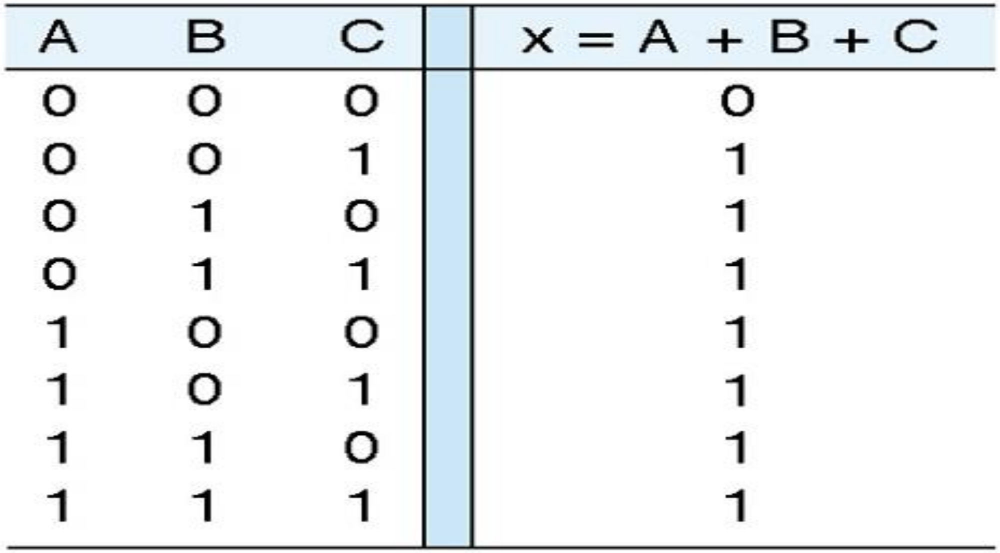
Consider three-input gates





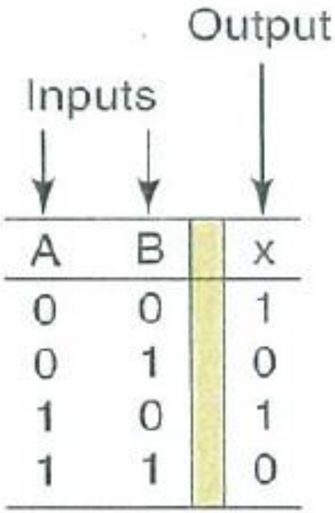
3 Input OR Gate

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

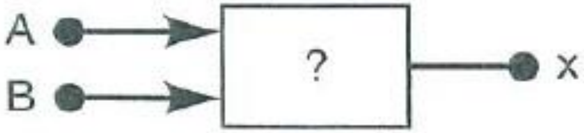
Consider many inputs!



Output

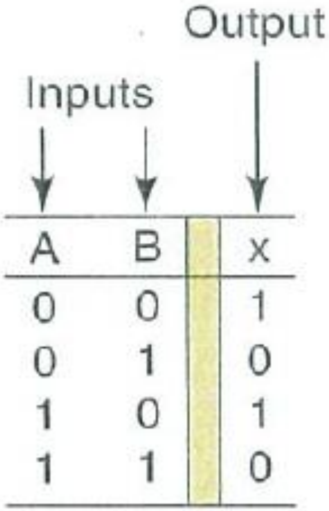
Inputs

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0



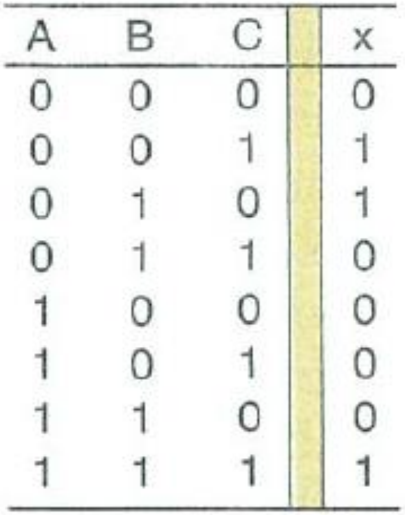
(a)

Consider many inputs!



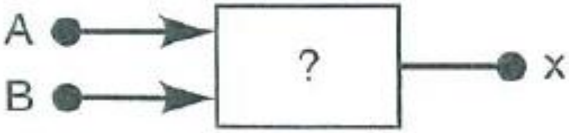
Output
Inputs

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0



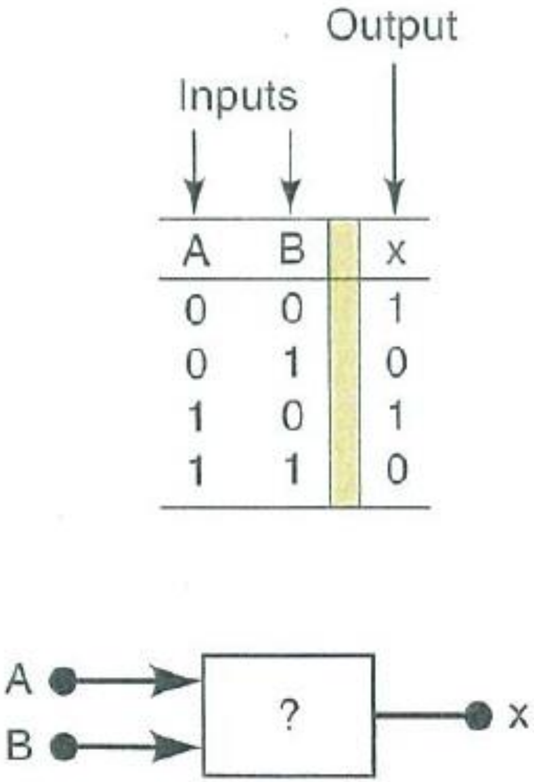
A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)



(a)

Consider many inputs!

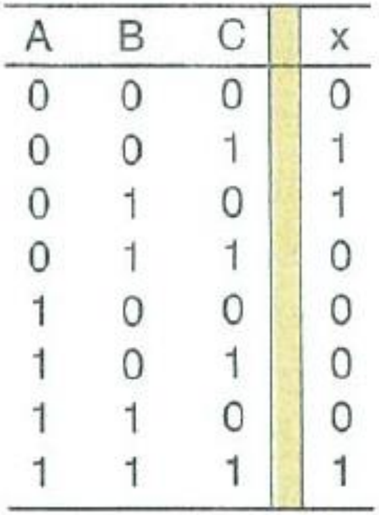


Output

Inputs

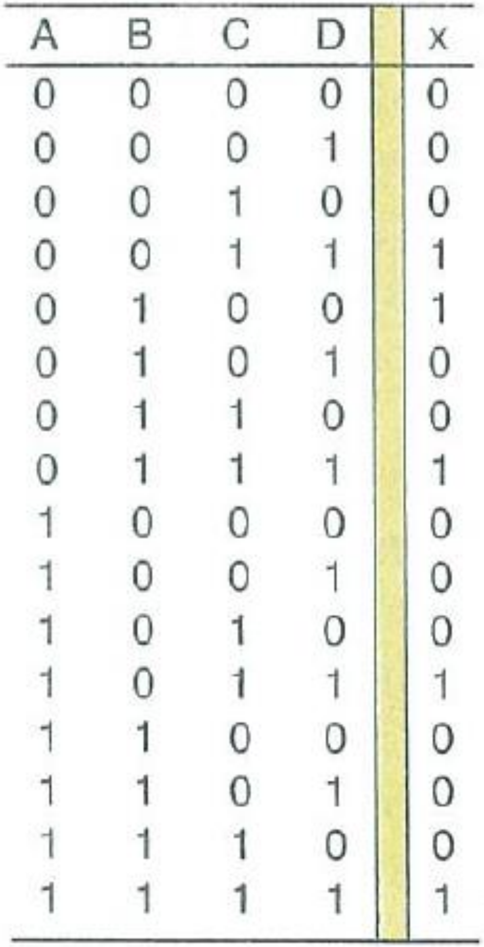
A	B	x
0	0	1
0	1	0
1	0	1
1	1	0

(a)



A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)

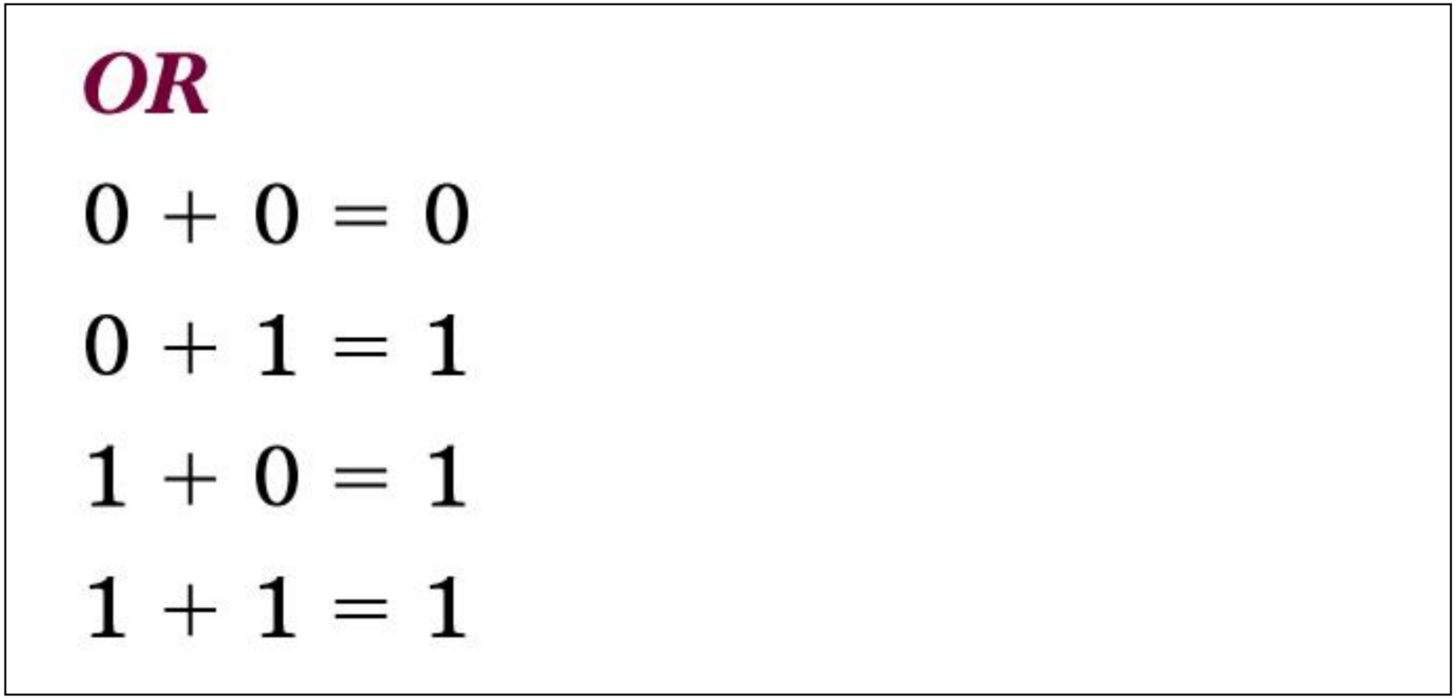


A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

Boolean Operators

Summarized rules for OR



OR

$$0 + 0 = 0$$

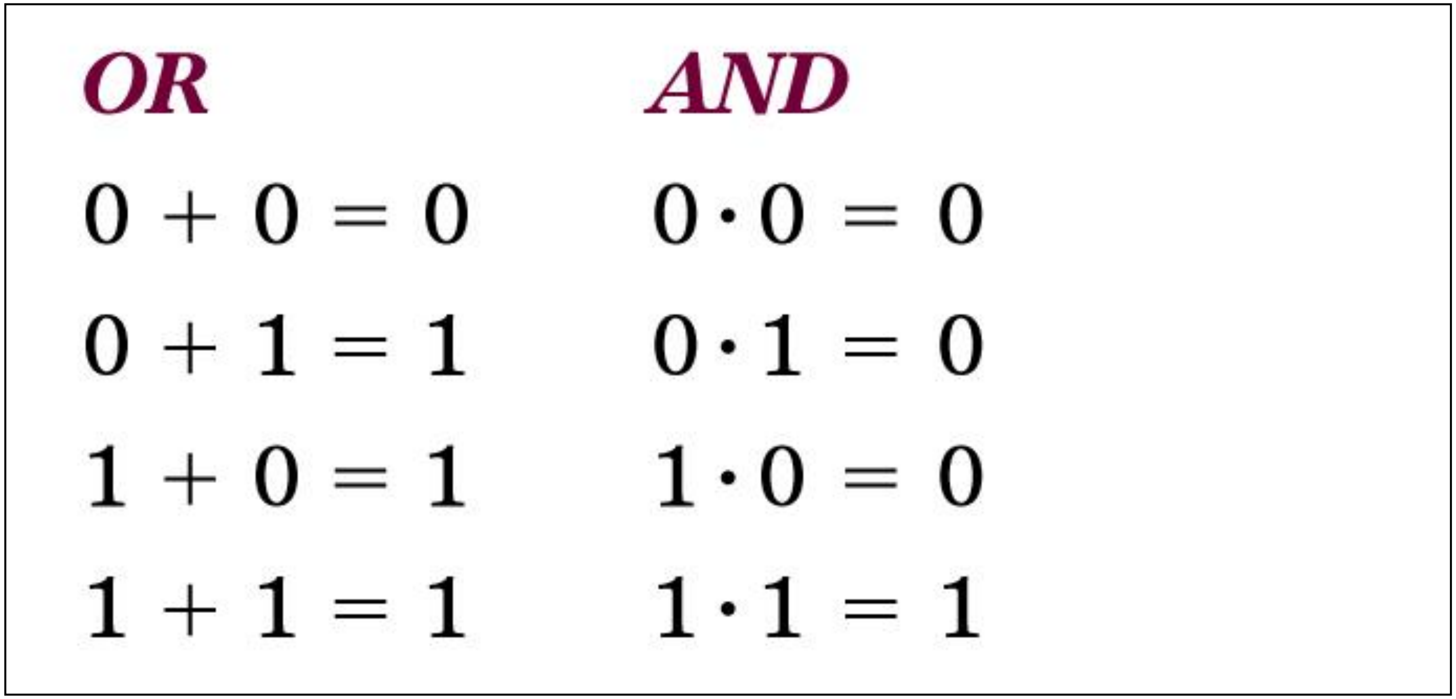
$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

Boolean Operators

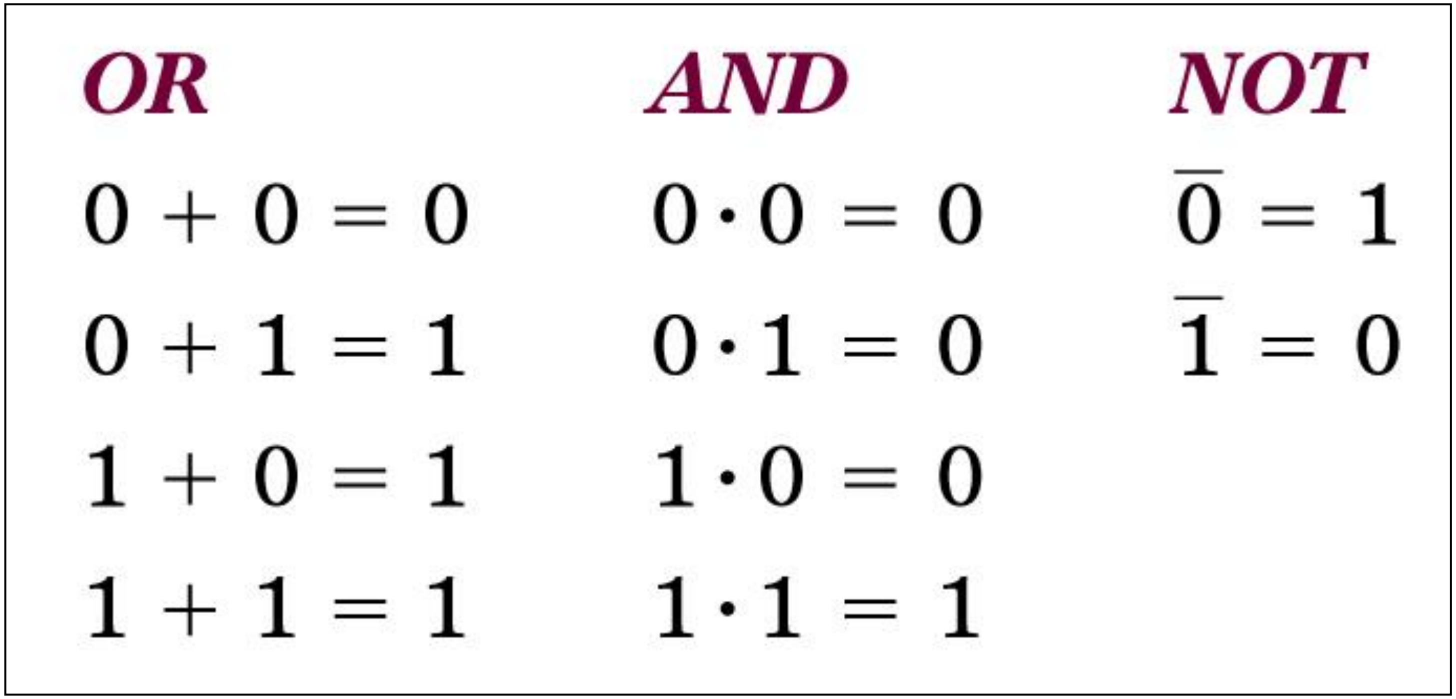
Summarized rules for OR and AND



OR	*AND*
$0 + 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$1 + 1 = 1$	$1 \cdot 1 = 1$

Boolean Operations

Summarized rules for **OR**, **AND** and **NOT**

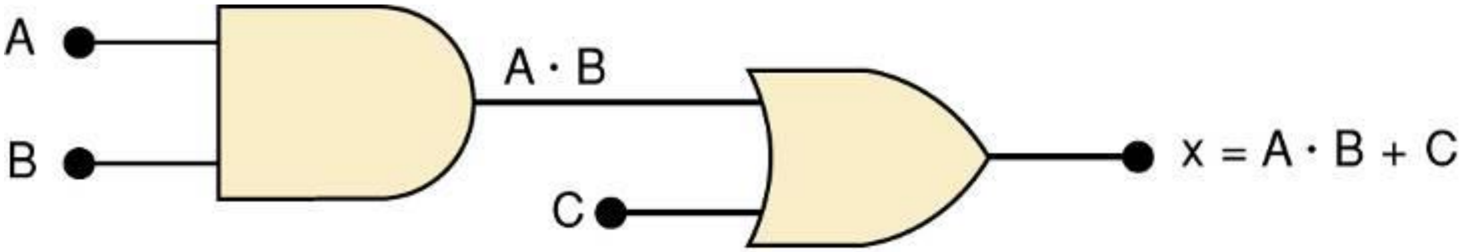


OR	*AND*	*NOT*
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\overline{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\overline{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

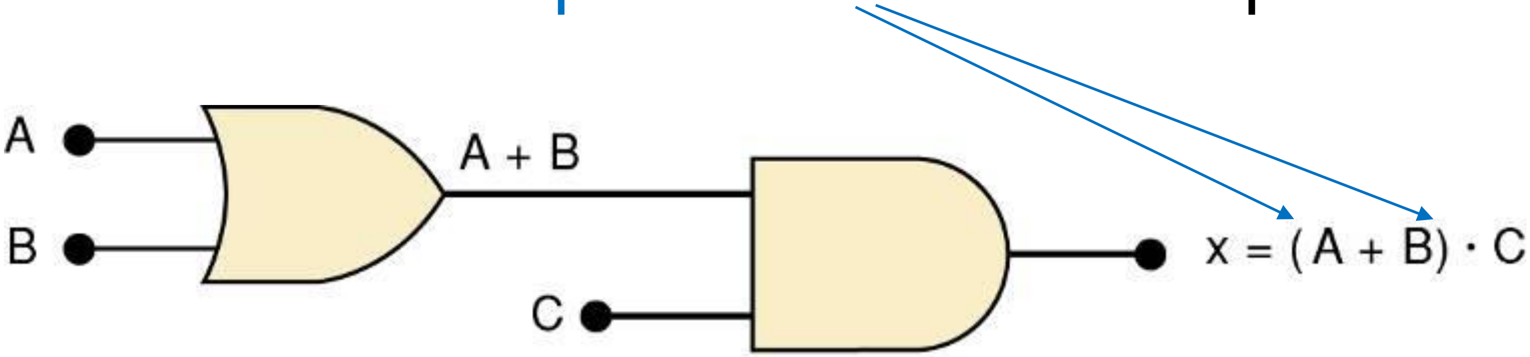
These three basic Boolean operations can describe any logic circuit.

Describing Logic Circuits Algebraically

- If an expression contains both **AND** and **OR** gates, the **AND** operation will be performed first.

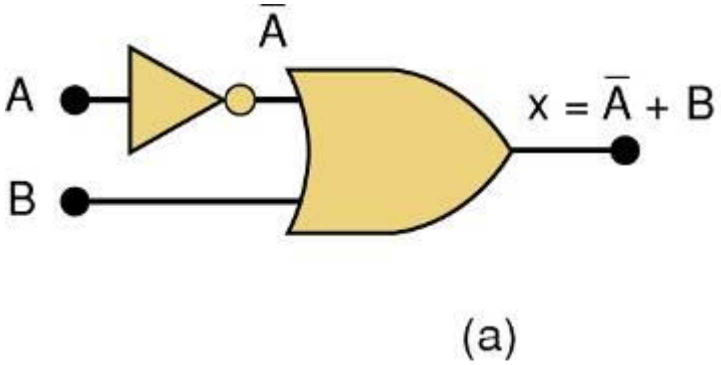


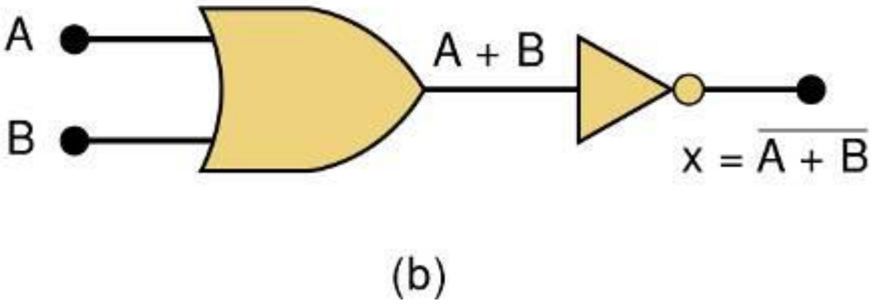
- Unless there is a parenthesis in the expression.



Describing Logic Circuits Algebraically

- Whenever an INVERTER is present, output is equivalent to input, with a bar over it.
  - Input *A* through an inverter equals $\overline{A}$.

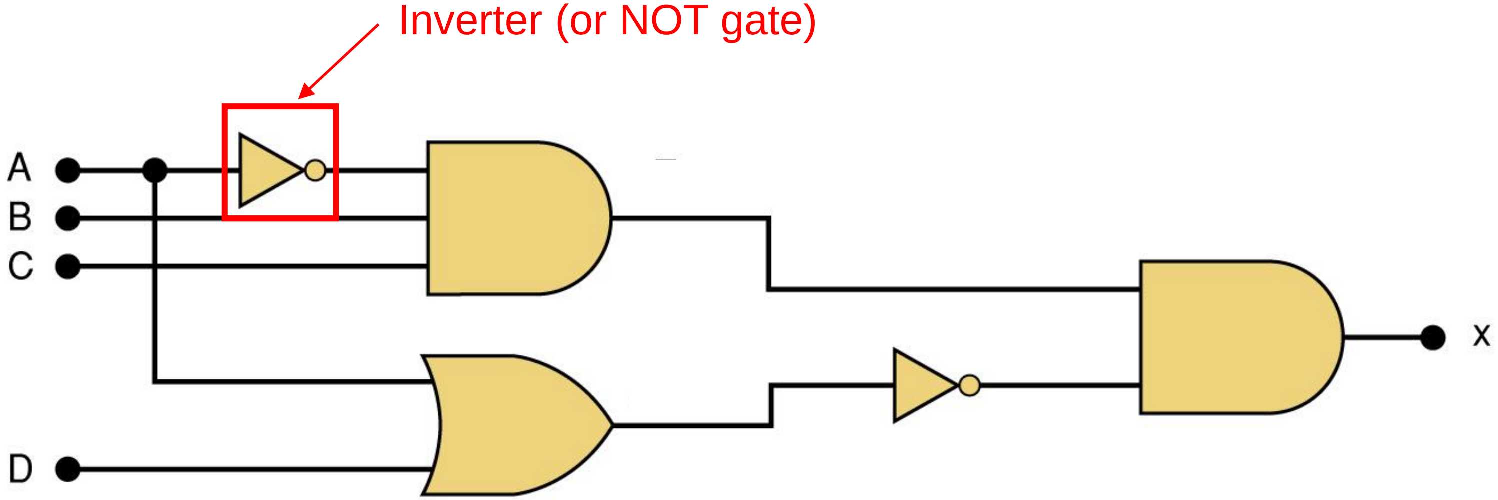




Describing Logic Circuits Algebraically

Example 1

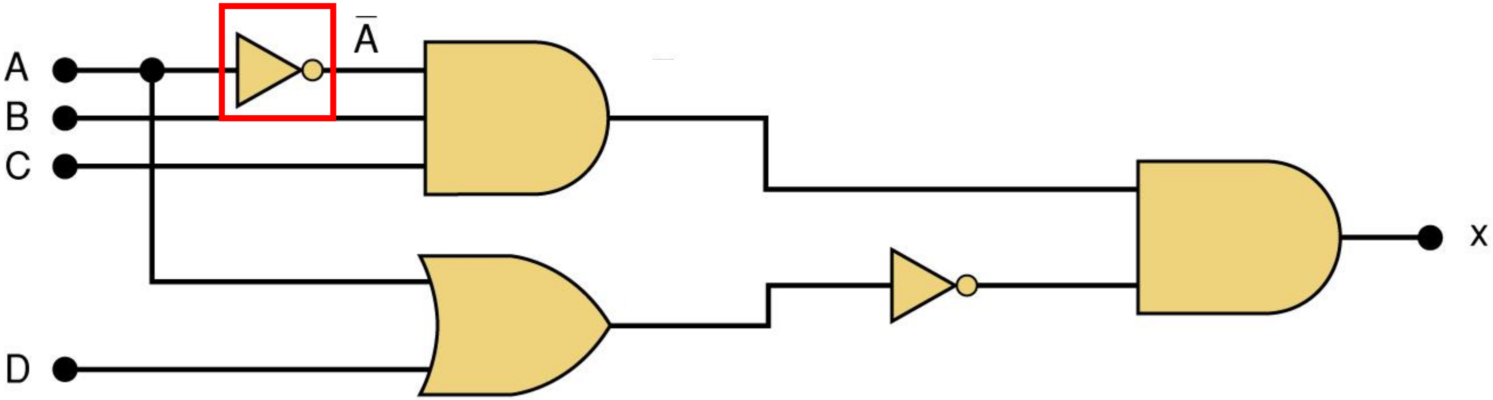
Inverter (or NOT gate)



Describing Logic Circuits Algebraically

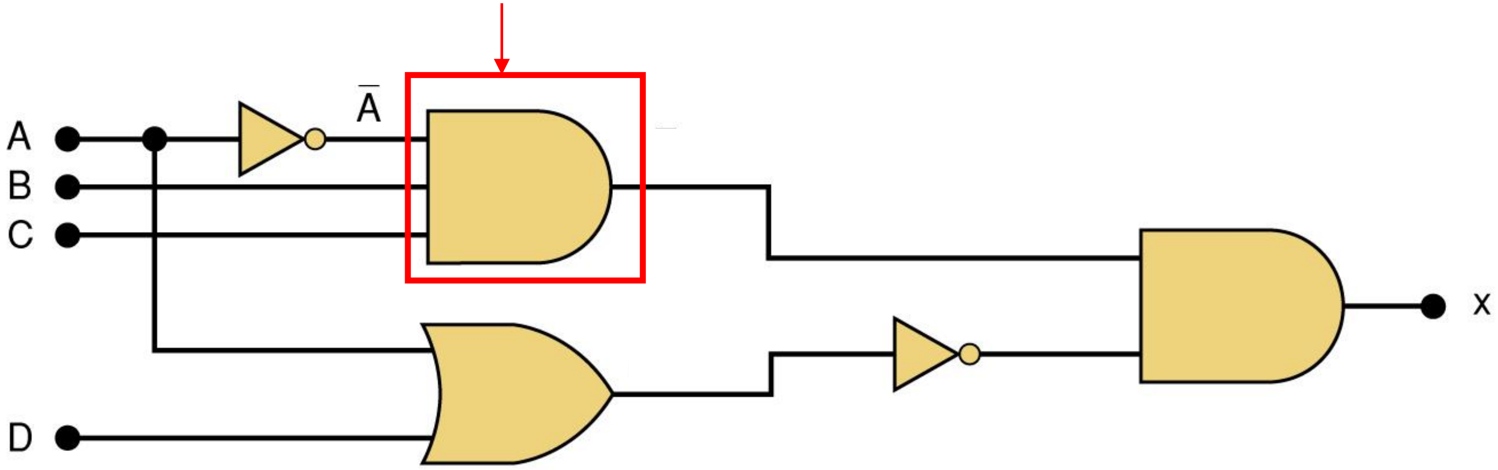
Example 1

Inverter (or NOT gate)



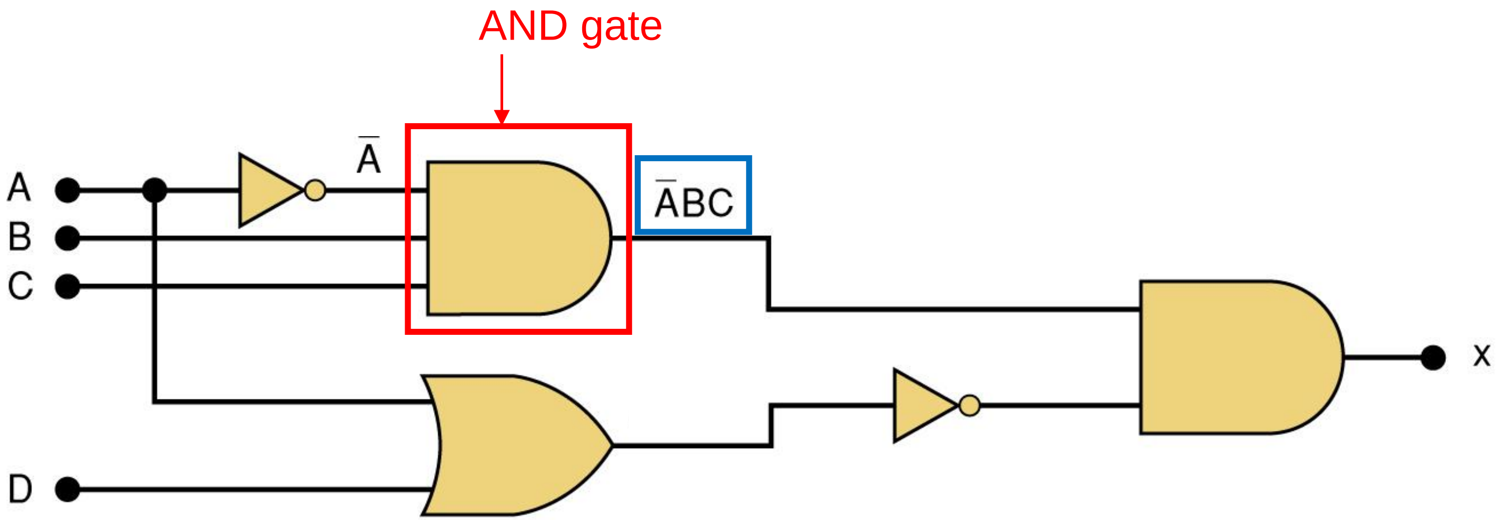
Describing Logic Circuits Algebraically

Example 1


AND gate

Describing Logic Circuits Algebraically

Example 1

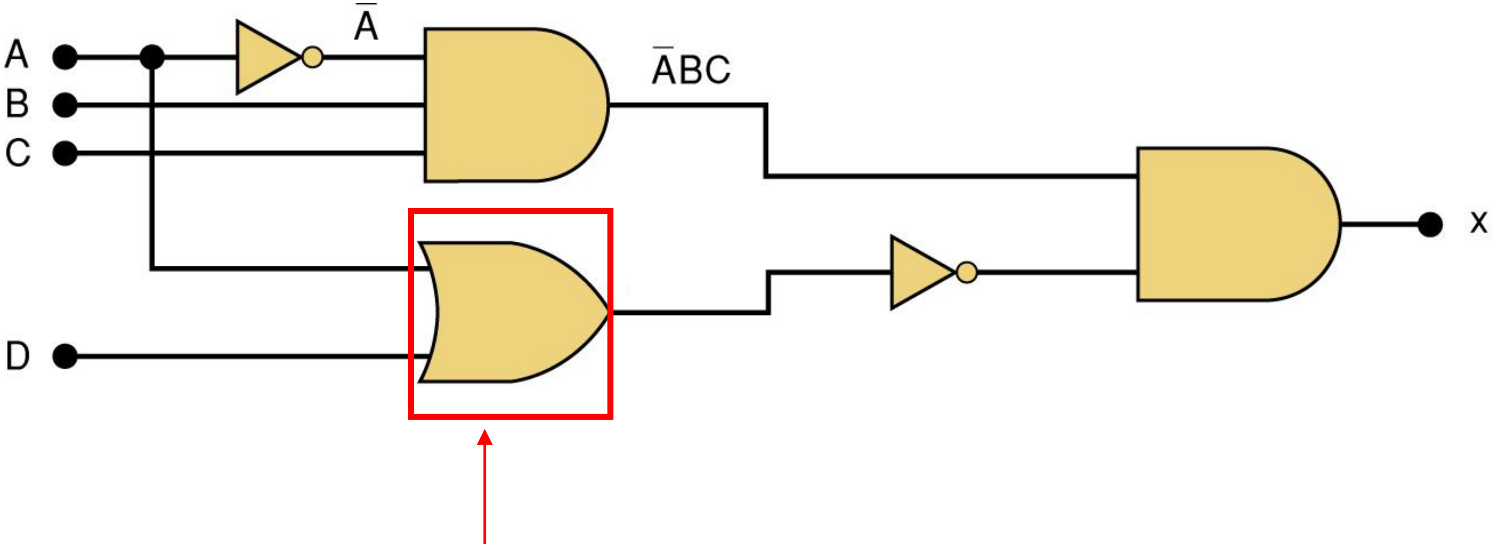


AND gate

$\overline{A}BC$

Describing Logic Circuits Algebraically

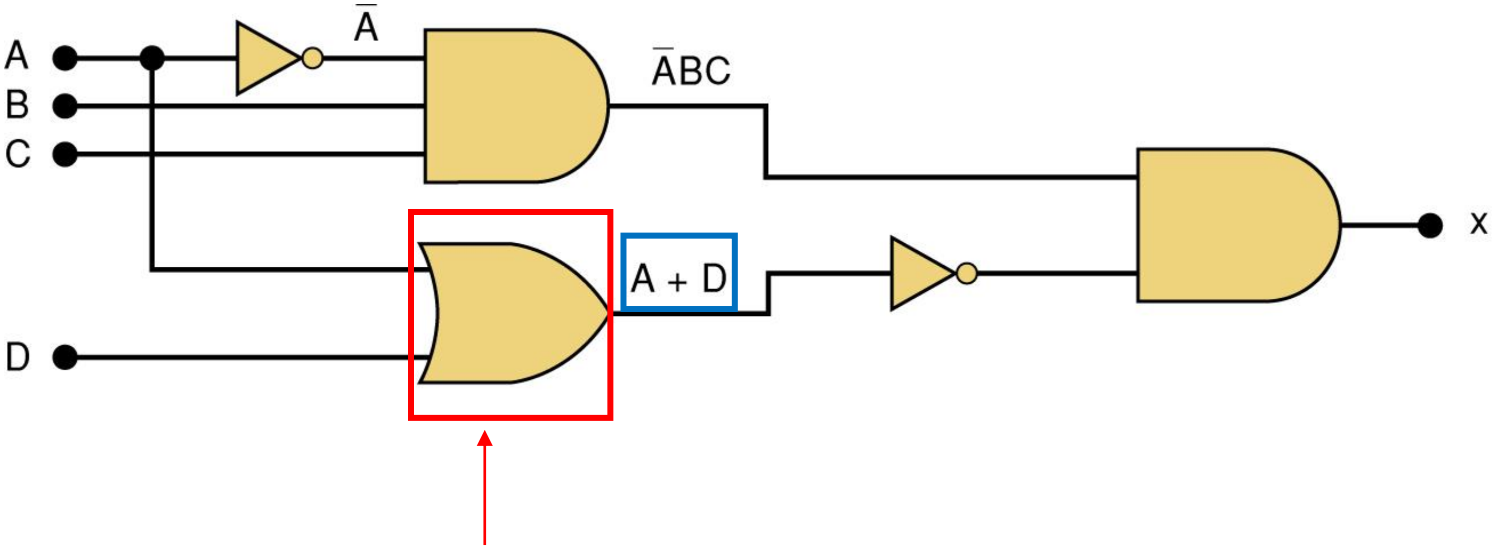
Example 1



OR gate

Describing Logic Circuits Algebraically

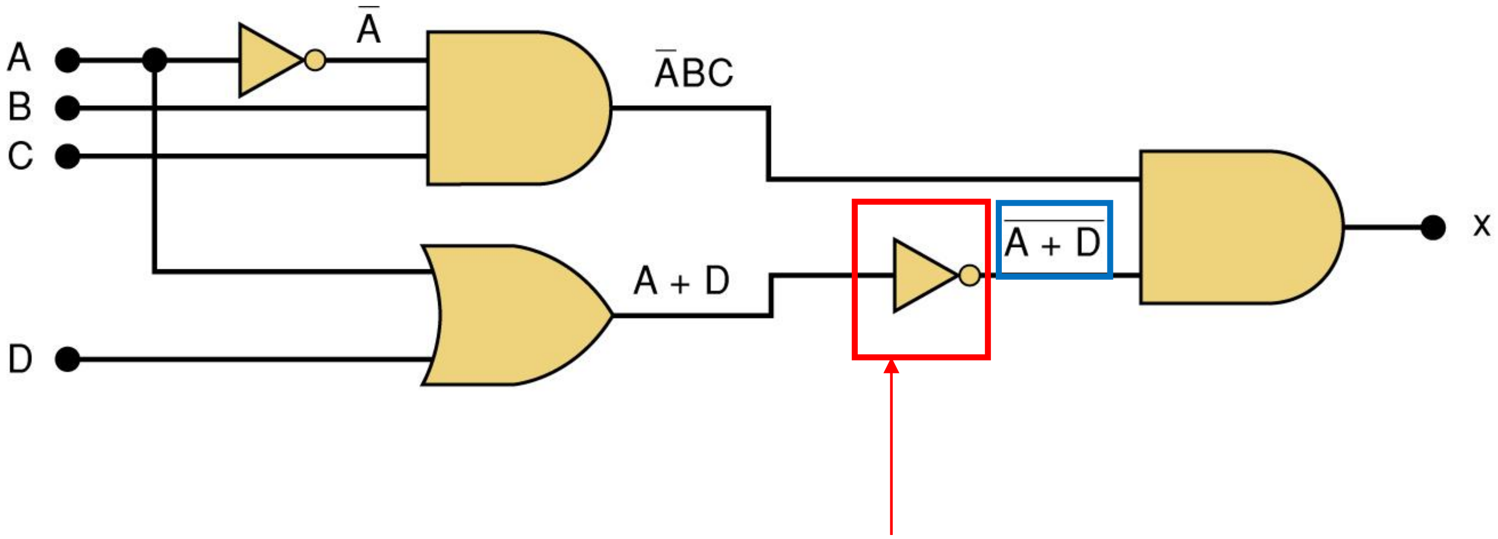
Example 1



OR gate

Describing Logic Circuits Algebraically

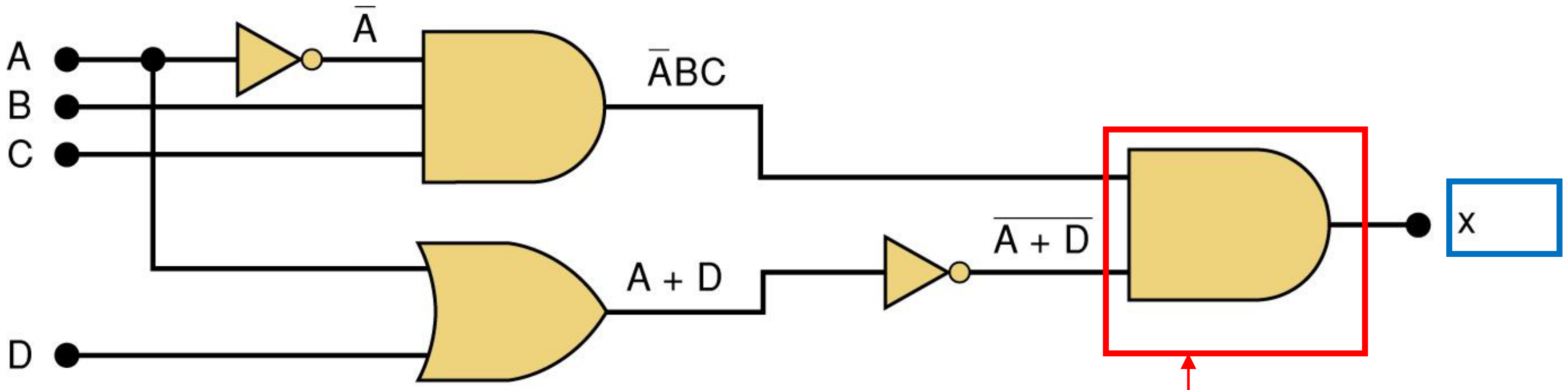
Example 1



Inverter (NOT gate)

Describing Logic Circuits Algebraically

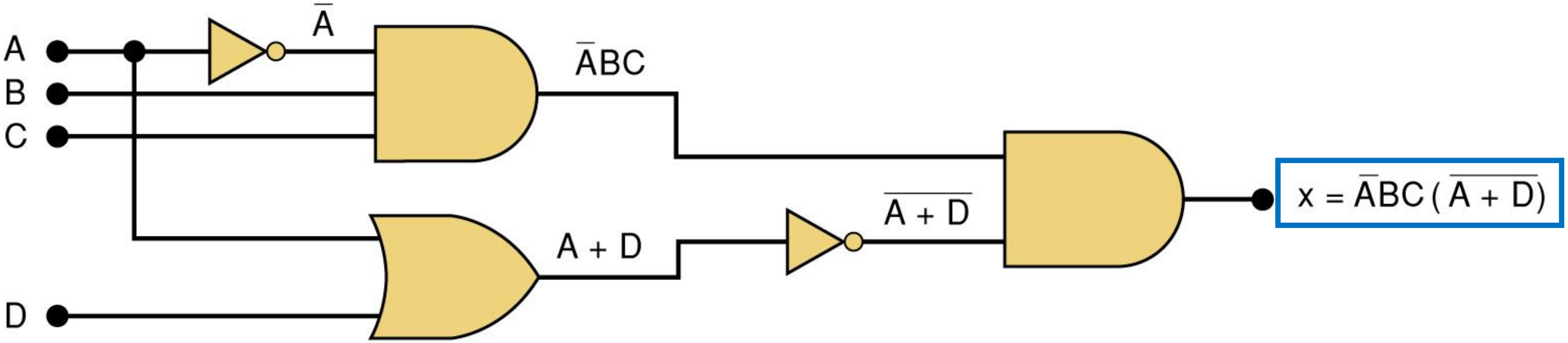
Example 1



AND gate

Describing Logic Circuits Algebraically

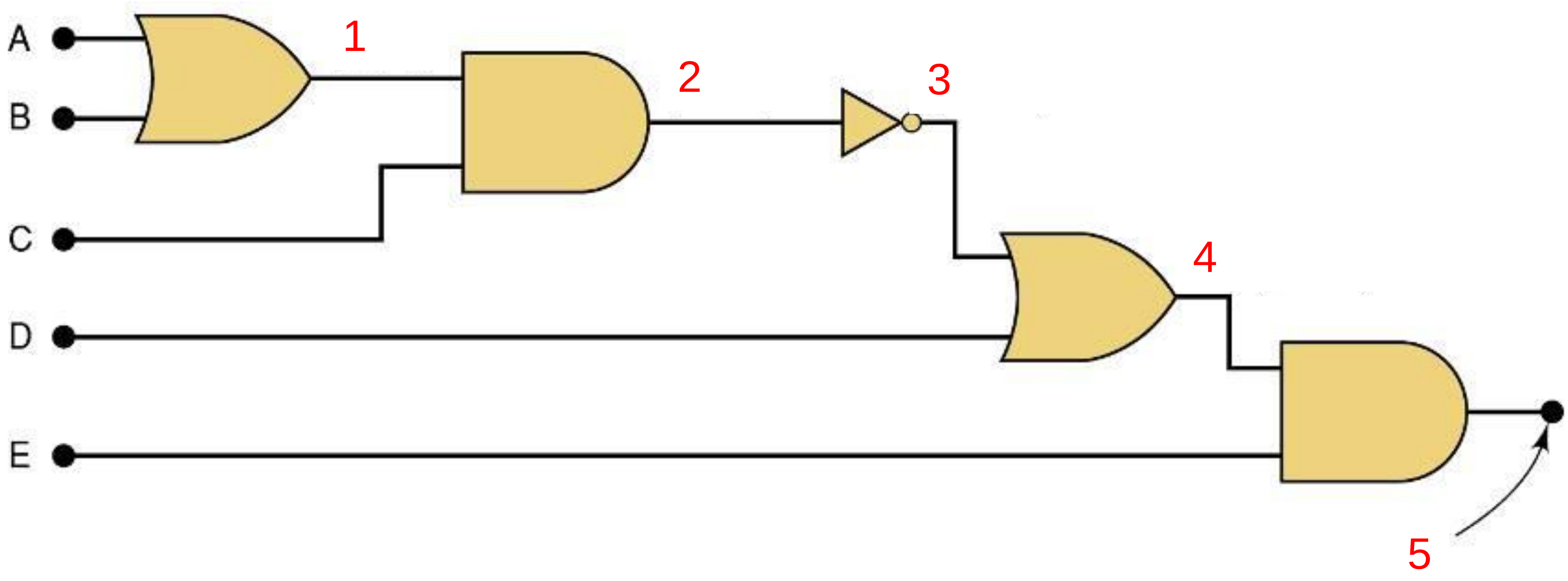
Example 1



$$x = \bar{A}BC(\overline{A + D})$$

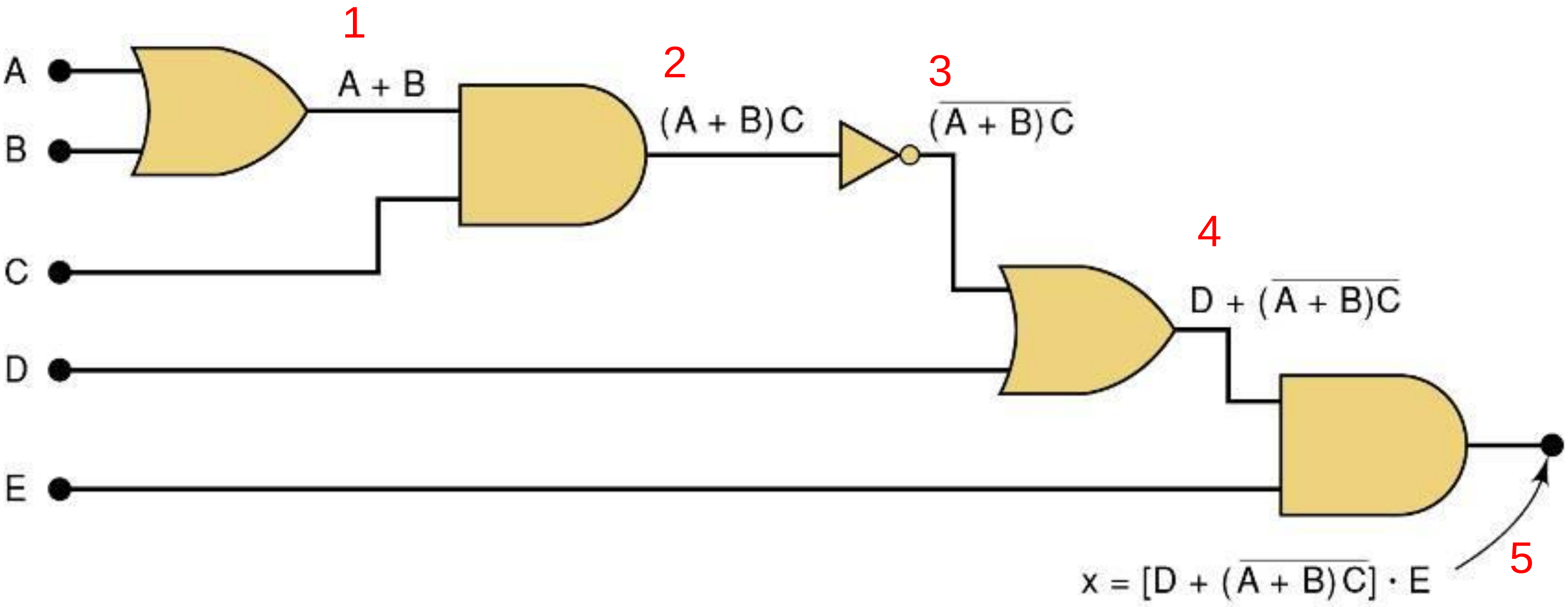
Describing Logic Circuits Algebraically

Exercise



Describing Logic Circuits Algebraically

Exercise - Solution



A
B
C
D
E
1
A + B
2
(A + B)C
3
$\overline{(A + B)C}$
4
$D + \overline{(A + B)C}$
5
$x = [D + \overline{(A + B)C}] \cdot E$