

ENGR 101

Engineering Technology

Dr. [Kerese Manueli](#)

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

Week 3 Lecture 5a

- Main topics
 - Introduction to Engineering Technology
 - Number system
 - Logic Gates
 - Boolean Algebra
- Course web page:
https://ecs.wgtn.ac.nz/Courses/XMUT101_2021T1/
- kerese@ecs.vuw.ac.nz



→ XMUT101 home

Course Outline

Lecture Schedule

Assignments and Labs

Submission

Moderation

Summary

↑ [School of Engineering and Computer Science](#) ▶ [Courses/XMUT101_2021T1](#) ▶ [XMUT 101 Course Outline](#) ▶ [LectureSchedule](#)

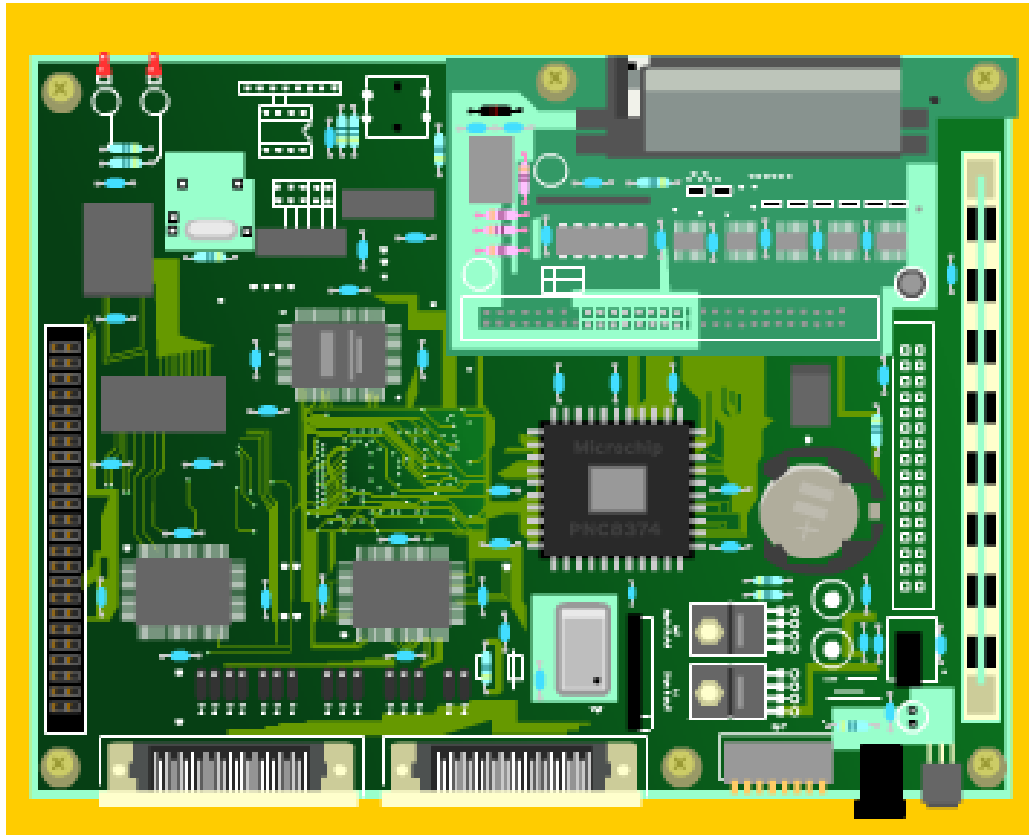
XMUT 101 Tentative Schedule

Lecture	Week 1: 1 - 7 March	Lecture Slides	Video (Zip) files
1	Introduction to the course	Wk01Lec01a Wk01Lec01b	Wk01Lec01a.zip Wk01Lec01b.zip
2	Computer architecture. Computer data.	Wk01Lec02a Wk01Lec02b	Wk01Lec02a.zip Wk01Lec02b.zip
Lecture	Week 2: 8 - 14 March		
3	Number systems: Decimal, binary, octal and hexadecimal	Wk02Lec01a Wk02Lec01b	Wk02Lec01a.zip Wk02Lec01b.zip
Lecture	Week 3: 15 - 21 March	Lecture Slides (pdf)	Video (Zip) files
4	Number system conversion	Wk03Lec04a Wk03Lec04b	Wk03Lec04a.zip Wk03Lec04b.zip
5	Logic Gates		

Week 3

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

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

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)


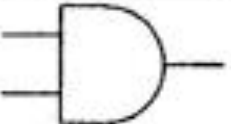
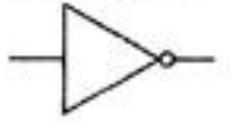
What are Logic Gates?

- 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


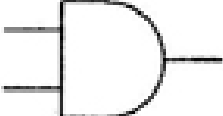
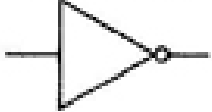
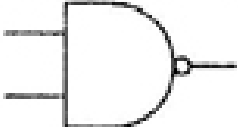



What are Logic Gates?

- 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

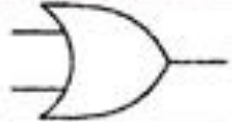

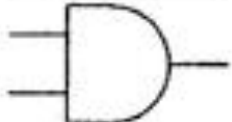
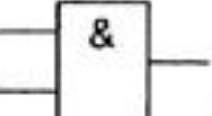
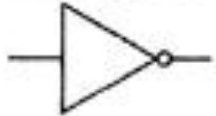
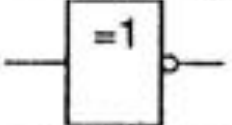

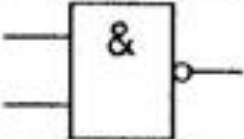

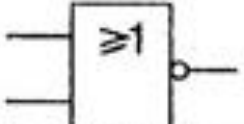

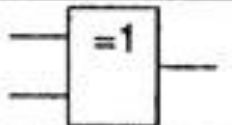

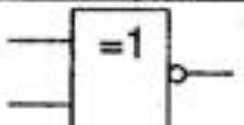
Gate	Symbol
OR	
AND	
NOT	

Logic Gates Symbols

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

Logic Gates Symbols

Symbol in your textbook

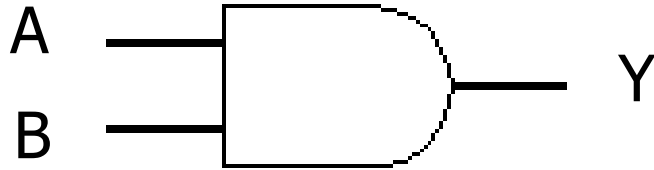
Gate	Symbol	Symbol in your textbook
OR		
AND		
NOT		
NAND		
NOR		
EX-OR or X-OR		
EX-NOR or X-NOR		

Three Basic Logic Gates

Three basic logic gates:

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

The AND Gate

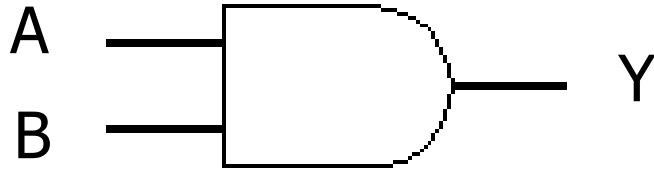


- If the 2 input signals are asserted (high), the output will also be asserted.
- If the 2 input signals are deasserted (low), the output will also be deasserted (low).

Truth Tables

- A **truth table** describes 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



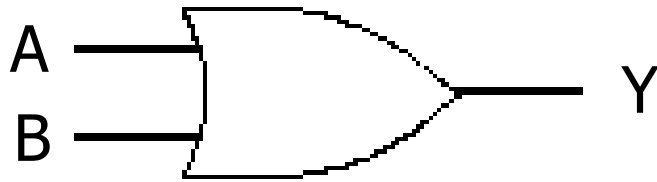
- If the 2 input signals are asserted (high), the output will also be asserted.
- If the 2 input signals are deasserted (low), the output will also be deasserted (low).

Truth Table

Inputs

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

The OR Gate



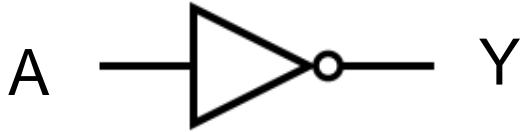
- If either of the two input signals are asserted (= high = 1), or both of them are, the output will be asserted.

Truth Table

Inputs

A	B	Y
0	0	0
0	1	1
1	0	1
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.

Describing Circuit Functionality: Waveforms

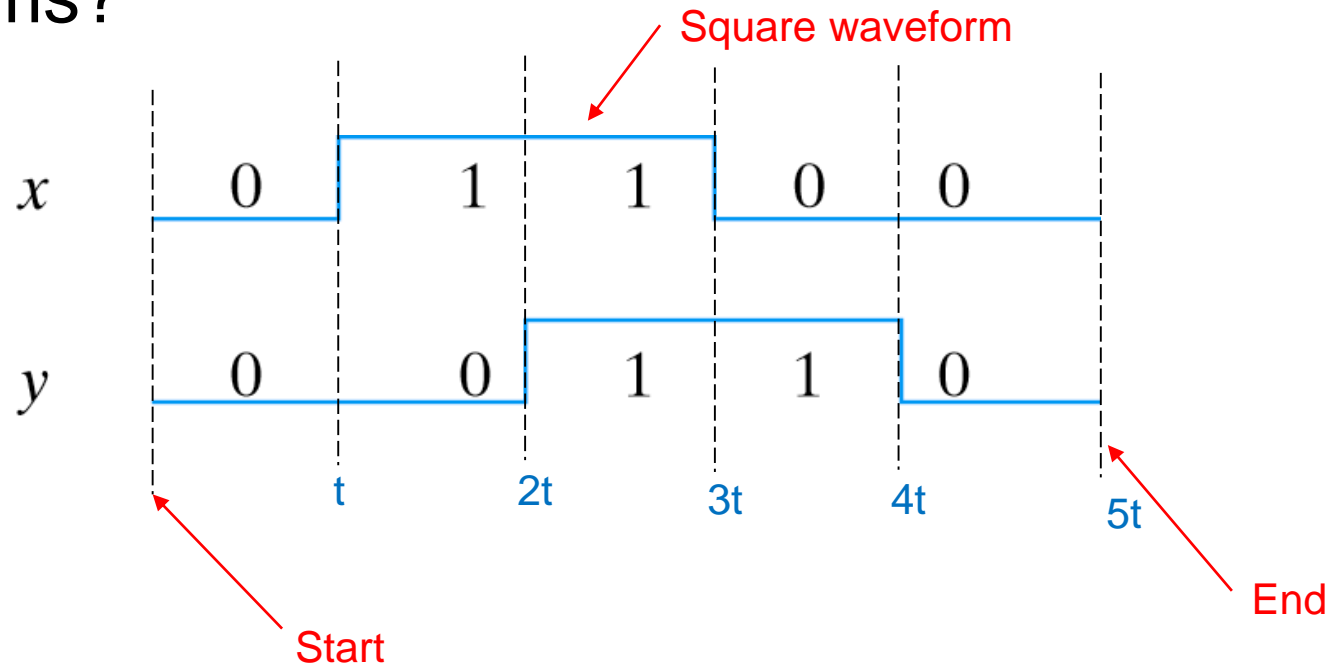
- Waveforms provide another approach for representing functionality.
- Values are either high (logic 1) or low (logic 0).

Describing Circuit Functionality: Waveforms

- Waveforms provide another approach for representing functionality.
- Values are either high (logic 1) or low (logic 0).
- Can you create a truth table from the waveforms?

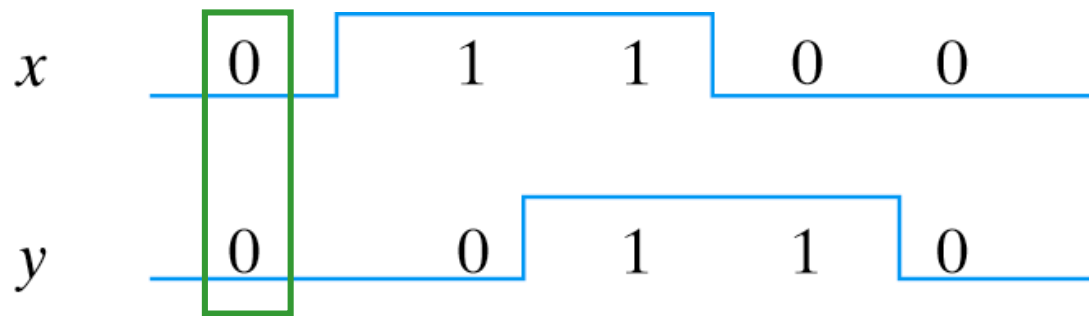
Describing Circuit Functionality: Waveforms

- Can you create a truth table from the waveforms?



Describing Circuit Functionality: Waveforms

- Can you create a truth table from the waveforms?

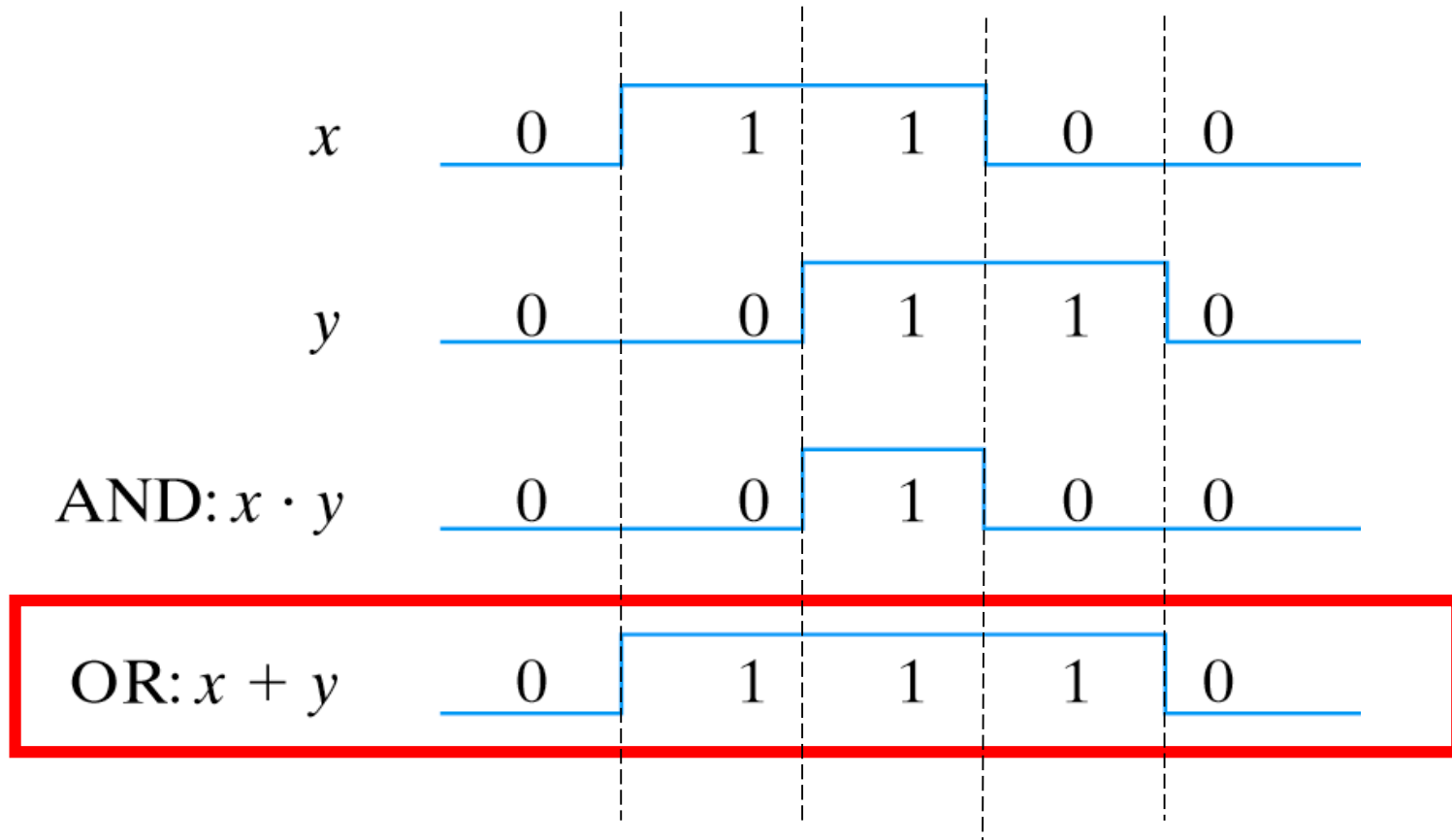


AND: $x \cdot y$



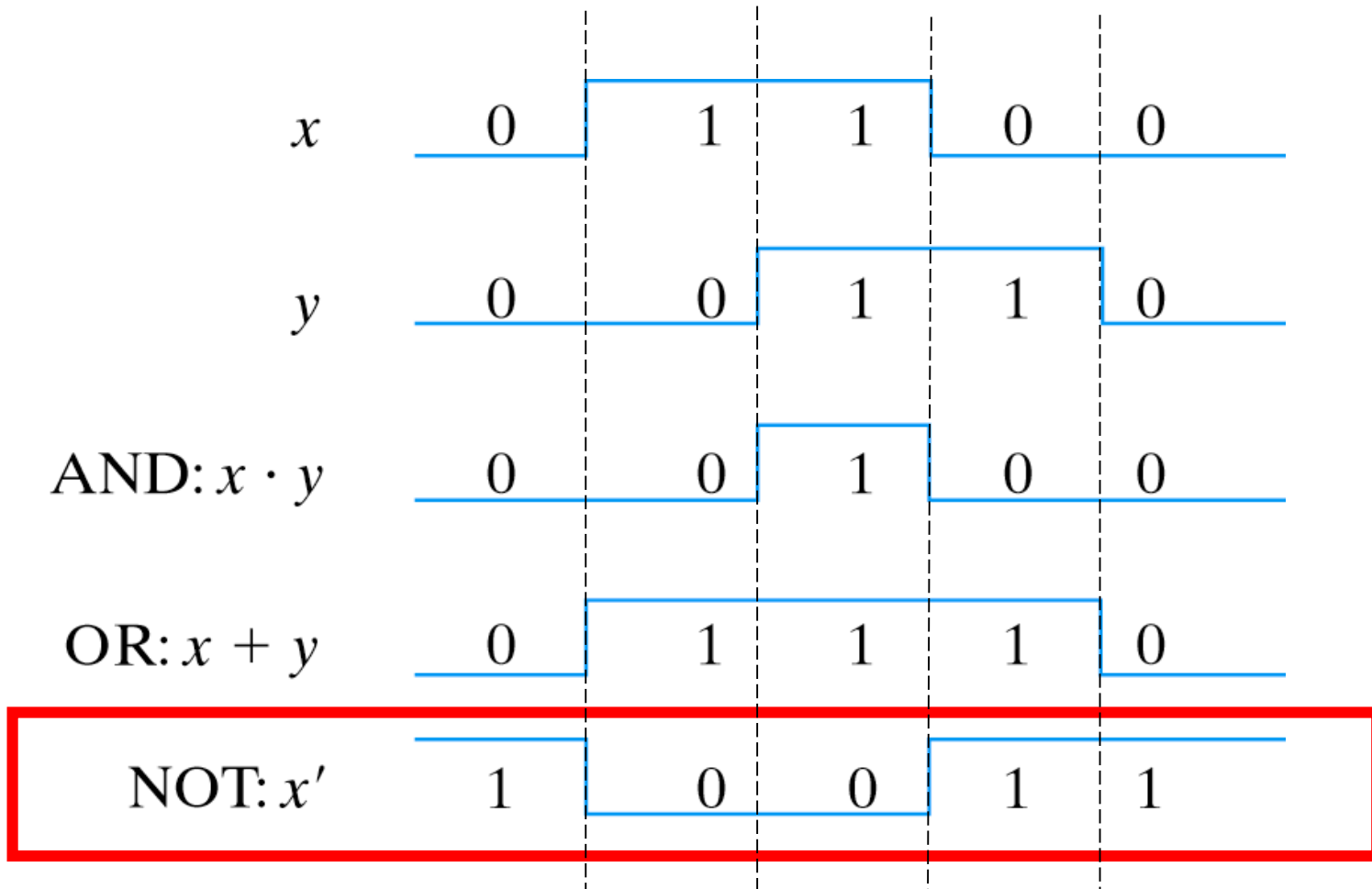
Describing Circuit Functionality: Waveforms

- Can you create a truth table from the waveforms?



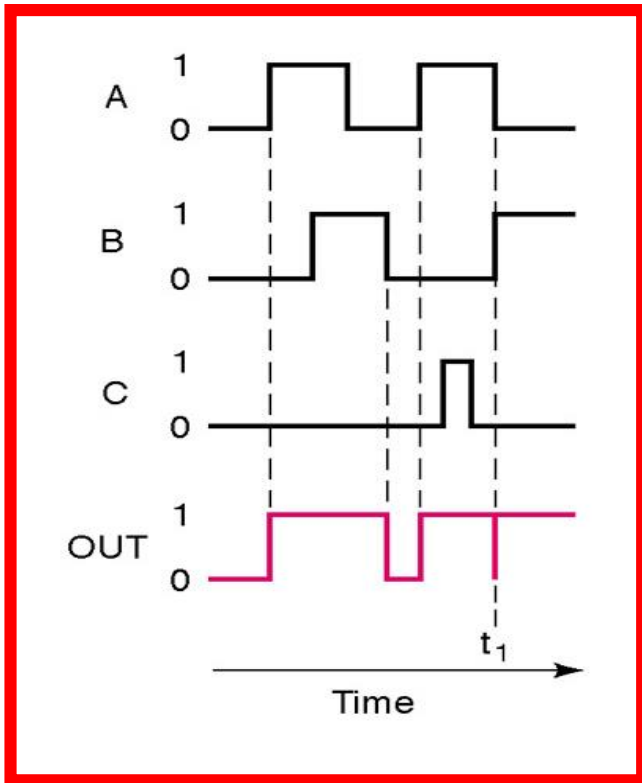
Describing Circuit Functionality: Waveforms

- Can you create a truth table from the waveforms?

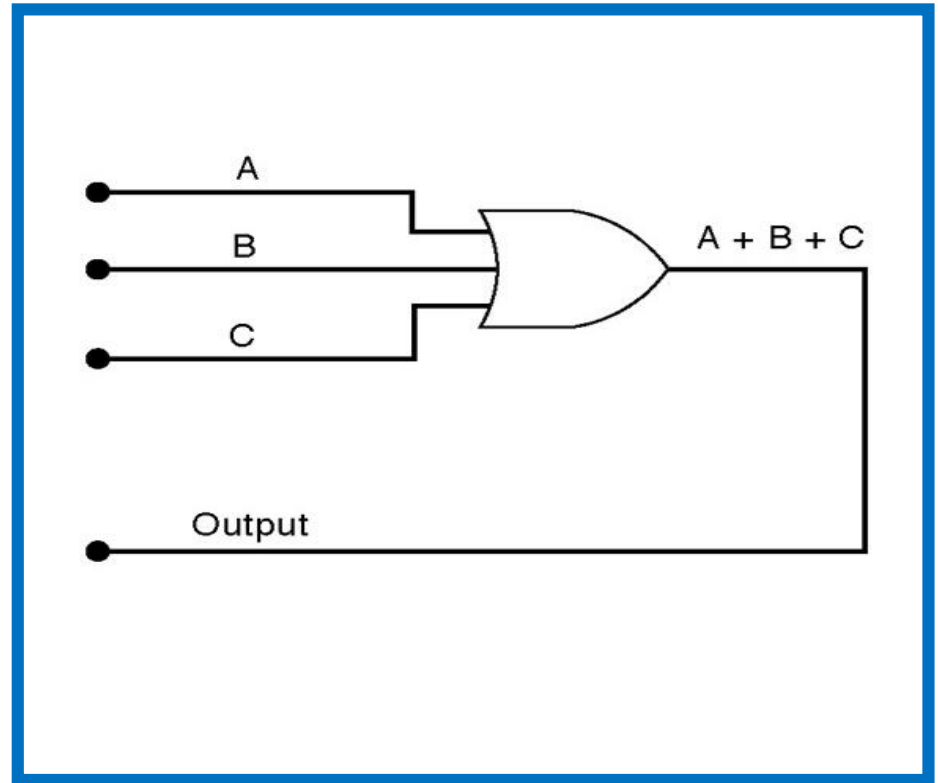


Consider three-input gates

3 Inputs OR Gate

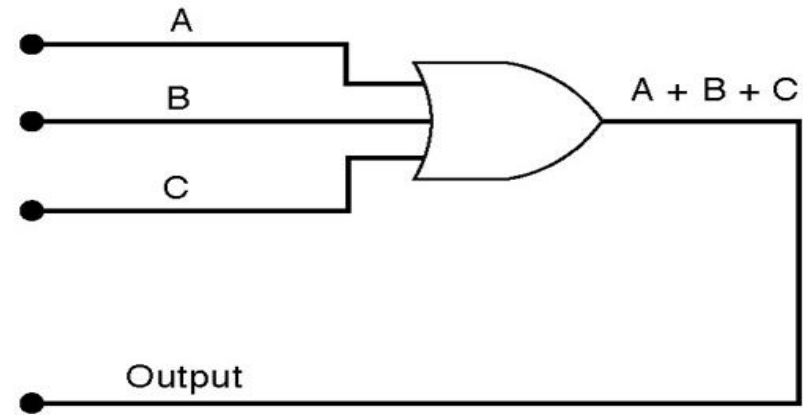
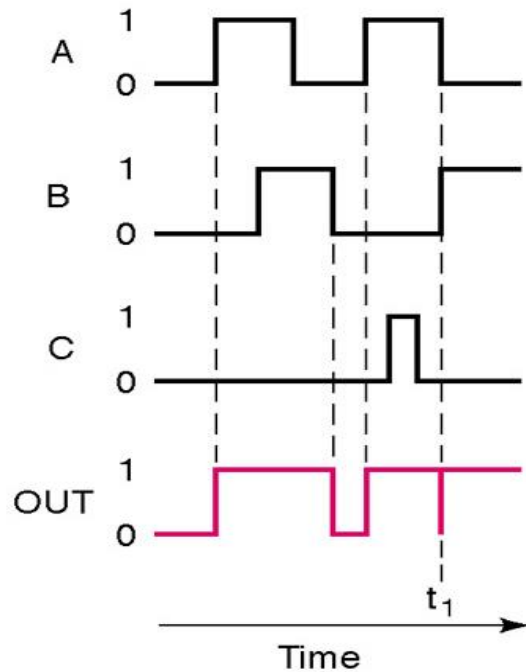


Waveform diagram



Logic symbol

Consider three-input gates



3 Input OR Gate

Truth table

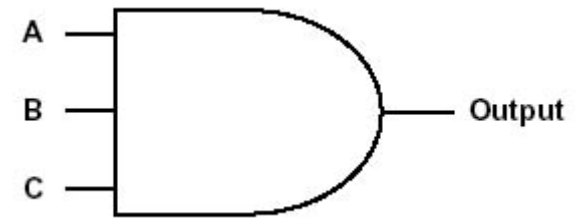
A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Exercise for you!!

3 Inputs AND Gate

Complete the following truth table for a 3-inputs AND gate

A	B	C	Y = A.B.C
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



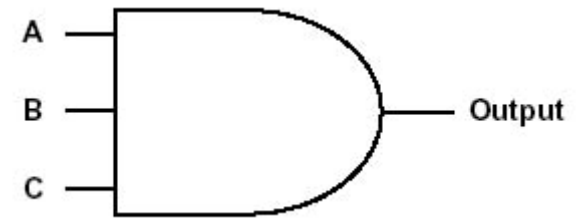
2 minutes ONLY...

Exercise for you!!

3 Inputs AND Gate

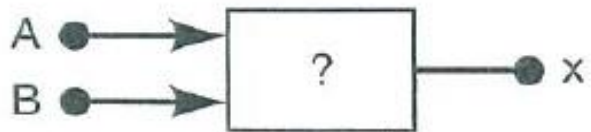
Complete the following truth table for a 3-inputs AND gate

A	B	C	Y = A.B.C
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



Consider many inputs!

Inputs		Output
A	B	x
0	0	1
0	1	0
1	0	1
1	1	0



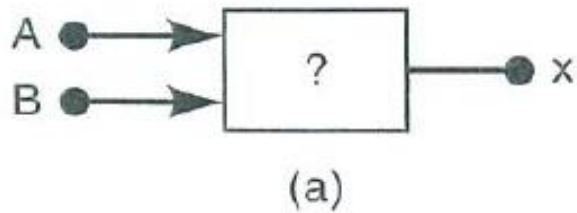
(a)

Consider many inputs!

Inputs		Output
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)



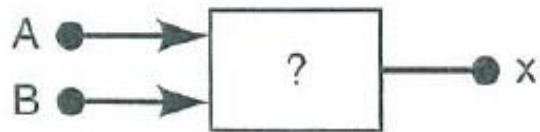
Consider many inputs!

Diagram illustrating a logic function with two inputs (A, B) and one output (x). The output is labeled "Output" and the inputs are labeled "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)

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

Line 1 → $0 + 0 = 0$

Line 2 → $0 + 1 = 1$

Line 3 → $1 + 0 = 1$

Line 4 → $1 + 1 = 1$

Boolean Operators

Summarized rules for OR and AND

	<i>OR</i>	<i>AND</i>
Line 1	$0 + 0 = 0$	$0 \cdot 0 = 0$
Line 2	$0 + 1 = 1$	$0 \cdot 1 = 0$
Line 3	$1 + 0 = 1$	$1 \cdot 0 = 0$
Line 4	$1 + 1 = 1$	$1 \cdot 1 = 1$

Boolean Operations

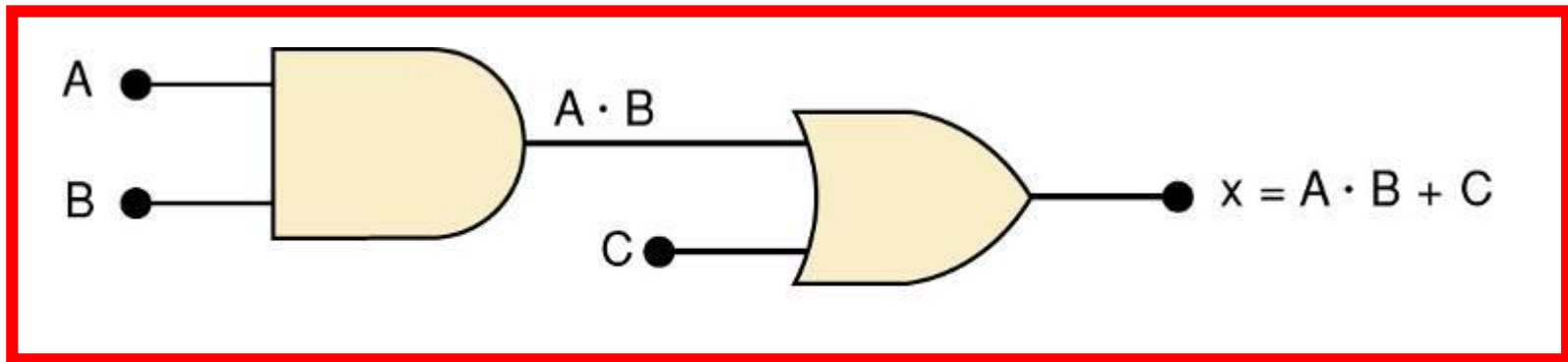
Summarized rules for OR, AND and NOT

<i>OR</i>	<i>AND</i>	<i>NOT</i>
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\bar{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\bar{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

These three basic Boolean operations can describe any logic circuit.

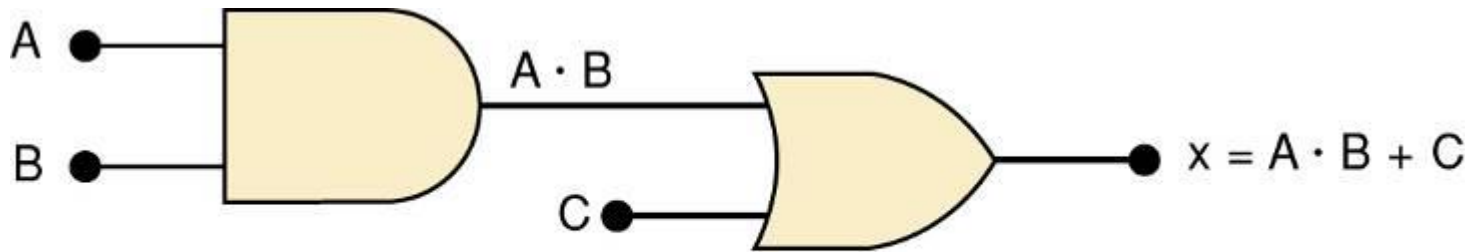
Describing Logic Circuits Algebraically

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

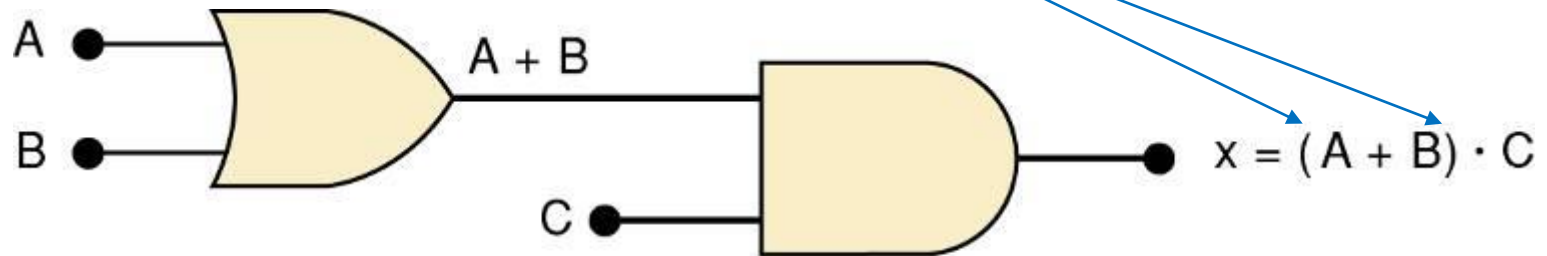


Describing Logic Circuits Algebraically

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



- Unless there is a **parenthesis** in the expression.



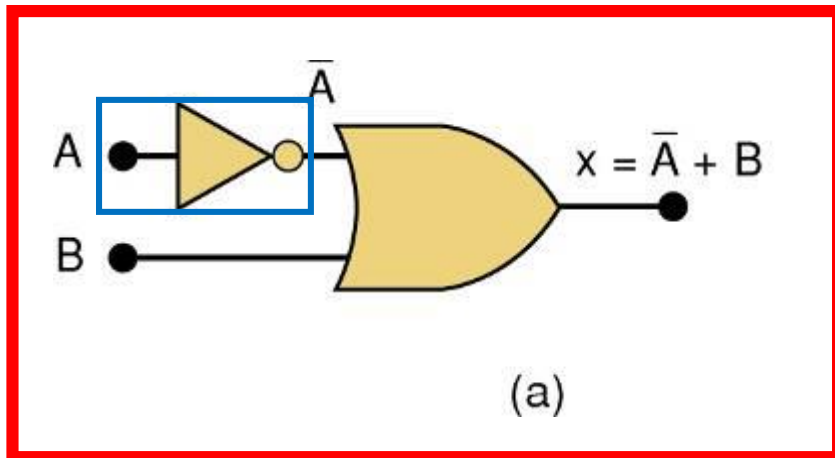
Describing Logic Circuits Algebraically

Whenever an INVERTER (ie NOT gate) is present, output is equivalent to input, with a bar over it.

- Input A through an inverter equals \bar{A} .

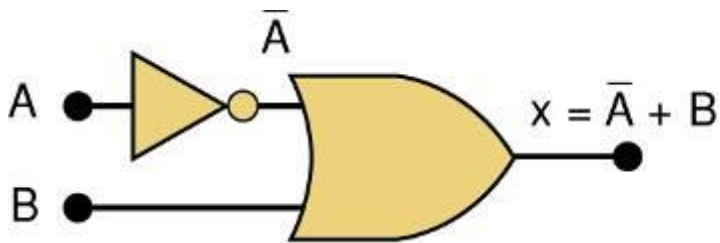
Describing Logic Circuits Algebraically

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

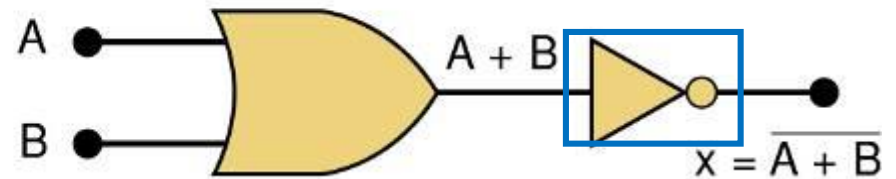


Describing Logic Circuits Algebraically

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



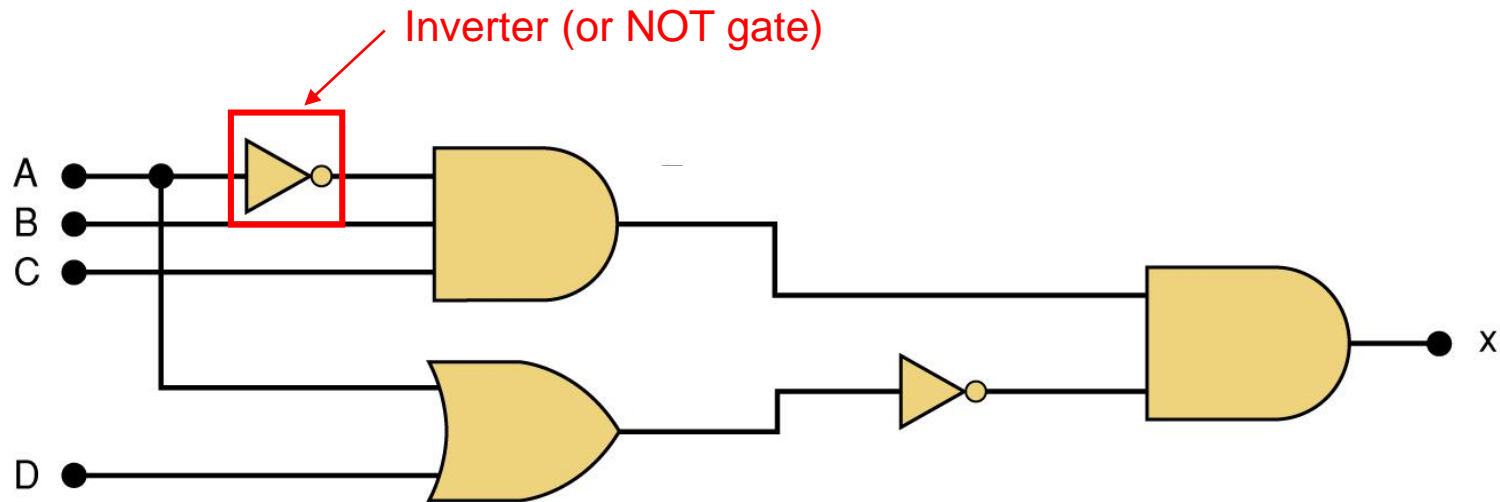
(a)



(b)

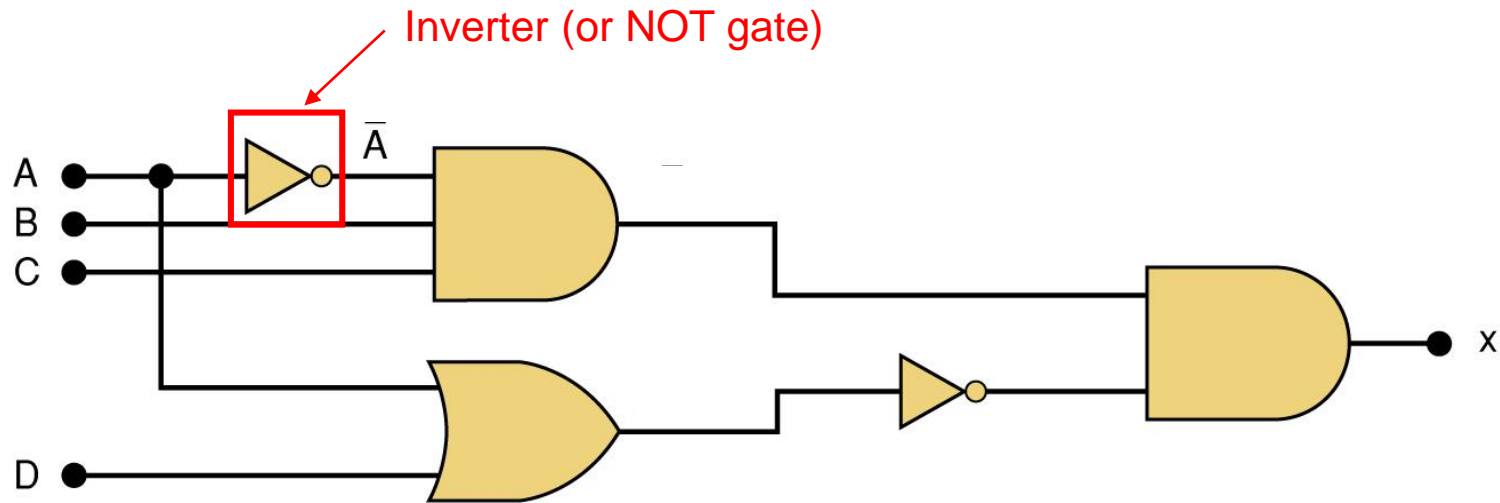
Describing Logic Circuits Algebraically

Example 1



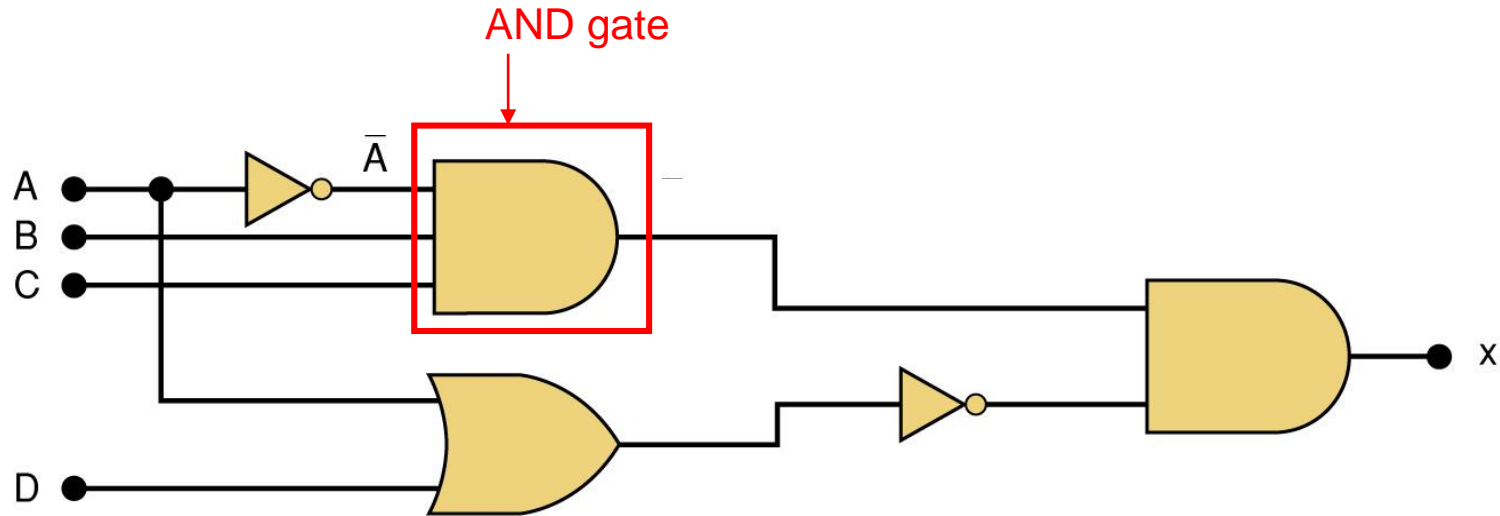
Describing Logic Circuits Algebraically

Example 1



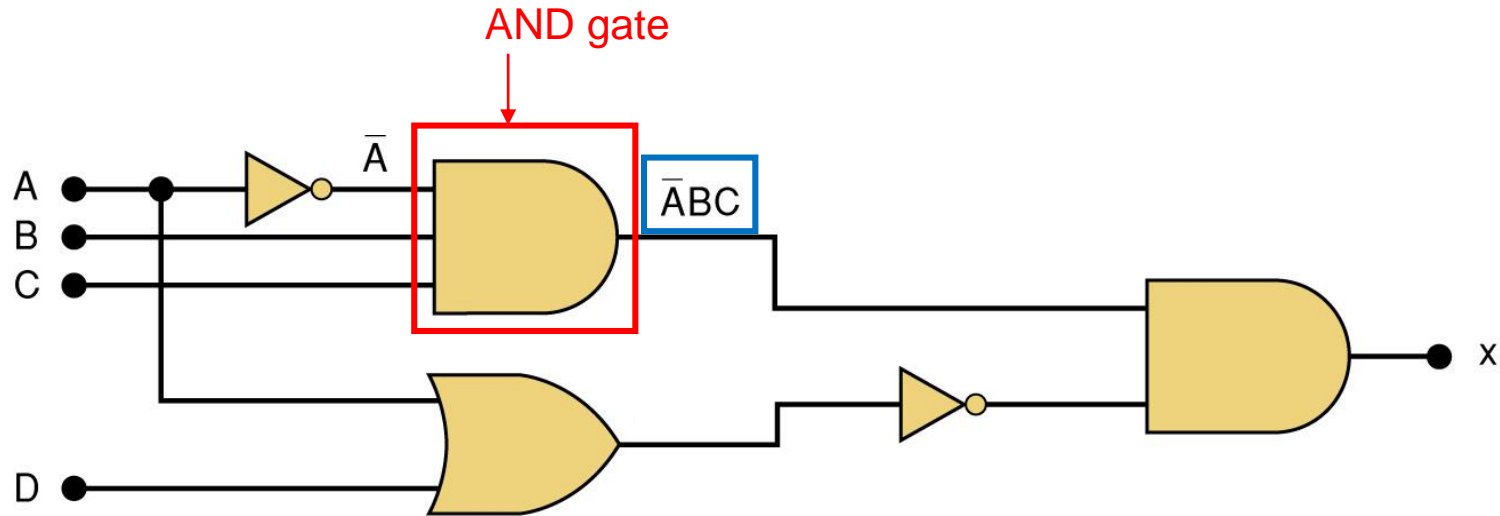
Describing Logic Circuits Algebraically

Example 1



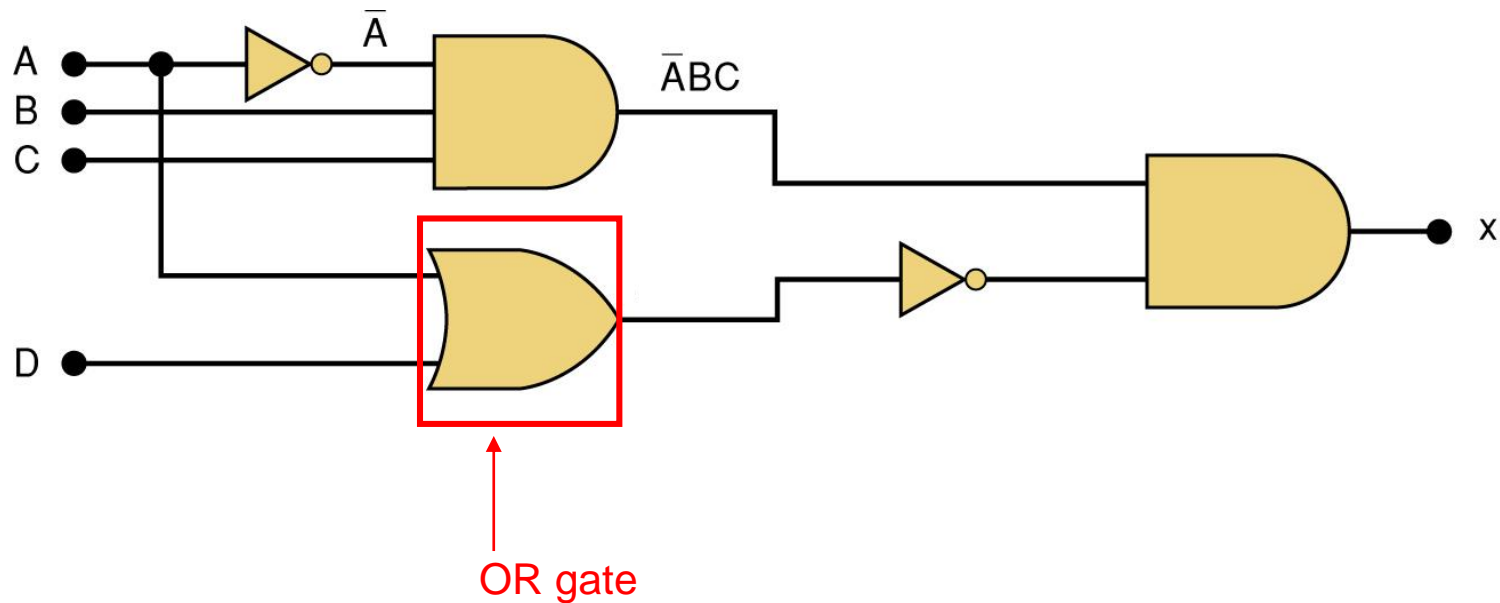
Describing Logic Circuits Algebraically

Example 1



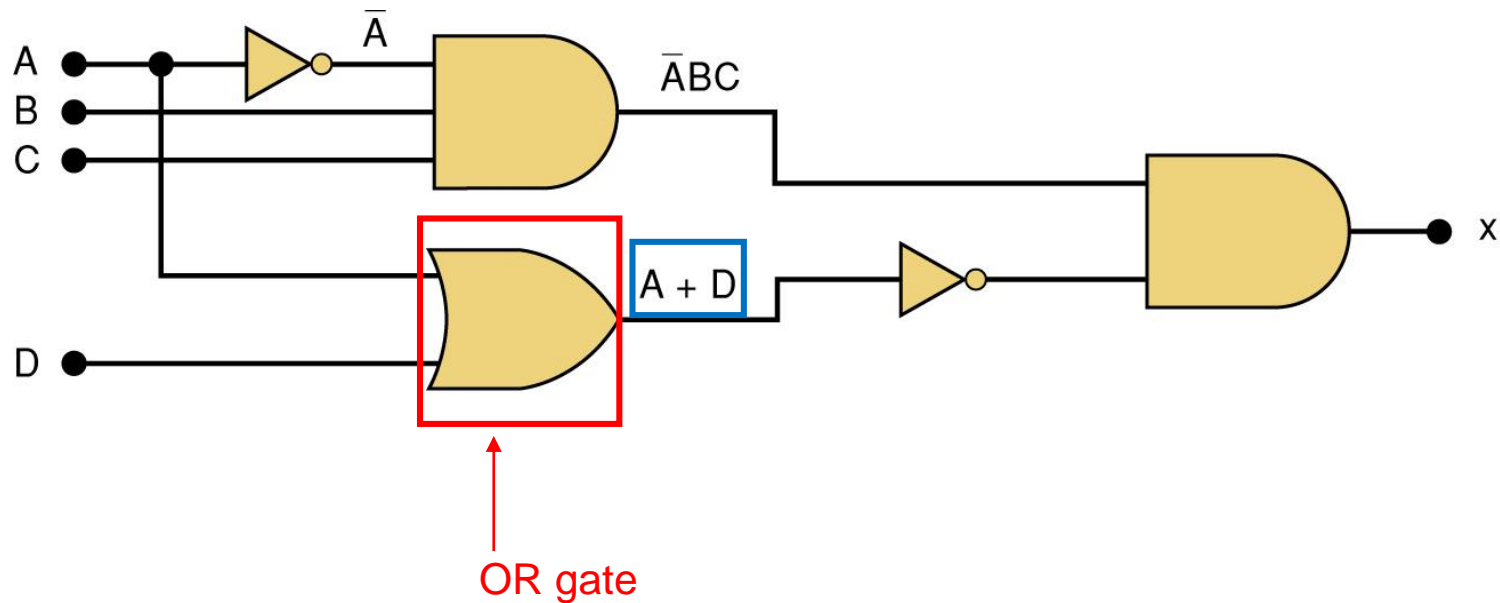
Describing Logic Circuits Algebraically

Example 1



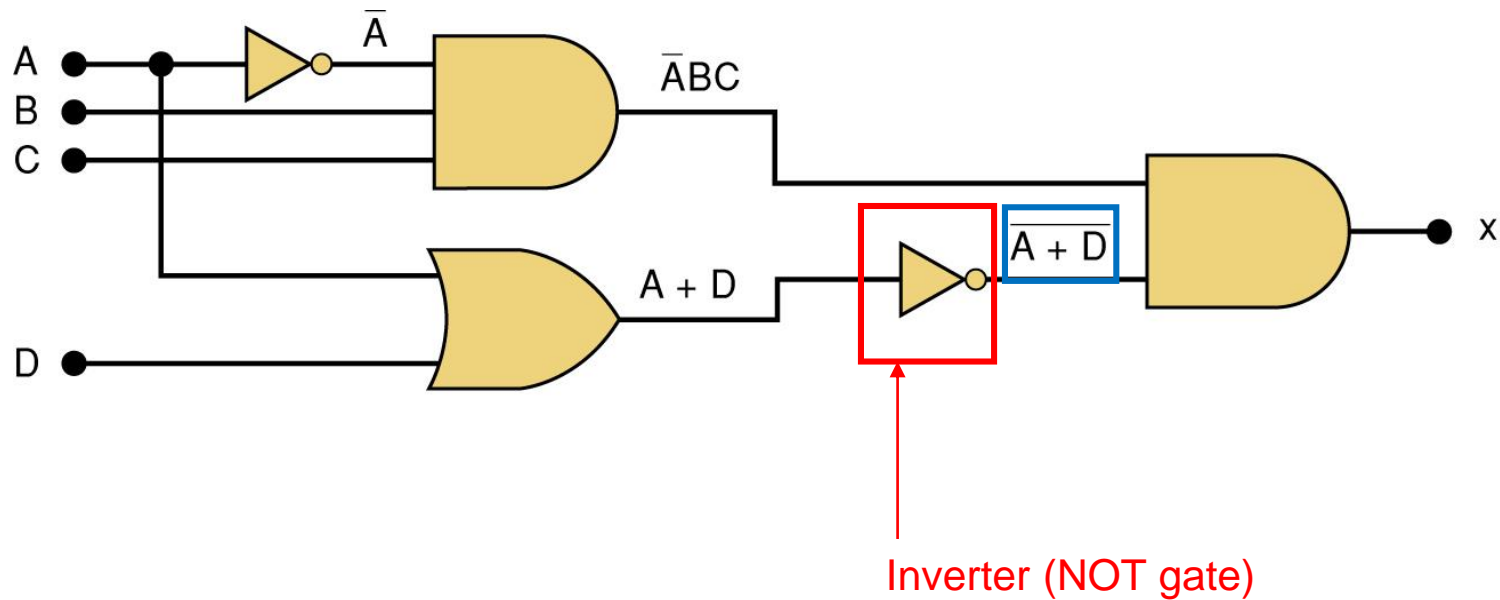
Describing Logic Circuits Algebraically

Example 1



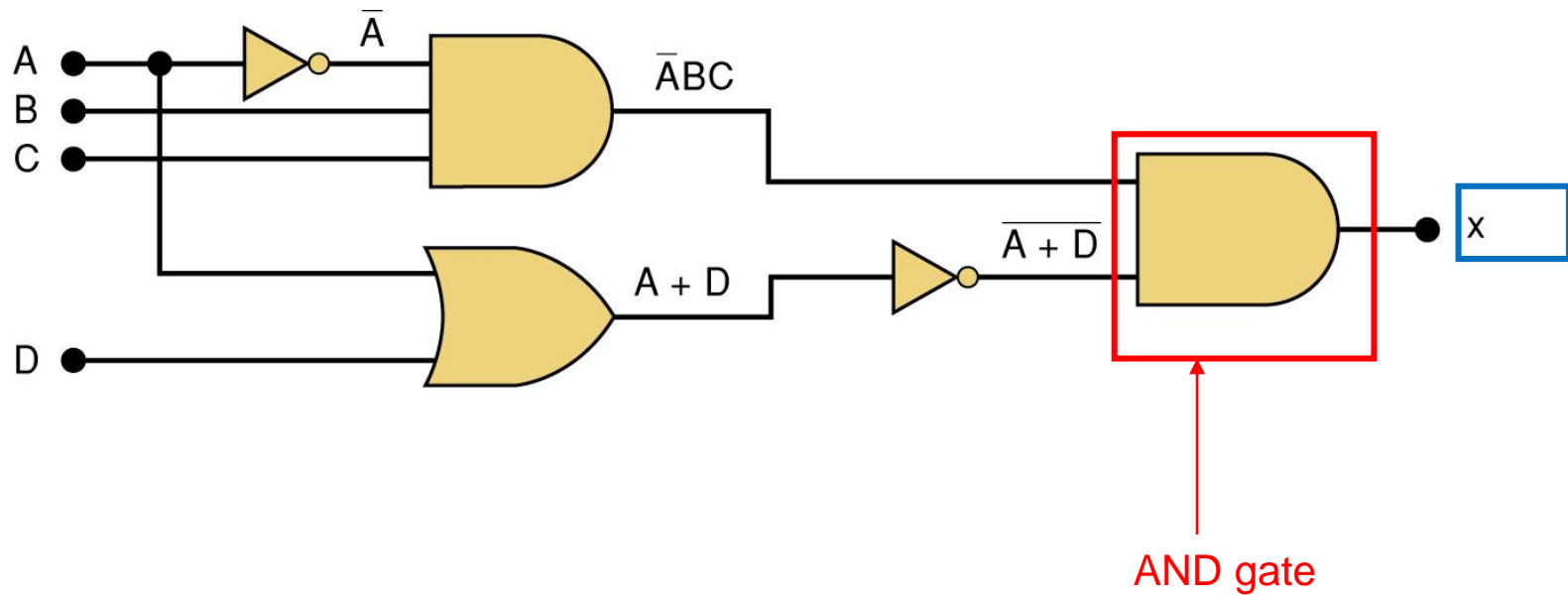
Describing Logic Circuits Algebraically

Example 1



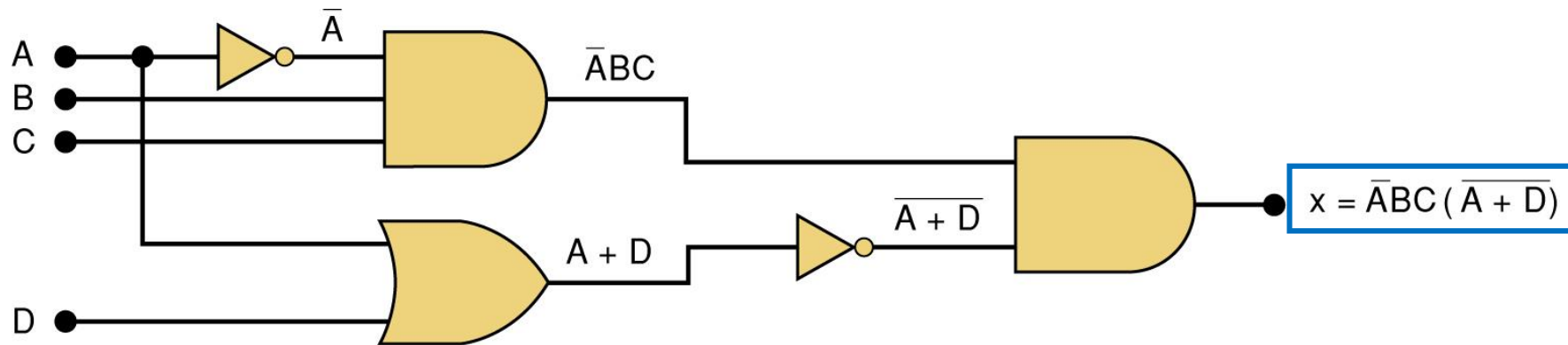
Describing Logic Circuits Algebraically

Example 1



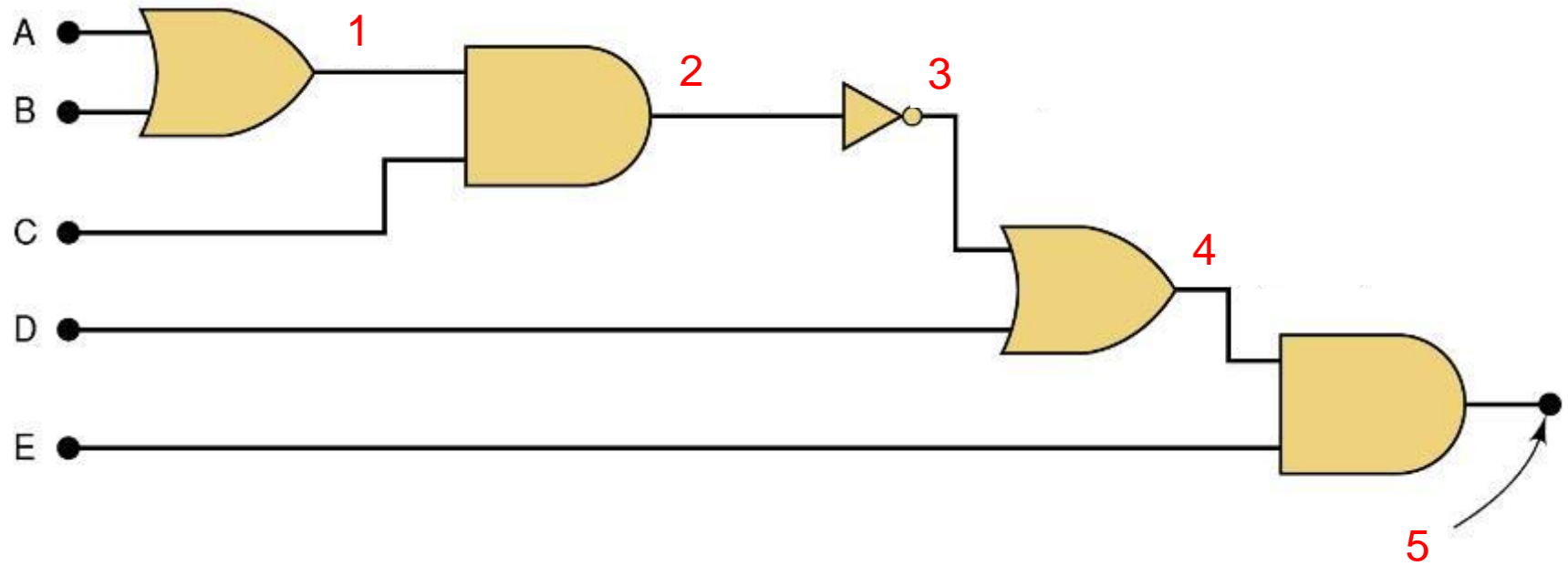
Describing Logic Circuits Algebraically

Example 1



Describing Logic Circuits Algebraically

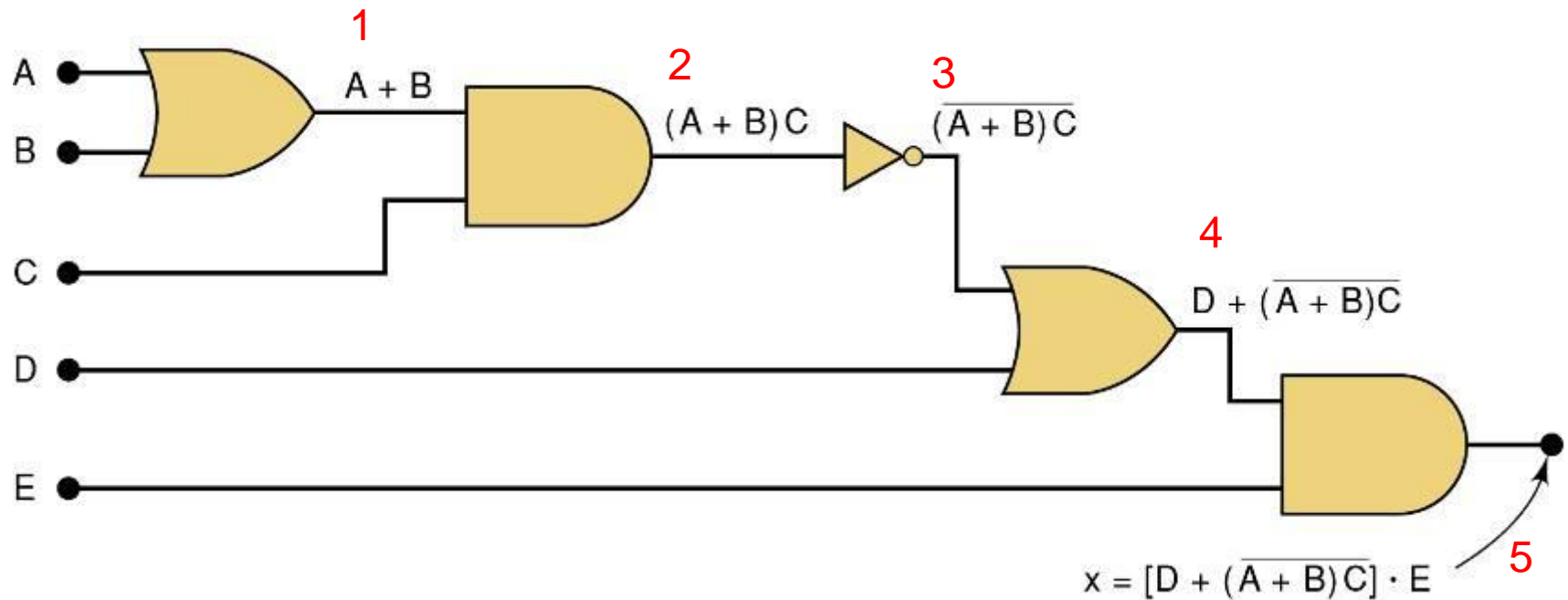
Exercise for you!!



5 minutes

Describing Logic Circuits Algebraically

Exercise for you!!



Week 3 Lecture 5a

- Basic Logic Gates
 - AND, OR and NOT (inverter)
 - Truth tables
 - Describing logic circuits algebraically

- Course web page:

https://ecs.wgtn.ac.nz/Courses/XMUT101_2021T1/

- kerese@ecs.vuw.ac.nz