Name: Index: Reg. No.: Class: Date:

Chapter 10
Boolean Logic

Logic Gates

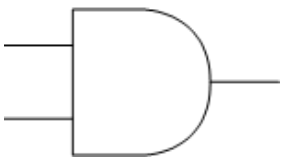
What is Boolean logic?

- Boolean logic is used in computer science and electronics to make **logical decisions**
- Boolean operators are either **TRUE** or **FALSE**, often represented as **1** or **0**
- Inputs and outputs are given **letters to represent them**
- To define Boolean logic we use **special symbols** to make **writing expressions** much **easier**

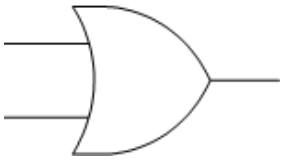
What are logic gates?

- Logic gates are a **visual way** of representing a **Boolean expression**
- The logic gates covered in this course are:

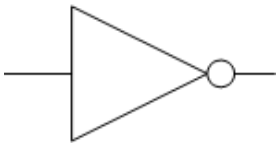
AND

Expression operator	Circuit symbol	Explanation
(A **AND** B)		Returns TRUE only if **both** inputs are TRUE **TRUE** AND **TRUE** = **TRUE** Otherwise = FALSE

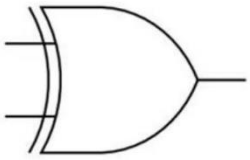
OR

Expression operator	Circuit symbol	Explanation
(A **OR** B)		Returns TRUE if **either** input is TRUE **TRUE** OR **FALSE** = **TRUE** **FALSE** OR **FALSE** = **FALSE**

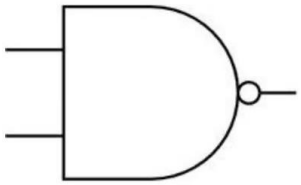
NOT

Expression operator	Circuit symbol	Explanation
(**NOT** A)		**Reverses** the input value NOT **TRUE** = **FALSE** NOT **FALSE** = **TRUE**

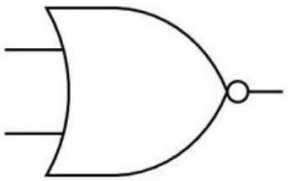
XOR (exclusive OR)

Expression operator	Circuit symbol	Explanation
(A **XOR** B)		Returns TRUE if **either** input is TRUE but **NOT** both **TRUE** OR **FALSE** = **TRUE** **FALSE** OR **FALSE** = **FALSE** **TRUE** OR **TRUE** = **FALSE**

NAND (not AND)

Expression operator	Circuit symbol	Explanation
(A **NAND** B)		Returns TRUE if **either or both** inputs are FALSE **TRUE** OR **FALSE** = **TRUE** **FALSE** OR **FALSE** = **TRUE** **TRUE** OR **TRUE** = **FALSE**

NOR (not OR)

Expression operator	Circuit symbol	Explanation
(A **NOR** B)		Returns TRUE if **both** inputs are FALSE **TRUE** OR **FALSE** = **FALSE** **FALSE** OR **FALSE** = **TRUE** **TRUE** OR **TRUE** = **FALSE**

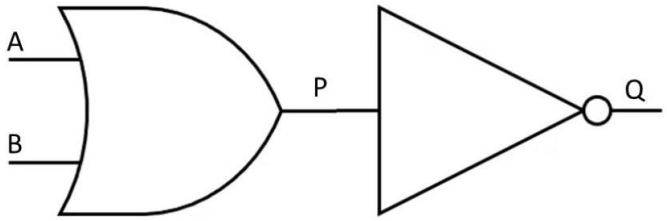
Examiner Tips and Tricks

You will need to either draw a diagram of a logic circuit using these symbols, or you will have to write the boolean expression from an existing diagram.

Logic Circuits

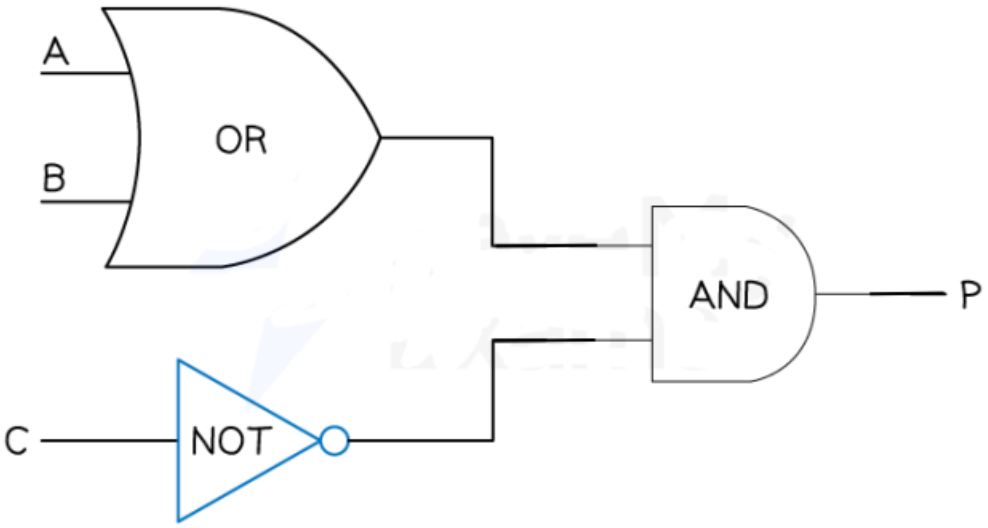
What is a logic circuit?

- A logic circuit **performs logical operations** on binary information
- Logic gates are **core components** of logic circuits
- Based on the **arrangement of logic gates** and the **input signals**, the circuit performs a specific function
- A logic diagram is a **visual representation** of combinations of logic gates within a logic circuit
- **Brackets** are used to clarify the **order** of operations
- An example logic diagram for the Boolean expression **Q = NOT(A OR B)** would be:



- In the logic diagram above, the **inputs** are represented by **A and B**
- **P** is the **output** of the OR gate on the left and becomes the input of the NOT gate
- **Q** is the fi**nal output** of the logic circuit

Example of combining logic gates



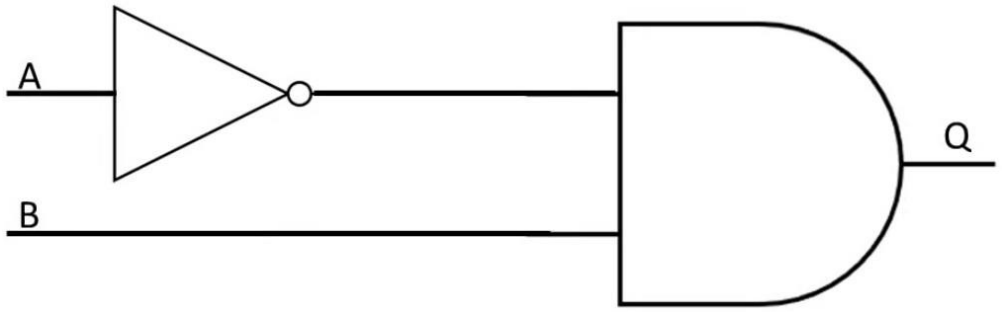
P = (A OR B) AND NOT C

Worked Example

A sprinkler system switches on if it is not daytime (input A) and the temperature is greater than 40 (input B)

Draw a logic circuit to represent the problem statement above **[2]**

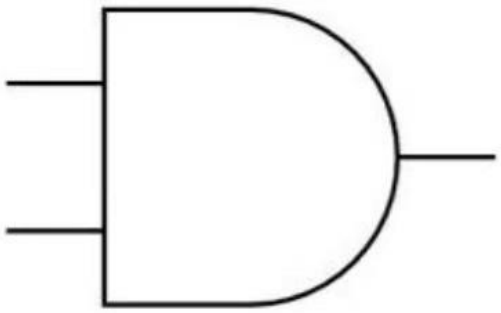
Answers



Truth Tables

- A truth table is a tool used in logic and computer science to **visualise** the **results** of **Boolean expressions**
- They represent **all possible inputs** and the **associated outputs** for a **given Boolean expression**

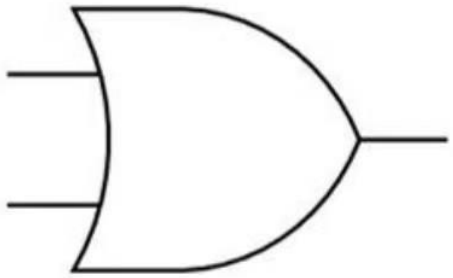
AND	
Circuit symbol	**Truth Table**



INPUTS		**OUTPUTS**
0	0	0
0	1	0
1	0	0
1	1	1

OR

Circuit symbol

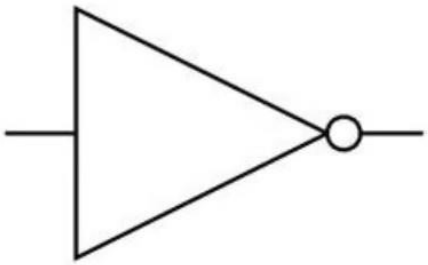


Truth Table

INPUTS		OUTPUTS
0	0	0
0	1	1
1	0	1
1	1	1

NOT

Circuit symbol

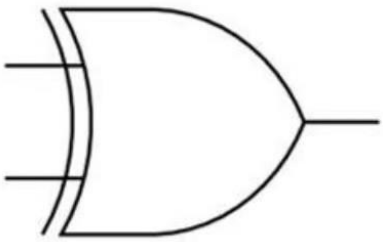


Truth Table

INPUTS	OUTPUTS
0	1
1	0

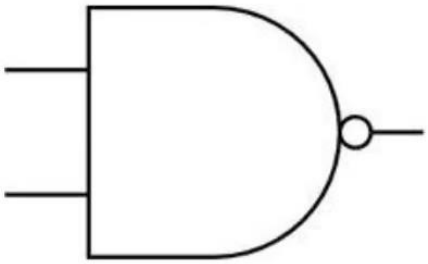
XOR (exclusive)

Circuit symbol

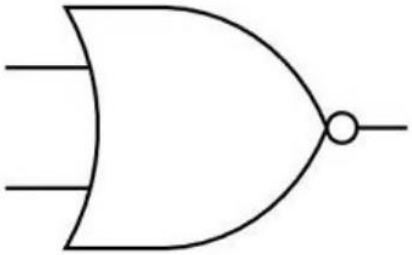


Truth Table

INPUTS		OUTPUTS
0	0	0
0	1	1
1	0	1
1	1	0

NAND (Not AND)	
Circuit symbol	**Truth Table**
	

INPUTS		OUTPUTS
0	0	1
0	1	1
1	0	1
1	1	0

NOR (Not OR)	
Circuit symbol	**Truth Table**
	

INPUTS		OUTPUTS
0	0	1
0	1	0
1	0	0
1	1	0

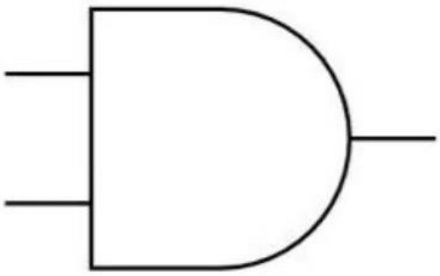
How do you create truth tables for logic circuits?

- To create a truth table for the expression **P = (A AND B) AND NOT C**
  - Calculate the numbers of rows needed ($2^{\textbf{number of inputs}}$)
  - In this example there are 3 inputs (**A, B, C**) so a total of **8 rows** are needed (2^3)
  - To not miss any combination of inputs, start with **000** and **count** up in **3−bit binary (0−7)**

Logic Expressions

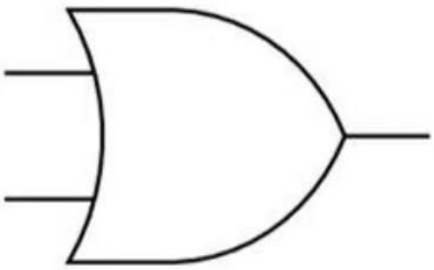
- A **logic expression** is a way of expressing a logic gate or logic circuit as an **equation**
- The **output** appears on the **left** of the equals sign with the **inputs** and **logic gates** on the **right**

AND

Circuit symbol	Truth Table	Logic Expression
		Z = A AND B

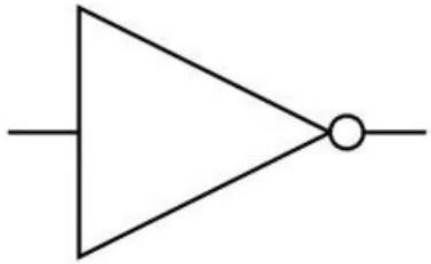
INPUTS		OUTPUTS
A	**B**	**Z**
0	0	0
0	1	0
1	0	0
1	1	1

OR

Circuit symbol	Truth Table	Logic Expression
		Z = A OR B

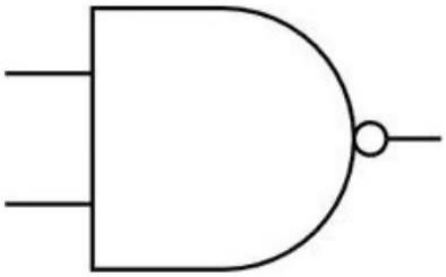
INPUTS		OUTPUTS
A	**B**	**Z**
0	0	0
0	1	1
1	0	1
1	1	1

NOT

Circuit symbol	Truth Table	Logic Expression
		Z = NOT A

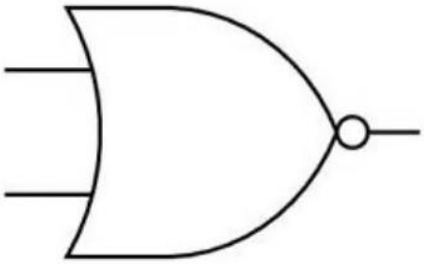
INPUTS	OUTPUTS
A	Z
0	1
1	0

NAND

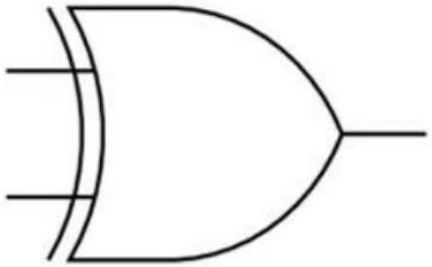
Circuit symbol	Truth Table	Logic Expression
		Z = A NAND B

INPUTS		OUTPUT
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

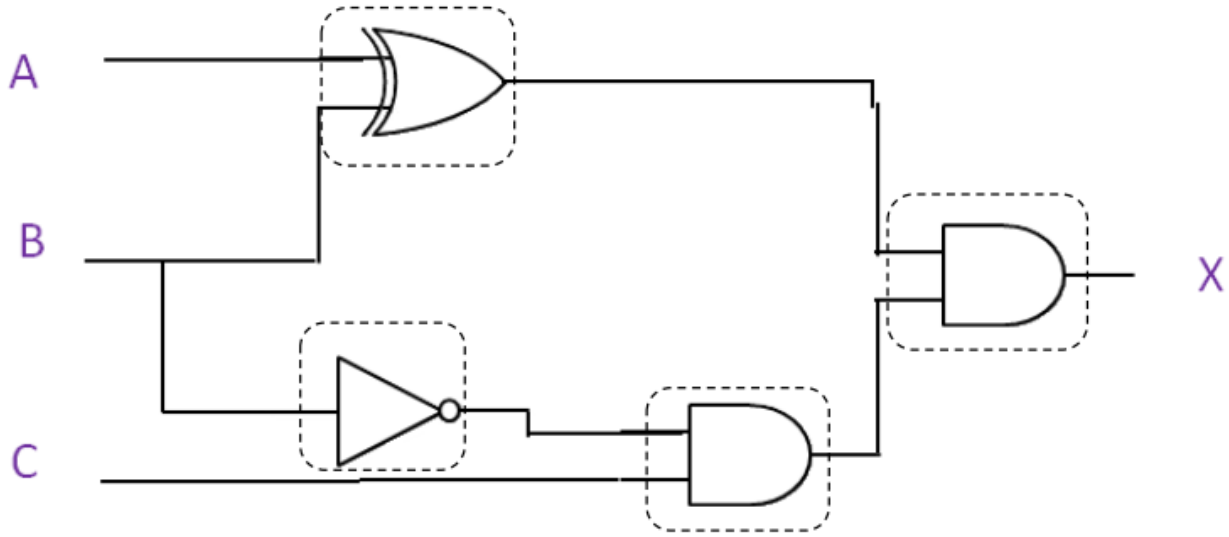
NOR

Circuit symbol	Truth Table	Logic Expression
		Z = A NOR B

INPUTS		OUTPUT
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

<table>
<tr><th colspan="3">XOR (Exclusive OR)</th></tr>
<tr><th>Circuit symbol</th><th>Truth Table</th><th>Logic Expression</th></tr>
<tr>
<td>  </td>
<td>
<table>
<tr><th colspan="2">INPUTS</th><th>OUTPUT</th></tr>
<tr><th>A</th><th>B</th><th>Z</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
<td>Z = A XOR B</td>
</tr>
</table>

From a circuit, build a Truth Table



Steps:

- Identify all the possible **combinations of input**
- Work out the solution for **intermediate output** (layer by layer)
- Using the intermediate outputs, construct the **final output**

Step 1:

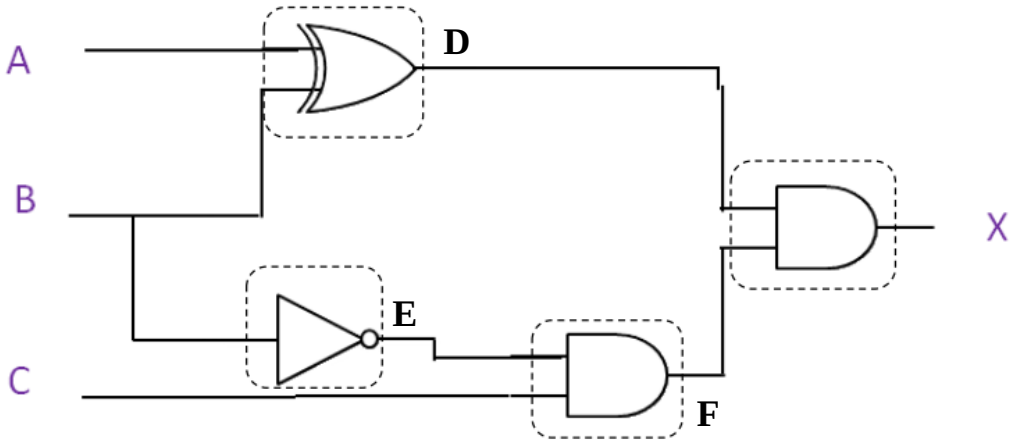
Identify all the possible combinations of input (we have 3 inputs)

$2^3 = 8$ rows

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

Step 2:

Identify the intermediate outputs. Solve them one by one.

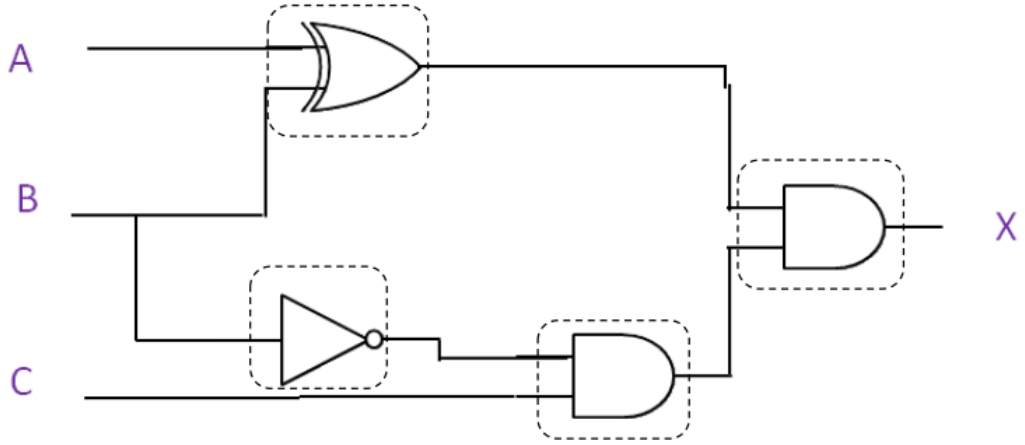


Step 3:

Using the intermediate outputs, construct the final output.

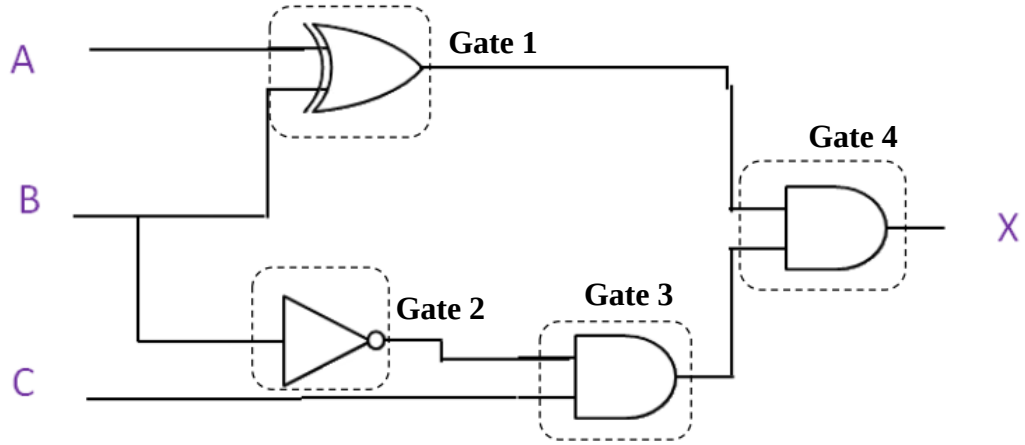
A	B	C	X
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**0**
1	1	1	**0**

From a circuit, build a logical expression



Step 1:

Look at the gates connected to A, B and C.



Step 2:

Then, form an expression with gate 1, gate 2 and gate 3.

Gate 1 = A XOR B

Gate 2 = NOT B

Gate 3 = Gate 2 AND C

↓ Substitution

Gate 3 = NOT B AND C

Step 3:

Then, form an expression with FINAL gate 4.

Gate 4 = Gate 1 AND Gate 3 → Substitution → **Gate 4 = (A XOR B) AND (NOT B AND C)**

Final logical expression → **X = (A XOR B) AND (NOT B AND C)**

From a logical expression, build a circuit

X = (A NAND C) XOR (NOT B OR C)

To solve this problem, we need to **break down** the statement into smaller pieces.

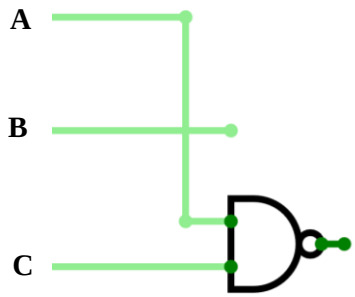
Part 1: (A NAND C)

Part 2: (NOT B OR C)

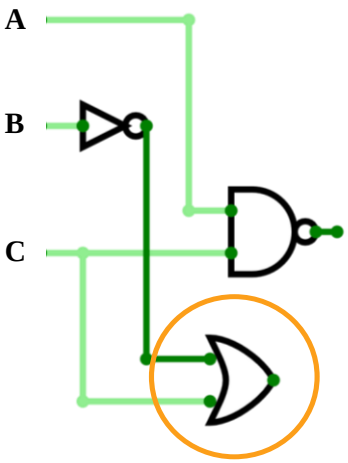
Part 3: (Part 1) XOR (Part 2)

Let's build the circuit piece by piece

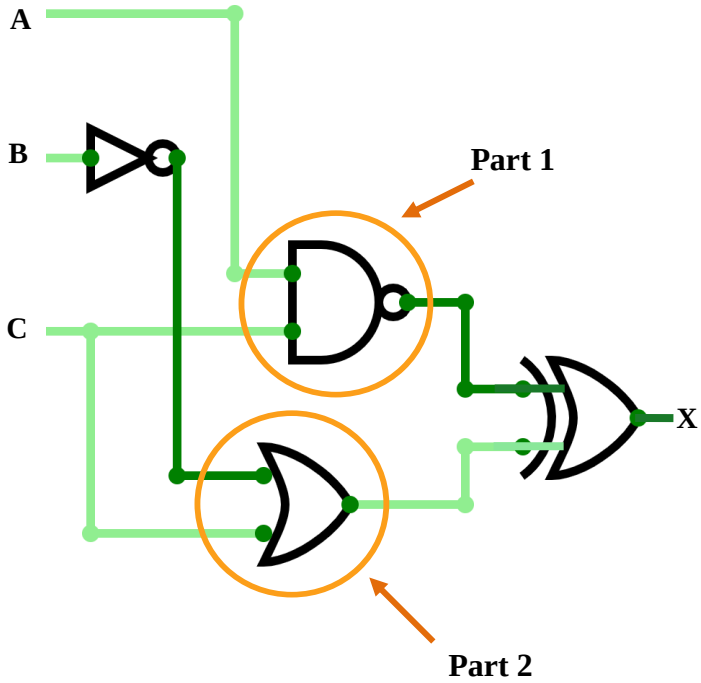
Part 1: (A NAND C)



Part 2: (NOT B OR C)



Part 3: (Part 1) XOR (Part 2)



From a truth table, build a logical expression and a circuit

INPUT		OUTPUT
A	B	X
0	0	0
0	1	0
1	0	1
1	1	0

Steps to follow to build logical expression

1. To produce the logical expression, look at only the row where the **output value is 1**
2. Combine each input with an **AND** statement
3. If the input value is **zero**, negate it using **NOT gate**

Negate B (which is 0)

X = A AND (NOT B)

Combine each input

If more than one row have output value 1.

1. Follow the **above steps** and create logical expression for **each row separately**
2. **Concatenate** these statements with **OR** gate

For example:

INPUT		OUTPUT
A	B	X
0	0	0
0	1	1
1	0	0
1	1	1

1st row: (NOT A) AND B
2nd row: A AND B

1st row **OR** 2nd row

Substitution

X = ((NOT A) AND B) OR (A AND B)

From a problem statement, draw a logic gate

(b) A water control system uses a switch and two pressure sensors.

The outputs of the switch and sensors are shown in the table.

Sensor or Switch	Output of 1	Output of 0
Switch (**S1**)	On	Off
Pressure Sensor (**P1**)	>= 3	< 3
Pressure Sensor (**P2**)	>= 3	< 3

Create a logic circuit that will produce an output (**X**) of 1 when:

The switch **S1** is on

and

either P1 is less than 3 **or P2** is less than 3, **but not both**.

Simplifying the problem statement

S1 AND (NOT P1 XOR NOT P2)

- Understand the problem first
- Identify the **inputs** (in this problem we have 3 inputs. S1, P1 and P2)
- Look for the **keywords**. For example **and, or**
- If any output is **zero**, then need to make a **NOT gate**

Sensor or Switch	Output of 1	Output of 0
Switch (**S1**)	On	Off
Pressure Sensor (**P1**)	>= 3	< 3
Pressure Sensor (**P2**)	>= 3	< 3

Output is 0 for this both value. That is why used **NOT gate**

Create a logic circuit that will produce an output (**X**) of 1 when:

The switch **S1** is on

and

either P1 is less than 3 **or P2** is less than 3, **but not both**.

S1 AND (NOT P1 XOR NOT P2)

Why not **OR gate**?

either P1 is less than 3 **or P2** is less than 3, **but not both**.

Read the statement carefully. It says not both.

XOR gate only gives output 1 if only one input is 1.

INPUTS		OUTPUT
A	**B**	**Z**
0	0	0
0	1	1
1	0	1
1	1	0