Chapter 10

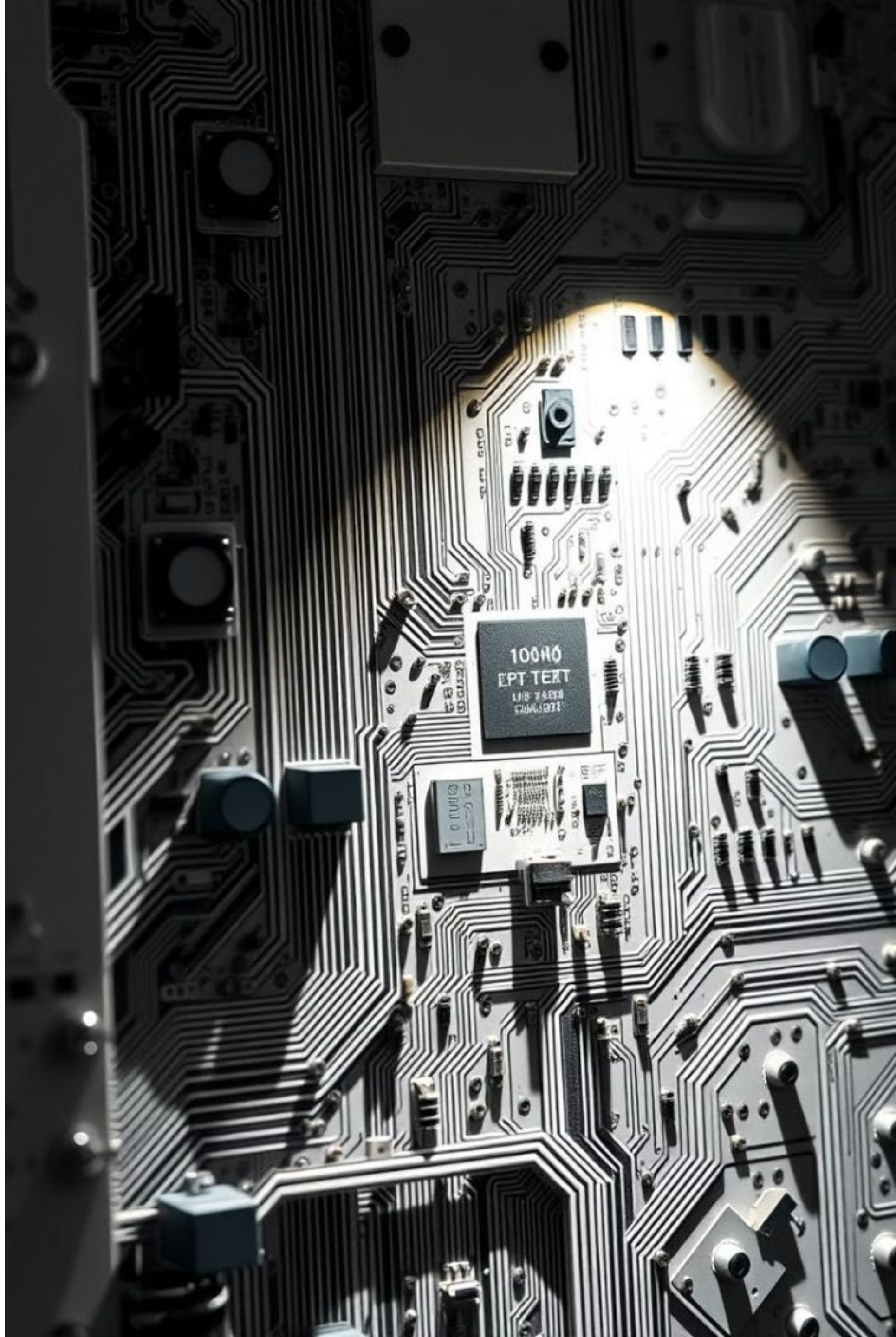
BOOLEAN LOGIC

Content overview *Candidates study the following topics:*	In this chapter you will learn about:
Computer systems 1. Data representation 2. Data transmission 3. Hardware 4. Software 5. The internet and its uses 6. Automated and emerging technologies **Algorithms, programming and logic** 7. Algorithm design and problem-solving 8. Programming 9. Databases 10. Boolean logic	**1. the identification, definition, symbols and functions of the standard logic gates:** **NOT, AND, OR, NAND, NOR and XOR** **2. how to use logic gates to create logic circuits from:** ○ a given problem ○ a logic expression ○ a truth table **3. how to complete truth tables from:** ○ a given problem ○ a logic expression ○ a logic circuit **4. how to write a logic expression from:** ○ a given problem ○ a logic circuit ○ a truth table.

What is Boolean Logic?

Boolean logic is a system of logical operations. It is named after George Boole, a 19th-century mathematician.

Boolean logic is used in computer science, mathematics, and philosophy.



Analogies for Boolean Logic

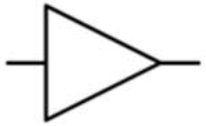
Light Switch

On represents true, off represents false. Simple and easily understood.

Voting

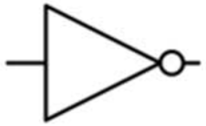
Yes is true, no is false. Illustrates decision-making processes.

Buffer



Input	Output
0	0
1	1

Inverter



Input	Output
0	1
1	0

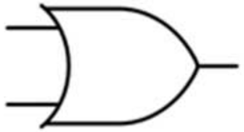
AND

A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

NAND

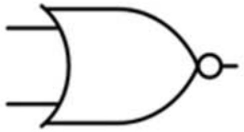
A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

OR



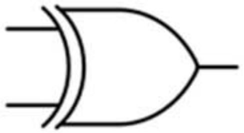
A	B	Output
0	0	0
1	0	1
0	1	1
1	1	1

NOR



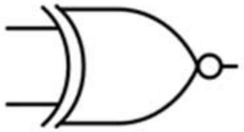
A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

XOR



A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

XNOR



A	B	Output
0	0	1
1	0	0
0	1	0
1	1	1

✅ Functions of Logic Gates

Logic Gate	Function Description	Boolean Expression	Truth Table (A, B → Output)
NOT	Reverses the input: 1 becomes 0, 0 becomes 1	`Q = ¬A` or `Q = NOT A`	A → Q: 0 → 1, 1 → 0
AND	Output is 1 **only if both inputs are 1**	`Q = A AND B`	00 → 0, 01 → 0, 10 → 0, 11 → 1
OR	Output is 1 if **at least one input is 1**	`Q = A OR B`	00 → 0, 01 → 1, 10 → 1, 11 → 1
XOR	Output is 1 if **inputs are different**	`Q = A XOR B`	00 → 0, 01 → 1, 10 → 1, 11 → 0
NAND	Opposite of AND: Output is 0 **only if both inputs are 1**	`Q = NOT (A AND B)`	00 → 1, 01 → 1, 10 → 1, 11 → 0
NOR	Opposite of OR: Output is 1 **only if both inputs are 0**	`Q = NOT (A OR B)`	00 → 1, 01 → 0, 10 → 0, 11 → 0
XNOR	Opposite of XOR: Output is 1 if **inputs are the same**	`Q = NOT (A XOR B)` or `A ⊙ B`	00 → 1, 01 → 0, 10 → 0, 11 → 1

Real World Example

NOT Gate (Inverter) **Real-World Example:**

Motion Sensor Lights at Night

When it's **not daytime (input = 0)**, the **light turns ON (output = 1)**.

Logic: Output = NOT Daytime

Real World Example

AND Gate Real-World Example:

Microwave Oven Safety

A microwave runs only if the door is closed **AND** the timer is set.

Logic: Run = DoorClosed AND TimerSet

Real World Example

OR Gate Real-World Example:

Car Alarm System

The alarm triggers if the door opens OR the window breaks.

Logic: Alarm = DoorOpen OR WindowBreak

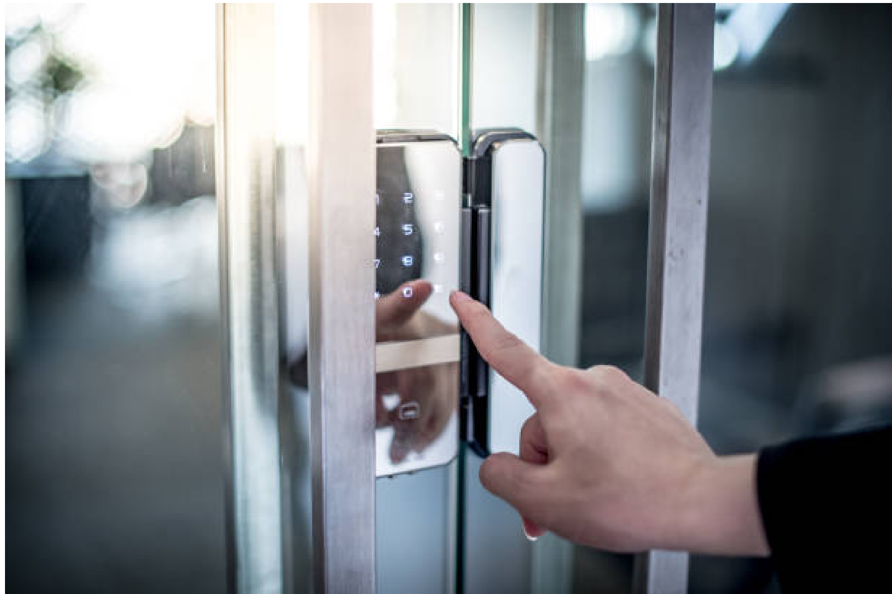
Real World Example

NAND Gate Real-World Example:

Security System Lock

A system stays locked unless both identity checks are passed.

Logic: Lock = NOT (ID1 AND ID2)



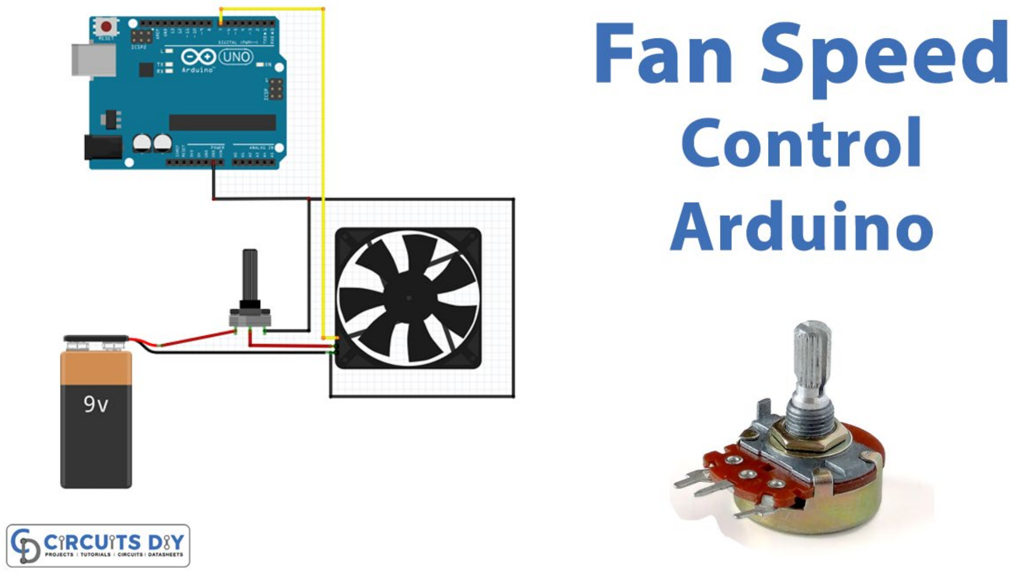
Real World Example

NAND Gate **Real-World Example:**

Automatic Fan Control

A fan turns on only when there is no heat OR humidity detected.

Logic: FanOn = NOT (Heat OR Humidity)



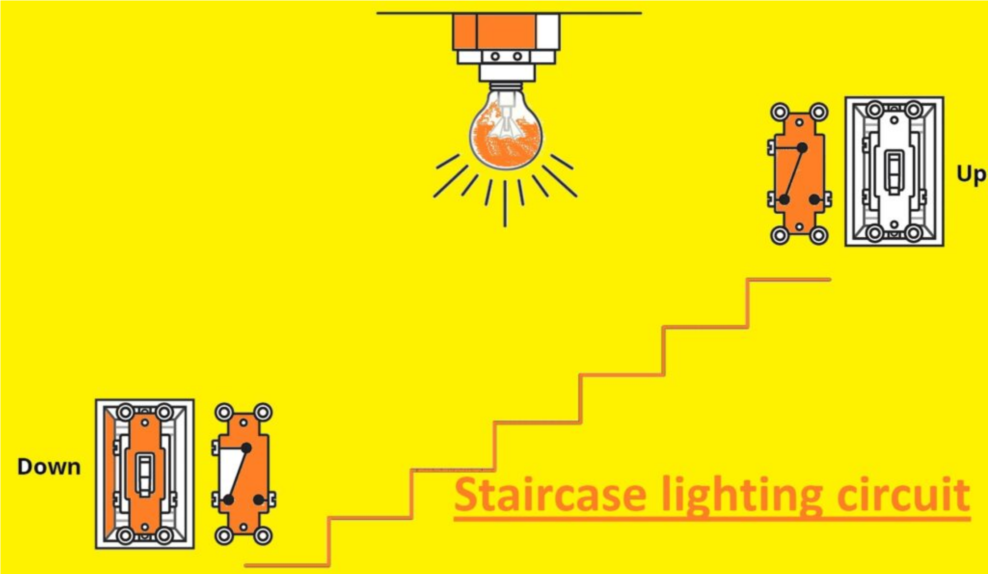
Real World Example

NAND Gate **Real-World Example:**

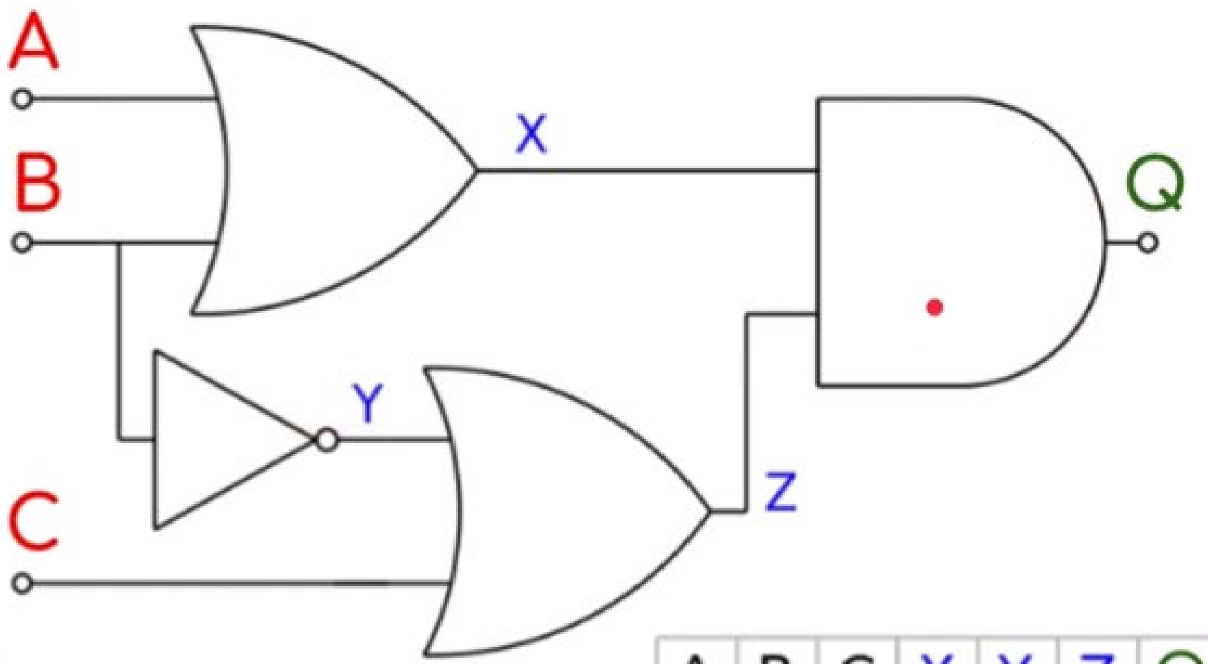
Two-Way Light Switch (Staircase Light)

Pressing either switch changes the light state (ON/OFF).

Logic: Light = Switch1 XOR Switch2



TRUTH TABLES



We mentioned before that multiple logic gates can be connected to form a circuit.

With this example we have 3 inputs (A, B and C) meaning there are a possible 8 combinations leading to the output.

X, Y and Z help us to work through the circuit and obtain the values before the final gate

A	B	C	X	Y	Z	Q
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	1	0	0	0
0	1	1	1	0	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	0	0	0
1	1	1	1	0	1	1

🧩 **Student Activity: "Design Your Own Logic Gate Circuit!"**

🎯 **Objective:**

Students will identify, define, and apply the functions of NOT, AND, OR, XOR, NAND, NOR, and XNOR gates to build a real-world logic circuit and justify its use.

📝 **Instructions:**

1. **Choose a Real-World Scenario**
   Select one from the examples (e.g. motion sensor lights, microwave safety, car alarm, etc.) **or create your own.**
2. **Design the Logic Circuit**
   1. Identify the required logic gates
   2. Draw the **circuit diagram** using correct symbols
   3. Write the **Boolean expression**
   4. Construct the **truth table** (up to 2 inputs)
3. **Explain Your Design**
   1. Why did you choose these gates?
   2. How does your circuit solve the problem?
4. **Bonus (Optional):**
   Create a **digital version** using free simulators like logic.ly or Tinkercad Circuits.

🧠 Example (Teacher Model):

- **Scenario:** Microwave Safety
- **Gates Used:** AND
- **Boolean Expression:** `Run = DoorClosed AND TimerSet`
- **Truth Table:**

DoorClosed	TimerSet	Run
0	0	0
0	1	0
1	0	0
1	1	1

Conclusion

Boolean logic is a powerful tool in computer science and many other fields.

From the basic gates that make up digital circuits to the advanced algorithms used in AI, Boolean logic is essential for understanding and manipulating information.

Sources

Watson, **David**, Williams, Helen. Cambridge **IGCSE computer science**

https://craigndave.org/videos/cambridge-igcse-topic-10-creating-logic-circuits/

https://www.youtube.com/@mrbulmerslearningzone

https://www.savemyexams.com/