# Logic Gate Puzzles

A Fun Resource for the Computing Workshop

## Introduction

This section contains some logic puzzles that help you cement your understanding of how using gates together can check inputs and provide outputs according to the questions you specify. The challenges start with simple examples and move to more challenging questions. Every puzzle has an answer provided in the answer key.

Each puzzle's objective is to solve the question asked, using as few gates as possible. There are many possible solutions, but the solutions focus on one straightforward answer, and sometimes a more clever answer where it exists. All puzzles have similar rules.

- The solution must only use OR gates, AND gates, and NOT gates
- There must be a single output to check for the result (that is, you can't ask the user to check three separate AND gates to see if $AB$, $AC$, or $BC$ are "true")

Some other parameters:

- You can connect signals to more than one gate at a time if you want to
- You can connect inputs to each other if it helps
- You may not have to build a circuit to handle every case

Your first move should be to understand the potential combinations of inputs and how to test them. Don't think about testing all of them at once. Start small. There may not be more than one thing to test, but if you need to, figure out the smallest test that helps you out, and then move on to the next one. You may have to figure out how the results of some tests must be compared: you may need to add AND, NOT, and OR comparisons to some of the results of individual tests.

## Puzzle 1

Which circuit will create the following truth table?

**Truth Table**

| **A** | **B** | **OUT** |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

## Puzzle 2

Design a circuit that tests to see if A is TRUE (has an output of 1) and B is FALSE (has an output of 0). You do not need to check to see if A is FALSE and B is TRUE.

## Puzzle 3

Design a circuit that tests to see if A is *different from* B, no matter what value A has. *Hint: the solution to Puzzle 1 is a huge chunk of the answer.*

## Puzzle 4

Say you had three inputs $A$, $B$, and $C$, and you want to know when a majority of them are ON or TRUE. The question is simple enough: with three inputs, use connected logic gates in a way that can tell when the majority of inputs (here, that's at least two of the three) have a 1 as their input value.

### Setting Up

Start by taking a minute to think of the way to test if some pair of inputs are both 1. We will need to test each possible pair of inputs. That may seem like a big deal, but we can break it down into small chunks.

There's a way to represent all of the possible conditions in a compact way. Let's start by saying a line over the letter indicates "NOT true", and no line over the letter indicates "true".

- If all of the inputs are off, then the condition can be represented by: $\overline{ABC}$.
- When all inputs are on, the shorthand would be $ABC$.
- When only A is off, the condition is represented by $\overline{A}BC$.

We must first find any combination of the three signals that has two or more of them reporting "yes". Then we can determine how we might test the signals. How might you break up the problem into tests of two inputs?

Returning to our setup: how many combinations are there in total?

1. Each input can be 1 or 0 – there are exactly two possible states
2. There are three inputs in total, and each can be on or off
3. Each input is not affected by any other input (the inputs are not tied together in any way)

We need a truth table! The truth table will have $2 \times 2 \times 2$ (that is, $2^3$) possible combinations. So let's make a table by counting from 0 to 7 in binary. We will assign one of the A, B, or C inputs to each column ($2^0$, $2^1$, and $2^2$).

| A | B | C | Symbol |
|---|---|---|---|
| 0 | 0 | 0 | $\overline{A}\overline{B}\overline{C}$ |
| 0 | 0 | 1 | $\overline{A}\overline{B}C$ |
| 0 | 1 | 0 | $\overline{A}B\overline{C}$ |
| 0 | 1 | 1 | $\overline{A}BC$ |
| 1 | 0 | 0 | $A\overline{B}\overline{C}$ |
| 1 | 0 | 1 | $A\overline{B}C$ |
| 1 | 1 | 0 | $AB\overline{C}$ |
| 1 | 1 | 1 | $ABC$ |

To cover every possible case, half of of each column will be "true" and half will be "false". For the left-most column ("A"), the first four column values are 0 and the next four are 1. The right-most column alternates "on-off" four times, and the middle column alternates off and on every other row. It looks like we are counting in binary from 0 to 7, because we are! That way we will know we have covered every possible condition that we have to test.

The table shows us all possible conditions for these three inputs. Now add a results column, showing the answer our logic circuit should produce. Remember, if any two inputs in the row are true, then the result column will be true. Otherwise, the result is false.

| A | B | C | Symbol | Result |
|---|---|---|---|---|
| 0 | 0 | 0 | $\overline{A}\overline{B}\overline{C}$ | 0 |
| 0 | 0 | 1 | $\overline{A}\overline{B}C$ | 0 |
| 0 | 1 | 0 | $\overline{A}B\overline{C}$ | 0 |
| 0 | 1 | 1 | $\overline{A}BC$ | 1 |
| 1 | 0 | 0 | $A\overline{B}\overline{C}$ | 0 |
| 1 | 0 | 1 | $A\overline{B}C$ | 1 |
| 1 | 1 | 0 | $AB\overline{C}$ | 1 |
| 1 | 1 | 1 | $ABC$ | 1 |

The fourth, sixth, seventh, and eighth rows all have two or more inputs set to "on". The results ("R") column is 1/"true" for these four rows, and 0/"false" for the rest.

The logic can be expressed like this:

($A$ AND $B$) OR ($A$ AND $C$) OR ($B$ AND $C$) OR ($A$ AND $B$ AND $C$).

So if any of these individual conditions is true, the result is true. Note that all three values do not have to be true for the result to be true. Any two true inputs gives the same result as all three inputs being on. So there is no need to specifically test for all three inputs being true.