Kozell YON-BROOKS

5727

51319

boolean logic circuit simulator

AQA Computer Science NEA

[Date]

Contents

[Analysis 2](#_Toc87990561)

**[The Problem](#_Toc87990562)** [2](#_Toc87990562)

# Analysis

## The Problem

GCSE Computer science is where a student’s introduction to logic gates may happen, this is a part of the curriculum but also helps to further a student’s logical thinking which can help in other areas of the subject, such as programming. A way to help me better understand logic gates were logic gate simulators, these simulators allowed me to create simple circuits using basic AND, OR, XOR, and NOT gates. These current simulators are too complex for what is needed for GCSE computer science and lack the teaching tools that my proposed system will include.

## Overview of Logic Gates

Logic gates are a model of computation that take one or two inputs and returns a single output based on the gate's logical operation / Boolean function, they are the fundamentals of logical circuits and physical logic gates made of diodes and transistor are what allow computers to work. Logic gates can be combined to produce a certain output based on the inputs of the circuit. An AND gate (Figure 1) for example will take two inputs and return a True output if both inputs are True, and a False output otherwise. Inputs and outputs can be True or False as they are Boolean, this is usually represented as a 1 and 0 for True and False respectively. Certain gates such as the Not gate will only need one input. There are other parts to the logic circuits besides the gates, switches, constant inputs, and clocks can provide initial inputs. Output can be handled by simple ‘bulb’ that’s on for True and off for False. A more complex output such as a 4-bit digit would produce an integer output based on a binary sequence from 4 Boolean inputs.



Figure 1: AND Logic Gate

A truth table displays all the possible inputs and outputs of a logical operation in tabular form. Figure 2 shows a truth table for the Logic operations AND, OR, XOR, and NOT. These are the only gates that are needed for the AQA course as per the specification (Figure 3).

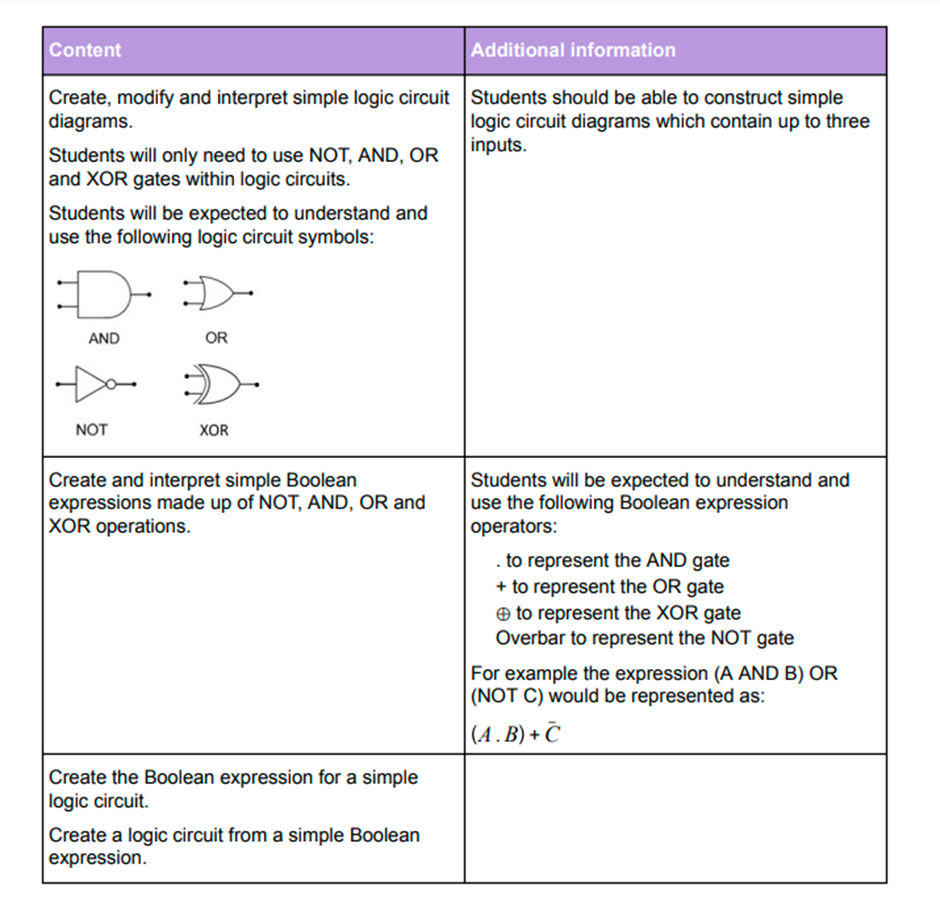
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** |  | **Output** |  |  |  |
| A | B | AND | OR | XOR | NOT (Input only B) |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

Figure 2: Truth Table

Figure 3: AQA GCSE Specification for Boolean Algebra 3.4.2

As these are the only gates needed for the AQA course, they will be the only gates included in order to simplify the program for the students. My solution would be more focused towards teaching, specifically what’s needed for the AQA GCSE specification. I had a short conversation with Mr Flynn about what features would be make my program relevant to the GCSE course. These include:

* Abstracting logic gates that aren’t needed.
* Including a check list to determine if the user has used each of the gates at least once.
* A truth table generator.
  + To display the results of a circuit as a truth table.
  + A truth table checker which will compare a truth table created by the student to the one produced by the program to check that the student has understood the circuit.
* Converting a written Boolean expression into a truth table.
* Displaying the Boolean expression for the created circuit.
* Allowing individuals to save and load circuits to and from their computer’s local storage.



Diagram

Description automatically generated

Figure 4: Object diagram for possible Logic Gate solution