*https://bournetocode.com/projects/AQA-A-NEA/pages/AQA%20Computer%20Science%20A%20level%20NEA%20Guidelines-%20Analysis.html*

## The Problem:

GCSE Computer science is where a student's introduction to Boolean logic happens. A visualisation of the logic helped me to understand how the basic gates interacted and combined to make complex circuits. An interactive Boolean Logic Circuit would help the students to further understand the aspects of Boolean circuits, and how the simple AND, OR, or NOT gates that they are introduced to can create multiple switches controlling one output to complex systems like computers.

A logic gate takes two inputs and returns an output based on the gate's logical operation. A truth table displays all the possible inputs and outputs of a logical operation. Here is a truth table for the Logic operations ‘And’, ‘Or’, ‘Xor’ and ‘Not’ to show what the outputs of each gate would be based on the inputs given.

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

The end-users will be Highdown computer science students. Highdown uses computers running Windows 10, this is the operating system that my program should be able to run on. The supervisor will

I searched the web for different logic simulations that were simple enough for someone just starting computer science to understand while also being user-friendly and able

***• Required documentation for ‘Analysis’: this section of the report should include:***

* ***A clear statement that describes the problem area and the specific problem being solved / investigated***
* ***An outline of how the problem was researched***
* ***A statement indicating who the problem is being solved / investigated for***
* ***Background in sufficient detail for a third party to understand the problem being solved / investigated***
* ***A numbered list of measurable, "appropriate" specific objectives, covering all required functionality of the solution or areas of investigation (‘appropriate’ means the specific objectives are single purpose and at a level of detail that is without ambiguity)***
* ***Any modelling of the problem that will inform the Design stage, for example a graph/ network model of Facebook connections or an E-R model, state diagrams, scientific / mathematical models or formulae, data flow diagrams.***

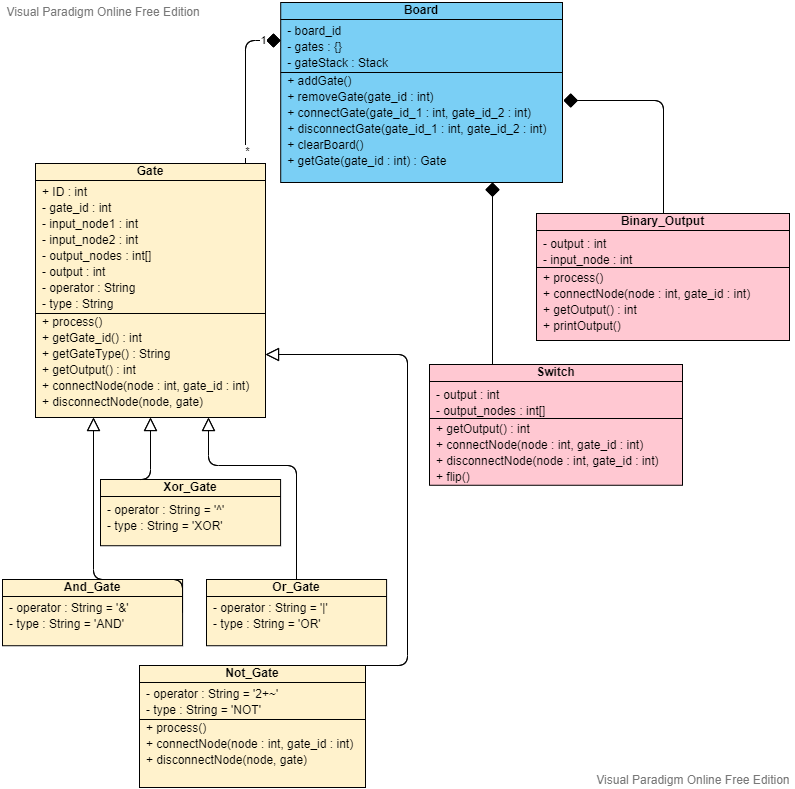
Solution would include an easy-to-understand user interface allowing user to drag and drop gates onto a canvas. Dragging lines between gates would link them. Other elements such as constant inputs, outputs, switches, and more complex gate combinations would also be available.

Canvas should be scalable and allow groupings of gates to be dragged together. Canvas should also produce a file of all gates used and how they link to allow saving.

* Make UML OOP Diagram
* Analysis redo
* Project is no longer web app
* Research kivy or alternative
* Potential solutions will be web application (use django or flask, have to learn javascript, html css), android application (have to learn javascript or kotlin or find way with python), desktop application (this probably, windows 10, has to run on school computers, most if not all computers used for computer science have python installed so will not need to make compiled version?, )
* Final solution will be desktop application
* Potential frameworks:
  + PyQt5.
  + Tkinter.
  + Kivy.
  + wxPython.
  + PySimpleGUI.

Chosen kivy, potential for multiple platforms

* Research clock for logic processing because flip flops don’t work because of recursion error because of the way it works atm because

Class Diagram for logic gates:

Objectives:

1. Make a Logic Gate object
   1. Should be able to take and store two inputs.
   2. Should be able to evaluate the Boolean equation based on its inputs and the gates type.
2. Make a Board
   1. Should be able to create and delete Gate objects.
      1. Creates gate object
      2. Adds gate to a dictionary with a gate\_id key
      3. Adds gate\_id to a stack
   2. Should be able to connect and disconnect Gate pairs.
   3. Should be able to use the stack of gate\_ids
3. D

Why my solution is better:

Check that they have used all the gates once

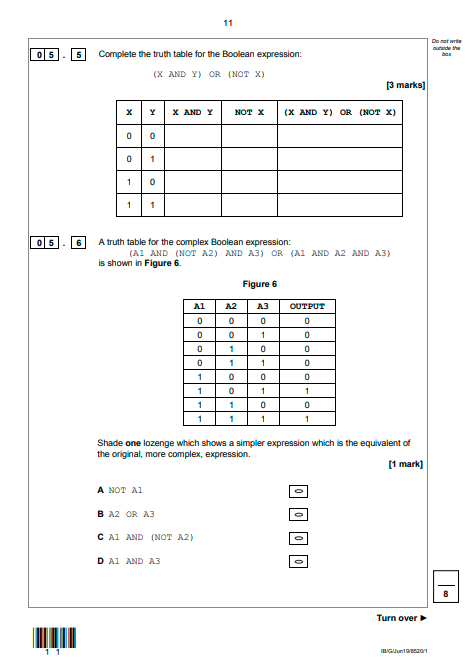
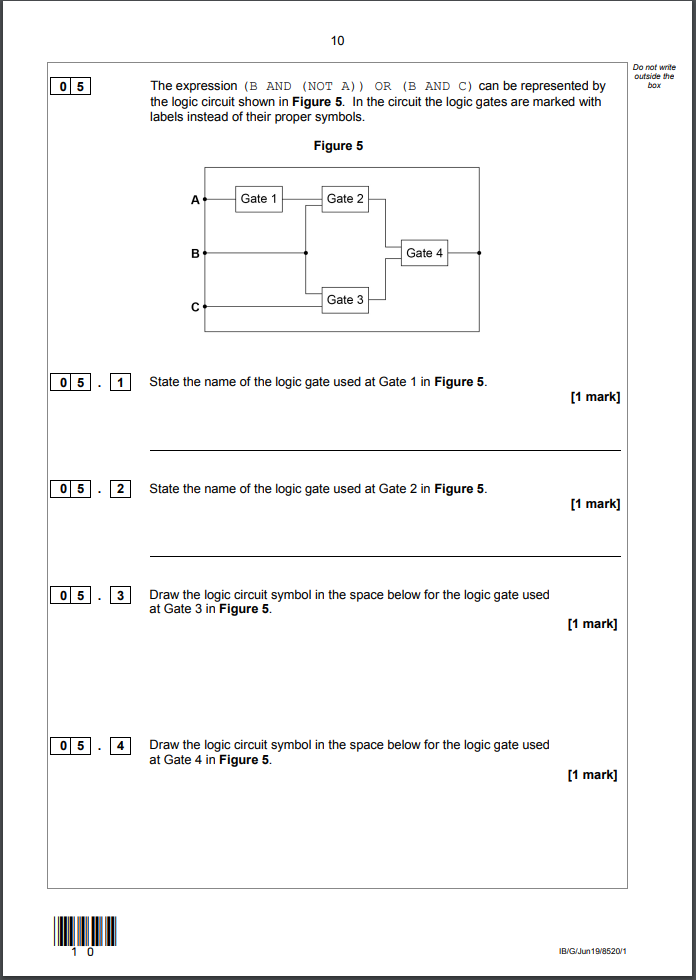
Truth table generator, Ask student to create logic circuit based on truth table, check if its correct, vice-versa

Convert written boolean expression to truth table, and maybe circuit

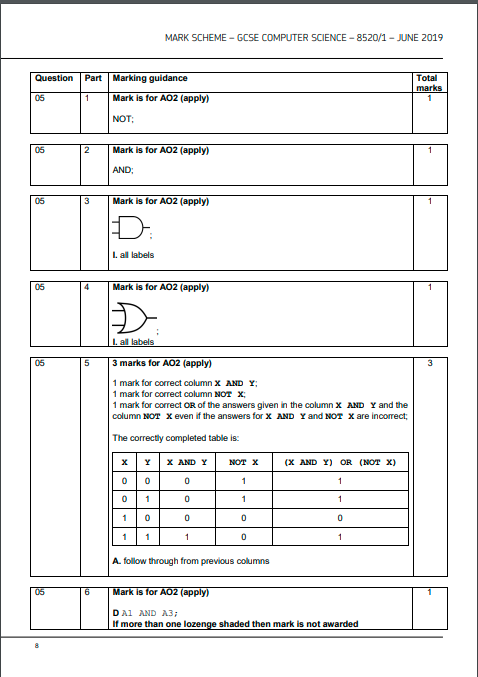
Check GCSE spec to find out what gates are needed

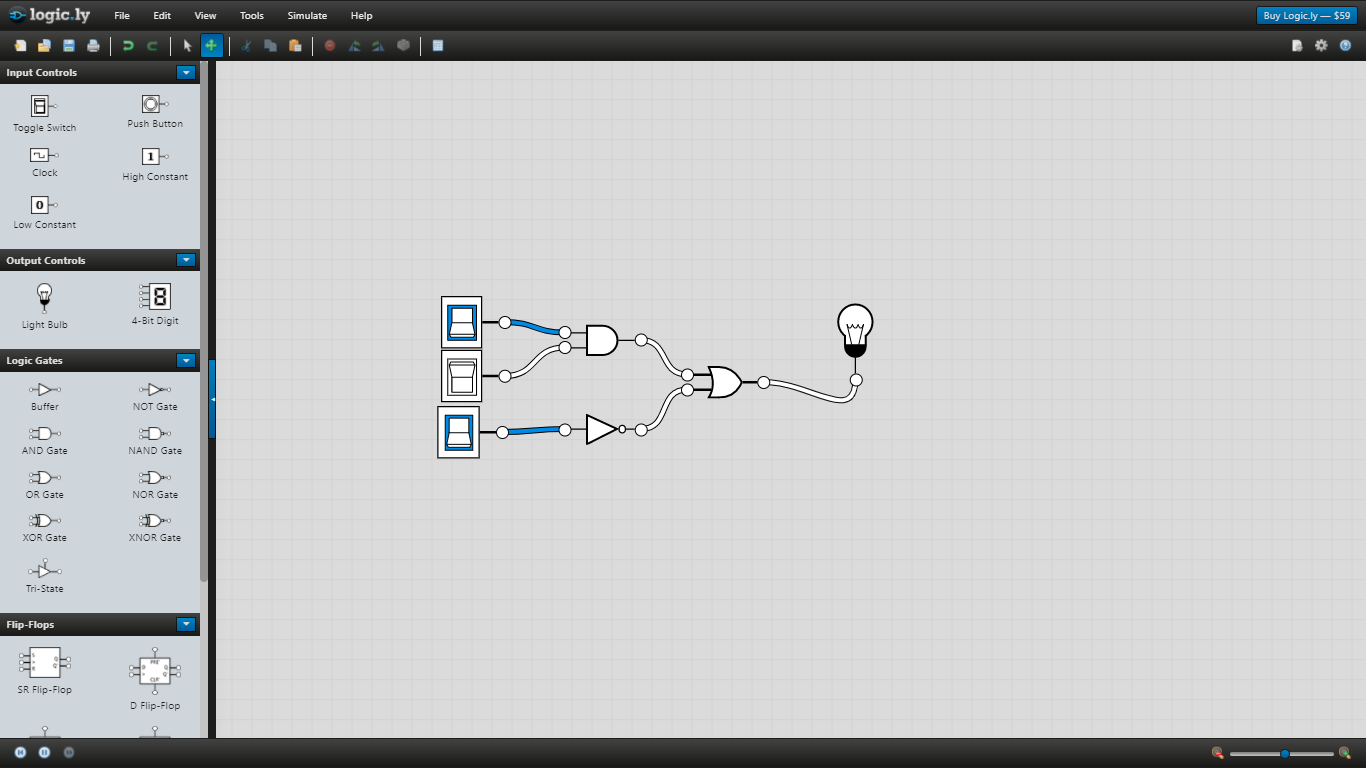
Don’t add gates which aren’t needed for the course

Add saving function

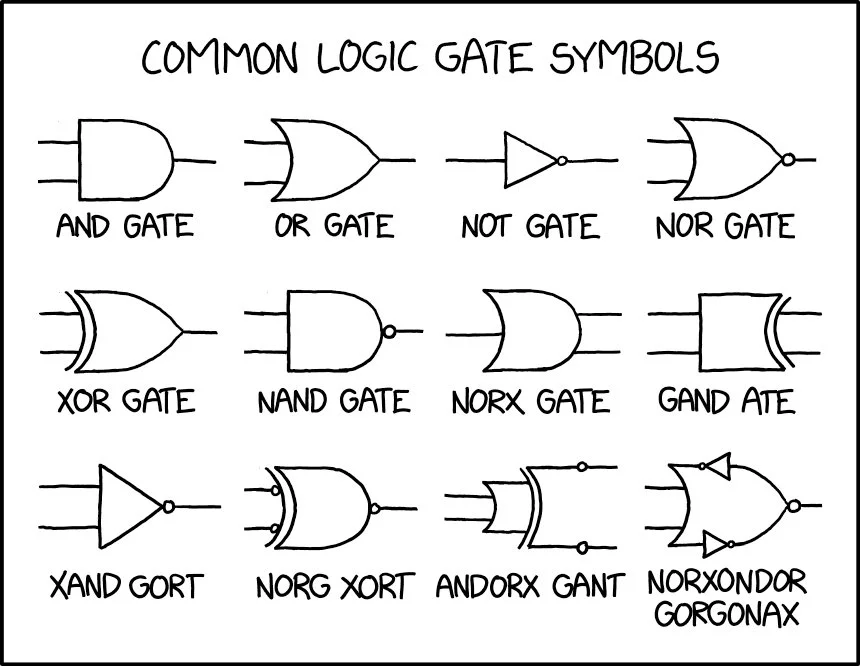


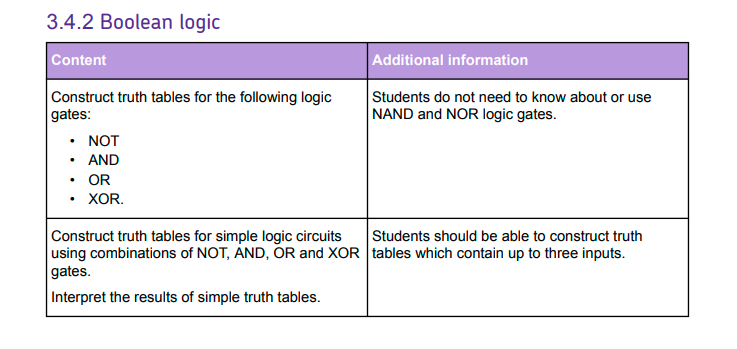
Example question on logic gates from AQA GCSE Computer Science June 2019 (8020/1) Paper 1 – Computational Thinking and Problem-Solving Question 5

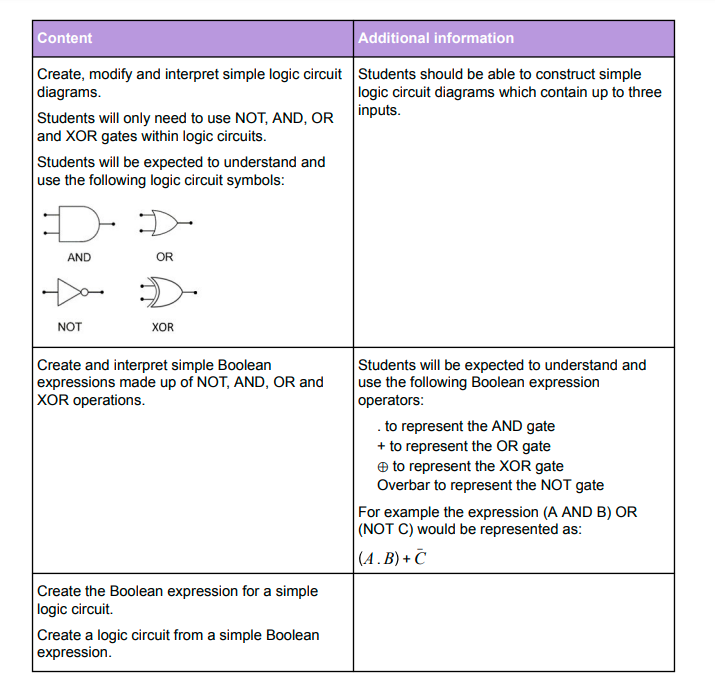
 mark scheme



<https://logic.ly/demo/> - Example of logic simulator





Aqa gcse spec for logic gates taught 2020 onwards