Requirements and Analysis Document for Open Logic Gate Simulator

Jimmy Andersson
Jacob Eriksson
Martin Hilgendorf
Elias Sundqvist

1 Introduction

1.1 Background

A huge benefit of choosing a software based career is that you can work from anywhere as long as you have a laptop and an internet connection. However, when it comes to hardware this is not always the case. In particular, when it comes to building computers it can get expensive real quick. For those students interested in learning how computers work, but do not want to invest in real hardware, there should be an application that helps understand how computers work.

1.2 Idea

Our approach to the problem is to develop a simulation software called Open Logic Gate Simulator. The purpose of this software is to simulate how low-level logic components work inside of a computer. It will try to emulate the reality as close as possible, without introducing the complexity of electronics theory to the user. It can be used by both students, teachers and hobbyists to enhance the learning experience, and to get a better intuitive understanding of how logic circuits work.

In practice, this is done by providing an interactive workspace where basic logic gates can be placed, moved and connected. Furthermore, more advanced components (such as clocks) can be placed, and the behaviour of the circuit can be simulated over time, allowing for the construction of more advanced circuits, such as small microprocessors.

The application will be a standalone desktop application with a graphical user interface for the Windows / Mac / Linux environment. It will be scalable and adapt to different screen sizes.

2 Requirements

2.1 User Stories

As a User I want to design and simulate basic electrical circuits to get an intuitive sense of what is happening at the lower level of computing.

---

1. As a User I want to see all components, so that I can get an overview of what's available.

- Create a rectangle looking panel that shows the name of the component
- Add a picture to the component
- Create a list object that appends these panels from top to bottom.
- Make the list object scrollable.

2. As a User I want to be able to see details about each available component, so that I can gain insight into what they're doing.

- Add name and title to components
- Create a GUI object that can display such information
- Show this object when a component is being hovered (from the list)

3. As a User I want to be able to search for a specific component, so that I can quickly find the component I need.

- We need to create a text field GUI object
- Add filter function to the list object that displays the components.
- Call the filter function onEnter input from the search field

4. As a User I want to be able to place components on a workspace, so that I can use them.

- We need a Workspace object that can store the components and their coordinates (so a wrapper will be needed)
- We need a Camera that shows a viewport of the workspace. The camera does NOT need to have movement yet (I.e. panning).

- We need to be able to drag components and show that we are dragging them, by attaching their icon to the mouse while dragging.
- Workspace will need to listen for mouse release in case a component is being dragged.
- On release, place component on workspace (I.e. add to list and such)

5. As a User I want to be able to get a sense of how big a component actually is and where exactly it will land before placing it, so that I can place components nicely and efficiently.

- When dragging over the workspace, show a “real size” scaled preview of the component.

6. As a User I want to be able to pan across the workspace so that I can utilize more space for my circuits.

- Add a tool to switch between normal/pan mode
- When in pan mode, move camera on mouse drag

7. As a User I want to be able to connect components, so that I create more complex structures.

- We create a Wire object that connects components and is created by clicking on 2 endpoints.
- We want to draw a wire between the connections
- We want to show a preview when the mouse is looking for an endpoint
- Pressing ESC should cancel the wire creation and thus remove the preview

8. As a User I want to be able to remove components, in case I mess up.

- Add wire tool that deletes entire wires
- Change so that the wire tool only deletes the selected wire until forks are reached.

9. As a User I want to get visual feedback from connections that are active (1/on) and ones that are not (0/off).

- We draw black lines to show off state
- We draw red lines to show off state

10. As a User I want to be able to include switches in my circuit, so that I can interactively connect/disconnect different components.

- We create a component that can either transparent the signal or block it (we determine which through code).
- We add the functionality to change the state by clicking on the switch.

11. As a User I want to have a component that gives a signal source, since I can power my circuits that way.

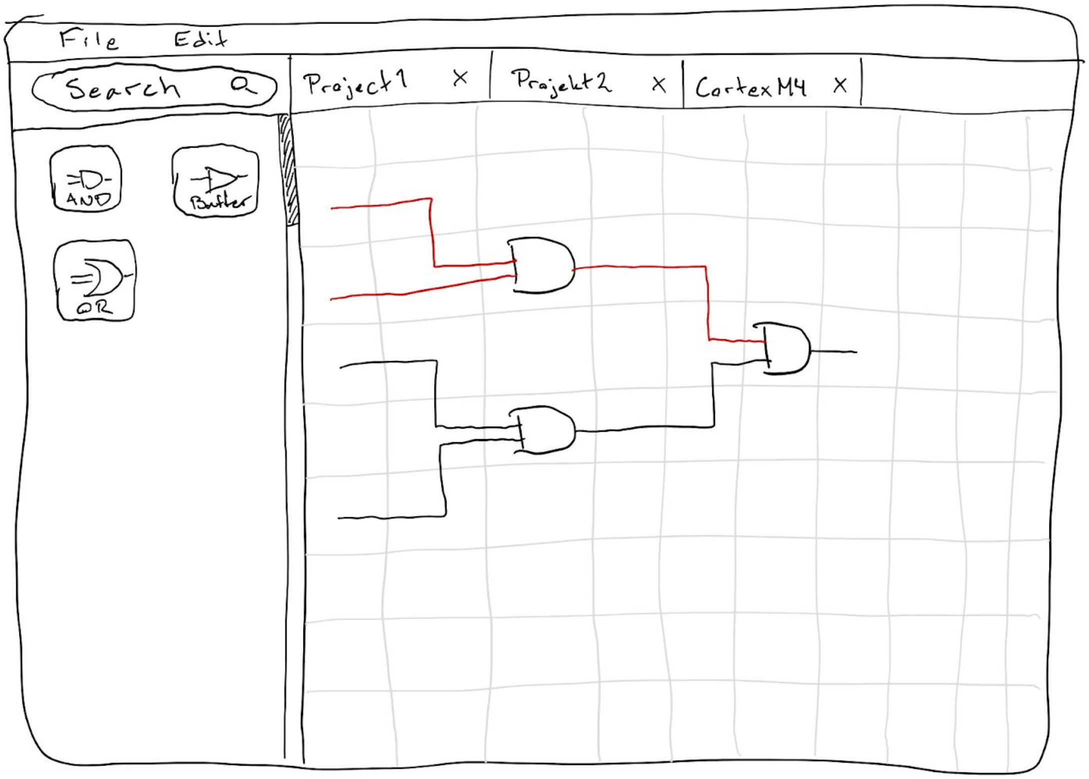
12. As a User I want to have a pulse clock so that I can produce dynamic circuits.

13. As a User I want to have a access to logic gates such as AND, OR, NOT, XOR, NOR, NAND.

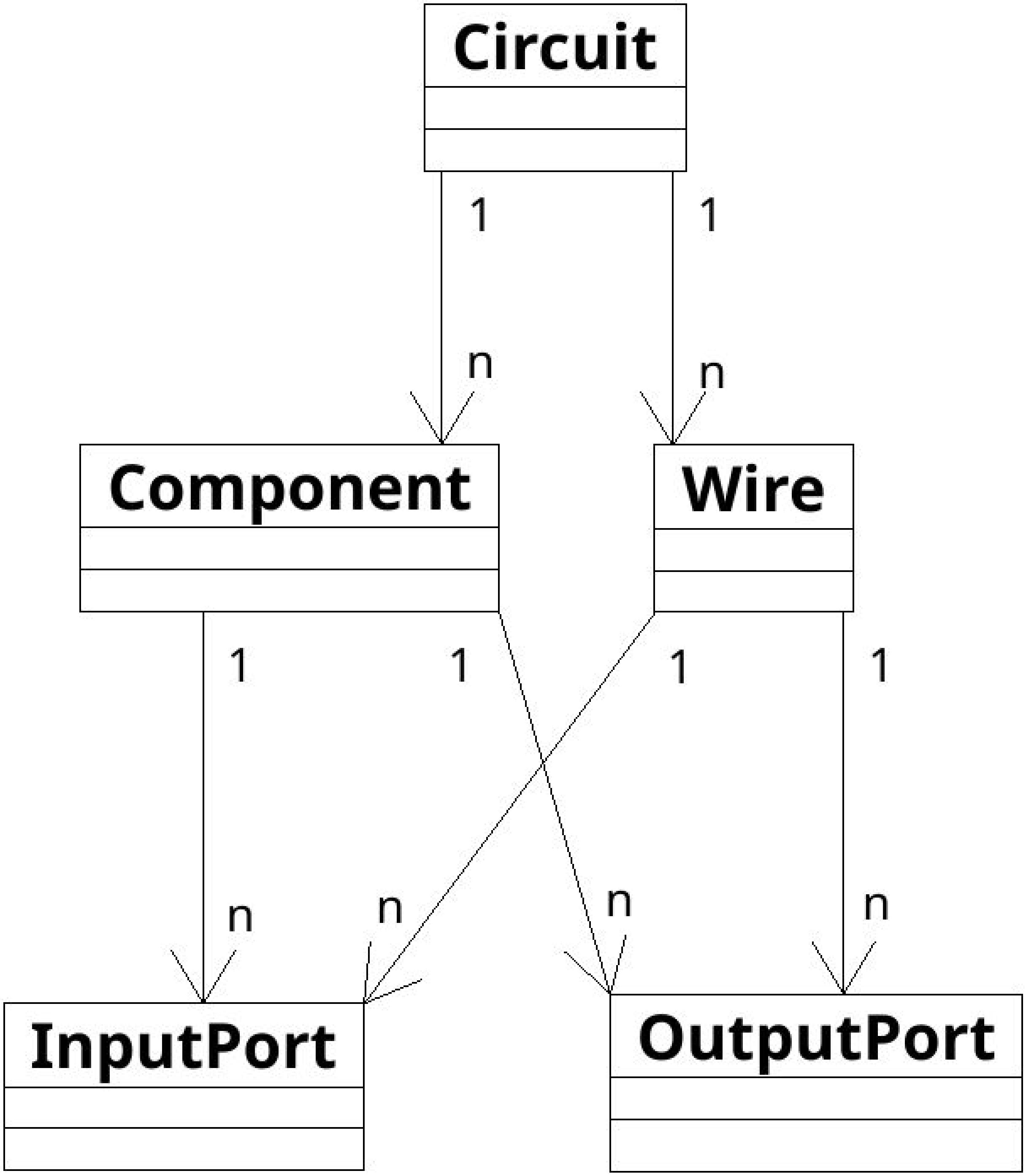
14. As a User I want to have display components so I that can show numbers.

15. As a User I want to have components with various abstraction levels so that I can quickly get started from the abstractions level I'm currently interested in.

2.2 User Interface



3 Domain model



3.1 Class responsibilities

- Circuit: Simulates the power flow between wires and components. Power flow in this sense is synonymous with logic state.
- Component: Modifies the logic state of output ports, based on input ports' logical states.
- Wire: Transmits logic states from all connected output ports to all connected input ports.
- InputPort: Reflects the logic state of the connected wire, without allowing modification to the wire's logic state.
- OutputPort: Sets the logic state of the connected wire.