

# Project's Name

## Functional Specification Document

DOCUMENT VERSION 2.0

02/28/2025

### Author

Name	Role
Alexis SANTOS	Program Manager

### Document History

Date	Version	Document Revision Description
02/24/2025	1.0	<ul style="list-style-type: none"><li>Created the document and wrote sections Overview, A to E.</li><li>Started the Glossary.</li></ul>
02/25/2025	1.1	<p>After meeting with the client, some parts have been updated.</p> <ul style="list-style-type: none"><li>Filled customers-relative sections</li><li>Partially wrote Functional Requirements.</li><li>Added Expected Deliverables.</li></ul>
02/27/2025	1.2	Made a lot of corrections and rewriting to have a clear document.
02/28/2025	1.3	<ul style="list-style-type: none"><li>Added the Test Cases Analysis</li><li>Added Personas</li></ul> <p>Added Non Functional Requirement Fix issues between Functional and Call for Tender</p>
03/03/2025	2.0	<ul style="list-style-type: none"><li>Rewrited A and B</li></ul>

### Approvals

Approval Date	Approved Version	Approver Role	Approver Name	Approvement Status

► Table of Content

- I. Overview
  - A. Product Description
  - B. Product Functional Capabilities

- Visualization
    - Examples
    - Time Control
    - Compatibility
    - Security
  - C. Deliverables
  - D. Project Organisation
    - Project Representatives
    - Stakholders
    - Project Roles
    - Expected Deliverables
  - E. Project Plan
    - Milestones
    - Ressources / Financial Plan
  - II. Requirements
    - A. Functional Requirements
      - FPGA Simulator
        - Teacher interface / Backend
        - Student interface / Frontend
      - Three Views
        - All Views
        - Sematics Representation
        - Signals Graph
        - FPGA board's Schematic
      - Personas Definition
      - Use Cases Analysis
    - B. Non Functional Requirements
  - Glossary
- 

## I. Overview

---

Our client, Florant MANNI has solicited us to develop a web interface for an FPGA Simulator<sup>[1]</sup>. This web interface<sup>[3]</sup> will be used to teach people how signals propagated inside an FPGA<sup>[2]</sup>. For simplicity, we will represente the board with differents views (define in the document) to understand the FPGA system through animations. Those animations mainly symbolize signals transfers.

### A. Product Description

The aim of this web application<sup>[4]</sup> is to help Florant MANNI and/or his team to teach to new FPGA users how the system works. The application has to combine a more or less realistic 2D representation of an FPGA with dynamic signal propagation over time. This includes integrating the layout resulting from the synthesis<sup>[5]</sup> and placement and routing (P&R)<sup>[6]</sup> processes with timing simulation data. The aim is to create an interactive and intuitive platform that allows users to observe and analyze signal propagation using a testbench<sup>[7]</sup> and a timing netlist<sup>[8]</sup>, both written in Verilog<sup>[9]</sup>.

## B. Product Functional Capabilities

The product, responding to customer demands, should be able to have a lot of functionality.

### Visualization

As a web application, the project must have an interface. This interface should be a 2D View where the user must have the ability to navigate through the view to visualize what happened in the **FPGA system**.

This graphical model should illustrated : \* An arrangement of basic elements(BELs)[^10], \* Connections dictated by a Verilog file,

Therefore, the model should reflect the post-synthesis and place-and-route(P&R) design.

However, to understand what happened in this system, every signals must be represented graphically with animations dictated by an SDF (standard delay file). These signals must be: \* The propagation of a clock signal \* The propagation of data

### Examples

An interface or view without content is useless. That's why we have examples. These examples are Verilog files that we can use to provide the views with something to focus on. Thanks to them, we can visualize the basic uses of the FPGA system.

However, these examples couldn't be enough. It is for that why the customer would to have the ability to add other Verilog files to become additional examples.

### Time Control

This web app has an educational goal. Hence, any animation should be able to be manipulate to watch again animations, slow it down, or speed it up. To acheive this objective, we should have a button(s) to **resume**, **pause**, and/or **reset** animations. To manipulate animations' speed, we should have an option to select which speed we want.

### Compatibility

Chrome, Edge, Safari, Firefox... A lot of browser exist in the market. As a consequence, our web application should be usable on every web browser existing on the market. For that, we should use in the web app only components and styles recognized in each browser. Therefore, the web app should be responsive to different screen sizes.

### Security

**If the web app will be share on the web**, we should implement security measures to protect user data and ensure the integrity of the application.

## C. Deliverables

Name	Type	Deadline	Link
------	------	----------	------

Name	Type	Deadline	Link
Functional Specifications Document	Document (Markdown)	03/13/2025	<a href="#">functionalSpecifications.md</a>
Technical Specifications Document	Document (Markdown)	03/25/2025	<a href="#">technicalSpecifications.md</a>
Test Plan Document	Document (Markdown)	03/25/2025	<a href="#">testPlan.md</a>
Development	Programing (HTML / CSS / JavaScript)	04/01/2025	
Weekly Reports	Document (Markdown)	Every Friday	<a href="#">Weekly Report Folder</a>

## D. Project Organisation

### Project Representatives

Project Owner	Represented by...
Florant MANNI	Represented by himself
Loïc NOGUES	Represented by Alexis SANTOS (Program Manager)

- Defining the vision and high-level objectives for the project.
- Approving the requirements, timetable, resources, and budget (if necessary).
- Authorizing the provision of funds/resources (internal or external) (if necessary).
- Approving the functional and technical specifications written by the team.
- Ensuring that major business risks are identified and managed by the team.
- Approving any major changes in scope.
- Receiving Project Weekly Reports and taking action accordingly to resolve issues escalated by the Project Manager.
- Ensuring business/operational support arrangements are put in place.
- Ensuring the participation of a business resource (if required).
- Providing final acceptance of the solution upon project completion.

### Stakeholders

Stakeholder	Might have/find an interest in...
Florant MANNI	Needs a tool to help him to explain Verilog/VHDL system to young FPGA users.
ALGOSUP Students	Learning the association between web languages (HTML/CSS/JavaScript)[^11] and Verilog.

### Project Roles

As defined at the beginning, the team is arranged as follows:

Role	Description	Name
Project Manager	Responsible for organization, planning, and budgeting. Keeps the team motivated.	Loïc NOGUES
Program Manager	Ensures the project meets expectations. Responsible for writing the Functional Specifications.	Alexis SANTOS
Technical Leader	Makes technical decisions in the project. Translates the Functional Specification into Technical Specifications. Conducts code reviews.	Yann-Maël BOUTON
Software Engineer	Writes the code. Participate in the technical design.	Lucas MEGNAN
Quality Assurance	Tests all functionalities of the product to find bugs and issues. Documents bugs and issues. Writes the test plan. Checks that issues have been fixed.	Mathis LEBEL
Technical Writer	Writes the user manual. Participate in the technical design.	Grégory PAGNOUX

## Expected Deliverables

For this project, the customer has requested us to meet the following deliverables:

- The source code need to be on a Git repository.
- An explainaton of how to run the software must be present on the repository and/or in the website.
- An explanation of how to add an application example must be present on the repository and/or in the website.
- The website must include at least two application examples:
  - One for the flipflop[^12].
  - One for the LUT4[^13].

## E. Project Plan

### Milestones

Milestones	Deadline
Functional Specifications V1	03/13/2025
Technical Specifications V1	03/25/2025
End of Development	04/01/2025
Oral Presentation	04/04/2025

### Ressources / Financial Plan

We have an estimated total of 70 man-hours to complete this project. We have access to the following resources:

- The team (6 people)
- Teachers
- A computer per team member

This project focuses on creating an animation of an FPGA Simulator to explain how an FPGA system works.

## II. Requirements

---

### A. Functional Requirements

This project focuses on creating an **animation** of an FPGA Simulator to explain how an FPGA system works.

#### FPGA Simulator

The project is based on an FPGA simulator. This one needs to follow these requirements:

- A **2D floor** plan with two/three views.
- The ability to see the path taken by signals in the board

However, the client requested two interfaces/main functionalities on this web app: The student interface and the teacher interface.

#### Teacher interface / Backend

The teacher interface is to be used by the teacher for several reasons:

- The ability to **add another example** with their own Verilog file.
- Access to a testbench.

#### Student interface / Frontend

The student interface allows the student to explore and discover FPGA properties and functionalities.

However, this view needs to follow these requirements:

- The 2D view of FPGA Simulator.
- Navigation within the view with zoom and move functionalities.
- Selection of the default example or the teacher's example and viewing it in a specific view.
- Selection of one of the three views: the components schematic, binary graphic, or board.
- A play button to pause or resume the animation.
- A button to select the speed of the animation.

#### Three Views

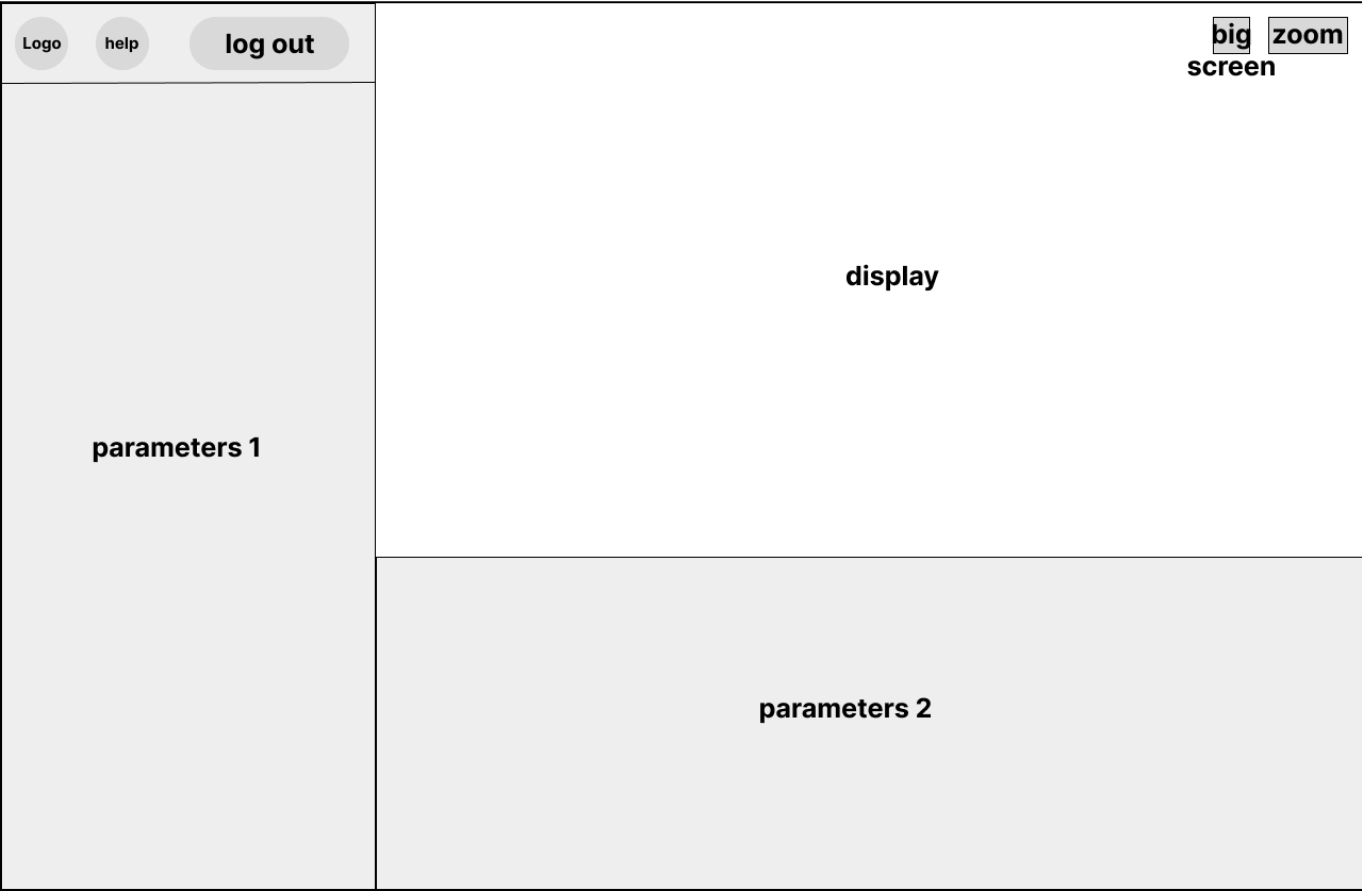
To provide a comprehensive amount of information, the student needs access to three views: a schematic representation of components, a graph of signals, and an FPGA board's schematic.

#### All Views

In this three views, the user has the following features :

2D View & Navigation	Time Control	Example used & Visualization
<ul style="list-style-type: none"><li>• Navigate trough the screen with delimitation.</li><li>• Zoom on the screen to see more details or zoom out to view the global schematic.</li></ul>	<ul style="list-style-type: none"><li>• Pause and resume button to stop or continue the animation.</li><li>• Controller/button to select the speed of the animation with the following choices: 0.5 / 0.75 / 1 / 1.25 / 1.5 / 2 / 4</li></ul>	<ul style="list-style-type: none"><li>• Choose the exercise between the default example and the example from the file provided by the teacher</li><li>• Select which view to display information from among the three views.</li></ul>

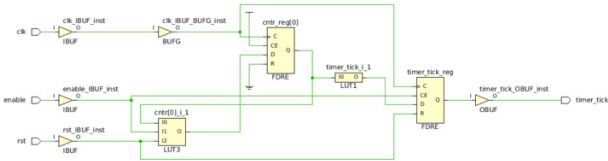
To organize the screen, we'll follow the model below.



For the next three views, they will displayed on the **display part**.

Shematics Representation

The schematic view is a **2D diagram** representing components such as flip-flops, BRAMs, or LUT4s. The aim of this view is to **understand** how data and clock signals move through the components and how the program interacts with them from start to finish.



The movement of data and clock signals is **illustrated** through animations. For each component, the following information must be displayed:

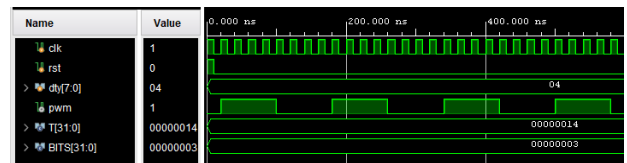
- The type of component (LUT, BRAM, Clock, etc.)
- The ID of the component (e.g., Q1, Clk, etc.)
- Inputs and outputs

Additionally, the following information must be displayed between components:

- The delay in signal transmission from one component to another
- The type of data being transferred (clock, status, etc.)

## Signals Graph

FPGAs, like other digital systems, use the **binary system**. Each component has a binary state or is used to change the binary state. Therefore, it is important to **monitor the status** of these components.



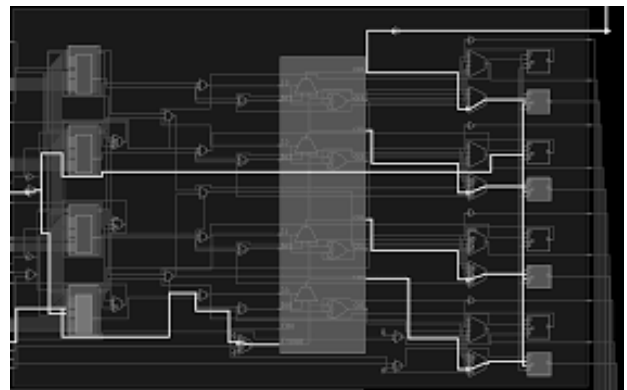
With this view, we should be able to see the clock status and the one of each component. This would be useful for observing how many clock cycles each step of the program takes. With this information, students can better **understand** why certain events occur.

## FPGA board's Schematic

The FPGA board's schematic is a **2D floorplan** used by the student to understand how the program sends data and clock signals.

In this view, the student needs the following features:

- An animation to show these clock and data signals.
- Different color between data and clock signals to distinguish them.
- Components used by the program would be glowing.



However, in the conception of the schematic, we need to strike the **best balance between reality and schematic representation** to make it realistic and user-friendly.

## Personas Definition

### Florent Manni

- **About**
  - **Job Title:** Electronic Engineer for over 20 years
  - **Experience:** 20 years of experience
- **Description**
  - Florent Manni is a Senior Electronic Engineer who has requested the team to develop the project. He has extensive experience in FPGA engineering. Typically, he teaches junior engineers the FPGA



system to enhance their productivity. He has requested this tool to upgrade his support capabilities.

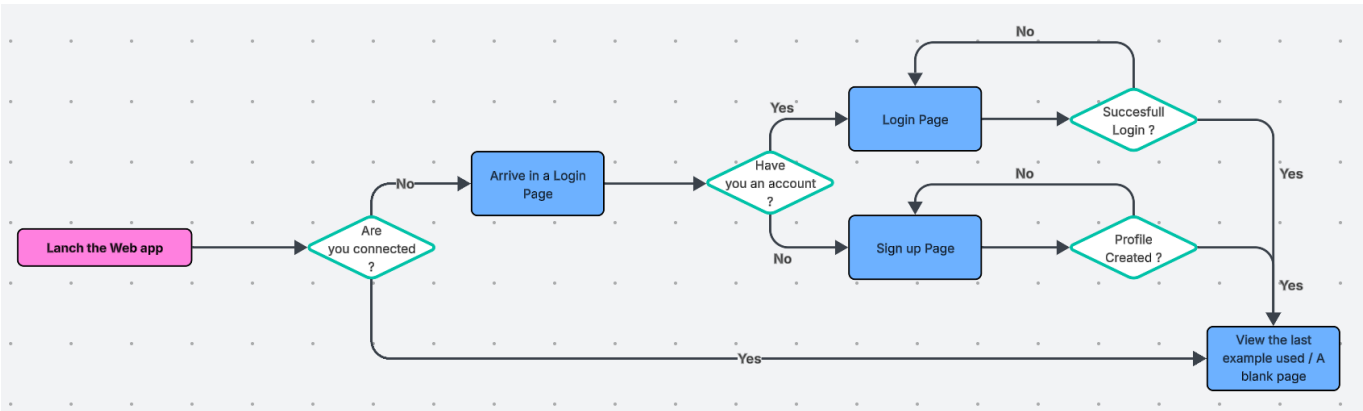
Alexandra Dupont

- **About**
  - **Job Title:** Junior Electronic Engineer
  - **Experience:**
    - Recently graduated with a Bachelor's degree in Electrical Engineering
    - Limited hands-on experience with FPGA, but eager to learn and grow in the field.
- **Description:**
  - Alexandra is a recent graduate who has joined the team as a Junior Electronic Engineer. With a strong foundation in electrical engineering concepts, she is excited to dive into the world of FPGA. Alex is highly motivated and eager to learn from experienced colleagues like Florant Manni. She is looking forward to contributing to innovative projects and expanding her skill set in FPGA design and implementation.

Use Cases Analysis

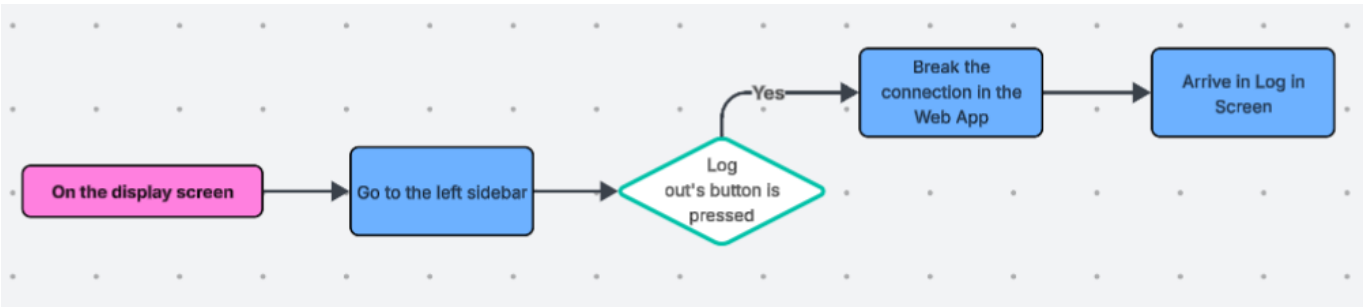
Start the Web App

When we start the Web app, firstly the program check if you are connected. If yes, you look at your last example used. If not, you arrive in a log in/sign up page.



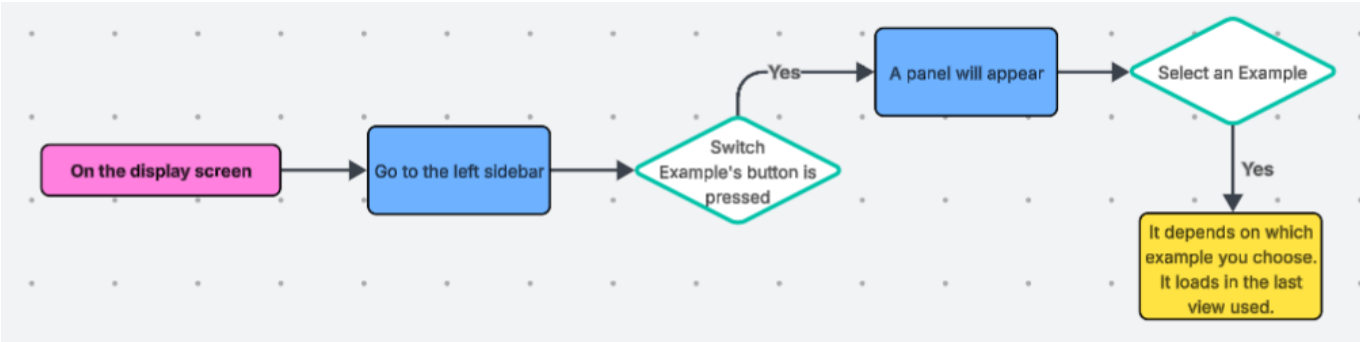
Log out

Above the sidebar located in parameter 1 of the model, we have a **log out button**. If we press it, we should be disconnected and a login page should be displayed.



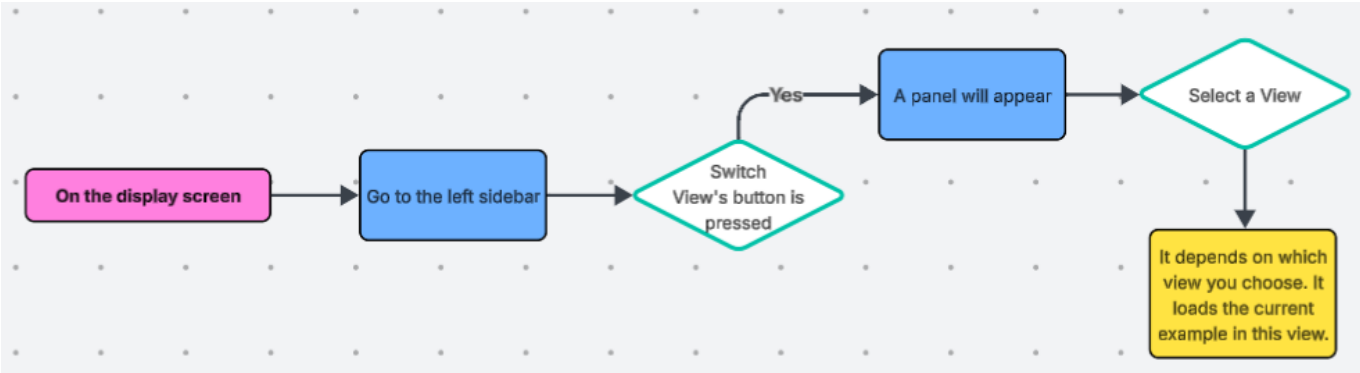
Switch Example

A sidebar located in parameter 1 of the model should have a lot of functionalities. One of them is the button **Switch Example**. With it, the user should select one of default examples or the teacher example.



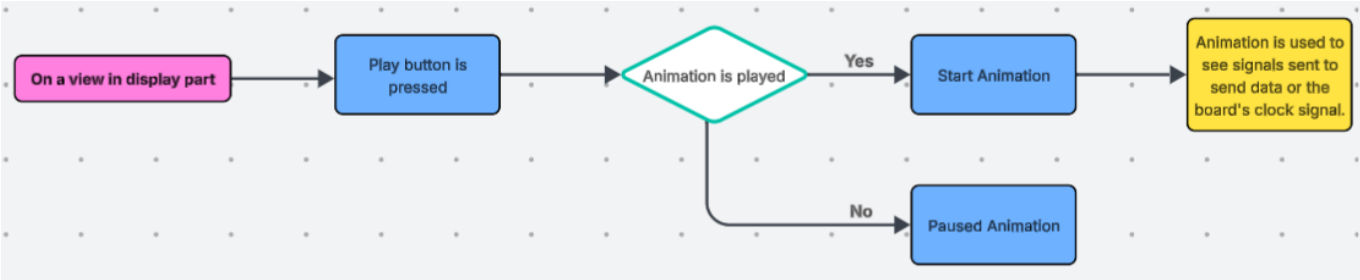
Switch View

The sidebar should have a button **Switch View**. With it, the user should select one of those three views.



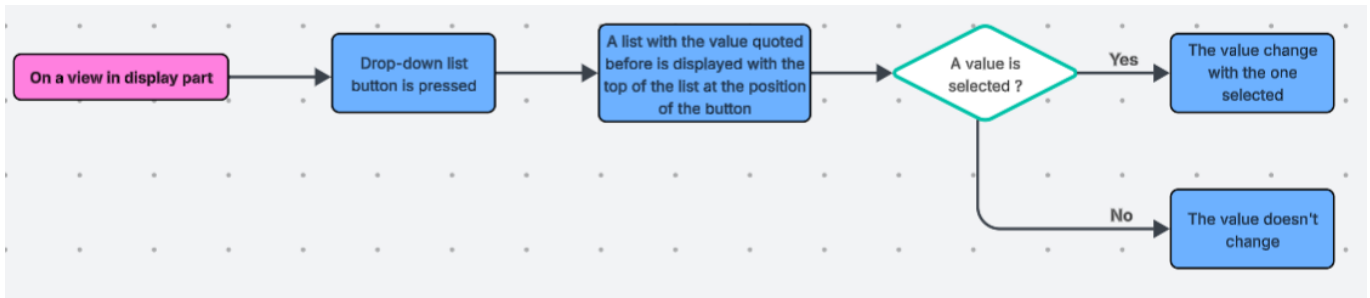
Animation

The project's goal is to have an animation to illustrate program action. So, with a **play button**, we should be allowed to pause and resume animation.



Animation Speed

Near to the play button, we should have a **drop-down list**. With it, the animation speed up or slow down. We should be allowed to select the speed between this speed values : 0.5, 0.75, 1, 1.25, 1.5, 2, 4. The **default value** is x1.



## B. Non Functional Requirements

### Input File

The software should not accept any input except for importing files for another example provided by the teacher. Therefore, the software must filter the data supplied by the teacher. Accepted entries are:

- A **schematic netlist** (in Verilog format) => elements to be represented on screen
- A **standard delay file** (SDF format) => signal propagation time between each element

Whole Example should be in a folder **Example**.

### Compatibility

The project is to create a web interface. Therefore, this interface should work with every computer (Windows, Linux, MacOS). However, compatibility with mobile devices is not a priority.

## Glossary

---

[^1]: An FPGA Simulator is a tool that helps designers test digital circuits before they are built. It uses special languages like Verilog or VHDL to simulate how signals move through an FPGA, ensuring everything works correctly.

[^2]: An FPGA (Field-Programmable Gate Array) is an integrated circuit with basic elements and preconfigured electrical signal routes between them.

[^3]: A web interface is a user-friendly website or online tool that lets you interact with a system or application through your web browser. It's like a control panel on the internet where you can click buttons, enter information, and see results without needing special software.

[^4]: A web app is like a website that you can interact with, it like an application you can use on your phone, hence instead of use an application store to download it, you use it through a web browser like Chrome or Safari.

[^5]: Translation of the application into an electrical equivalent. It creates a netlist (which can be exported as a netlist).

[^6]: Place and Route is the packing of the netlist component in the FPGA available BEL (Place). Then a route for signals between each BEL is selected (Route). A timing netlist is created and can be exported in verilog.

[^7]: A testbench is a crucial tool in the design and verification process, helping to ensure that digital circuits and systems function correctly before they are manufactured.

[^8]: A netlist is like a recipe for building an electronic circuit. It tells you what components you need and how to put them together (connections) to create a functioning circuit.

[^9]: Verilog is a language used to program a system using the FPGA layout. It use to writting instruction to the system.

[^10]: These are the hardware electrical ressources available inside the FPGA like fliflop, Look-Up-Table (LUT), Block RAM....

[^11]: HTML, CSS and JavaScript are pagramations languages usually used for the web. It is what we use for this project.

[^12]: A flip-flop is a tiny switch in circuits that store one bit of data (0 or 1) and helps control when data changes.

[^13]: A LUT4 is a small table that can execute simple logical operations, such as AND, OR or NOT, on up to four inputs.