



User Manual

Table of Contents

Introduction	3
Getting Started	4
Interface Overview	5
Tool Bar.....	6
Component Color Code	7
Running a Simulation.....	8
Example	9
Troubleshooting	10
FAQ	11
Contact	12

Introduction

Welcome to **FPGA Explorer**, a web-based simulation tool for visualising and analysing an FPGA's internal chip flows. Designed for students, teachers, and engineers, this platform provides an interactive, user-friendly experience for exploring FPGA concepts through **SDF file uploads, clock speed adjustments, and step-by-step simulations**.

What You'll Learn in This Manual?

- **Navigating the Interface** – How to use the FPGA Explorer efficiently.
- **Uploading & Simulating FPGA Designs** – Step-by-step file uploads and visualisation.
- **Controlling Simulations** – Start, pause, reset, and step through processes.
- **Adjusting Clock Speeds** – Modify simulation speed in real time.
- **Troubleshooting** – Common issues and solutions.

Who Is This Manual For?

- **Teachers** – For interactive FPGA instruction.
- **Students** – To explore and understand FPGA behaviour.
- **Developers & Engineers** – For a web-based FPGA simulation tool.

Keep this guide handy for reference while using **FPGA Explorer**. For further assistance, check the **FAQ** or **contact** support.

Getting Started

The web interface is accessible via our GitHub repository and a hosted link.

Local Access

To run the web interface locally:

1. **Clone the repository**

[GitHub Repository](#) *

2. **Open a terminal**

3. **Navigate to the directory**

```
cd src
```

4. **Start the application**

```
npm start
```

The web interface will open in your default browser.

Online Access

You can also access the interface online at:

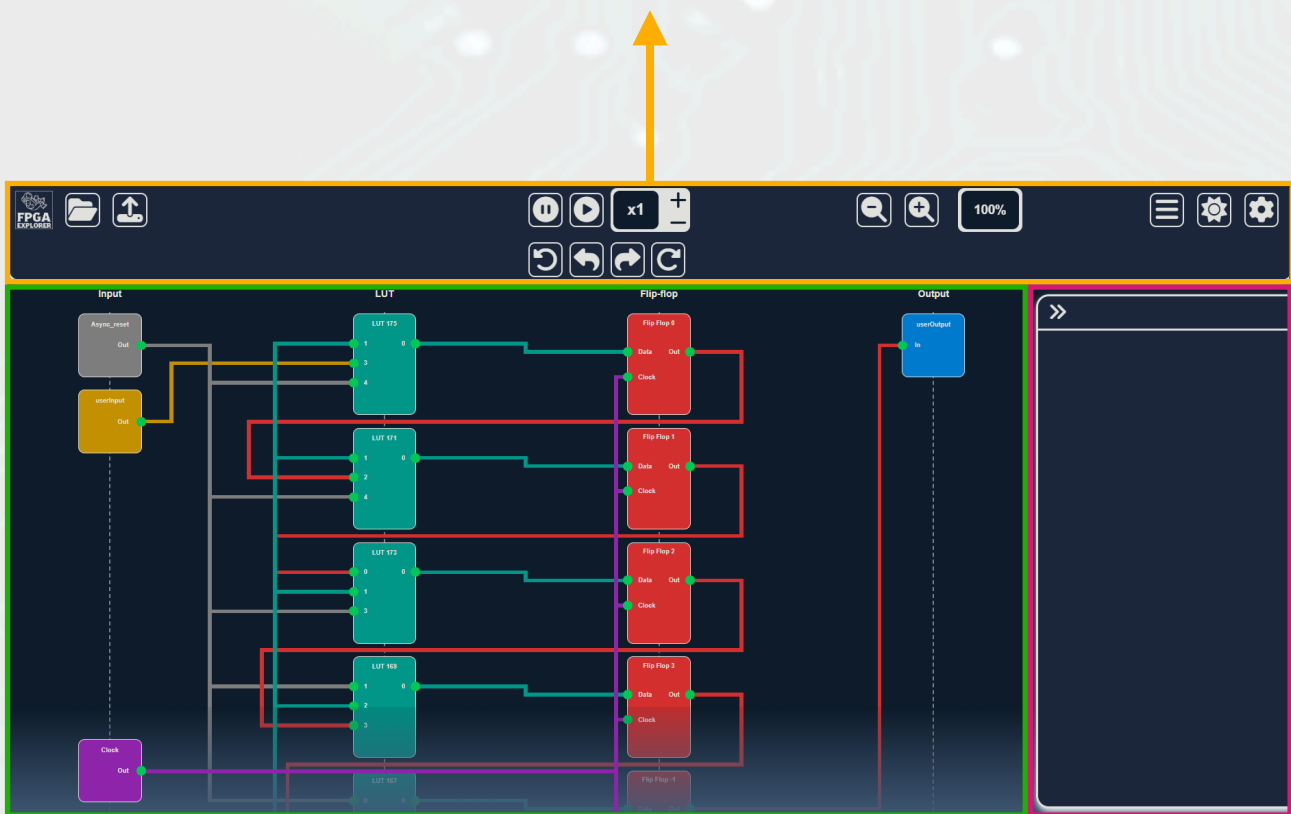
[FPGA Explorer](#) **

*<https://github.com/algosup/2024-2025-project-4-web-fpga-team-4/tree/main>

**<https://two024-2025-project-4-web-fpga-team-4.onrender.com/client.html>

Interface Overview

Tool Bar



Visualisation Area

Drawer

Tool Bar

The **toolbar** in FPGA Explorer provides quick access to essential functions, including file management, simulation controls, zoom options, and display settings.



File Management



Load File – Opens an example file.



Download JSON – Downloads the pivot file.

Simulation Control



Play – Starts or resumes the simulation.



Stop – Stops the simulation temporarily.



Step Back – Moves the simulation one step backwards.



Step Forward – Advances the simulation by one step.



Reset – Resets the entire simulation.



Final Step – Moves to the end of the simulation.

Zoom & Navigation



Zoom In – Enlarges the simulation view.



Zoom Out – Reduces the simulation view.

Options & Settings



Show/Hide – Expands or collapses the real-time clock signal data panel.



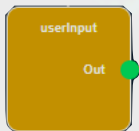
Light/Dark Mode – Toggles between light and dark themes.



Settings – Opens the configuration panel for additional options.

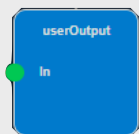
Component Color Code

The visualisation area is the main part of this interface. It shows all components and signal flows, each with a unique colour and shape for clarity.



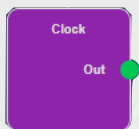
Input

- Inputs *send data* into the circuit (like buttons or lights)



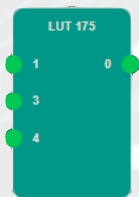
Output

- Outputs *display results*



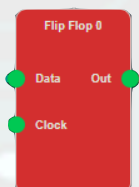
Clock

- Drives the simulation timing
- Feeds pulses to *synchronise Flip-Flops*



Look-Up Table (LUT)

- Executes logical operations
- Used for *combining and transforming input* values

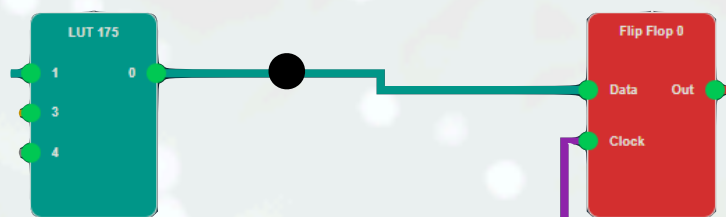


Flip-Flop

- Stores data and *introduces a delay in circuits*
- Controlled by clock edges

Wire

- Transfers logic signals* between components
- Matches the component it originates from



A **white or black circle** moves along the wire to represent real-time signal flow.

Running a Simulation

Here, you will see how to launch and interpret a simulation.

Example

Todo: think about an example

Troubleshooting

The simulation doesn't start

The file may be missing components or incorrectly formatted.

- Use a .sdf or .json file
- Include at least one LUT or Flip-Flop
- Try uploading a known working example

No wires between components

Connections might be missing or malformed in the .json file.

- Each connection should include `input`, `output`, and `delay`
- Check that port numbers and types match

Play/Pause/Step buttons don't respond

The simulation may not be properly loaded.

- Upload a valid file
- Refresh the page

Layout looks broken or components overlap

Rendering or browser zoom issue.

- Set browser zoom to **100%**
- Use Chrome, Firefox, Safari, or Edge
- Avoid mobile devices

Signal animation isn't visible

Simulation may be paused or the delay is too short to notice.

- Press Play
- Lower speed to x1
- Check that delay values exist in your connections

Error when uploading a file

The file is not readable or exceeds size limits.

- Make sure the file extension is .sdf or .json
- The file must be under 10MB
- If the error persists, try another file or refer to the example files

FAQ

Can I write Verilog or VHDL inside the tool?

No — the tool is for visualisation only. You must upload `.SDF` or `.JSON` files.

What files are supported?

Only `.SDF` and `.JSON` are supported at the moment. Verilog must be compiled externally.

How do I control the simulation speed?

Use the speed control buttons (x1, x2, x4) in the toolbar.

👉 See *"Running a Simulation" for more.*

What do the different block colours mean?

Each component has a colour (LUT = green, Flip-Flop = red, etc.)

👉 See *"Component Color Code" for the full legend.*

What is the `.json` file supposed to contain?

The `.JSON` file must describe the FPGA circuit in a structured way, including:

- Components (LUTs, Flip-Flops, I/Os) with their IDs and ports
- Connections between components, including delays
- A list of I/Os with their types (input/output)

If you're unsure, you can:


- Upload a `.SDF` file and download the generated `.JSON` to use as a template.
- Refer to the [Technical Specifications](#) or example files included in the [repository](#).

Contact

If you would like to get in touch with the development team — whether to report a bug, ask a question, or share feedback — please use our GitHub repository. We manage all inquiries directly through GitHub Issues to ensure transparency and traceability.

How to Contact Us

1. Navigate to the issue tab of the [GitHub repository](#)*
2. Open a new issue.
3. In the title, include the word contact.

 **Tip:** Please be as clear and specific as possible when describing your request so we can assist you efficiently.

We monitor GitHub regularly and will get back to you as soon as possible.

**<https://github.com/algosup/2024-2025-project-4-web-fpga-team-4/issues>*