Logic Simulator: User Guide

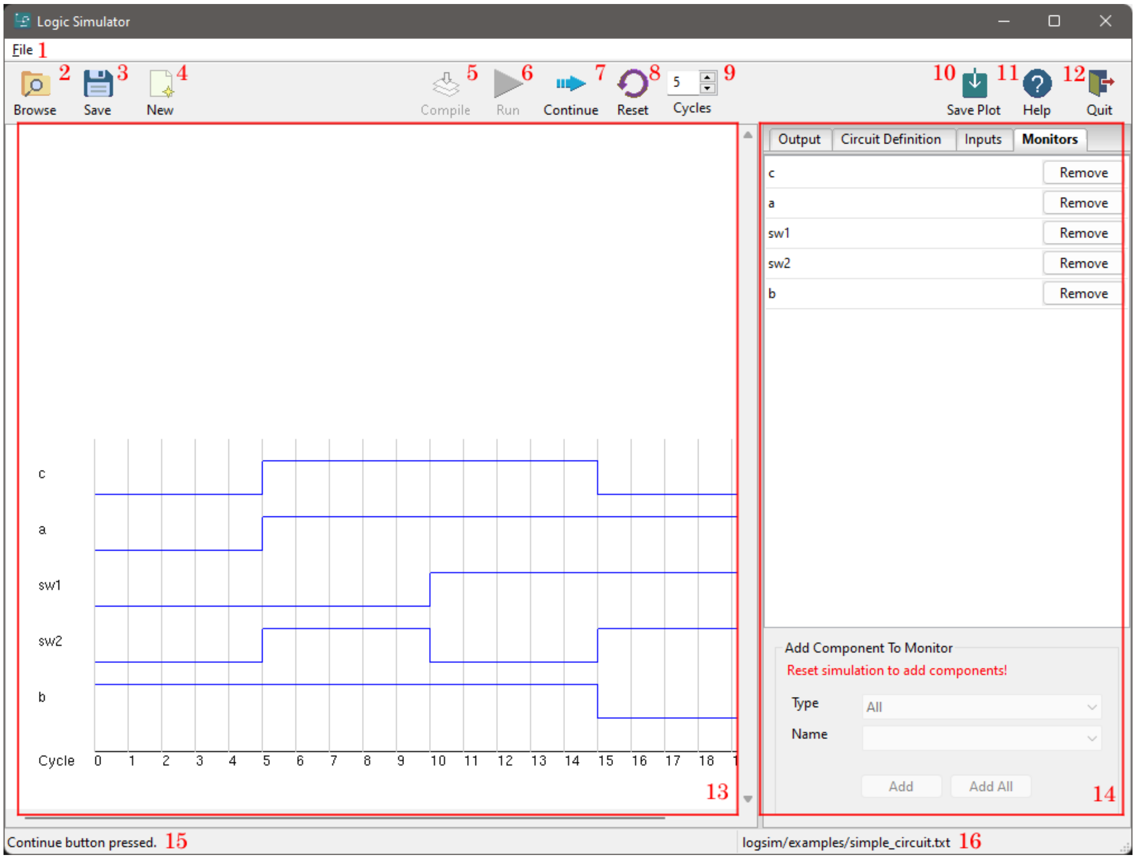

Figure 1: Screenshot of the GUI

Logic Simulator is a tool that allows the user to read in, edit and simulate a logic circuit defined in a user-provided circuit definition file[1]. Figure 1 shows a screenshot of the Graphical User Interface (GUI) labelled with the following features:

1. Menubar containing 3 options: *About*, *Save As* and *Quit*.
2. Opens a definition file
3. Saves the current definition file
4. Creates a new definition file
5. Compiles the edited definition file in the *Circuit Definition* tab for any errors and initialises the inputs and monitors
6. Runs the code for the specified number of simulation cycles from scratch.
7. Continues the simulation for the specified number of cycles.
8. Resets the simulation.
9. Allows the user to specify the number of simulation cycles.
10. Saves the plot as an image.
11. Displays the user guide.
12. Quits the application.
13. Signal trace plot.
14. Side panel containing four tabs:

    Output: Console log. Text commands can also be run here.

    Circuit Definition: Editable area for the loaded definition file.

    Inputs: List of input switches. Has buttons which allow the switch states to be toggled ON or OFF.

    Monitors: List of signals to be monitored. Allows components to be added or removed.
15. Statusbar.
16. Path name of the current definition file.

A command-line interface is also available by running `python logsim/logsim.py -c <pathname>`. Typing `h` will display a list of possible commands. Example definition files can be found in the `./logsim/examples` folder[2]. For further information on how to install and run the Logic Simulator, please consult the `README.md`.

[1]Definition files follow the EBNF grammar defined in the first interim report.

[2]See overleaf for the circuit diagrams of the available example definition files.

Example Circuit A — Simple Circuit

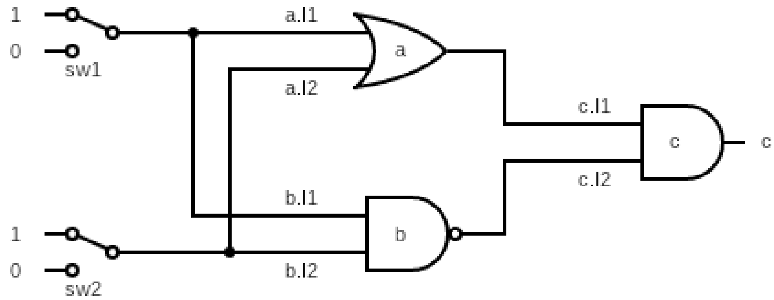

Figure 1: XOR Gate

```
#Example circuit - Simple circuit (As per first interim report);
#XOR gate;

devices(
    a is OR;
    b is NAND;
    c is AND;
    sw1, sw2 are SWITCH;
)

initialise(
    sw1, sw2 are HIGH;
    a, b, c have 2 inputs;
)

connections(
    a(
        sw1 is connected to a.I1;
        sw2 is connected to a.I2;
    )

    b(
        sw1 is connected to b.I1;
        sw2 is connected to b.I2;
    )

    c(
        a is connected to c.I1;
        b is connected to c.I2;
    )
)

monitors(
    c;
)
```

Example Circuit B — Complex Circuit

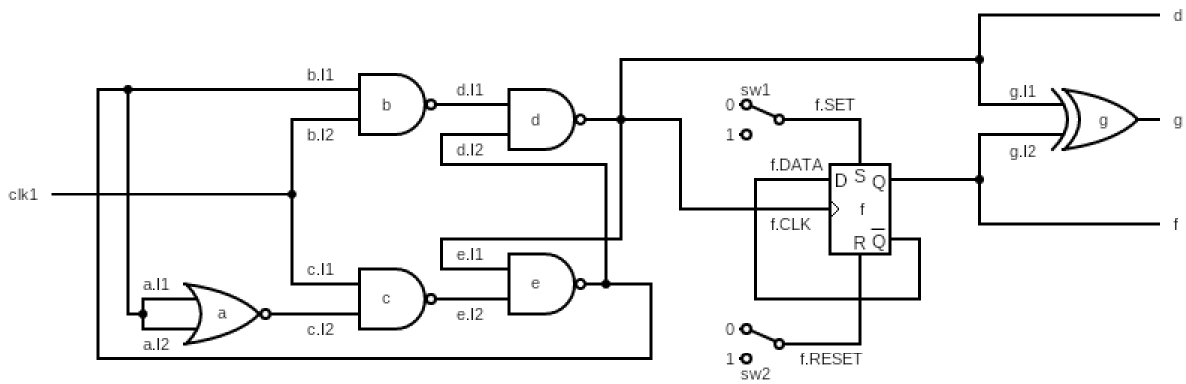

Figure 2: 2 Bit Counter with Outputs XOR

```
#Example circuit - Complex circuit (As per first interim report);
#2 bit counter with outputs XOR;

devices(
    a is NOR;
    b, c, d, e are NAND;
    f is DTYPE;
    g is XOR;
    sw1, sw2 are SWITCH;
    clk1 is CLOCK;
)

initialise(
    a, b, c, d, e, g have 2 inputs;
    sw1, sw2 are LOW;
    clk1 cycle length 5;
)

connections(
    a(
        e to a.I1;
        e to a.I2;
    )

    b(
        e to b.I1;
        clk1 to b.I2;
    )

    c(
        clk1 to c.I1;
```

```
            a to c.I2;
        )

        d(
            b to d.I1;
            e to d.I2;
        )

        e(
            d to e.I1;
            c to e.I2;
        )

        f(
            f.QBAR to f.DATA;
            d to f.CLK;
            sw1 to f.SET;
            sw2 to f.CLEAR;
        )

        g(
            d to g.I1;
            f.Q to g.I2;
        )
    )

    monitors(
        d, f.Q, g;
    )
```

Example Circuit C — 50 Switches

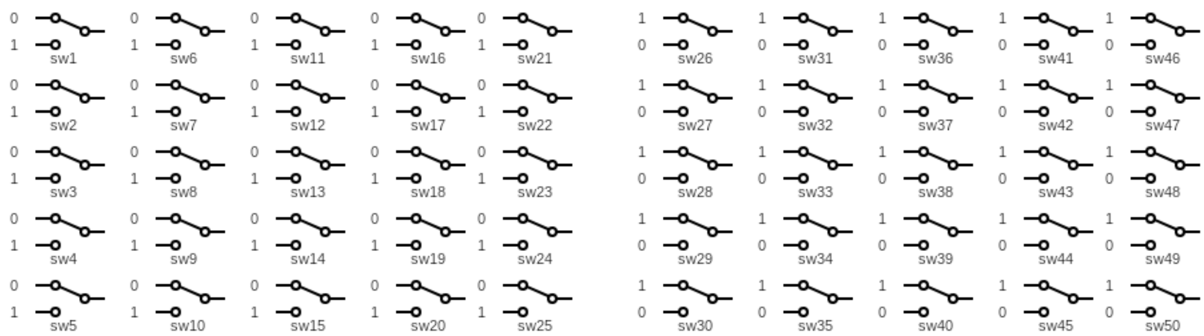

Figure 3: 50 Switches

```
#Example circuit - 50 switches;

devices(
    sw1, sw2, sw3, sw4, sw5, sw6, sw7, sw8, sw9, sw10,
    sw11, sw12, sw13, sw14, sw15, sw16, sw17, sw18, sw19, sw20,
    sw21, sw22, sw23, sw24, sw25, sw26, sw27, sw28, sw29, sw30,
    sw31, sw32, sw33, sw34, sw35, sw36, sw37, sw38, sw39, sw40,
    sw41, sw42, sw43, sw44, sw45, sw46, sw47, sw48, sw49, sw50 are SWITCH;
)

initialise(
    sw1, sw2, sw3, sw4, sw5, sw6, sw7, sw8, sw9, sw10,
    sw11, sw12, sw13, sw14, sw15, sw16, sw17, sw18, sw19, sw20,
    sw21, sw22, sw23, sw24, sw25 are LOW;
    sw26, sw27, sw28, sw29, sw30, sw31, sw32, sw33, sw34, sw35,
    sw36, sw37, sw38, sw39, sw40, sw41, sw42, sw43, sw44, sw45,
    sw46, sw47, sw48, sw49, sw50 are HIGH;
)

connections(
)

monitors(
    sw1, sw2, sw3, sw4, sw5, sw6, sw7, sw8, sw9, sw10,
    sw11, sw12, sw13, sw14, sw15, sw16, sw17, sw18, sw19, sw20,
    sw21, sw22, sw23, sw24, sw25, sw26, sw27, sw28, sw29, sw30,
    sw31, sw32, sw33, sw34, sw35, sw36, sw37, sw38, sw39, sw40,
    sw41, sw42, sw43, sw44, sw45, sw46, sw47, sw48, sw49, sw50;
)
```

Example Circuit D — SR Bistable

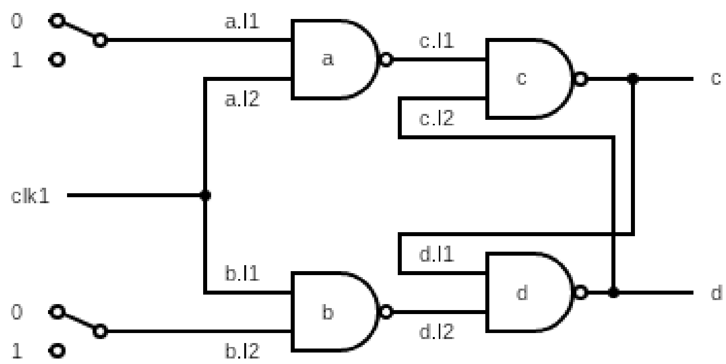

Figure 4: SR Bistable

```
#Example circuit - SR bistable;

devices(
    a, b, c, d are NAND;
    sw1, sw2 are SWITCH;
    clk1 is CLOCK;
)

initialise(
    sw1, sw2 are LOW;
    a, b, c, d have 2 inputs;
    clk1 cycle 5;
)

connections(
    a(
        sw1 to a.I1;
        clk1 to a.I2;
    )

    b(
        clk1 to b.I1;
        sw2 to b.I2;
    )

    c(
        a to c.I1;
        d to c.I2;
    )

    d(
        c to d.I1;
```

```
            b to d.I2;
        )
    )

    monitors(
        c, d;
    )
```

Example Circuit E — Divide by 3 Circuit

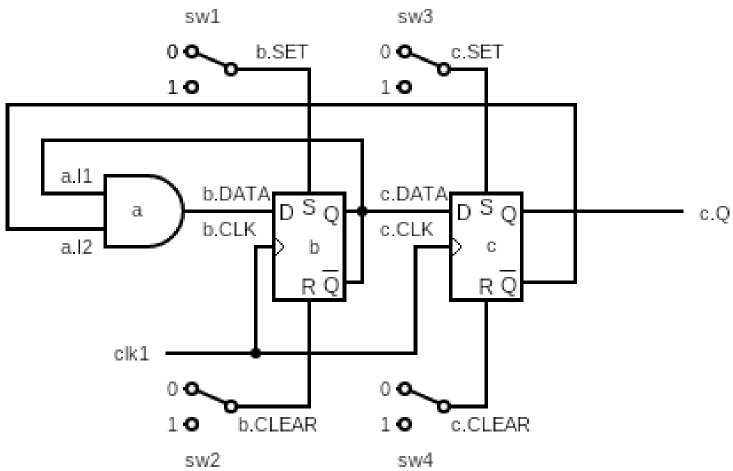

Figure 5: Divide by 3 Circuit

```
#Example circuit - divide (clock frequency) by 3;

devices(
    a is AND;
    b, c are DTYPE;
    sw1, sw2, sw3, sw4 are SWITCH;
    clk1 is CLOCK;
)

initialise(
    a has 2 inputs;
    sw1, sw2, sw3, sw4 are LOW;
    clk1 cycle 9;
)

connections(
    a(
        b.QBAR to a.I1;
        c.QBAR to a.I2;
    )

    b(
        a to b.DATA;
        clk1 to b.CLK;
        sw1 to b.SET;
        sw2 to b.CLEAR;
    )

    c(
```

```
            b.Q to c.DATA;
            clk1 to c.CLK;
            sw3 to c.SET;
            sw4 to c.CLEAR;
        )
    )

    monitors(
        c.Q;
    )
```

Example Circuit F — Ring Oscillator

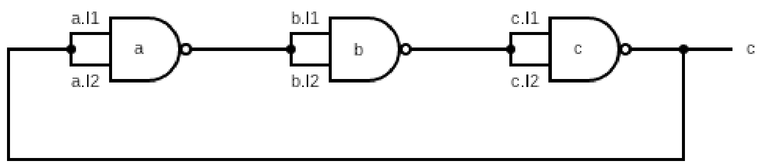

Figure 6: Ring Oscillator

```
#Example circuit - Ring Oscillator;

devices(
    a, b, c are NAND;
)

initialise(
    a, b, c have 2 inputs;
)

connections(
    a(
        c to a.I1;
        c to a.I2;
    )

    b(
        a to b.I1;
        a to b.I2;
    )

    c(
        b to c.I1;
        b to c.I2;
    )
)

monitors(
    c;
)
```

Note that this circuit will not be able to be simulated as it involves oscillating signals.

Example Circuit G — Blank Circuit

```
#Example circuit - Blank circuit;

devices(
)

initialise(
)

connections(
)

monitors(
)
```

Alternatively, a blank .txt file will also suffice