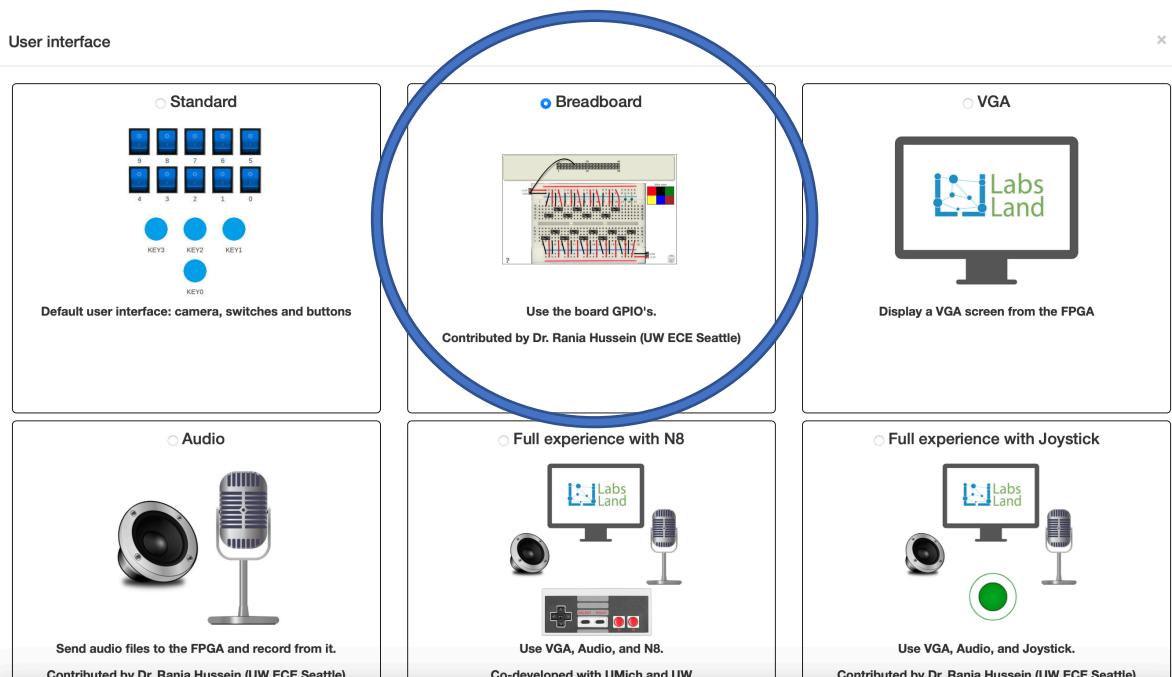


# EE/CSE 371:

## Design of Digital Circuits and Systems

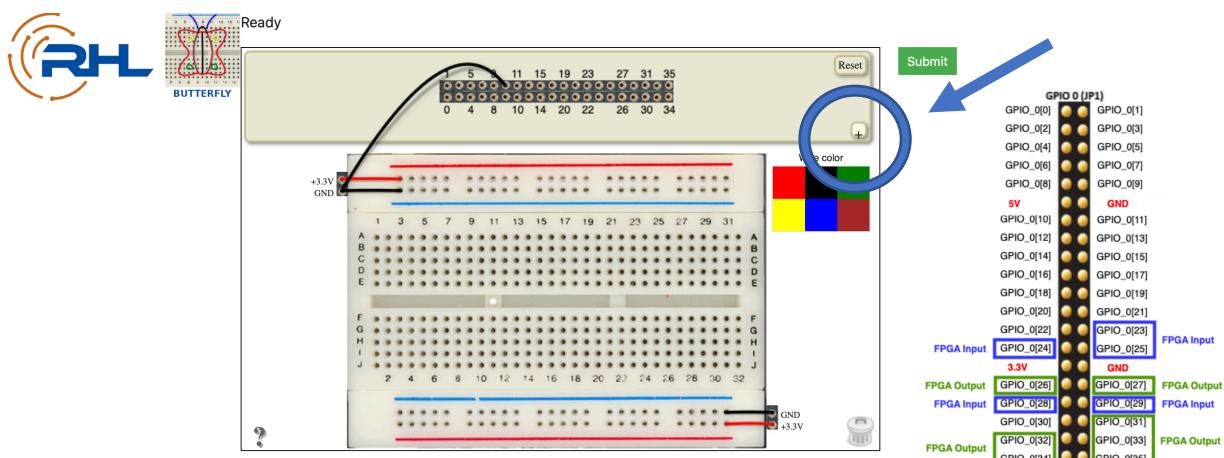
### Updated Breadboard Guide

Under “User interface”, click “Edit” and choose the “Breadboard” option.



### Adding Components

The first time the breadboard loads, it will not be populated. To add components, click on the “+” button to bring up a list of available components.



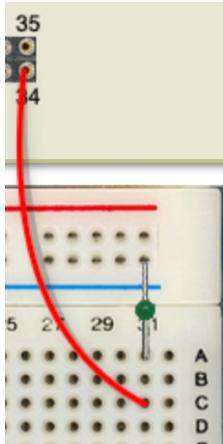
Click on the switches and LEDs to add these components.

The screenshot shows a breadboard simulation interface. On the left, there's a logo for "RHL BUTTERFLY". In the center, a logic component library window is open, displaying four logic gates: NOT, AND, OR, and XOR. Below them are a virtual LED and a switch. A "Wire color" palette is visible. To the right, a breadboard area has a red wire connected from the output of the XOR gate to the anode of a virtual LED. A blue wire connects the cathode of the LED to ground. A green wire connects the output of the XOR gate to the input of a switch. A red wire also connects the output of the switch back to the breadboard. A "Submit" button is at the top right. On the far right, a pinout table for "GPIO 0 (J1)" is shown, mapping pins 0-35 to various functions like 3V, GND, and FPGA Input/Output.

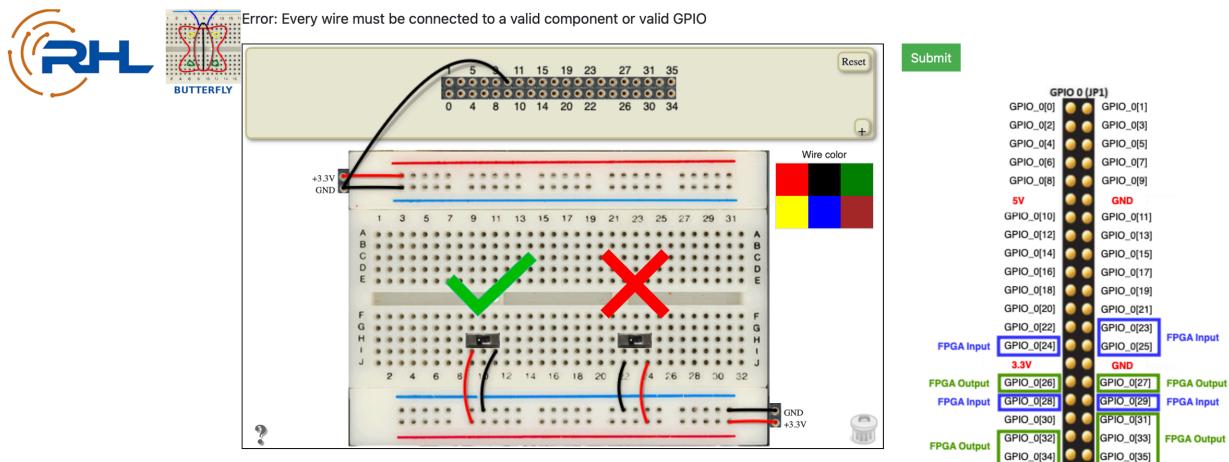
Drag each component to the desired location on the breadboard. For a virtual LED, one end must be connected to GND.

This screenshot shows a breadboard simulation interface. A logic component library window is open, showing a 7400 quad NOR gate. A red wire connects the output of the NOR gate to the anode of a virtual LED. A blue wire connects the cathode of the LED to ground. A green wire connects the output of the NOR gate to the input of a switch. A red wire also connects the output of the switch back to the breadboard. A "Submit" button is at the top right. On the far right, a pinout table for "GPIO 0 (J1)" is shown, mapping pins 0-35 to various functions like 3V, GND, and FPGA Input/Output.

**Note: Do not put an LED on the right-most edge of the breadboard:**

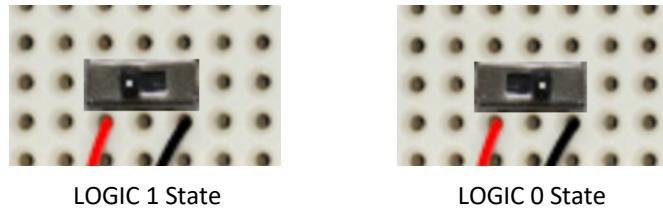


For a switch, the left pin must be connected to +3.3V, while the right pin must be connected to GND.

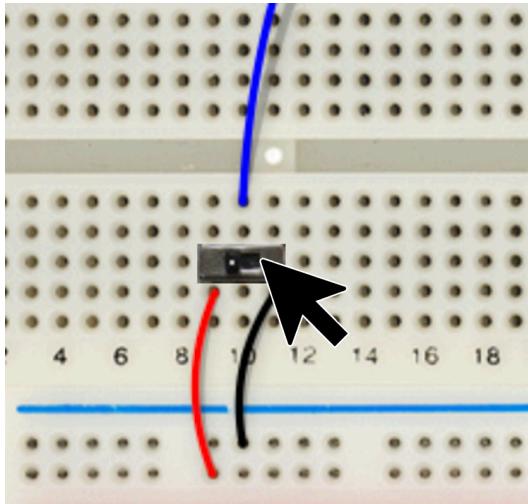


## Using the Virtual Switches

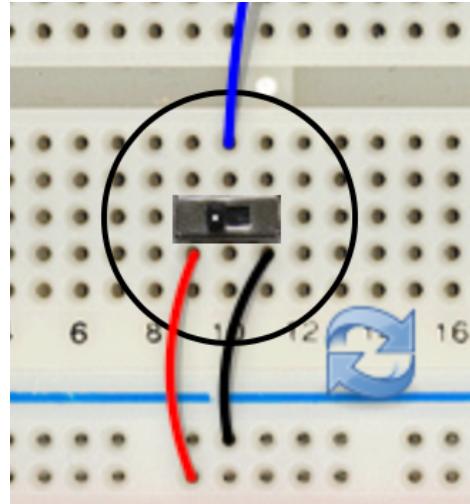
When the switch is on the left position, it is in the LOGIC 1 state. When the switch is on the right position, it is in the LOGIC 0 state.



To change the state of the virtual switch, **press and hold** for at least half a second to change its state, or do two fast clicks. One fast click will bring up the circle element, used if you want to delete the component.

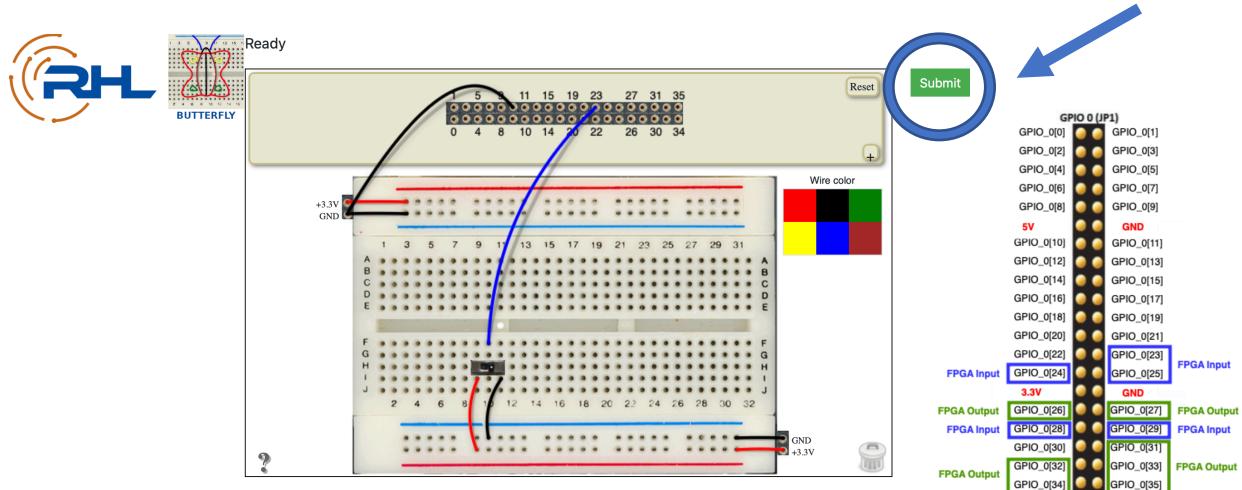


Press and hold the virtual switch to change states



Quick press brings up the circle element

When the states of the switches are in their preferred location, press the green “Submit” button. This will send the breadboard information to the DE1-SoC.



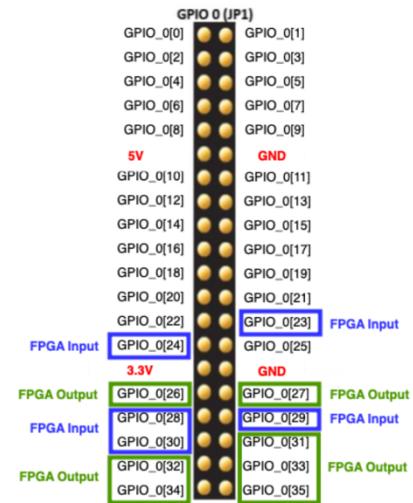
## GPIO Mappings

FPGA Inputs (i.e. where to connect the switches):

- V\_GPIO[23]
- V\_GPIO[24]
- V\_GPIO[28]
- V\_GPIO[29]
- V\_GPIO[30]

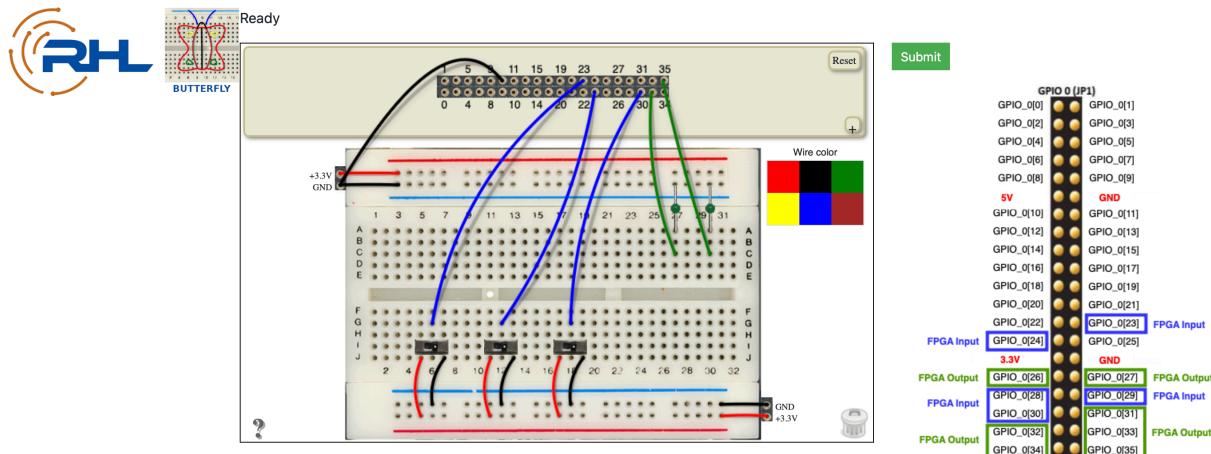
FPGA Outputs (i.e. where to connect the breadboard LEDs):

- V\_GPIO[26]
- V\_GPIO[27]
- V\_GPIO[31]
- V\_GPIO[32]
- V\_GPIO[33]
- V\_GPIO[34]
- V\_GPIO[35]



## Example Configuration

An example configuration for lab6 can be:



## Sample Code

```
// DE1_SoC.sv
// The objective of this sample code is to show how to map a virtual breadboard
// switch logic directly to a virtual led.
```

```
// Virtual switch 1 is connected to V_GPIO[23], and will control virtual led 1, which
// is connected to V_GPIO[32].
// Virtual switch 2 is connected to V_GPIO[24], and will control virtual led 2, which
// is connected to V_GPIO[35]
module DE1_SoC (V_GPIO);

    // define ports
    inout logic [35:23] V_GPIO;

    // Assign virtual switch 1 to virtual led 1, and virtual switch 2 to led 2
    assign V_GPIO[32] = V_GPIO[23];
    assign V_GPIO[35] = V_GPIO[24];

endmodule // DE1_SoC
```