ELEC3342: Digital System Design
First Semester, 2022–23

Lab 2

Objective: To build simple combinational circuits with Logisim and Vivado.

1 Getting the files

Download the source file for this lab from:

`http://www.eee.hku.hk/~elec3342/fa22/handout/elec3342lab2.tar.gz`

and extract it.

2 Building Combinational Circuits with Logisim-evolution

In this part, we will use a software called Logisim-evolution to experiment with simple combinational logic circuits. Logisim-evolution is an educational tool for designing and simulating digital circuits. It started as a fork to the original and very successful educational software called Logisim. The original creator of Logisim ceased support for Logisim in 2011 and Logisim-evolution was born as fork from the original code. Although Logisim-evolution has made many substantial changes, you will probably find the online help for the original Logisim software useful:

`http://www.cburch.com/logisim/docs/2.7/en/html/guide/index.html`

In particular, the section on "Beginner's Tutorial" can be very helpful to learn the very basic. (Note, some of the icons have changed over the years.)

Alternatively, some documents on the new Logisim-evolution can be found from their github repository:

`https://github.com/logisim-evolution/logisim-evolution/blob/master/docs/docs.md`

2.1 Installing the Software

Logisim-evolution is free and you can find it from:

`https://github.com/logisim-evolution/logisim-evolution`

The latest release as of this writing is version 3.7.2. You can find all the releases from:

`https://github.com/logisim-evolution/logisim-evolution/releases`

There are prepackaged installation files for Windows, Mac and Linux from the github page. You can also run the original JAVA jar file from the release page as well. You are strongly encouraged to install it on your own computer to avoid congestion. As a fall backplan, we have also provided `logisim-evolution` on `tux-2`. You can launch it through a terminal in a X2Go session by typing:

```
> source ~elec3342/elec3342.bashrc
> logisim-evolution
```

Once started, you should see the main window of logisim-evolution:

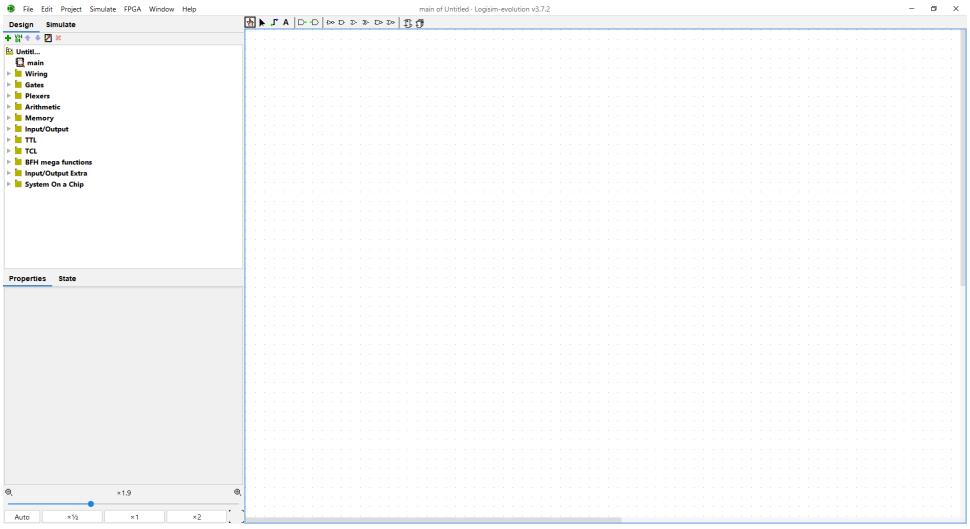

Figure 1:

You will be building circuits on the blank canvas.

2.2 Warmup Exercise On your newly opened window, construct the following circuit:

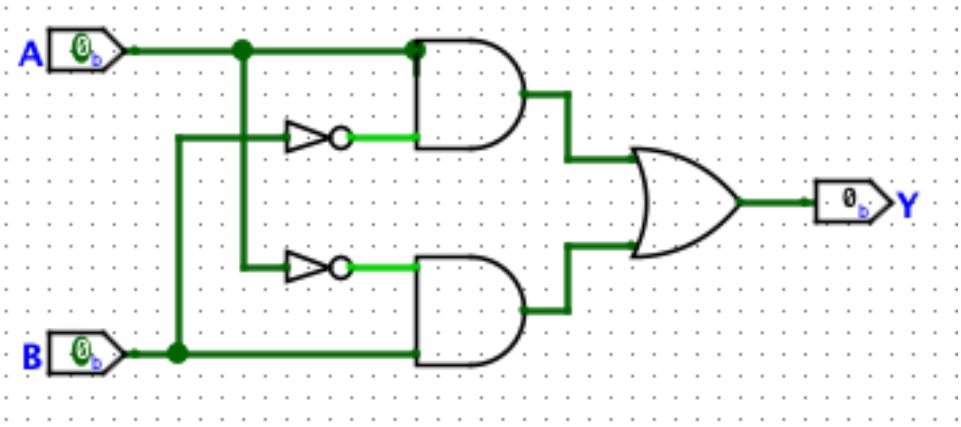

Figure 2:

Most of the common tools you need to construct a circuit can be accessed from the toolbar:

Figure 3: Toolbar in Logisim-evolution.

To add a logic gate, select the desired logic gate (icon 6 to icon 11) from the toolbar. Your mouse cursor will turn into the shape of the gate. Place the component on the canvas by clicking at the desired location. Alternatively, you can also locate a gate by expanding the folder **Gates** from the left of the screen.

Every component has properties. You can open the properties window by clicking icon 2 and selecting a component on the canvas. The properties window of the selected component will show on the left. An

important property for a gate is its **Number Of Inputs** property, which control the number of input for the gate (default is 2). You will see the number of connection point on the component (a blue dot) changes when you modify its number of inputs.

Similarly, you can add an input pin by selecting icon 4 and add an output pin by selecting icon 5. For I/O pins, you must modify its **Label** properties so you can correctly identify them later. To modify the label of a pin, double click on the pin. Alternatively, click on the field **Label** under the **Properties** panel.

To connect the components, you need to click on the icon 3 to draw wires. Wires can be drawn by dragging your mouse from the output of one component to the input of another component.

Once you have connected the circuit, you will notice the color of the wire changes. A *dark green* color means that the current value on the wire is a logical '0', while a *light green* color signifies a '1'. Other wire colors: *blue* = unknown value, *gray* = unconnected, *red* = conflict.

Select the poke tool (icon 1) so you can test the circuit manually. You can toggle the value of an input pin by clicking on it using the poke tool. Note the values of all subsequent wires change instantly.

2.3 Check Yourself

By testing all combinations of input `A` and `B` using the Poke tool, construct the truth table of the circuit. What function does this circuit implement?

2.4 Building a Combinational Circuit Now, start a **New** circuit in `logisim-evolution` and build the circuit described by the following boolean expression:

$$X = \overline{A} \cdot \left(\overline{B} + (C \oplus D)\right) + B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C \tag{1}$$

Note that in this case, you have 4 inputs (`A`, `B`, `C`, `D`) and one output `X`. For this part, simply implement the expression directly using the method we discussed in class. We will experiment further with the expression in later part of this lab.

2.5 Checkoff 1

Save your circuit by clicking **File** → **Save**. Name it as `comb_logic.circ`

3 Analyzing Circuits with Logisim-Evolution

One of the many features of `logisim-evolution` is its ability to analyze a circuit. You will explore its function in this part using the circuit you have just created in Checkoff 1.

3.1 Using Analyze Circuit Open your circuit from Checkoff 1 and save it with a **NEW** name `comb_logic_simplified.circ`, using the **File→Save As...**

Now, with the new file, you can analyze your circuit by selecting:

Project → Analyze Circuit

A new window will pop up. Explore the various functions in this tool. It analyzes the circuit that is currently at the top level (i.e., the circuit you just drawn), and can produce results such as the Boolean expression of the circuit:

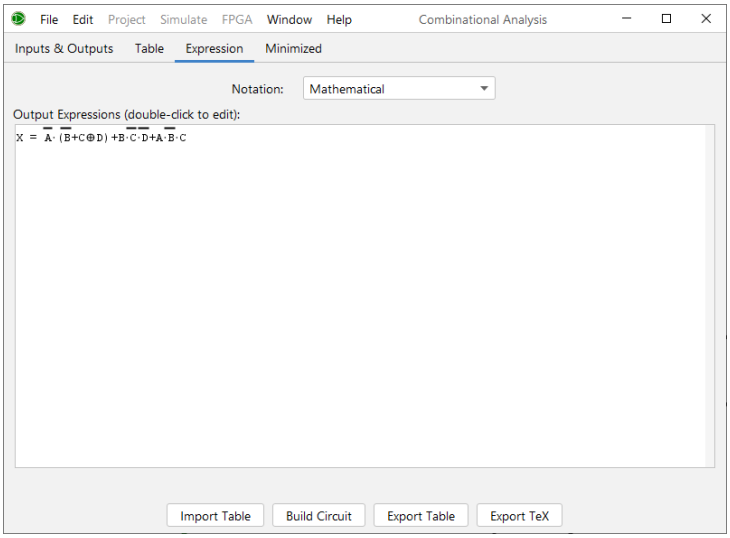

Figure 4: **Expression** Tab in Combinational Analysis Window.

3.2 Check Yourself

Verify the analyze circuit function is producing the same Truth Table as equation 1. Is the Boolean expression the same as the original equation 1? Why or why not?

3.3 Minimize Circuit Now, minimize the circuit by selecting the **Minized** tab. You will see the K-Map and simplified expression of your circuit:

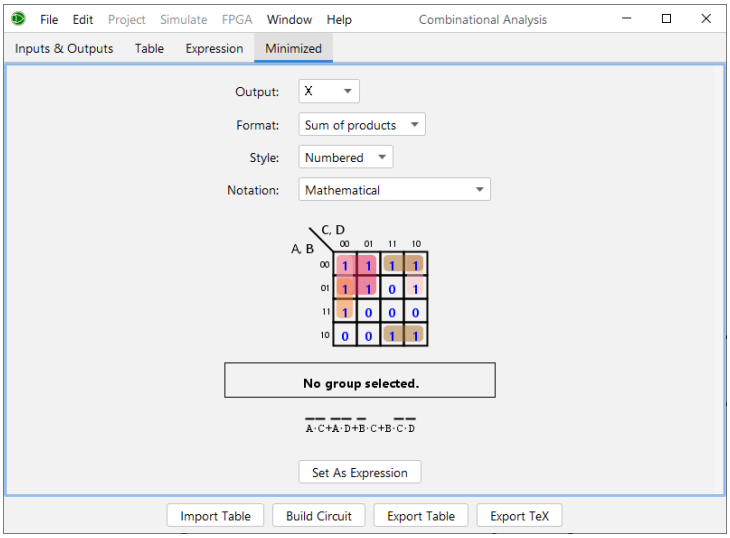

Figure 5: **Minized** Tab in Combinational Analysis Window.

Click on **Set As Expression**. The expression in **Expression** tab will now be replaced by this simplified expression. Then click on **Build Circuit** to let Logisim-evolution generate the circuit based on the simplified expression.

3.4 Checkoff 2

Save this simplified circuit to `comb_logic_simplified.circ`

4 Implementing the Circuit in Vivado

4.1 Create a Vivado Project. Create a new Vivado project and add the `comb_logic_blank.vhd` to the source file. You may refer to Lab 1 on how to create a new project.

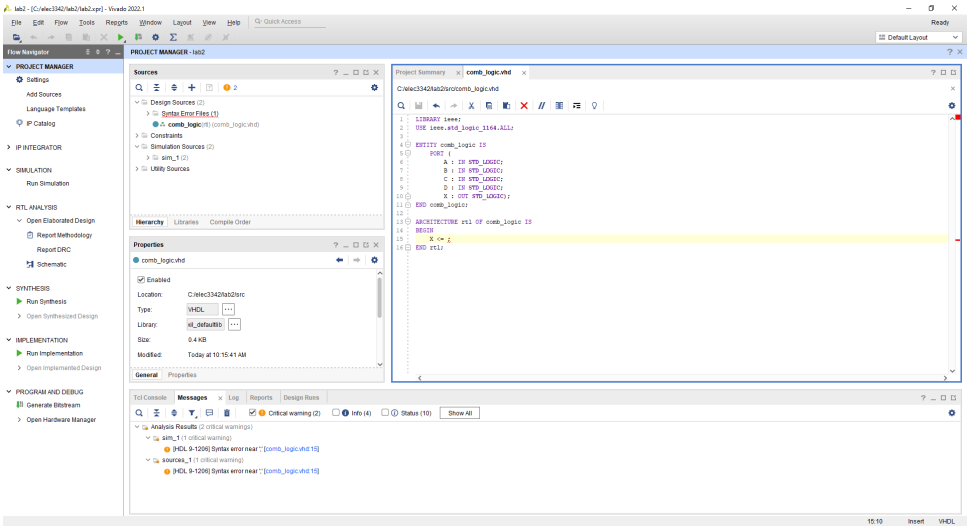

Figure 6: Vivado Project Window.

You will see that Vivado reports a syntax error in the source file. It is because the source file we have provided is not completed. Now complete line 15 of the source file with equation 1. Make sure Vivado doesn't report any errors before you continue.

4.2 Check Elaborated Design. Vivado can analyze your VHDL code. This process is called "Elaborate" in Vivado. You can open the elaborated schematic from **Flow Navigator** by:

RTL ANALYSIS → Open Elaborated Design → Schematic

Click on **OK** if this window pops up:

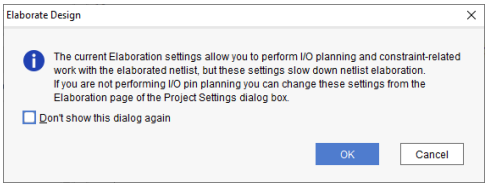

Figure 7: Vivado Elaboration Pop-up.

4.3 Check Yourself

Compare the elaborated schematic from Vivado and the circuits from logisim-evolution. Are they the same? Why or why not?

4.4 Checkoff 3

Save your modified source file to `comb_logic.vhd`

5 Submission

Compress your files from previous checkoffs to a zip file and upload it to Moodle:

```
<your-zip-file>
├─ comb_logic.circ
├─ comb_logic_simplified.circ
└─ comb_logic.vhd
```