Lab1 Cover Page

Digital Systems 1

Student Information (Print)

Braelyn
First Name

Rotman
Last Name

1006740
Student ID

Introduction: This laboratory exercise introduces you to the process of building digital-logic circuits using the schematic capture tool, *LogicWorks*. Make sure that you understand all of the steps in this lab, as you will be repeating them throughout the remainder of the course. Should you run into any difficulties with the lab material, ask your lab instructor during your lab session for clarification. Should you have trouble connecting to the Windows' server, review the instructions available on the School of Computer Science Wiki wiki.socs.uoguelph.ca. If the problem persists, please contact help@socs.uoguelph.ca with a description of the problem that you are encountering.

Submission: Submission of this assessment is a two-step process. First, you will need to complete this cover sheet with your personal information. (There is a video uploaded to CourseLink that shows how you can use the free version of Adobe Acrobat Reader DC to create both text and graphics within an existing PDF document.) Once completed, you will need to upload the PDF to a Dropbox labeled Lab 1. The second step involves completing the lab and then uploading the LogicWorks files to the previous Dropbox. You may make multiple submissions to the Dropbox, but only the last submission will be marked. Failure to submit both of the previous items in your last submission will result in an automatic grade of zero.

A couple of things to note:

- When working remotely, you should save your files on your own local machine. This way you always have a backup regardless of the state of the server.
- When uploading work to a Dropbox, it is your responsibility to check that you have uploaded the right items. Remember, what "goes up" must "come down." So, verify that you have uploaded the correct items for marking by downloading them and checking their contents. Failure to submit the correct files will result in an automatic grade of zero.

Lab 1 is due no later than the end of your regularly scheduled lab section. All submissions must be fully uploaded and verified (as described above) before the end of the lab session, otherwise, a grade of zero will be assigned. Therefore, don't leave things to the last minute.

1. A First Design

In this first lab, you will gain experience with LogicWorks by implementing and simulating a 2-input multiplexor (or mux). A 2-input multiplexor is a simple component of digital circuitry that passes (i.e., multiplexes) one of two input signals, IN0 or IN1, onto the output line, OUT. A third input signal, SEL, selects whether IN0 or IN1 is passed.

Specifically, the mux takes three input signals: IN0, IN1, and SEL. The mux generates one output signal: OUT. OUT has the value of IN0 when SEL is low and the value of IN1 when SEL is high.

The mux's functionality is described by the following Boolean equation:

$$OUT = (\overline{SEL} \bullet IN0) + (SEL \bullet IN1) + (IN0 \bullet IN1)$$

Though this is not the simplest way to implement a mux, it will be illustrative for our tour of LogicWorks. Later in the course, you will learn how to derive equations and then simplify them leading to more area efficient implementations.

We are now ready to build the actual circuitry of the above equation. The following section will guide you through the design of this simple equation using LogicWorks.

2. Design Using LogicWorks

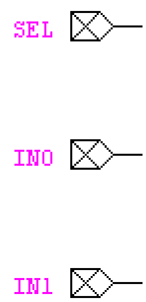
- Startup LogicWorks. Five windows should appear. The **LogicWorks** window is where all of the application's main functions can be selected, such as opening a new schematic. The **Palette** contains shortcuts for many of the common circuit editing commands, such as drawing a signal (i.e. a wire). The **Parts** window contains libraries of pre-built components that come with the software package. Later on, you will add your own parts as well as your own libraries. The **Design** window is where the actual circuit design is built. The final window is the **Timing** window, where you can see the results of simulations of your design.

- Your next step is to create your own library. Go to the **Parts** window and press the right mouse button. Click on the **New Lib…** option, name your new library "3120" by replacing the "*" in the **File Name** field of the popped-up window. Make sure that you have a directory on your local disk where you can store all of your designs. (And remember to store these designs on your own local machine when working on the server.) A library is automatically opened when created, from now on, however, you will need to open your library each time you start up LogicWorks. You can list all the libraries by clicking the left button to the right of the current library name (i.e. 3120.clf). You can select a library from the list by clicking on it.

- You can now move to creating the actual design. Begin by adding the input pins to your design. To do this, switch to the **Connect** library and double-click on **Port In**. When you move to the **Design** window, the symbol for a port should appear. If the symbol is not facing the way you would like it to, you can change its orientation by using the arrow keys before placing the symbol on the schematic. To place the symbol on the schematic, click the left mouse button. Place three ports on the left side of the window. When done, you can press the spacebar to indicate to the program that you are finished placing ports.

- You should assign your ports to our three inputs, SEL, IN0, and IN1. To name an input port, simply select and click the right mouse button. Choose the **Name** option. Name your ports SEL, IN0, and IN1 respectively. Also be sure to click the **Visible** option, which indicates that the names should be displayed on the schematic.

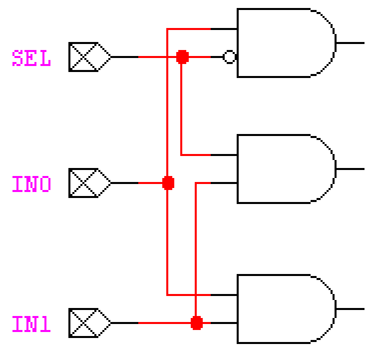
- Now save your design. Click on the **Design** window and, from the **File** menu, choose **Save As**. Call your design "tutorial". Your design should now look like this:



- You can now add the necessary gates. From the Boolean equation given on page 2, you can tell that you will first need to use three AND gates in your design. To do this, select the **Simulation Gates** library. Select the AND-2 gate by double clicking. The "-2" indicates a 2-input AND gate.

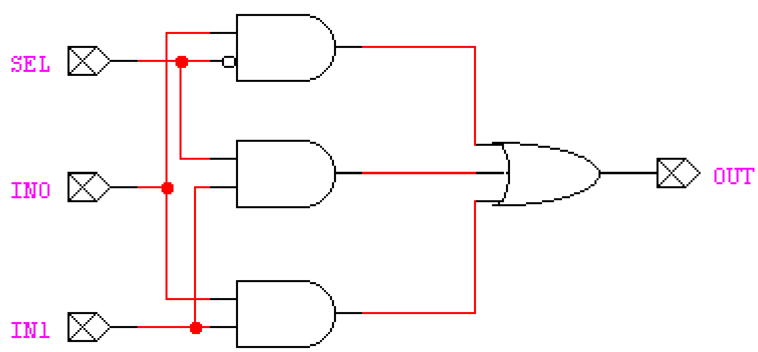
  Place two of the gates towards the bottom of your design. Again, you may have to change the orientation of the gate using the arrow keys before placing it. Use the space bar to indicate that you are done placing library components. You will use a different AND gate for the top one. Notice that one of its inputs, SEL, is inverted. Instead of adding an inverter, you can simply use the **AND-2 (1-INV)** gate, which indicates a 2-input AND gate with one inverted input. Place this gate at the top of your design.

- We can now wire the gates to the appropriate inputs. To add wires, select the **Draw Signal** option from the **Edit** menu of the **Schematic** window. As you go over the Schematic window, your cursor should take the shape of a cross. Simply tie together the inputs to the appropriate gates by clicking the left button wherever you wish to make a connection. Double-click to terminate the line. Be sure to tie SEL to the input of the first AND gate with the little bubble in front it; this means that the input will be inverted. Note that though two wires may cross on your screen, they only connect if a bubble appears. Your circuit should now look like the following:



- The remaining gate that you will need to add to your implementation is an OR gate. You could add two 2-input OR gates. However, a simpler way is to add one 3-input OR gate, which can be found in the **Simulation Gates** library. Go ahead and add it to your schematic, you can also tie it to its inputs.

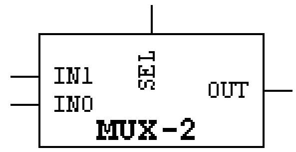
- The only remaining part of the design that needs to be added is the output port. To do this, use the same procedure as the input ports but use the **Port Out** component instead. Place the port on the schematic at the output of the OR gate and name it OUT. Your completed design should look like the following:



- Your circuit is now fully functional. However, it would not be very practical to keep having to redraw the circuit every time you need a MUX-2. Not only that, but the circuitry takes up a lot of space on your screen. To address these issues, you can use LogicWorks' **DevEditor** to create a symbol for your circuit. To do this, keep your schematic window open, then select the **New** command from the **File** menu and choose the **Device Symbol** document type. A new window should appear.

- From the **Options** menu, select the **Subcircuit and Part Type…** option. Select the **Create a subcircuit symbol and select an open circuit to attach to it** option. Choose the tutorial circuit that we have been working on. Click on **OK** then click **Done**.

- From the **Options** menu, select the **Auto Create Symbol** option. Type in "MUX-2" as the part name. Now click on the **Extract Pin List** option. You should see the names of all your ports appear. LogicWorks assumes that input pins are located on the left or top side of a design, while output pins are located on the right or bottom side of a design. Note that the following notation IN0..INn indicates all the inputs IN0, IN1 through INn. Move the SEL pin to the top by deleting it from the list of **Left Pin Names** and adding it to the list of **Top Pin Names**. Once you have done this, press the **Generate** button. Your symbol should now appear on your screen like this:

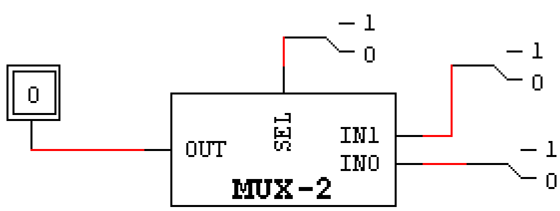


- Now that you have created a multiplexor circuit and created a symbol for it, save the multiplexor for future use in your 3120 library. Inside **DevEditor**, click on **File** and choose **SavePartAs**. Choose the 3120 library from the list of available libraries.

3. Simulation using LogicWorks

You have now completed the design component of your lab. What remains is to simulate your design in order to confirm that it works correctly.

- Open a new design by clicking on **File** in the **LogicWorks** window. Select **New** and choose **Design** as the document type. Place your 2-input multiplexor in your design. To simulate your multiplexor, you need a way to drive input signals and observe output signals. LogicWorks uses a **Binary Switch** to drive a signal and a **Binary Probe** to observe a signal. Both of these can be found in the **Simulation IO** library. Attach three switches to your inputs and one binary probe to your output. Your design should look like the following:



NOTE: To drag a switch in the **Design** window, hold down the Shift key and use the left mouse button.

- Now test your design by switching the inputs and watching the output. To 'switch' a switch, simply click on it. Is your design correct? In other words, does it produce the right output for each possible input?

A binary probe only displays the circuit's output at one instance in time. To get a more detailed understanding of your circuit's behavior, you can simulate your circuit for some time and display the input and output waveforms in the **Timing** window.

- Start by assigning names to your input and output signals. Click the right mouse button on the **IN0** signal. The signal will turn yellow and a menu will appear. Select the **Name** option. Name your signal "IN-0" and check **Visible**. The word **IN0** now appears on the left axis of the **Timing** window. Repeat for the **IN1, SEL**, and **OUT** signals.

  NOTE: If the signals do not automatically appear in the **Timing** window, pull down the **Timing** menu in your design window and make sure that **Add Automatically** is enabled.

- To start the simulation, choose **Simulator** from the pull-down menu. Now test your circuit by clicking on **Run** and switching your inputs to all eight possible settings. If the waveforms are too small, ask your TA about how to make them larger.

4. Other Important Things

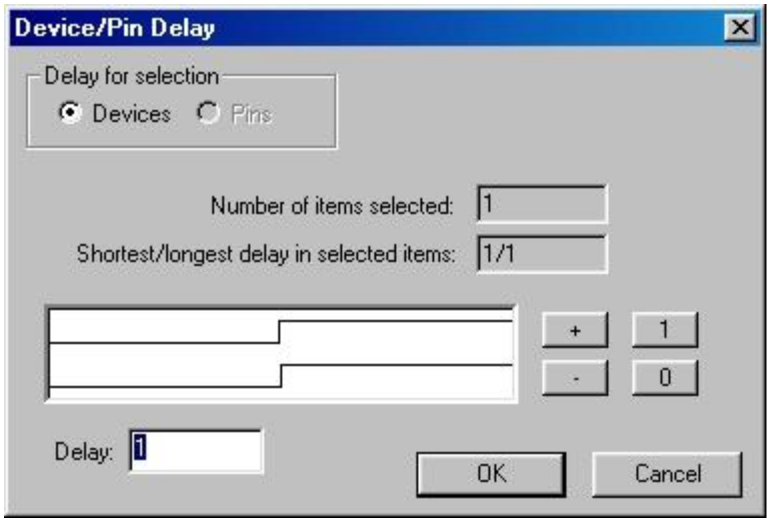
4.1Hidden Wires

Connecting many devices together using visible wires can be tedious, fraught with errors, and visually unappealing. <u>Therefore, I strongly suggest that you get into the habit of using Hidden Wires to connect devices.</u> Two unconnected signals labeled exactly the same (case matters) will behave AS IF THEY ARE CONNECTED. You should definitely use hidden wires to connect two or more signals that are far away on the screen. Visible wires should only be used to connect two pins that are within close proximity to one another.

LogicWorks also supports an Auto-Wiring feature: A module placed next to an existing module while pins are touching will automatically connect. Each module can be moved if so desired using the mouse or preferably the arrow keys.

4.2 Delays

Even in a simulated environment, changes to a device's inputs do not result in an instantaneous change in output. Rather, devices have a fixed delay. To change device delay, you should choose the device and select Simulation Parameters in Simulation Menu. Next, type a new delay time in the delay box or click the "+" button to increase the propagation delay in the device. Remember to press OK once finished. The figure below illustrates the process.



Note that the Simulation Parameters are used to view and set delays associated with both pins and devices. Pin delays are normally zero, but can be modified to fine tune the delays for different paths through a device.

4.3 Short Cuts

Below, you will find a list of useful short cuts:

- Ctrl-X -- Cut
- Ctrl-C -- Copy
- Ctrl-V -- Paste
- Ctrl-Shift-E -- Enlarge
- Ctrl-Shift-R -- Reduce
- Ctrl-S -- SAVE!!!
- Ctrl-Z -- Undo (oops)
- Ctrl-Shift-Z -- redo (un-oops)
- Ctrl-B -- New Breakout
- Arrow Keys -- Rotate a part before dropping it (check Num-lock)
- Ctrl-E -- Edit Text Tool
- Ctrl-H -- Zapper (delete) Tool
- Ctrl-Shift-S -- Signal Drawing Tool
- ESC -- Pointer Tool (usually)

5. Conclusion

You are now done with the first lab! Congratulations! You now understand the basics of using LogicWorks and are ready to design and simulate digital circuits. Remember to upload both the signed version of this lab document and your LogicWorks files to the Dropbox. By signing your name, you are guaranteeing that the work that you are submitting is solely your own.