Introduction to Computer Engineering
Design Project 0: Using your Hardware Design Tools

Objective

After completing this project, you should be able to start using the hardware design tools for this class. You will learn how to use LogicWorks to simulate basic circuits, and you will learn the correct circuit layout and wiring practices for your A&D Trainer Kit.

Preparation

You will need access to a computer that can run the LogicWorks simulation software package.

You will also need the A&D Trainer Kit and a 2504 chipset that contains HC series chips. If you purchased your chipset from the appropriate source, then you should have the correct chips for your trainer kit. Make sure that you always use the chips that are appropriate for your trainer kit. Older chipsets that contain LS series chips are not compatible with the A&D Trainer Kit. If you have questions, ask your instructor.

Before starting the lab assignment, read the appropriate sections of the 2504 Lab Manual and the A&D Board User Manual and Test Procedure for your trainer kit. Consult the sections of the Lab Manual on Circuit Breadboard and Wiring and Project Grading. If you have questions on the use of your trainer kit or your parts, see your a CEL GTA or your instructor.

Complete the diagnostic for your trainer kit. You will find the diagnostic in the A&D Board User Manual and Test Procedure, which is on the CEL website. Your instructor may ask that you submit documentation of having completed the diagnostic for your trainer kit. Follow your instructor's guidance.

Review this specification. You will carry out a series of activities in LogicWorks that will culminate in a wiring exercise. You will have to take your wired circuit to the CEL for validation. Begin early, as the lines for validation in the CEL will be long as the validation date approaches.

Getting Started

Install LogicWorks 5 on your computer from the CD, and then install any LogicWorks updates and patches that are posted on the LogicWorks 5 website (http://www.logicworks5.com).

Download the 7400 Actual.clf LogicWorks chip library from the CEL webpage. Unzip the file and put the library into the "Libs" directory of LogicWorks 5. Restart LogicWorks if necessary.

The .zip file that contained this specification also contains three LogicWorks circuit files, **P0_Gates.cct**, **P0_Bubble.cct**, and **P0_XOR.cct**. You will use each of these files to complete one activity in this project.

Open the circuit file called **P0_Gates.cct**. To do this, first start LogicWorks. A pop-up menu should appear when the program starts. To open the circuit file, choose the Create option from the pop-up menu to open a new circuit diagram. Then choose the Open option under the File menu, and search for the circuit file. Or, choose the Browse option from the pop-up menu, and search for the circuit file.

Navigating the LogicWorks Screen

Usually, when you open LogicWorks, you will see three tiled windows below the toolbars. Towards the upper left, you should see the **Drawing Window**. This is where the circuit diagrams appear. Towards the upper right, you should see the **Parts Palette**. This is where you will choose the parts that you want to place in the Drawing Window. Towards the bottom, you should see the **Timing Window**. When you use LogicWorks to analyze circuits in simulated time, the waveform results will appear in the Timing Window.

You will not be using the Timing Window in this assignment. Close it by clicking the "X" in the Timing Window's upper left corner. Depending on the size and resolution of your monitor, you may see a new part of the circuit appear. This is okay – this part of the circuit was always there. You may still not have enough room on your screen to see the entire circuit at once. Use the scroll bars at the right of the Drawing Window to move up and down, or left and right in the Drawing Window. When you make larger schematics in Projects 1 and 2, you will use the scroll bars to move around in your schematic.

If you close a window that you want to re-open, open the "View" toolbar menu and make sure that a check appears next to that window's name. Select a window that is unchecked to open it. Select a window that is checked to close it.

Part I: Functional Completeness

P0_Gates.cct contains four circuit diagrams. Each of these circuit diagrams corresponds to a logic function that you have studied in class. You may find the layout to be strange, since each circuit diagram shows a function that has been created using NAND gates only.

The NAND gate is a **functionally complete gate**. In the most basic sense, this means that we can implement each of the other logic functions using only NAND gates. In this section of the project, you will explore this concept by determining the logic function that each of the four circuits implements.

Each of the circuits has one or two inputs and one output. Each circuit receives its inputs using a LogicWorks part called a **binary switch**. You can click on the binary switch to change the input value that it supplies. The simulator must be running to change the value on a switch. If the simulator is not running, you will receive a pop-up asking if you want to run the simulator. Select "OK" and LogicWorks will start the simulator. Now you will be able to change the switch inputs.

Note that the labels for each input actually label the wires that connect the switches to their gates. Right-click on one of the wires, and you should see the signal name highlighted along with the wire. You can choose options from the pop-up menu that will allow you to view or change the attributes of the wire, including its name and its color. When you simulate circuits in Projects 1 and 2, you will name signals by naming the wires that carry them, and not the switches that you might use to test your circuit.

A part called a **binary probe** monitors the inputs and outputs of the circuits. Binary probes display the value that an input or an output carries. They are useful debugging tools, as you can use them to detect input or output values that do not correspond to your expectations. Be careful, because wiring or physical problems with your circuit can result in other values. Look at the binary probes that are below the circuit diagrams:

- A **high-impedance**, or "Z" value results when a binary probe is unconnected to any input or output.
- A **conflict**, or "C" value results when a binary probe is connected to both logic-0 and logic-1.
- A **don't know**, or "X" value results when the inputs are insufficient to determine a gate's output value.

If these values occur on inputs or outputs of gates that you are using in a circuit, you should inspect the circuit for connection problems. They typically indicate that a problem exists somewhere in the circuit.

Let's begin with Figure 1. Page 1 of the Validation Sheet has a truth table that corresponds to each of the figures. Use the first truth table to record your observations for each input that you apply to the circuit in Figure 1. Apply an input by clicking the switch so that the appropriate value appears on the input. Use the binary probe on the input to ensure that you have applied the correct value. Apply each input value, and record the value that you observe on the output binary probe. Use the information in the truth table to identify the gate implemented in the figure.

Repeat these steps for the circuits in each of the other figures. Each of the gates in the subsequent figures has two inputs, so make sure to apply all four combinations of inputs. Record the output for each input combination. Use the information to identify the gate implemented in the figure.

Part II: Bubble Propagation

Follow the instructions in the "Getting Started" section to open the circuit file **P0_Bubble.cct**. After you open the circuit diagram, follow the instructions in the "Navigating the LogicWorks Window" section to close the Timing Window.

Even though we can implement each of the gates contained in P0_Gates.cct using only NAND gates, it is not particularly efficient to do so using the circuits shown there. Fortunately, we have a way to transform sum-of-products circuits directly into a form that uses only NAND gates.

P0_Bubble.cct contains three circuit diagrams. Figure 1 shows a two-level (plus inverters) sum-of-products circuit. First, trace the circuit and determine the function of A and B that the circuit implements. Then, use the switches to apply all four combinations of the inputs A and B to the circuit. Use the binary probes on the inputs to ensure that you have applied the correct value. Finally, use the information you have obtained to identify the gate implemented in the figure.

Figure 2 shows a modified version of the circuit in Figure 1. Think of the bubbles as being inverters. The output of the top AND gate has been inverted once after it leaves the AND gate, and again before it enters the OR gate input. **Does inverting a signal twice change its value?** Use the inverter circuit of Figure 3 to verify this. **Which of your Boolean algebra principles expresses this fact?** The bottom AND gate of Figure 1 has been "changed" in a similar fashion; one bubble inverts the output after it leaves the AND gate, and a second one inverts it again before it enters the OR gate.

The circuit of Figure 1 has been changed into the circuit of Figure 2 through **bubble propagation**. Imagine that we have placed two inverter bubbles on the wire that connects each AND gate output to one of the OR gate inputs. These bubbles have propagated through the circuit, until one of them reached the AND gate output, and the other reached the OR gate input. It should be obvious that the two first-level AND gates have been changed into NAND gates. But what about the second level OR gate? **What kind of gate does an OR gate become if we invert <u>all</u> of its inputs? Which of your Boolean algebra principles expresses this fact?**

Use the switches to apply all four combinations of the inputs C and D to the circuit. Use the binary probes on the inputs to ensure that you have applied the correct value. Finally, use the information you have obtained to identify the gate implemented in the figure.

Try replacing the "invert-OR" gate with its equivalent gate. Highlight the invert-OR gate by clicking on it, then hit the Delete key. If you delete the wrong item, choose "Undo" under the "Edit" menu, or press Control-Z to undo the last action. Now, find the replacement part on the Parts Palette. Select the "Simulation Gates" library on the pull-down menu at the top of the Parts Palette, then scroll down the library until you find the right part called. The number after the gate name indicates the number of gate inputs. Make sure you choose a gate having the right number of inputs.

Place the gate into the place that the invert-OR gate occupied. It's okay if the inputs of the gate overlap with some of the wire, but make sure that the wires connect with the inputs and outputs of the gate. If you see an unspecified logic value on the output binary probe, delete and replace the gate.

Use the switches to apply all four combinations of the inputs C and D to the circuit. Use the binary probes on the inputs to ensure that you have applied the correct value. Finally, use the information you have obtained to identify the gate implemented in the figure. If you have replaced the invert-OR gate with the right kind of equivalent gate, you should find that the circuit still implements the same function.

Bubble propagation is a simple way to transform circuits into a form that uses only one kind of gate. You must be careful when using bubble propagation, as it only works for circuits that are organized in specific ways. You can learn more about bubble propagation from your text and your class lectures.

Part III: The XOR Gate Circuit

Follow the instructions in the "Getting Started" section to open the circuit file **P0_XOR.cct**. After you open the circuit diagram, follow the instructions in the "Navigating the LogicWorks Window" section to close the Timing Window. Also, print a copy of the diagram (File → Print), since you will have to take a printout of the circuit with you to the CEL for validation. Print this circuit on a landscape page. Check the print options (File → Print Setup) to make sure that you use the right orientation.

Use LogicWorks to view the circuit schematic. P0_XOR depicts a chip-level implementation of an XOR gate that uses only NAND gates. The pins and gates on the LogicWorks chips match the pins and gates on the chips in your chipset. However, since wiring in LogicWorks can be different than wiring on your trainer kit, review the wiring guidelines in the 2504 Lab Manual.

Use LogicWorks to verify that the circuit in the diagram operates as a two-input XOR gate. Apply the input combinations shown on the validation sheet one at a time. Each input case should cause the output logic indicator to take on the right value. Even though you will not use the timing window in this project, you can find more information on how to use the timing window to create and observe simulation output in the LogicWorks notes that are posted to the CEL webpage.

Before you begin wiring your circuit, consult the additional notes on your printed circuit schematic. Use these notes to determine which of your switches supply the inputs, and which of your logic indicators display the outputs. One logic indicator will display the output, and two other indicators will display the values on the input switches.

Choose a location on your trainer kit to place the chips shown in the circuit file. Make sure that you make all of the exact pin-to-pin connections that are shown in the circuit diagram. The wire colors shown in the circuit is only one example of a color-coding that you might use. When wiring your circuit, follow the wiring guidelines and grading criteria discussed in the Lab Manual and on the validation sheet for Part III of the assignment.

Note that each chip must be connected to power (+5 volts) and ground on your trainer kit. However, LogicWorks does not require power and ground connections for the chips to work in simulation. For your convenience, this circuit diagram shows the power and ground connections. Not every circuit diagram that you use will show this same information. In general, you should refer to the Lab Manual for chip diagrams to identify the power and ground pins.

Project Validation

Complete the parts of this assignment in order. After you build your circuit and verify that it works, take the wired circuit, a copy of your circuit diagram, and all three pages of your validation sheet to the CEL during operation hours. Any GTA on-duty in the CEL should be able to validate your lab.

A GTA will validate your circuit and evaluate your wiring. Wire the circuit carefully the first time. You are encouraged to rewire the circuit until you achieve a perfect wiring score. In addition to validating your project, GTAs are available to provide wiring assistance.

Honor Code Requirements

You must complete this project individually. Do not discuss any aspect of your solution or approach with any other student. Copying or using another student's design is a violation of the Virginia Tech Honor Code, and will be prosecuted as such. You may discuss general features of LogicWorks, your trainer kit and parts kit. Direct all other questions to your GTA or to your instructor.

Project Submission Requirements

You must complete the entire validation sheet. Your instructor may require additional elements. Follow your instructor's guidance. Turn in your validation sheet to your instructor on the due date announced by your instructor.

Introduction to Computer Engineering
Project 0 Validation Sheet – Page 1

The complete Validation Sheet for Project 0 is three pages long. You only need the last page for your CEL validation, but the GTA will not validate your circuit until you have completed Parts I and II of the project. Make sure that you take all pages of the Validation Sheet to the CEL when you go to have your trainer kit validated.

Name: _____ ID Number: _____

Pledge: I have neither given nor received unauthorized assistance on this assignment.

_____

Part I: Functional Completeness
The student should complete this portion of the Lab assignment at home. Open the circuit diagram **P0_Gates** in LogicWorks. Determine the output for each set of input cases, and complete each truth table using your observations. Then, identify each gate based on the truth table you complete.

Use Figure 1 to complete the truth table below.

Input A	Output
0	
1	

Figure 1 is a/an _____ gate.

Use Figure 2 to complete the truth table below.

Input B	Input C	Output
0	0	
0	1	
1	0	
1	1	

Figure 2 is a/an _____ gate.

Use Figure 3 to complete the truth table below.

Input D	Input E	Output
0	0	
0	1	
1	0	
1	1	

Figure 3 is a/an _____ gate.

Use Figure 4 to complete the truth table below.

Input F	Input G	Output
0	0	
0	1	
1	0	
1	1	

Figure 4 is a/an _____ gate.

Introduction to Computer Engineering
Project 0 Validation Sheet – Page 2

Part II: Bubble Propagation
The student should complete this portion of the Lab assignment at home. Open the circuit diagram **P0_Bubble** in LogicWorks. Use the circuits to answer the questions below. Then, determine the output for each set of input cases and complete each truth table using your observations.

Use Figure 1 to answer these questions.

The circuit in Figure 1 implements the function *f*(A,B) = _____.

Input A	Input B	Output
0	0	
0	1	
1	0	
1	1	

Figure 1 is a/an _____ gate.

Use the inverter circuit and your knowledge of Boolean algebra to answer these questions.

Does inverting a signal twice affect the signal's value? _____

Which of your Boolean algebra principles expresses this fact? _____

Use Figure 2 to answer these questions and your knowledge of Boolean algebra to answer these questions.

What kind of gate does an OR gate that has **all** of its inputs inverted represent? _____

Which of your Boolean algebra principles expresses this fact? _____

Input C	Input D	Output
0	0	
0	1	
1	0	
1	1	

Figure 2 is a/an _____ gate.

Does changing the invert-OR gate into its equivalent change the function? _____

Introduction to Computer Engineering
Project 0 Validation Sheet – Page 3

Students and GTAs must complete this validation sheet in ink.

Part III: The XOR Gate Circuit

The student used a LogicWorks circuit diagram to implement a two-input XOR gate. In addition to all three pages of this validation sheet, the student should have a copy of the circuit diagram used to wire the circuit.

The GTA should complete the following table to verify the correct operation of the circuit. Consult the circuit diagram for the location of the inputs and outputs. Perform the circuit validation or have the student carry it out under your supervision.

Input A	Input B	Observed Output X
0	0	
0	1	
1	0	
1	1	

Circuit Wiring Validation

Evaluate the student's wiring efforts. The student starts with 10 points. Deduct **one point** for each item on the following list that you find. If you decide that a fault exists, show it to the student and mark the appropriate space.

a. Wires are more than one inch above the breadboard. _____
b. Wires are cut to random lengths. _____
c. Wires are stripped too short. _____
d. Wires are stripped too long. _____
e. Wires go over the tops of chips. _____
f. All of the wires are one color. _____
g. Power or ground wires are not correctly color-coded. _____
h. Daisy-chained power distribution. _____
i. Daisy-chained signal distribution. _____
j. Wired pins in the circuit don't match those in the schematic _____

Final Wiring Score _____

Before signing the validation sheet or allowing the student to leave the CEL, make sure that the student removes the wires from the trainer kit. The student may leave power and ground wires on the board.

Validated by: _____ (Print and sign your name.)

Date and Time: _____