CURIE Academy, Summer 2021 Lab 1 Handout: Computer Engineering – Hardware Perspective

Prof. Christopher Batten School of Electrical and Computer Engineering Cornell University

In this lab assignment, you will work in a group of three scholars to explore the field of computer engineering from the hardware perspective by assembling basic logic gates to implement a simple "calculator" for adding small binary numbers. Feel free to consult the lab notes as you work through the lab. In Part 1, you will experiment with basic circuits before developing a NAND gate. In Part 2, you will experiment with basic logic gates before developing a parity checker. In Part 3, you will gradually develop the full-adder described in the lab notes. After completing all three parts, all of the scholars will come together to create a multi-bit adder unit where we will use Zoom to implement the carry chain. For each part and subparts you will need to have an instructor observe the desired milestone and initial the appropriate box on the shared Google sign-off spreadsheet. Here is a list of each milestone:

- Part 1: Understanding Basic Digital Circuits
 - Part 1.A: Experiment with LED
 - Part 1.B: Experiment with Inverters
 - Part 1.C: Develop NAND Gate
- Part 2: Understanding Basic Logic Gates
 - Part 2.A: Experiment with Logic Gates
 - Part 2.B: Develop Parity Checker
- Part 3: Building a Simple Calculator
 - Part 3.A: Experiment with Half-Adder
 - Part 3.B: Develop Full-Adder
 - Part 3.C: Share Photo or Video of Full-Adder on Slack

Before beginning, remove the following materials from your electronics prototyping kit and put all of the remaining items aside (see Figure 1).

- Breadboard-based prototyping platform
- Two red LEDs
- Two resistors
- Pre-cut wires and jumper wires
- Integrated full-adder board
- Black power adapter with barrel connector
- 9 V battery in battery holder

Part 1. Understanding Basic Digital Circuits

In this part, you will wire up some simple circuits using the prototyping platform. You can use either the precut wire or the jumper wires. Consult the lab notes for more information on the connectivity that is inside the breadboard. Before getting started make sure your prototyping platform is *not* plugged in (i.e., the barrel plug from the wall adapter or the 9V battery should not be plugged into the breadboard power supply on the right of the prototyping platform).



Figure 1: Materials Required for Lab 1

Part 1.A Experiment with LED

Wire up the simple LED circuit shown in Figure 2 using the breadboard diagram in Figure 3 as an example. You will need to use a red and black jumper wire to connect VDD and ground from the white breadboard at the bottom to the green circuit board at the top, and you will also need to find some free space on the breadboard to insert the LED and resistor. Once you have wired everything up, show your circuit to the other scholars in your group to verify that things are connected correctly. Once all scholars are confident things are connected correctly, plug in either the wall adapter or the 9V battery. Test the circuit by turning the board on/off using the switch on the breadboard power supply. Try putting the LED in both directions.

Sign-Off Milestone: Show an instructor that the LED lights up when you turn the board on using the switch on the breadboard power supply. Ideally, all scholars in the group should sign-off this milestone before moving on to the next milestone. Work together to help each other achieve the milestone!

Critical Thinking Questions to Discuss Within Your Group: What do you think would happen if we used a resistor with higher resistance? This would be equivalent to using an even narrower pipe in our water circuit analogy. What do you think would happen if we used a resistor with lower resistance? Would would happen if we put two resistors in series or in parallel?



Figure 2: Simple LED Circuit



Figure 3: Simple LED Circuit Implemented on Prototyping Platform

Part 1.B Experiment with Inverters

Turn off your prototyping platform using the switch on the breadboard power supply. Wire up a PMOS and an NMOS transistor to create a single inverter. Use the circuit in Figure 4 and the breadboard diagram in Figure 6 as a guideline. Note how the points A, B, and C in Figure 4 correspond to points A, B, and C in Figure 6. We have already inserted two PMOS transistors at the top of your breadboard and two NMOS transistors at the bottom of your breadboard. We have also already wired up VDD and ground to the transistors, and we have inserted two orange jumper wires.

We are using a digital input switch to send a digital value into the circuit and a digital output LED to observe the output of the circuit. Once you have double checked your wiring, turn on the board and try toggling the input to the inverter. When the input to the inverter is one is the output zero? When the input to the inverter is zero is the output one?

Turn off your prototyping platform using the switch on the breadboard power supply. We now wish to wire up four transistors to create two back-to-back inverters as shown in Figure 5. Discuss how



Figure 6: Breadboard Diagram for Testing Inverter Circuit

to implement this circuit on the breadboard with the other scholars in your group. Once you have come up with a plan as a group implement the circuit on the breadboard. Once you have wired the two back-to-back inverters up, show your circuit to the other scholars in your group to make sure things are connected correctly. Then turn on the board and test the functionality of the circuit.

Sign-Off Milestone: Show an instructor that the back-to-back inverters have the expected behavior. Ideally, all scholars in the group should sign-off this milestone before moving on to the next milestone. Work together to help each other achieve the milestone!

Critical Thinking Questions to Discuss Within Your Group: What do you think would happen if we cascaded three inverters? Four inverters? What would happen if we tied the output of a three inverter chain to the input of the chain?

Part 1.C Develop a NAND Gate

In this subpart, you will develop a more complex circuit that implements a new kind of logic gate. Before beginning, make sure your prototyping platform is off using the switch on the breadboard power supply.

Consider the circuit shown in Figure 7. Before continuing can you determine what this circuit does using pencil-and-paper? Try all four possible combinations of input values and derive the expected truth table. Discuss with the other scholars in your group. Once you have studied the circuit using pencil-and-paper, discuss how to implement the circuit on the breadboard with the other scholars in your group. You can use Figure 8 to help get you started. We have removed one yellow jumper, moved one orange jumper, and shown where to connect input A and input B. Once you have come up with a plan, then as a group implement the circuit on the breadboard. Once you have wired the logic gate, show your circuit to the other scholars in your group to make sure things are connected correctly. Then turn on the board and test the functionality of the circuit. Does the circuit have the functionality you expected based on your pencil-and-paper analysis? If not, try to debug what is going wrong.

Sign-Off Milestone: Show an instructor the truth table you have written on paper. Show an instructor that the circuit correctly implements a NAND gate. Ideally, all scholars in the group should sign-off this milestone before moving on to the next milestone. Work together to help each other achieve the milestone!

Critical Thinking Questions to Discuss Within Your Group: Why is this gate called a NAND gate? Can we use a NAND and the inverter from the previous part to implement an AND gate? Can you think of how to implement a NOR gate?



Figure 7: NAND Circuit (P = PMOS transistor, N = NMOS transistor)



From Digital Input Switch (this is input A)

Figure 8: Initial Breadboard Diagram for Testing NAND Circuit

Part 2. Understanding Basic Logic Gates

In the previous part, we learned about basic circuits and how to use these circuits to implement logic gates. In this part, we will begin working with the logic gate abstraction. Using this abstraction, we no longer need to worry about the details involved with transistors. We can simply focus on the logical operation of each gate.

Part 2.A Experiment with AND, OR, XOR Gates

Turn off your prototyping platform using the switch on the breadboard power supply. Wire up the AND gate that is contained within the second chip as shown in Figure 9. Turn on the board and try all four input combinations. Does the output correspond to the expected truth table? Wire up the other two chips using the digital input switches and digital output LEDs in a similar fashion and test their functionality. Using a truth table and the information from the lab notes work together to determine which chip contains XOR gates and which chip contains OR gates.

Sign-Off Milestone: Once you have determined which chip contains the each type of gate, show an instructor your truth tables and demonstrate the operation of either the XOR or the OR gate. Ideally, all scholars in the group should sign-off this milestone before moving on to the next milestone. Work together to help each other achieve the milestone!

Critical Thinking Questions to Discuss Within Your Group: So far we have see four two-input logic gates: NAND, AND, OR, and XOR. How many different logic gates are possible if we limit ourselves to gates with just two inputs and one output?



Figure 9: Breadboard Diagram for Testing AND Gate

Part 2.B Develop Parity Checker

In this subpart, you will develop a more complex *parity* hardware unit for a four-bit input. A parity unit should produce a zero when the number of ones in the input is even and should produce a one when the number of inputs in the input is odd. Parity units are actually quite common to help detect errors when sending messages between systems. The sender can calculate the parity of a message and send the parity bit along with the message. The receiver can then also calculate the parity of the message and compare it to the parity bit sent along with the message. If the parity bits do not match then it is likely that one of the bits in the message was corrupted.

Figure 10 illustrates how we can implement a parity hardware unit using three XOR gates. Wire up the three XOR gates on your breadboard, and verify its functionality by filling out the truth table below. You will need to carefully consider how the various points in the circuit in Figure 10 should correspond to wires on your breadboard. Check with the other scholars in your group to compare truth tables.

Sign-Off Milestone: Once you have a working parity unit, show an instructor its operation and verify the corresponding truth table. Ideally, all scholars in the group should sign-off this milestone before moving on to the next milestone. Work together to help each other achieve the milestone!

Critical Thinking Questions to Discuss Within Your Group: How would we implement a parity hardware unit for five, six, or more bits?



A B C D out

Figure 10: Four-Bit Parity Hardware Unit

Part 3. Building a Simple Calculator

In the previous part, we started building more interesting hardware units using logic gates. In this final part, we will incrementally build a simple binary adder unit. We will begin by implementing a half-adder and then a full-adder. You should review the background material on binary arithmetic and implementing adders in the lab notes.

Part 3.A Experiment with Half-Adder

Work with the other scholars in your group to wire up the half-adder shown in Figure 11. Make sure it works as expected.

Sign-Off Milestone: Once you have a working half-adder, show an instructor its operation and verify the corresponding truth table. Ideally, all scholars in the group should sign-off this milestone before moving on to the next milestone. Work together to help each other achieve the milestone!

Part 3.B Develop Full-Adder

Wire up the full-adder shown in Figure 12. This unit is complicated enough, that you should plan out your wiring ahead of time using the template in Figure 13. Draw lines to represent where you plan to place each wire on your breadboard.

Sign-Off Milestone: Once you have wired up the full-adder, show an instructor its operation and verify the corresponding truth table. Ideally, all scholars in the group should sign-off this milestone before moving on to the next milestone. Work together to help each other achieve the milestone!



Figure 11: Half-Adder



Figure 12: Full-Adder



Figure 13: Planning Diagram for Breadboard Implementation of Full-Adder

Part 3.C Share Photo or Video of Full-Adder on Slack

Now that you have your full-adder working, take a photo of your final circuit and/or record a short video of your full-adder in operation. Then upload your photo and/or video to Slack using the #lab1-final-media-milestone channel. It would be great if you could maybe even have someone else take a photo of you either debugging your final circuit or showing it off for the camera!

Sign-Off Milestone: Show your uploaded photo and/or video to an instructor.

Next Steps

Parts 1–3 are the required portions of the lab. If you complete these parts quickly, you can start to experiment with a multi-bit adder. Note that once everyone in the academy has completed Parts 1–3, all of the scholars will come together to experiment with the multi-bit adder so your goal here is really just to start to explore this idea.

We could continue to add more and more logic gates to implement a multi-bit adder, but we will instead leverage modularity to simplify building a ripple-carry adder. As a first step, take a look at the integrated full-adder board which is basically a printed circuit-board that contains the exact same circuit you implemented in the previous part. You will see that it has three input switches corresponding to the three one-bit inputs. It also includes a "mode switch" which can be set to either "independent mode" or "chain mode". Set the integrated full-adder board to "independent-mode" and test its operation. Power the integrated full-adder board using either the black power adapter with barrel connector or the 9 V battery. Verify that it produces the same truth table as the full-adder you implemented in the previous part.

Now work together as a group to implement the ripple-carry adder in Figure 14. Start by picking two numbers between 0–4 and then determine their binary representation. Each scholar's integrated full-adder board will be used to calculate one bit in the multi-bit addition. So one scholar will calculate the first bit and then tell the next scholar whether the carry out bit is zero or one. The next scholar will set her carry in bit appropriately. Once all scholars have finished we will have calculated a three-bit binary addition implemented using basic logic gates!



Figure 14: Four-Bit Ripple-Carry Adder (FA = full-adder)