LAB 1: Mixed-Logic Design and Quartus

OBJECTIVES

- Understand the operation of Quartus Prime as a digital design and simulation tool.
- Learn how to construct mixed-logic circuit designs using discrete components (IC's, LED's, switches, and resistors).

INTRODUCTION

There are many ways to go about building a combinational logic circuit. Many introductory digital logic classes design circuits using a purely positive-logic (p-logic) technique. This means that a high voltage always corresponds with a logical value of true, and a low voltage always corresponds with a logical value of false. This is an easy concept to grasp when you are starting out in digital design, however it can be a detriment for more complex circuits. For example, reset and enable signals commonly activate when a low voltage their input value. For these types of signals, p-logic requires the use of level-shifters (i.e., inverters or NOT gates), which can be inefficient when many active-low inputs and outputs exist in your circuit.

Mixed logic is a powerful circuit design methodology that decouples the ideas of voltage levels and truth levels. In mixed logic, there are two different types of signals: active-high and active-low. Active-high signals are true (i.e., have a truth value of '1') when the signal has a high voltage and are false (i.e., have a truth value of '0') when the signal has a low voltage. Active-low signals are the opposite; they are true when the voltage is low and false when the voltage is high. Logical negations arise from a mismatch between the activation level of two connected devices (or a device and an input or a device and an output). This flexibility allows us to use the same gate to implement many different logical operations depending on the activation levels of the input and output signals.

LAB STRUCTURE

In this lab, you will become familiar with the process of implementing logic circuits using the mixed-logic convention. In § 1, you will be introduced to logic circuit design using only the positive-logic interpretation of gates. In § 2, you will explore the benefits of using mixed logic to implement a circuit. In § 3, you will prove that positive-logic and mixed-logic implementations of an equation can be equivalent if you convert all activation levels to match between the two circuits. Finally, in § 4 you will design multiple mixed-logic circuits and optimize them as one unit.

REQUIRED MATERIALS

- *Quartus 22.1 Tutorial*
- Pinouts of common 74'xx Parts
- Hardware – Getting Started
- Specs for 74HC00 Quad 2-input NAND
- Specs for 74HC02 Quad 2-input NOR

SUPPLEMENTAL MATERIALS

- How to Use a Breadboard – YouTube Video
- Protoboard layout for PowerPoint
- Protoboard layout for Acrobat (.pdf)
- Switch Bank Figure
- Legends for DIP switches and LEDs (.pptx)
- Honorlock and Lab Quiz info

REVIEW FROM LAB 0

Your lab PI demonstrated the functioning of some of the components in your toolbox during your Lab 0, including SPST switches, LEDs, SIP resistor packs, DIP resistor packs, and breadboards. **Before** your Lab 1, verify that you can repeat what your PI demonstrated in Lab 0. If you are uncomfortable with any of these parts, reread the relevant class notes and the *Hardware: Getting Started* document, and review the relevant course notes and videos. You can also go to any office hour to get additional help. You are expected to know how to do all of the following (from Lab 0).

1. Know how to place an IC (integrated circuit) device on the breadboard, where power/ground (i.e., Vcc/GND) are located, and how power/ground are connected. You should be able to find pin 1 on all their 74'xxx ICs. You should understand how to find the pinouts of all of your 74'xxx ICs.
2. Know how DIP and SIP resistor packages work and how to use them. You should also be able to sketch the internal resistor circuit for each.
3. Know how to construct switch circuits using a SIP resistor package. You should also be able to sketch a switch circuit diagram. Only use the SIP as pull-up resistors in your switch circuits. (Switch circuits can be made with pull-down resistors, but these circuits should use different value resistor values than circuits with pull-up resistors. Since you only have single value SIP resistor packs, do **NOT** use switch circuits with pull-down resistors in EEL 3701 during the entire semester.)
4. Know how to construct a 2-input NAND-gate circuit (using a 74HC00) on your breadboard (implementing the equation Y = /(A * B)), with two active-high inputs, each connected to a switch circuit, and an output active-high high output, connected to an active-high LED circuit.

- Know how to use your multimeter to verify the voltage supplied from your DE-10 Lite board. (If you don't yet know how to do this, please ask your PI to demonstrate during your Lab 1 or office hours.)
- Know how to use your multimeter to verify electrical node connectivity of your breadboards. This includes the power/ground buses and the general signals. (If you don't yet know how to do this, please ask your PI to demonstrate during your Lab 1 or office hours.)

5. As hopefully discussed in Lab 0 and as discussed in class, you should know how to construct active-low and active-high LED circuits using DIP resistors**. The LED in each of these circuits should be lit (on) when the relevant signal is true**, i.e., an active-low LED circuit should light up when the signal is low (ground) and an active-high LED circuit should light up when the signal is high (Vcc). You should also be able to sketch an LED circuit diagram.

PRE-LAB PROCEDURE

During Lab 0 you received your lab kit. For documentation purposes, take pictures of all of the parts in your kit (chips on the foam, PCBs, breadboards, etc.) and include these images in your Pre-Lab Report for Lab 1.

1. INTRODUCTION TO POSITIVE LOGIC

Most of the world uses positive-logic only; and so will you in this section.

1. Using only AND gates with **NO** bubbles, OR gates with **NO** bubbles, and level-shifter (NOT) gates, design and draw the circuit for the following logic equation.

   V = /[(A + /B) * /(C * /D)] (1)

   Do **NOT** simplify the equation. For this part only, do **not** use the alternative way to draw the AND and OR gates. The output and all the inputs are active-high. Do this by hand, i.e., without Quartus. Write the equations for intermediate inputs next to the input pins of each gate. Since you will not build this circuit, you are not limited (in this section) to parts in your lab kit. Take a screenshot and submit this in your report (as specified in the *Lab Submission Template)*. All hand-drawn circuits must be submitted in all lab reports.

2. Draw the circuit for the equation below by hand and then with Quartus **using only the positive-logic logic gates** (i.e., AND, OR, NAND, NOR, and NOT gates) and **active-high** inputs and output. Positive-logic gates do **not** have bubbles at input pins. Use a project and filename of `Lab1a`.

   W = /[(A + /B) * /(C * /D)] (2)

   Note that this equation has the same right-hand side as the equation (1). Do **not** simplify or otherwise manipulate the equation. **Always add your name on each schematic that you design in Quartus** (as described in the *Lab Rules and Policies*). All hand-drawn circuits must be submitted in all lab reports.

3. Generate a truth table (using 0's and 1's) for equation (2) and then make an accompanying voltage table (using L's and H's). Both tables **MUST** be in counting order. All truth and voltage tables must be submitted as part of your Pre-Lab Report, as shown in the required *Lab Submission Template*.

4. Generate a complete timing (**not** functional) simulation for this circuit in Quartus. Note that with the present version of Quartus, there is no option available for obtaining a timing simulation when using the MAX 10 FPGA, so for **THIS LAB ONLY**, **use any of the MAX V devices**. **All other simulations** in this lab can be functional (not timing) simulations. **Every simulation that you turn in this semester must be annotated** and **include a caption**. Unannotated simulations will receive no credit. Annotations are meant as a guide to help you and others reading your Pre-Lab document (like your PI) understand the simulation. **It should help explain what is going on and prove that your circuit works.** Simple circuits may need only a few annotations, while more complicated circuits (like those that you will design later in the semester) may need many more. The caption references the appropriate part of the lab document.

5. Since the simulator in Quartus generates a voltage simulation (not a logic simulation), use your voltage table to verify your simulation outputs. Unfortunately, in Quartus simulation a "0" represents a low voltage and a "1" represents a high voltage. (This is the purpose of the annotation.)

6. Take a screenshot of the simulated results. Annotate the simulation screenshot, i.e., add notes with arrows pointing to important results, and add the annotated simulation to your pre-lab report. **Every simulation that you turn in this semester must be annotated** and **include a caption**. Unannotated simulations will receive no credit. Annotations are meant as a guide to help you and others reading your Pre-Lab document (like your PI) understand the simulation. **It should help explain what is going on and prove that your circuit works.** Simple circuits may need only a few annotations, while more

complicated circuits (like those that you will design later in the semester) may need many more. The caption references the appropriate part of the lab document.

7. Submit your Quartus archive file to Canvas with both the design and the simulation (use the filename `Lab1a.qar`). Also include your circuit diagram screen shots, truth/voltage tables, and your annotated simulation results in your prelab document.

2. TOTALLY OPTIONAL PRE-PRE-LAB

1. This section is for your eyes only, i.e., it will **not** be turned in; it **will** help you understand the big picture, so I **strongly** recommend that you do it!
2. With A, B, and C all active-high, make a truth table and a voltage table for C = /A*B. Design the circuit in Quartus and verify that the Quartus voltage simulation matches the voltage table. Now change A to active-low in Quartus by just changing the name of the signal from A to A_L. Do not change the circuit design in Quartus. Does the voltage simulation change? How about the truth table and voltage table? What is the equation of this "new" circuit?
3. See also the hint that corresponds to the above at mil.ufl.edu/3701/labs/quartus_mixed-logic.html.

3. INTRODUCTION TO MIXED LOGIC

1. Again draw (by hand and then with Quartus, using project and filename `Lab1b`) a circuit design for equation (3). Note that this equation has the same right-hand side as the equations (1) and (2). In this design you are **not** limited in your gate selection.

   X = /[(A + /B) * /(C * /D)] (3)

   Choose activation-levels for the inputs and outputs that will allow you to minimize the number of **gates** necessary. Do **NOT** simplify the equation; draw the gates in such a way that the logic of the equation is apparent in the circuit, e.g., if you need a 2-input AND gate with bubbles at the inputs (a BAND2 in Quartus), draw the gate this way instead of as a NOR gate. This is called a direct implementation of an equation. Show intermediate inputs as in part 1. **Unless otherwise stated, all hand-drawn circuits this semester must be drawn as described above (direct implementation with intermediate input equations).** Quartus-generated circuits do not need to include intermediate input equations but should otherwise be created as described above.
2. Note that Quartus includes only some of the mixed-logic circuit elements in its libraries. For instance, Quartus does not have a level-shifter with the bubble on the input side.
3. Generate a voltage table for this design. (Hint: The truth table should be the same as the one made in the previous section.) Truth tables depend only on the logic equations. Note that the voltage table and the simulation for this design may **NOT** match the voltage table and the simulation for the design of parts 1-4.) Submit (through Canvas) the archive file with your design and simulation, named `Lab1b.qar`. Remember to put the screen shot of the design, annotated simulation, and the truth and voltage tables in your pre-lab document file. **For this entire semester, when using Quartus to draw a circuit, label all active-low inputs or outputs with _L at the end** (since (L) and .L will not work), e.g., R(L) will be written as R_L. **All active-high signals will either have no ending or end in _H (your choice)**, e.g., S(H) will either be written as S or S_H.

4. CONVERSION BETWEEN ACTIVATION LEVELS

1. Make a new project with filename `Lab1c`. Copy the logic diagrams in Quartus for equations (2) and (3) and place them in a single new file, i.e., this file will have circuits for both W and X.
2. Generate a complete Quartus simulation timing diagram for these two circuits and verify that the voltage tables for the two circuits match the simulation outputs.
3. Since all the inputs for the positive-logic circuit are active-high and some of the inputs for the mixed-logic solution are probably active-low, if you just copy the two files you will have eight inputs. In order to directly compare the outputs,

use the four inputs for **either** W or X and **create the other necessary inputs (with opposite activation-levels) using level-shifters** as shown in Figure 1. For example, using the signal names in Figure 1, you can**not** have a switch for G(H) **and anothe**r switch from G(L); you can have one switch for either G(H) **or** G(L), and you will get the other signal with a level shifter.

4. **Submit** (through Canvas) the archive file (including the design and simulation, with filename `Lab1c.qar`).

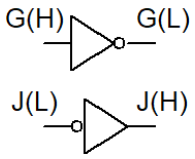

Figure 1: Use only four inputs. Create alternate activation-level inputs with level-shifters.

5. LOGIC MINIMZATION USING MIXED LOGIC

1. Make a new project and filename `Lab1d`. Design mixed-logic implementations of the equations for Y and Z in equations (4) and (5) below. The goal in this part is to choose input and output activation-levels to minimize the **total number of gates and chips** used in the overall design of these two circuits. Choose only one of the two possible activation-levels for each of the inputs, i.e., A(H), B(L), etc. [but not A(H) for one and A(L) for the other]. If you need both A(H) and A(L), that will cost one gate, a level-shifter or its equivalent, as shown in Figure 1. One last requirement is that Y and Z have **opposite** activation levels. Create the design by hand and then do it with Quartus in a single file. Do **not** simplify. (Hint: No more than two chips and seven gates are needed. Also, since you will **need to build this circuit** and you only have 74HC00 and 74HC02 chips available [and allowed], you should strongly consider using only the gates available on these chips.)

   Y = /[(A + /B) * /(C * /D)] (4)

   Z = /[(/A * B) * (/C + D)] (5)

2. Create a **truth** table for equations (4) and (5). Note: Activation-levels have **no** effect when constructing a truth table from a logic equation.

3. Create a **voltage** table for equations (4) and (5). Simulate your Quartus circuit and compare the simulation to your voltage table to verify that your circuit works as expected.

4. **Add pin number and chip labels to each gate** to change the Quartus schematic in part **1** into a wiring diagram. While looking at your pinout sheet, label the inputs and outputs of your gates using the text tool with the pin numbers of the corresponding IC chips (For example, if using an AND gate from a 74'08 chip, you could use pins 1 and 2 for inputs and pin 3 for the output.) You should also use the text tool to add the chip number, e.g., label the gate **08** for the 74'08 chip. If there are multiples of the same part needed, i.e., if you need five 2-input AND gates when the 74'08 only has four gates per chip, then label the two 74'08's differently, e.g., **08_A** and **08_B**.

5. **For this lab only**, you will also draw a layout of your circuit as it would appear on your breadboard. A layout shows each of the parts (ICs, switches, LEDs, SIP and DIP resistor packs) as they appear on the breadboard. Include the needed switches, resistors (SIP and/or DIP), and LED's. You should use one of the posted files on our website for creating your layout. Both a PowerPoint and pdf version is available.

   In future labs you will **ONLY** draw logic circuit diagrams (also known as schematic circuit diagrams) with pin numbers added as you did in part 4 above, but you will **NOT** draw layout diagrams.

6. Implement the pre-lab's wiring diagrams for both circuits in part **1** on your large breadboard. You must wire your breadboard with your own wires **at home**, i.e., **before** your lab. All inputs must come from switches and all outputs must go to LED's (as described in class notes and in the *Hardware: Getting Started* document). Vcc and GND will be provided from your DE10-Lite by connecting the rails of your breadboard to the female headers on your DE10-Lite. These are

also called Arduino headers in the *DE10-Lite Pins* document. Use the 3V3 pin as VCC and any GND pin for ground. You should use the Arduino headers on your DE10-Lite for any connection you must make to your breadboard.

7. Make **a switch legend on your logic circuit schematic** that indicates the switch position for each input signal when it is true (as shown in Figures 2). For every circuit diagram that you build in lab this semester you are **REQUIRED** to draw a legend on your logic circuit design, like

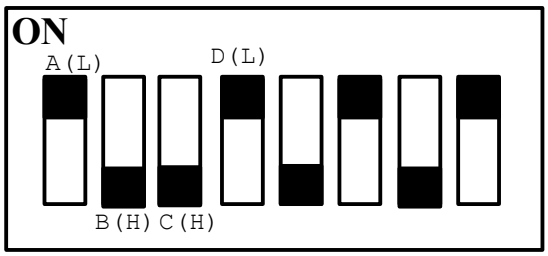

Figure 2: Switch layout (legend) with true switch positions labeled. A and D are active-low signals; B and C are active-high signals. All switches are shown in their true positions.

the one shown in Figure 2. Be sure to include the word **ON**.

8. **Make an LED legend on your logic circuit diagram** that indicates the location of each LED on an LED DIP chip (as shown in Figure 3).

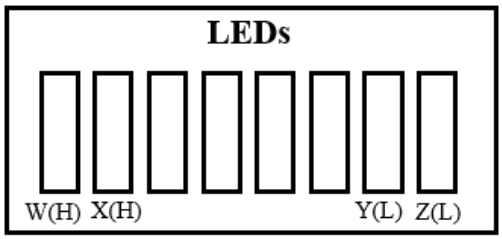

Figure 3: LED layout (legend). Note that each LED should be on (lit) when the corresponding signal is true.

PRE-LAB PROCEDURE SUMMARY

1. Learn about positive logic by implementing two simple logic circuits in § 1.
2. Learn about the benefits of mixed-logic circuit implementations in § 2.
3. Understand the relationship between active-high and active-low inputs/outputs in § 3.
4. Design and optimize a larger-scale logic circuit in § 4.

IN-LAB PROCEDURE

There will be a 15-minute to one hour quiz relating to Lab 1's pre-lab at the beginning of this lab. Quizzes will happen at the beginning of every lab period. The goal of a lab quiz is to verify that you understand what you did during the pre-lab assignment and can replicate a portion of it without help within a time constraint. In general, lab quizzes will relate to the current lab's prelab material but may also cover homework and in-class material.

You will not be told the contents of the lab quiz prior to your lab period, but generally you can guess what the quiz will focus on based on the pre-lab assignment. For this quiz only, we will tell you that the lab quiz will involve the following:

- You will get a logic equation similar to the equation in Parts 1-3. You must then:
  - Create truth and voltage tables for the equation.
  - Draw a logic diagram for the equation by hand, using appropriate mixed-logic notation.
  - Design the circuit in Quartus.
  - Verify the correct operation of the circuit by comparing a Quartus simulation of your circuit to your voltage table.
  - Build the circuit and demonstrate its correct operation.

After the quiz finishes, your PI will ask you to demonstrate your quiz solution. After everyone in your section has demoed their quiz solution, your PI will have you demonstrate a portion of the pre-lab. For this lab, your pre-lab demo will consist of demonstrating the operation of the circuits you built in Part **5** of the pre-lab assignment by switching through various inputs and comparing the output LED results to those obtained in the pre-lab simulation.

IN-LAB PROCEDURE SUMMARY

1. Complete the lab quiz.
2. Demonstrate the operation of the circuits you built in Part **5** of the pre-lab.

TIPS FOR DEBUGGING CIRCUITS

In your future breadboard designs, your circuit will probably not work the very first time you build it. Therefore, troubleshooting is a basic skill that you will have to learn. The following are some helpful suggestions. Please feel free to ask your PI any questions you may have on this subject. Keep this handout available for use in future labs.

- Come prepared with a large, neat Pre-Lab schematic diagram. **This is REQUIRED to get help from Dr. Schwartz or a PI**. (I suggest that you print this and bring it to your lab, with room for notes and corrections, but this is **not** required.)

- When creating a large circuit, you should identify, build, and debug small portions of the circuit, one at a time, to isolate potential problems. In other words, decompose your design and your construction. (Be the Tortoise and not the Hare!) Several links to versions of the story are available below.
  - http://read.gov/aesop/025.html
  - Find one video at https://youtu.be/2DrKmpuKhKE.

- Start troubleshooting at the point at which the voltage value is wrong; continue to check your circuit backward, making sure that at each point the value agrees with your prediction.

- Construct your circuit neatly. Use appropriate length wires for connections, i.e., use short wires for short distance connections. Your circuit should not look like a bowl of spaghetti or a bird's nest.

- Use your multimeter or an LED circuit with a long wire as a logic probe. Ask your PI to demonstrate this if you do not understand.

- Check IC insertion to make sure all the pins are in the correct holes and that the IC does not have any missing or bent legs.

- Check power and ground connections (and other connections) **before** applying power to the circuit.

- Make sure that the power and the ground are properly connected to all IC's before applying power to the circuits.

- **DO NOT** short (connect) the power supply outputs (Vcc and GND) together, i.e., do not allow the exposed wires to touch each other. This will cause permanent damage to the power supply. If the green light on your PLD board does not turn on when you connect your PLD board to power, there is most likely a short.

- **DO NOT** connect the power supply to the breadboard with reverse polarity. This could cause permanent chip damage.

- **DO NOT** connect an output of any gate to the output of another gate, to a switch circuit output, to Vcc, or to GND. These situations will cause excessive currents and result in permanent damage to the chip or chips involved.