LAB 2: MSI Circuits

OBJECTIVES

The objective of this lab is to familiarize students with the structure, design, and operation of several MSI (medium scale integrated) devices, i.e., a multiplexer, decoder, and encoder. Students will design and build the first two of these devices using SSI (small-scale integrated) components and the last with SSI and MSI devices.

MATERIALS

- Prototyping board (bread board)
- Wires, Switches, LED's, Resistor packs
- IC's: Any SSI (no MUX or Decoder) chips available in your base lab kits (but **not** your quiz kit)
- *7-segment LED data sheet*
- *7-segment PCB schematic*
- *7-segment Display Board Info*
- You must adhere to the *Lab Rules and Policies* document for **every** lab. Re-read, if necessary. Documents must be submitted through Canvas and on paper for every lab. All pre-lab files must submitted **BEFORE** the beginning of your lab.

PRE-LAB REQUIREMENTS

- You must have your CPLD PCB tested before walking into the door for your Lab 2. The TAs have a way to test your PCB in our lab. In fact you should have it tested prior to walking into the door for your Lab 1!
- Only program pins that are available externally on the CPLD, as shown in the Appendix.
- A single pdf document (for this lab, called **`Lab2.pdf`**) of all design files (bdf or vhdl) and simulation files is required for this and all pre-labs. I suggest that you capture screen shots of each design and simulation (as they are generated) into a MS-Word (or equivalent) file and then save it as a pdf file. This file should also include any other required items including truth/voltage tables, or anything else specifically requested in the lab document or required in the *Lab Rules and Policies*.
- As usual, submit through Canvas your *Pre-Lab Report* pdf file (**`Lab2.pdf`**) and the three Quartus archive files that you generate in the pre-lab sections.

Part A. Multiplexer Design

The **multiplexer (MUX)** is a device that acts as a multi-position switch. See Figure 1. A number of DATA inputs are applied to this device (D0-D3) and one of the inputs is switched to the output (Y) of the device. A binary number applied to the SELECT (S1 & S0) lines controls which input is passed to the output. For example, when S1=S0=0, D0 is "logically" connected to the output Y. When S1=0 & S0=1, D1 "logically" connected to Y and so forth for D2 & D3 to connect to Y. Notice that some of the inputs are active-low and some are active-high; the output is active-low.

Part A. Mux Pre-Lab Requirements

1. Draw a **truth table** for the 4-input multiplexer (6 inputs and 1 output) using "wild cards" as appropriate. Wild cards, usually designated with an "*" (or a "–"), indicate that any value can replace it. If one * exists in a row of a truth table, the one row represents two rows; one with the * replaced by a 0 and the other with the * replaced by a 1. If there are two asterisks in a single row, that row is an abbreviation for four rows.

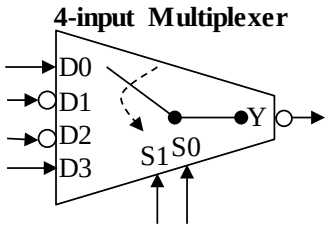

Figure 1: 4-input MUX.

2. Derive the **logic equation** for this MUX from the truth table (both an SOP and a POS).
3. Draw a **voltage table** for this MUX. The figure shows that some of the D inputs and both of the S inputs are **active-high**; some of the D inputs and the output are **active-low**. Draw a functional block diagram of the MUX; draw signal definitions (i.e., activation levels and signal names **outside** the MUX).
4. Design a circuit for this MUX using logic gates available in the ICs listed above in the "Materials" section. Use project name **`lab2_MUX`**. (It is usually easier to design first on paper.) There is no requirement, in this lab, to minimize the number of gates or chips; but of course a simpler design will be easier to build.)
5. If you designed the circuit in part 4 on paper, now design it using Quartus and then simulate the complete mixed-logic circuit in Quartus. ~~Normally~~ As usual, when using discrete logic, i.e., 74'xx ICs, ~~you would~~ add pin numbers and chip labels to the logic circuit diagram to make this a wiring diagram. Do so for this the design of item 7. When you are doing designs entirely inside the CPLD (as in item 8), it is **NOT** necessary to use 74'xx pin and chip labels, but you should use CPLD pin labelswhen you are using the CPLD only. ~~but since you will use the CPLD for this design, this is not necessary~~. Annotate the simulation in your lab report.
6. In addition to the requirements specified above, items to be included in your Pre-Lab Report always include logic equation truth table, voltage table, circuit schematics (with chip and pin numbers), and simulation results. As usual, all pre-lab files must be submitted prior to the start of your lab.
7. Build your MUX circuit on your bread boards with parts in your lab kit (but do not use the parts reserved for your lab quiz specified here). You will need switches for the inputs and an LED for the output. Test this design with discrete elements. Use the truth table

to demonstrate (to yourself) that your MUX circuit operates correctly by connecting appropriate switch circuits to all six inputs and an **active-low** LED circuit to the output. Use your truth table

8. In item 7 above, you built a MUX with 74'xx gates; in this part you will build the MUX with your CPLD PCB. Program your CPLD with your MUX design. Test your CPLD solution using the **same inputs** as your solution from the previous section (i.e., use the same switches for both the part 7 and part 8 designs). You must use different outputs (two LEDs, one for part 7 and one for part 8). Use your truth table to demonstrate (to yourself) that your MUX circuit (using the CPLD PCB) operates correctly (and the same as the circuit from item 7).
9. Submit your archive file through Canvas. It should have filename **`lab2_MUX.qar`**.
10. **Always** make a legend (as described in lab 1) for the input switches. **Always** include the legend in your pre-lab report and have it available to your TA during your demonstrations. Note that LED's should always be lit when the related signal is true.

Part B. Decoder Design

In this part of the lab you will design a **Hex to 7-segment Decoder**. This decoder will take 4-inputs (a hexadecimal number) to drive 7 active-high outputs that will in turn drive a 7-segment LED. (Two 7-segment LED devices were provided in your lab kit.) Figure 2 shows a block diagram of the part that you are required to design.

Figure 3 shows the arrangement of each of the LEDs on the 7-segment LED device. Each of the LEDs are active-high. The T input is used to test the outputs; when T is true, all of the outputs A through G are true.

Part B. Decoder Pre-Lab Requirements

1. Draw a truth table for the above decoder.
2. Derive the logic equations for each of the outputs from the truth table (either a SOP or a POS).
3. Draw voltage table for the above decoder.
4. Draw a functional block diagram of the Decoder; draw signal definitions (i.e., activation levels and signal names **outside** the Decoder).
5. Design a circuit to implement the decoder using SSI parts only (i.e., AND, NAND, BNAND, NOT, …). Use project name **`lab2_decoder`**. (It is usually easier to design first on paper.).
6. If you designed the circuit in part 5 on paper, now design it using Quartus and then simulate the complete mixed-logic circuits in Quartus. There is no requirement, in this lab, to minimize the number of gates or chips. Annotate the simulation in your lab report.
7. Each of the above items, including the truth tables, logic equations, voltage tables, circuit schematics, simulation results, and your Quartus simulation files should be part of the Pre-Lab Report. As usual, all pre-lab files must be submitted prior to the start of your lab.
8. Submit your archive file through Canvas. It should have filename **`lab2_decoder.qar`**.
9. Add your decoder design to the MUX design from Part A so that your CPLD contains both designs. Add the necessary switch circuits to your breadboard to drive (generate the required inputs) for your Hex to 7-segment Decoder. Use your DAD/NAD to verify the decoder outputs.
10. Add the 7-segment LED PCB to your breadboard.
11. See Figure 4 (also in the *7-segment Display Board Info* document) for 7-segment display pin placements. You will use the left 7-segment LED display (not the right one) in this lab by connecting EN0 to GND; also connect EN1, LE0, and LE1 to Vcc. (You will learn what these signals mean in a few weeks.)
12. Use Figure 4 to correctly place the decoder outputs A through G.

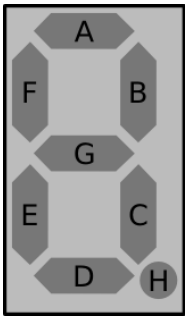

Figure 3: LED arrangement on 7-segment display.

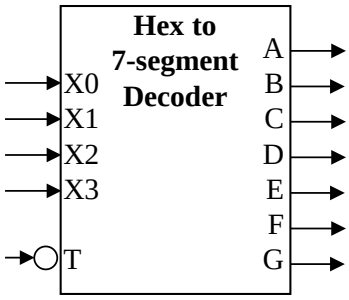

Figure 2: Hex to 7-segment decoder.

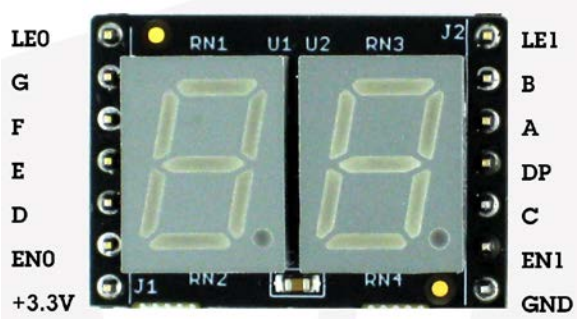

Figure 4: Pin placements for the two 7-segment LED displays on 7-segment PCB.

13. Demonstrate (to yourself) that your MUX circuit and Hex to 7-segment Decoder circuit operates correctly
14. **Always** make a legend (as described in lab 1) for the input switches. **Always** include the legend in your pre-lab report and have it available to your TA during your demonstrations. Note that LED's should always be lit when the related signal is true.

Part C. Encoder Design

In this part of the lab, you will design a **priority encoder**. The encoder has four input lines (I1-I4) and three output lines (C0:C2) as shown in Figure 5.

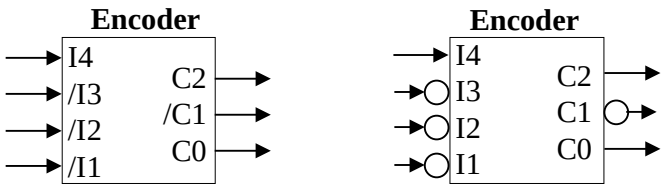

Figure 5: Priority Encoder

Note: The notation in the left block diagram of Figure 5 (with no bubbles for active-low inputs or outputs) is common. Sometimes the /I3 will appear as I3 with a bar over it (i.e., $\overline{I3}$). I suggest you use drawing like the one shown on the right for your designs in this course and in the future.

When one of the four inputs is asserted (i.e., is true), the encoder outputs a binary code corresponding to the active input. For example, if I3 is the only input asserted, the output code will be "011", or a binary "3". If no inputs are asserted (i.e., true), the encoder will output "000". If more than one input is true at the same time, the encoder will output a code corresponding to the ***highest*** numbered true input. For example, if both I2 and I4 are true, the encoder will output a "100". In this case, the code for I4 is output because it has the highest *priority*.

You will design this encoder using two 4-input MUX components (available in a 74'153) and some SSI gates.

Part C. Encoder Pre-Lab Requirements

1. When you use Quartus to design with a 74'153 MUX (available under `others|maxplus2|74153`), you will find that the select lines are labeled A and B. Which select inputs correspond to A and B for you MUX? Verify the functioning of a 74'153 MUX (and if A B =S_1 S_0 or A B = S_0 S_1) by creating a new Quartus circuit (in a new file, **`Lab2_MUX_TEST`**) using only this chip, inputs, and outputs. Simulate this design using enough different input combinations to prove that the 74'153 works as you suspect it should. Note: For every "new" part you use (like the 74'153), if you are not sure of a pins function, you should **verify its operation before** using it in a circuit. Include this design and simulation in your pre-lab document. Submit the archive file **`Lab2_MUX_TEST.qar`** through Canvas.
2. Create a truth table for the above encoder.
3. Derive **both** MSOP **and** MPOS logic expressions for each of the encoder outputs. (You may use K-maps, if desired.)
4. Design a circuit (**`lab2_ENC`**) to create the encoder with a single 74'153 chip (in Quartus) and (only if necessary) the gates available on 74'00, 74'02, and/or 74'04 ICs. Your design must (non-trivially) use a 74'153. Note that the 74'153 includes two 4-input MUX's. Simulate your designs using Quartus to verify that your design works. Submit your archive file (**`lab2_ENC.qar`**) through Canvas.
5. Add your encoder design to the MUX and Decoder designs from Parts A and B so that your CPLD contains all three designs in new project called **`lab2_ALL`**. Add the necessary switch circuits to your breadboard to drive (generate the required inputs) for your encoder. Use LEDs to verify the encoder outputs. Note that LED's should always be lit when the related signal is true. Simulate your designs using Quartus to verify that your design works. Submit your archive file (**`lab2_ALL.qar`**) through Canvas.
6. Demonstrate (to yourself) that your MUX circuit, Hex to 7-segment Decoder circuit, and Encoder circuit operates correctly.
7. **Always** make a legend (as described in lab 1) for the input switches. **Always** include the legend in your pre-lab report and have it available to your TA during your demonstrations.

LAB PROCEDURE

Demonstrate the functioning MUX circuit, your Hex to 7-segment Decoder circuit, and your Encoder circuit to your TA.

APPENDIX

The CPLD pins that you can use are shown in Figure 6.

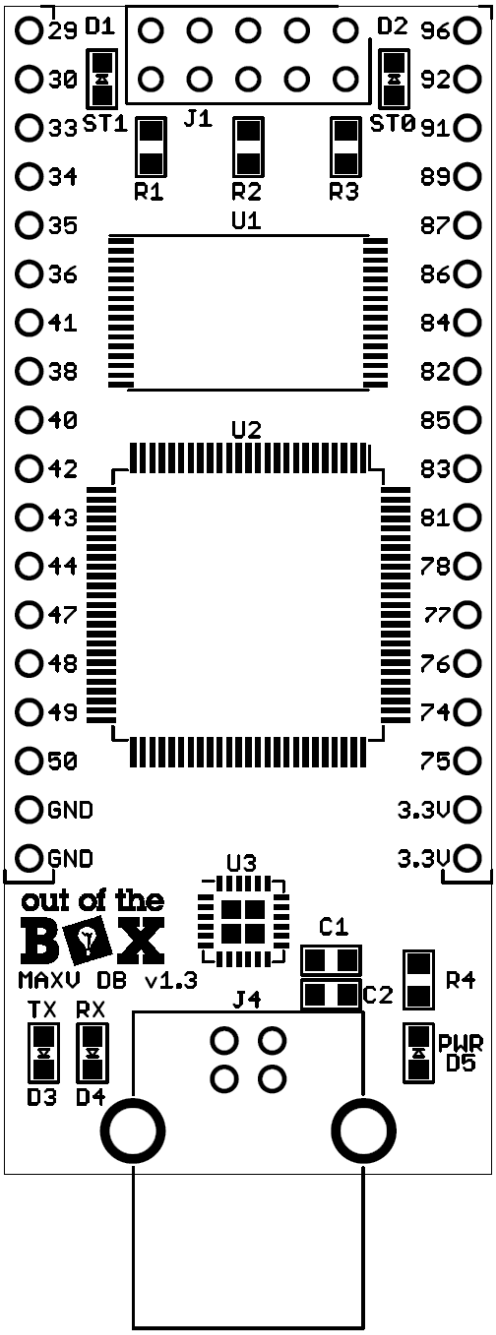

Figure 6: CPLD PCB top layer with available CPLD pins shown.