

LAB 2: MSI Circuits

OBJECTIVES

The objective of this lab is to familiarize students with the structure, design, and operation of two MSI (medium scale integrated) devices, i.e., a multiplexer and a decoder. Students will design and build these two devices using SSI (small-scale integrated) components and then design a circuit to implement an equation 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 DISPLAYS ON MAX 10 PCB](#)
- 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 be submitted **BEFORE** the beginning of your lab.

PRE-LAB REQUIREMENTS

- You must have your **MAX 10** PCB tested before walking into the door for your Lab 2. The PIs have a way to test your PCB in our lab. In fact, if possible, you should have it tested prior to walking into the door for your Lab 1!
- Only program pins that are available externally on the **MAX 10**, as shown in the Appendix. When using the 7-segment displays, you can also use **MAX 10** pins specified in the [7-SEGMENT DISPLAYS ON MAX 10 PCB](#) document.
- A single pdf document (for this lab, called **Lab2.pdf**) of all design files (bdf or vhd) 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.

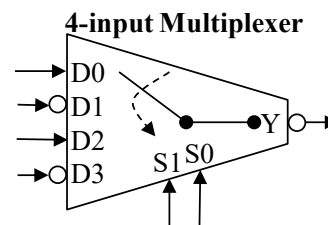


Figure 1: 4-input MUX.

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.
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 (with filename **lab2_MUX.bdf**) and then simulate the complete mixed-logic circuit in Quartus. As usual, when using discrete logic chips, i.e., 74'xx ICs, add pin numbers and chip labels to the logic circuit diagram to make this a wiring diagram. Use this circuit design to build the circuit in item 7, below. When you creating designs exclusively with the **MAX 10** (i.e., without other ICs, as in item 8, below), it is **NOT** necessary to use 74'xx pin and chip labels. But when the design is done for the **MAX 10**, put the design **AFTER** adding the **MAX 10** pin labels into your Pre-Lab Report. 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(s), voltage table(s), circuit schematic(s) (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 switch circuits for the inputs and an LED circuit 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 gates on 74'xx ICs; in this part you will build the MUX with your **MAX 10** PCB. Program your **MAX 10** with your MUX design. Test your **MAX 10** 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 LED circuits, one for item 7 and one for item 8). Use your truth table to demonstrate (to yourself) that your MUX circuit (using the **MAX 10** 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 PI during your demonstrations. Note that LED's should always be lit when the related signal is true (whether that means that the signal is high or that it is low).

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.

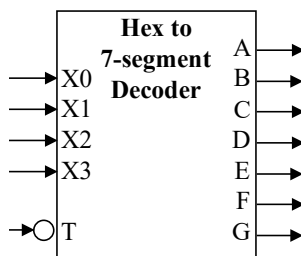


Figure 2: Hex to 7-segment decoder.

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).

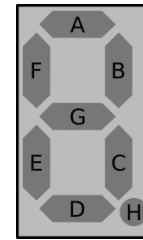


Figure 3: LED arrangement on 7-segment display.

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.) A partially completed and well-formatted design, [hex to 7seg.bdf](#), is available on our website. You may copy the designs in this file and augment them with your own designs or design the entire circuit yourself. Note that you are **required** to design a **POS** circuit for the decoder's A and B outputs.
6. If you designed the circuit in part 5 on paper, now design it using Quartus (with filename **lab2_decoder.bdf**) 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. In addition to the requirements specified above, items to be included in your Pre-Lab Report always include logic equation truth table(s), voltage table(s), circuit schematic(s), and simulation. 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 **MAX 10** contains both designs. Add the necessary switch circuits to your breadboard to drive (generate the required inputs) for your Hex to 7-segment Decoder.
10. Add the 7-segment LED PCB to your breadboard.
11. See the [7-SEGMENT DISPLAYS ON MAX 10 PCB](#) document for information to correctly link the decoder outputs A through G to the appropriate **MAX 10** pins. You may use either of the two 7-segment displays.
12. Demonstrate (to yourself) that your MUX circuit and Hex to 7-segment Decoder circuit both operate correctly.
13. **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 PI during your demonstrations. Note that LED's should always be lit when the related signal is true.

Part C. MUX Implementation

In this part of the lab, you will design two circuits, each to implement the equation,

$$F_0 = \overline{W}X(Y + Z) + X\overline{Y}\overline{Z}.$$

LAB 2: MSI Circuits

For each of the circuits, you will use a 4-input MUX, along with a few SSI gates. When you build the circuits, first in Quartus and then on your bread board, you will use a 74'153 dual MUX device inside Quartus, i.e., not with a 74'153 chip on your breadboard. First you will design a circuit to implement this equation without using the MUX enable and then you will design a second circuit to implement the equation using the MUX enable.

Part C. MUX Implementation Pre Lab Requirements

1. When you use Quartus to design with a 74'153 MUX (available under `others\maxplus2\74153`, and shown in Figure 4), 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 $A \cdot B = S_1 \cdot S_0$ or $A \cdot B = S_0 \cdot S_1$) by creating a new Quartus circuit (in a new project and 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.

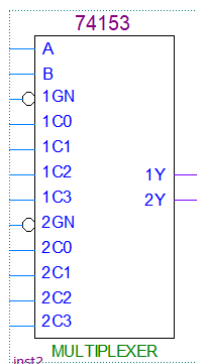


Figure 4: 74'153 in Quartus.

- Create a truth table for the above equation.
- Derive **both** MSOP and MPOS logic expressions for the equation. (You may use K-maps, if desired.)
- Design a circuit (**lab2_EQU.bdf** in project **lab2_EQU**) to implement this equation 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. But do not use the MUX enable in this design. Your design must (non trivially) use a 74'153. Use the MUX select lines as follows: $S_1 = W(H)$ and $S_0 = Y(H)$. Note that the 74'153 (see Figure 4) includes two 4 input MUX's.
- Make a voltage table from the truth table and the activation levels that you chose above.
- Simulate your design using Quartus to verify that your design works (by comparing it to the voltage table).
- Design another circuit in the same **lab2_EQU.bdf** that implements the same equation (below), but calls the output F_I (instead of F_0) and this time uses the MUX enable.

$$F_I = \overline{W} X (Y + Z) + X \overline{Y} \overline{Z}$$

- Add the new F_I output to your previous Quartus simulation to verify that both of your design works. Submit your archive file (**lab2_EQU.qar**) through Canvas.
- Add your second design to the MUX and Decoder designs from Parts A and B so that your MAX 10 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 this equation. Use LEDs to verify the F_I 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.
- Demonstrate (to yourself) that your MUX circuit, Hex to 7 segment Decoder circuit, and F_I circuit operate correctly.
- **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 PI during your demonstrations.

LAB PROCEDURE

Demonstrate the functioning MUX circuit, your Hex to 7-segment Decoder circuit, and your F_I circuit to your PI.

APPENDIX

The **MAX 10** pins that you can use are shown in Figures 5 and 6. Do **NOT** use the clock pins for general purpose inputs or outputs, e.g., C2P (i.e., CLK2P, MAX 10 pin G9) and C3P (i.e., CLK3P, MAX10 pin F13); use these pins only for clock inputs. These figures are also available in the Quartus Tutorial on our website. Information about the 7-segment displays are also available on our website in the document [7-Segment Displays on MAX 10 PCB](#).

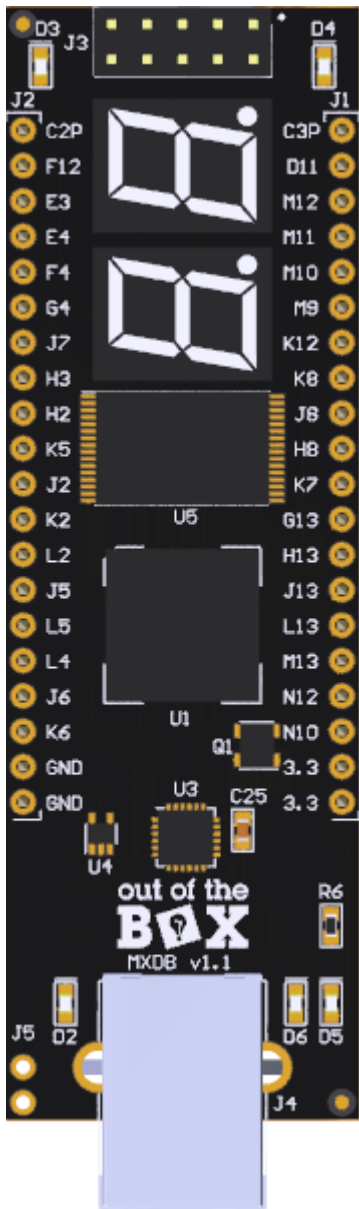


Figure 5: PLD PCB top layer with available FPGA pins

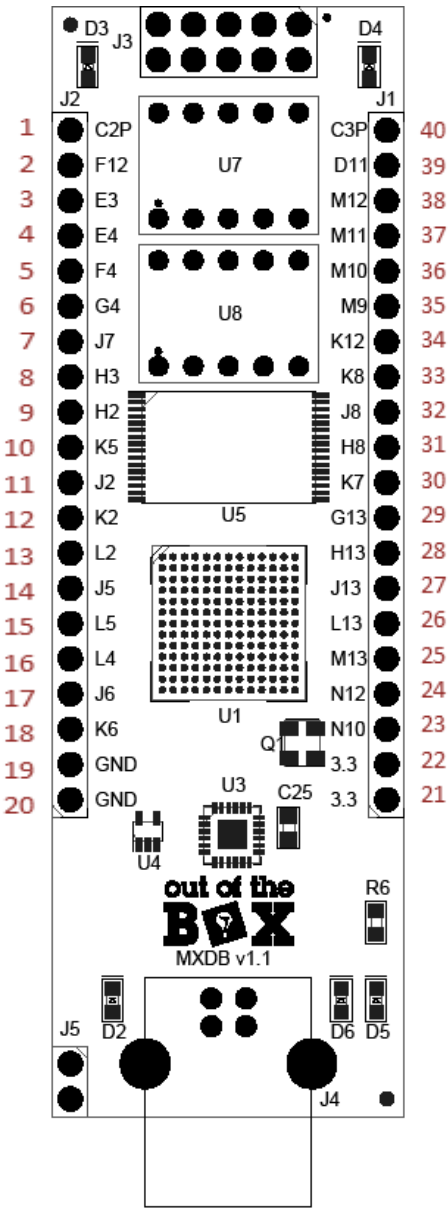


Figure 6: Available FPGA pins

Pin as Chip #	Available FPGA Pins
1	C2P (CLK2P) G9
2	F12
3	E3
4	E4
5	F4
6	G4
7	J7
8	H3
9	H2
10	K5
11	J2
12	K2
13	L2
14	J5
15	L5
16	L4
17	J6
18	K6
19	GND
20	GND
21	3.3 V
22	3.3 V
23	N10
24	N12
25	M13
26	L13
27	J13
28	H13
29	G13
30	K7
31	H8
32	J8
33	K8
34	K12
35	M9
36	M10
37	M11
38	M12
39	D11
40	C3P (CLK3P) F13

Available FPGA Pins	Pin as Chip #
C2P (CLK2P) G9	1
C3P (CLK3P) F13	40
D11	39
E3	3
E4	4
F4	5
F12	2
G4	6
G13	29
H2	9
H3	8
H8	31
H13	28
J2	11
J5	14
J6	17
J7	7
J8	32
J13	27
K2	12
K5	10
K6	18
K7	30
K8	33
K12	34
L2	13
L4	16
L5	15
L13	26
M9	35
M10	36
M11	37
M12	38
M13	25
N10	23
N12	24
GND	19
GND	20
3.3 V	22
3.3 V	21