UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering

EEC 18 **DIGITAL SYSTEMS I** **Fall 2024**

Lab 3: Combinational Network Design Using Muxes

Objective

This lab is an exercise in designing combinational circuits using multiplexers. It includes creating symbols to make your schematic hierarchical and easier to understand, simulating your design in ModelSim, and verifying your circuit on the DE10-Lite board.

Preparation (Pre-lab)

Before coming to the first lab session, complete the following tasks:

- Read handouts and textbook material covering adders (Unit 4.7) and multiplexers (Unit 9.2)
- Generate a truth table showing inputs vs outputs for the following circuit blocks in Part I: Comparator, 7-segment decoder, Circuit A, and Circuit B.
- Use the truth tables to produce minimized SoP (sum of products) for the Comparator, 7-segment decoder, Circuit A and Circuit B.

I. Binary to BCD Conversion and Display Control

Table 1. Binary-to-decimal Conversion Values

Binary value	BCD digits
0000	00
0001	01
0010	02
0011	03
…	…
1001	09
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Design Specifications

Design a circuit that converts a four-bit binary number $v = \{v3, v2, v1, v0\}$ into its two-digit Binary Coded Decimal (BCD) equivalent and displays it on two seven-segment LED displays. Table 1 shows the required output values. A high-level design of this circuit is shown in Figure 1; it includes the following major blocks, all of which you must design and implement from basic logic gates (including the multiplexers):

Comparator: Checks the value of v. If v is less than 10, the output z is 0, otherwise if the input is 10 or greater, the output z is 1.

Multiplexer: This block is a standard 2-to-1 multiplexer with 2 data inputs, 1 input selector, and 1 output.

Circuit A: Determines the BCD value of the "ones" digit to be displayed on d_0 when the value of $v \geq 10$. For example, if $v = 12$ (i.e., 1100), the output of Circuit A should be 2 (i.e., 0010). When the value of $v < 10$, the output of Circuit A is not used.

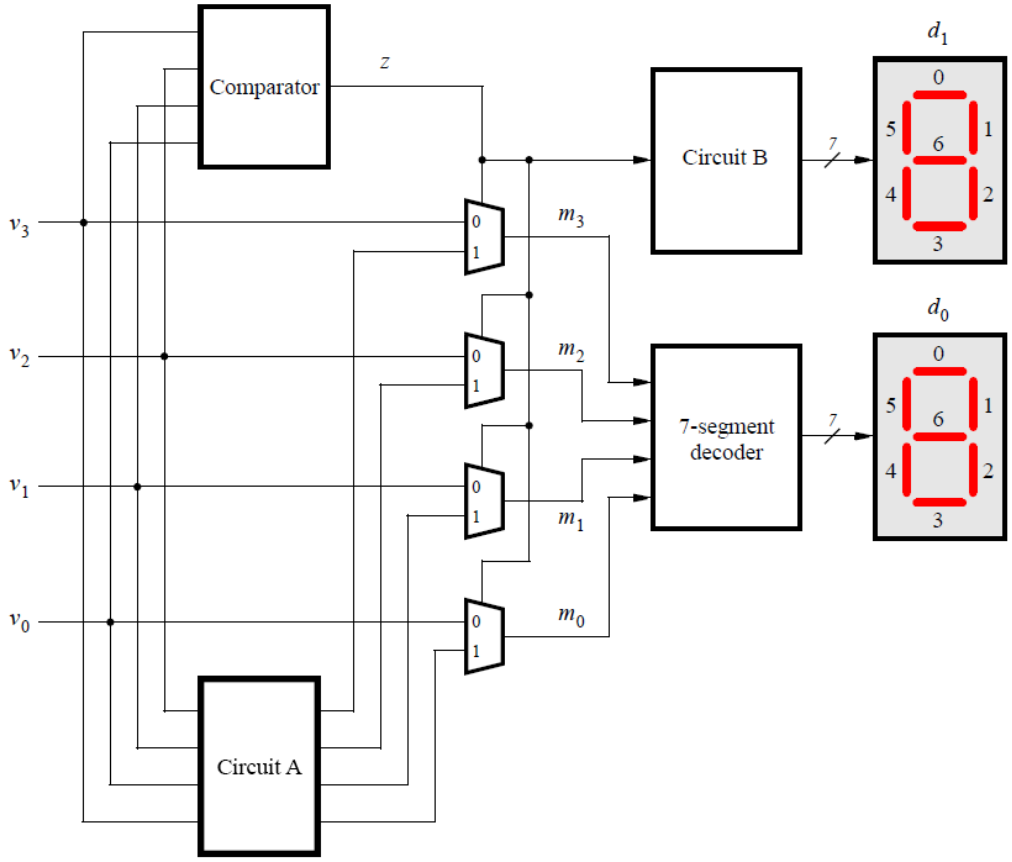


Figure 1. The overall design of the binary-to-decimal conversion circuit where d_1 is the "tens" digit and d_0 is the "ones" digit, which display the input $v3$, $v2$, $v1$, $v0$.

Circuit B: A simplified 7-segment decoder that decodes only 0 and 1. It has one input and seven outputs (each output is one segment of HEX display d_1).

7-Segment Decoder: Uses the 4-bit input binary to turn on the appropriate segments on d_0. For example, when the input is equal to 7 (i.e., 0111), segments 0, 1, and 2 on d_0 should turn on. The numbers should be displayed exactly as shown in Figure 2—note some patterns are non-standard.

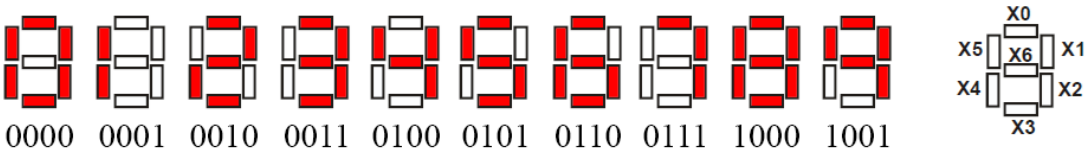


Figure 2. Output from 7-segment Decoder based on input binary.

The number displayed in d_0 is represented by $\{m3, m2, m1, m0\}$. When the input number is less than 10 (i.e., $z = 0$), the four multiplexers pass the input directly to the 7-segment decoder to be displayed on d_0. When the input number is greater than or equal to 10 (i.e., $z = 1$), the four multiplexers select the number from Circuit A. For example, when the input number $v = 10$ (i.e., 1010), the Circuit A output should be 0000; when the input number $v = 15$ (i.e., 1111), the Circuit A output should be 0101, since the number "5" should be displayed on d_0. The equations of Circuit A can be derived using a truth table and K-maps.

Reminder: the 7-segment displays are active-low, meaning segments turns on when their input is 0.

Complete the design of this circuit and create its schematic.

Design Procedure

1. Make a new Quartus project, such as **lab3**, for your design.

2. Draw a schematic for the complete circuit given in Figure 1, observing the following points:

   - Use four **SW** switches as inputs for *v*3, *v*2, *v*1, *v*0.

   - Use two **HEX** displays for d_0 and d_1.

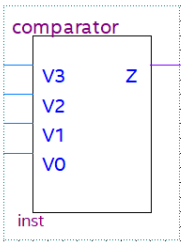
   - Create a symbol for the 2-to-1 multiplexer, Comparator, Circuit A, and the 7-Segment Decoder by:

     i. Create a new Block Diagram/Schematic File (.bdf) and enter the circuitry for one of the blocks, such as the Comparator. The inputs and outputs of the circuit must use INPUT and OUTPUT components, but these should NOT be names of FPGA pins such as SW[3]…SW[0].

     ii. When your circuit is complete, save it with the name of the block such as comparator.bdf.

     iii. From the Quartus pull-down menus, select **File > Create/Update > Create Symbol Files for Current File**. This will generate a symbol file, comparator.bsf, for your schematic.

     iv. In the top-level design, you can use the Symbol Tool (or double-click on open area) and specify the Comparator component that you just created. It should be in your project directory. (Try not to use a name of an existing Altera component for your symbol/circuit.) Your component should look similar to the one shown below.



   - Since the circuitry for Circuit B will be very simple, you can keep it on the top-level schematic rather than creating a symbol for it, if you prefer.

   - All circuit I/O pins, such as HEX1[6]..HEX1[0], SW[3]..SW[0], HEX0[6]..HEX0[0] should be in the top-level schematic.

   - Import the pin assignments for the DE10-Lite board. You can use the lab1.qsf file that you created using System Builder for lab 1, or you can generate a new file following the procedure described in Lab 1.

3. Compile your circuit in Quartus and fix any errors.

4. Run a ModelSim simulation of your design to verify functionality by selecting **Tools > Run Simulation Tool > Gate Level Simulation** in Quartus and then select **Simulate > Start Simulation** in ModelSim. Select the **altera_ver** and **fiftyfivenm_ver** libraries, as in previous labs.

   - Right-click on SW in the Objects window and select **Add Wave**. Do the same for both HEX displays.

- Set the Radix for the HEX displays to hexadecimal.

- Right-click on SW in the Wave window and select **Force…** and force the appropriate SW signals to 0000. Simulate for 20 ns and verify the output.

- Continue to simulate the circuit through the full sequence of SW inputs, 0000 – 1111. Print a simulation waveform showing just one complete cycle of the inputs.

- Demonstrate your simulation to your TA and have him or her sign your verification sheet.

5. Download your circuit to your DE10-Lite board and verify your circuit using the SW switches you chose. Verify the HEX display outputs.
   Demonstrate your circuit to your TA and have him or her sign your verification sheet.

II. Multiple-of-4 Detector

Design Specifications

Design a circuit that detects whether a four-bit binary number $v = v3, v2, v1, v0$ is a multiple of 4 (i.e. 4, 8, 12), not including 0. If v is a multiple of 4, turn on a LEDR of your choice. Otherwise, keep the LED off. Add this module to the design from Part I.

Design Procedure

1. Add on to your previous design schematic.

2. Create a new schematic (.bdf) for the circuit required to achieve the results in Figure 3. The inputs and outputs of the circuit must use INPUT and OUTPUT components, but these should not be names of FPGA pins.

   - From the Quartus pull-down menus, select **File > Create/Update > Create Symbol Files for Current File**. This will generate a symbol file for your top-level schematic.

3. In the top-level design, you can use the Symbol Tool (or double-click on open area) and specify the Comparator component that you just created. It should be in your project directory.

   - The inputs of this sub-circuit should be the switches, SW[3:0], and the output should be a LEDR.

4. Compile your circuit in Quartus and fix any errors.

5. Do a ModelSim simulation of your design by selecting **Tools > Run Simulation Tool > Gate Level Simulation** in Quartus and then select **Simulate > Start Simulation** in ModelSim. Select the **altera_ver** and **fiftyfivenm_ver** libraries, as in previous labs.

   - Right-click on SW in the Objects window and select **Add Wave**. Do the same for LEDR.

   - Right-click on SW in the Wave window and select **Force…** and force the SW signal to 0000. Simulate for 20 ns and verify the output.

   - Continue to simulate the circuit through the full sequence of SW inputs, 0000 – 1111. Print a simulation waveform showing just one complete cycle of the inputs.

   - Demonstrate your simulation to your TA and have him or her sign your verification sheet.

6. Download your complete circuit to your DE10-Lite board and verify your circuit using SW3 – SW0. Verify the output on the LEDR. Demonstrate your circuit to your TA and have him or her sign your verification sheet.

III. Lab Report

Each individual will be required to submit a lab report. Use the format specified in the "Lab Report Requirements" document available on the class web page. Be sure to include the following items in your lab report:

- Lab cover sheet with TA verification for circuit simulation and circuit performance
- Graded pre-lab
- Quartus schematics for Parts I and II
- Simulation waveforms for Parts I and II
- Explain any significant optimizations you found to simplify Circuit A.

Acknowledgement: This lab is based on a Laboratory Exercise developed by Altera Corp. Used by permission.