

# EE334 PRACTICAL EXERCISE: Digital Logic III

Name: \_\_\_\_\_

Section: \_\_\_\_\_

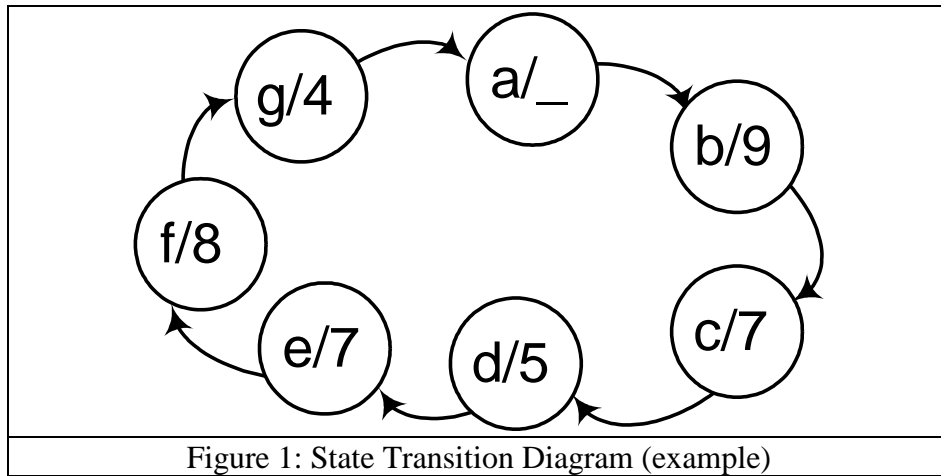
Date: \_\_\_\_\_

**I. Goal:**

- a. Design a multi-stage sequential logic circuit that includes a state machine and logic.
- b. Implement and simulate the results of digital logic circuits in the Quartus II software.
- c. Test and transfer logic circuits onto the Altera DE-1 board.
- d. Utilize pre-constructed device to drive the seven-segment display.

**II. Equipment:** Laptop with Altera DE-1 Board and Quartus II software.

**III. Discussion:** During the next three lab periods you will design a “state machine” that drives a seven-segment LED display to show your alpha code, one digit at a time with a blank at the beginning. For example, given an alpha code 975784, the display pattern will be “\_”, “9”, “7”, “5”, “7”, “8”, “4”, repeat. See Figure 1 below.



Your logic capstone project will consist of the following devices (See Figure 2):

1. a one-second clock
2. a 3-bit mod-7 counter that sequences from 000 to 110
3. a minimized logic circuit that produces BCD numbers for the alpha code as the counter increments
4. A “7447 Decoder” block that converts the BCD number to its corresponding seven-segment display input

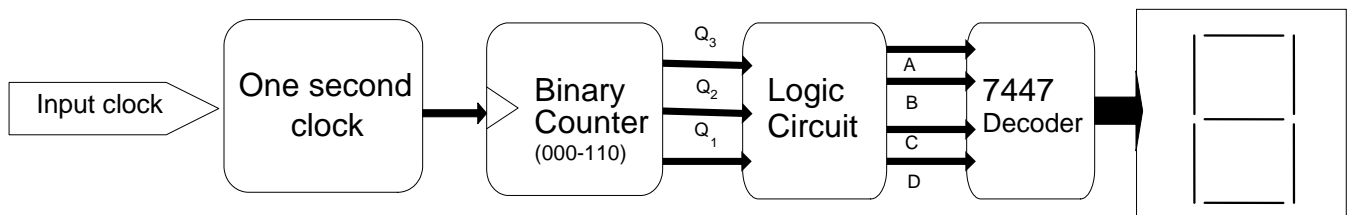


Figure 2: Block Diagram

## EE334 PRACTICAL EXERCISE: Digital Logic III

### IV. Minimum Logic for BCD State Generation

- a. Develop the logic circuit truth table such that the output BCD number has 4-bits, where D is MSB and A is LSB. The leading “blank” display is driven using a “1111” output. A BCD number is simply the decimal number you want to display translated into binary.

State	Counter Value			Decimal # (alpha)	BCD Digits			
	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>		D	C	B	A
a	0	0	0	{blank}	1	1	1	1
b	0	0	1					
c	0	1	0					
d	0	1	1					
e	1	0	0					
f	1	0	1					
g	1	1	0					
h	1	1	1		X	X	X	X

“Don’t  
Care”  
state

Table 1

- b. Determine the K-maps for each of BCD digit, and write them below.

**D =**

**C =**

**B =**

**A =**

- c. Use the above K-maps to simplify each expression into minimum logic expressions for D, C, B & A. Have your instructor review your minimum logic before proceeding to *step d*.

**D =**

**C =**

**B =**

**A =**

Instructor Initials \_\_\_\_\_

- d. Start a new project in Quartus II. A suggested name and directory for the project are “Alpha” and c:\ee334\\Alpha
- e. Implement the four minimized logic expressions in the Quartus II software environment using **inputs Q<sub>2</sub>, Q<sub>1</sub> and Q<sub>0</sub>** and **outputs D, C, B and A**. (\*\*Sort your inputs top to bottom, Q<sub>2</sub>, Q<sub>1</sub>, Q<sub>0</sub> when running the Simulator Tool – Highlight all inputs then use the sort button on the tool bar to sort “Descending”\*\*)
- f. Compile your design and verify the BCD output sequence using the waveform simulator.

## EE334 PRACTICAL EXERCISE: Digital Logic III

### V. Designing the 3-Bit Mod-7 Counter

- a. Draw the State Diagram for a 3 bit mod 7 up counter (counts from '000' to '110'. Note that the counter may accidentally start up in state '111'. Therefore you should have state '111' feed into another state such as '000')
  
- b. Assuming the use of T flip-flops, complete the State Transition Table for the mod-7 counter below:

Present State	Next State	Required Flip-Flop Inputs
$Q_2 \ Q_1 \ Q_0$	$Q_2^+ \ Q_1^+ \ Q_0^+$	$T_2 \ T_1 \ T_0$
0 0 0		
0 0 1		
0 1 0		
0 1 1		
1 0 0		
1 0 1		
1 1 0		
1 1 1		

- c. Using K-Maps, determine the simplest implementation for the 3 flip-flop inputs ( $T_2, T_1, T_0$ ) in terms of the flip-flop outputs ( $Q_2, Q_1, Q_0$ )

$T_2 =$  \_\_\_\_\_,  $T_1 =$  \_\_\_\_\_,  $T_0 =$  \_\_\_\_\_

## EE334 PRACTICAL EXERCISE: Digital Logic III

- d. Draw your mod-7 up counter below using the input “clk” as the timing signal. Use T flip-flops.

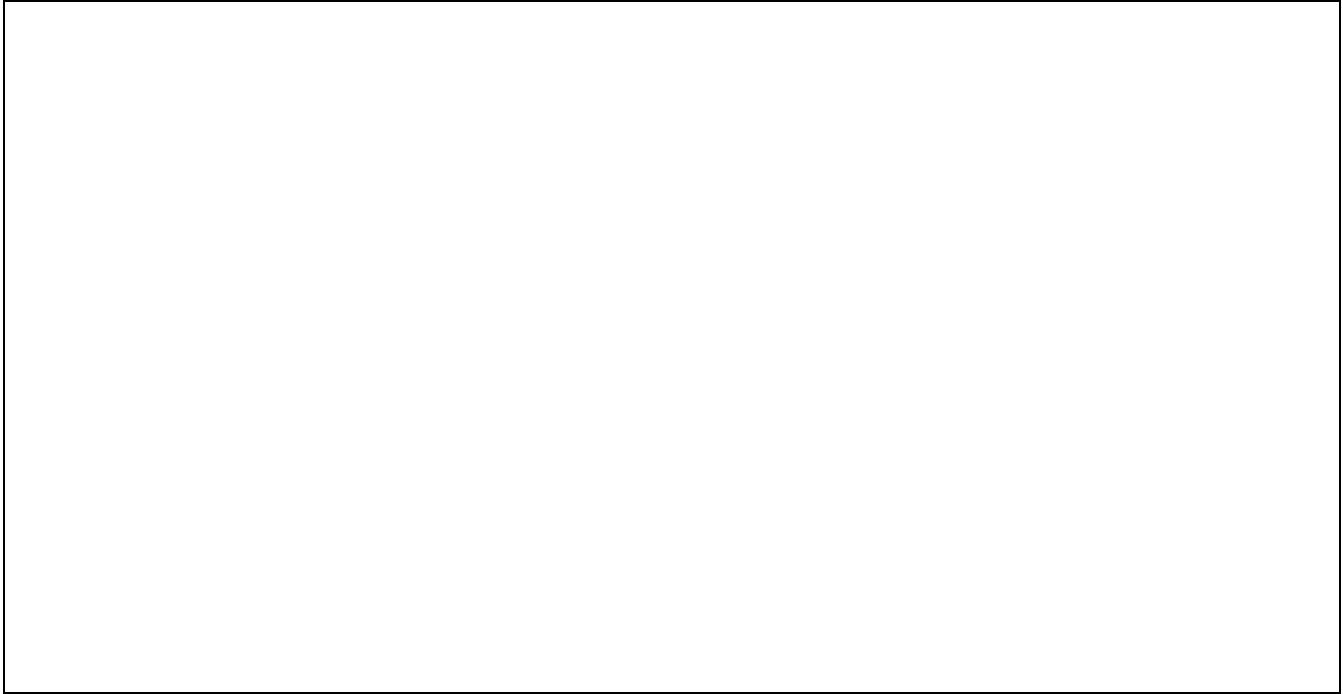


Figure 3: Counter Design (blank)

- e. Have your design checked by your instructor.

Instructor Initials \_\_\_\_\_

- f. Open your Alpha project file and add the mod-7 counter to the design.
1. Use an “input pin” at the clk input as with the logic diagram, also use an “output pin” at the three TFF outputs ( $Q_1$ ,  $Q_2$  and  $Q_3$ ). \*\*Note, T Flip-Flop – TFF is found in the “Storage” folder after double clicking on the BDF).
  2. Outputs Q1, Q2 and Q3 should also drive the BCD circuitry you developed in Section IV.
  3. Be sure to wire any unused asynchronous inputs (i.e. PRN and CLR) to +VCC (found in “other” folder after double clicking BDF).
- g. Compile and simulate your counter.
- h. Verify the output sequence of your counter is correct.

## EE334 PRACTICAL EXERCISE: Digital Logic III

### VI. Assembling the Alpha Code State Machine

- Download “1-sec Clock” from the EE334 website and copy all of the files into your current project directory. Insert the “one-second clock” symbol into your project’s directory.
- Assign the input pin on “one\_sec\_clock” to PIN\_A12 (DE-1 Board). Send the “one\_sec\_clock” output to the clock inputs of your mod-7 counter (you will need to remove the “input pin” from the previous version of your circuit).
- Add a 7447 decoder/driver chip to your circuit. The 7447 chip will drive one of the 7-segment displays on the Altera board, making binary numbers appear as decimal numbers on the 7-segment display. The 7447 symbol is included in Quartus’s list of circuit symbols.
- Wire the outputs of the BCD generator to the 7447 and connect the outputs of the 7447 to output pins using Figures 2 and 4 as guides. You may delete the ABCD output pins leftover from Section IV from the circuit at this time

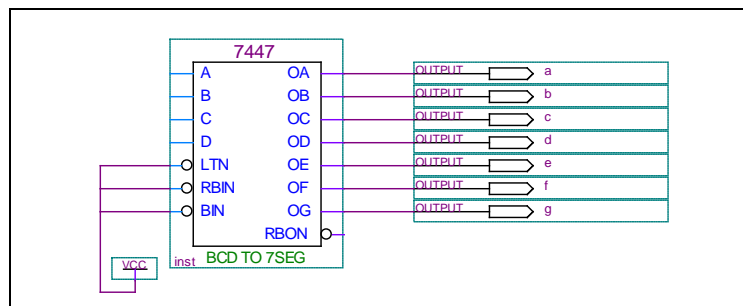


Figure 4: 7447 Decoder Device

- Referencing the pin tables in Section IV of the *Altera Quartus II Supplement* assign the correct pins for the seven-segment display (PIN\_J2 for a output, PIN\_J1 for b output, etc)
- Save, compile and program your board using the JTAG mode. Have your instructor verify successful operation of your alpha code display.

Instructor Initials: \_\_\_\_\_

### VII. Optional: Scrolling Alpha Code

- Change your Alpha counter circuitry to one that will scroll your alpha code across the four 7-segment displays. **Hint:** Starting with the circuit from Section VI you’ll need to add more 7447 decoders and some method to store/shift numbers from one 7-segment display to another.
- Program your Altera board. Once it works, have your instructor initial that your project is complete.

Instructor Initials: \_\_\_\_\_