Lab 3
DESIGN A KEYPAD ENCODER SYSTEM and Custom Display
This is a group lab assignment (3-4 students)

1. Learning Objectives

In this lab, you will practice working with an encoder system for a keypad. First, you will learn how to design a keypad encoder and simulate it to verify that it works. Second, you will learn how to design a decimal display decoder and simulate it to verify that it works. Third, you will learn how to combine designs with some additional modules to obtain a flat design for a complete system. Then, you will download and test your design in hardware to verify that the system works. This is summarized as follows:

1. Design a keypad encoder and simulate it to verify that it works.
2. Design a decimal display decoder and simulate it to verify that it works.
3. Obtain a structure design for the complete keypad encoder system.
4. Download and test your design in hardware.

Lab Preparation

- Read this document completely before you start on this experiment.
- Review lecture notes and the Lab 3 PowerPoint.

Equipment and Materials
Access to Xilinx software

Software needed	Quantity
Download the ISE® WebPACK™ software from the Xilinx website, www.xilinx.com.	1

Additional References
Current Xilinx ISE Software manuals found on Xilinx web site: www.xilinx.com.

Part 1. Keypad Encoder System with Its Inputs and Outputs

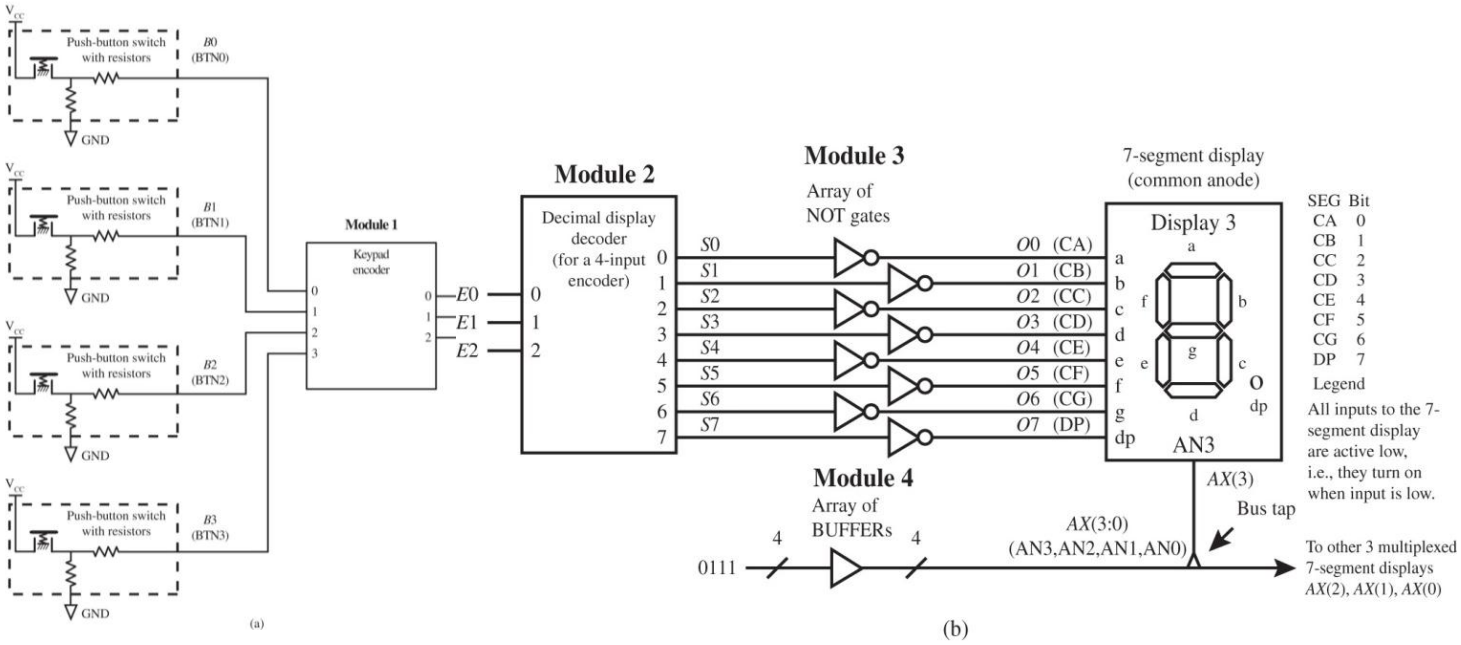

Figure 1 shows annotated schematic for the keypad encoder system.
Annotated schematic for the keypad encoder system: (a) push-button inputs and keypad encoder; (b) decimal display decoder, array of NOT gate, array of buffers, and 7-segment display

Table 1. Truth table for the keypad encoder (Module 1)

B3	B2	B1	B0	E2	E1	E0	
0	0	0	1	0	0	0	Only press B0 so output E2 E1 E0 is 000 or 0
0	0	1	0	0	0	1	Only press B1 so output E2 E1 E0 is 001 or 1
0	1	0	0	0	1	0	Only press B2 so output E2 E1 E0 is 010 or 2
1	0	0	0	0	1	1	Only press B3 so output E2 E1 E0 is 011 or 3
0	0	0	0	1	1	1	E2 is 1 to indicate no button is pressed (We elected to make E1 and E0 also 1s because it doesn't matter what their values are when E2 is 1)

Notice that a keypad encoder has a restriction placed on its inputs just like the restriction placed on the inputs of telephone keypads and computer keyboards. The restriction is simply that you are only allowed to press one key at a time.
Notice in Figure 1 that the outputs of the keypad encoder are connected to the inputs of the decimal display decoder. The truth table for a decimal display decoder with active high outputs is shown in Table 2.

Table 2. Truth table for the decimal display decoder with active high outputs

E2	E1	E0	S7	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	0	0	1	1	0
0	1	0	0	1	0	1	1	0	1	1
0	1	1	0	1	0	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0

7-segment display

Blank the display
(turn off all segments)

Please note:

S7	S6	S5	S4	S3	S2	S1	S0
dp	g	f	e	d	c	b	a

Tasks:

1. Create a new project and write complete VHDL code for Figure 1(a) for the keypad encoder (module 1). Simulate the design to verify that it follows the keypad encoder truth table. [Note: review previous labs on how to create VHDL Test Bench Program]. If the outputs of the keypad encoder do not follow the truth table, then you know that the VHDL code for module 1 has an error. You must find the error or errors and fix them.

2. Create a new project and write complete VHDL code for Figure 1(b) for the decimal display decoder (module 2). Simulate the design to verify that it follows the decimal display decoder truth table. If the outputs of the decimal display decoder do not follow the truth table, then you know that the VHDL code for module 2 has an error. You must find the error or errors and fix them.

3. Create a Structural new project design and write complete VHDL code for the keypad encoder system. Use the designs for modules 1 and 2 as components and include them in a single architecture, then add the VHDL code for modules 3 and 4. Be sure that all external input and output signals are declared in the entity and that all internal signals are declared between the different components.

4. Complete the design cycle for your keypad encoder system by doing the following:
   a. Assign package pins for all the port signals in the entity for your design.
   b. Generate a programming file, then check to see if your VHDL code needs to be corrected based on reported errors and warnings; if so, correct your VHDL code, then rerun Generate Programming File.
   c. Download the programming file into the FPGA on your Spartan board.

5. Check to see if your keypad encoder system works in hardware that is, when BTN0 is pressed the 5-segment display should display a 0, when BTN1 is pressed, the 7-segment display should display a 1; when BTN2 is pressed, the 7-segment display should display a 2; and when BTN3 is pressed, the 7-segment display should display a 3; and when no push button is pressed, the 7-segment display should be blank. The push buttons may be pressed in any order, but only press one push button at a time. If

your keypad encoder system design does not perform in this manner, you have made a mistake that you must find and correct.

Part 1. Lab Report Requirements:

1. To receive full credit you must demonstrate your final working design and get it signed off by your lab instructor/TA. First print out a cover page with only the following information: course title, experiment number, your name, and your partner's name(s). Then invite your lab instructor/TA to come to your bench to observe our final working design. Your final working design for this experiment is task 5 (Keypad Encoder System) and 7 (Custom Decimal Display Decoder System).
2. Include the complete VHDL code for your (Keypad Encoder Design) as well as the (Custom Decimal Display Decoder System).
3. Include a printout of the simulation waveform diagram for all components and the entire designs. Identify the inputs on the waveform diagram.
4. Include a printout of the Edit Constraints (Text), which shows the package pin assignments for your designs. Make sure you include your comments.
5. For the screen captures, make sure the signals are clear and the signal values are legible. It is ok if it on the black background. Put in figure captions explaining what the figures show. The screenshots should clearly show the signal values.
6. Do not attach your code separately but paste the code at the end in your report.
7. Write a short paragraph summarizing the work you did for this experiment, and describe any problems you may have encountered while obtaining your solutions. You may include any helpful hints and improvements you may think of for this experiment.

<u>Part 2.</u> Annotated schematic for the custom display decoder system

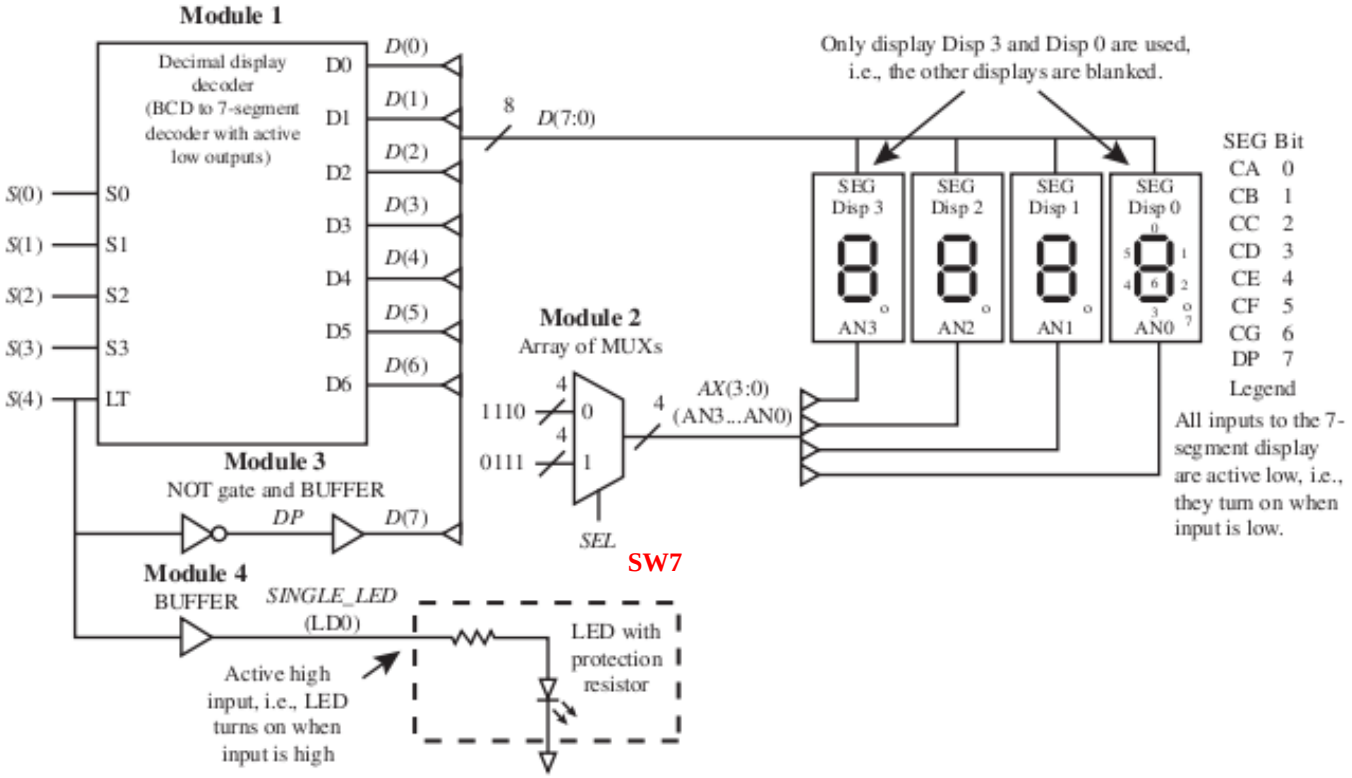

Figure 1. **Annotated schematic for the custom display decoder system**

Figure 1 shows an annotated schematic for the custom decimal display decoder system. Notice that the four slide switches SW3, SW2, SW1, and SW0 are used for the BCD inputs and one push-button switch BTN3 is used for the lamp test (LT) input. Notice that SW7 is used to switch between Disp 3 (display 3) and Disp 0 (display 0) of the 7-segment display.

Please note:

D(6)	D(5)	D(4)	D(3)	D(2)	D(1)	D(0)
g	f	e	d	c	b	a

BCD # (0-9)

Decimal display decoder

S(4)	S(3)	S(2)	S(1)	S(0)	D(6)	D(5)	D(4)	D(3)	D(2)	D(1)	D(0)	7-segment display
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	1	1
0	0	0	1	0								2
0	0	0	1	1								3
0	0	1	0	0								4
0	0	1	0	1								5
0	0	1	1	0								6
0	0	1	1	1								7
0	1	0	0	0								8
0	1	0	0	1								9
0	1	0	1	0								Blank*
0	1	0	1	1								Blank
0	1	1	0	0								Blank
0	1	1	0	1								Blank
0	1	1	1	0								Blank
0	1	1	1	1								Blank
1	X	X	X	X								8

* All segments turned off

Table 1. Partially filled in truth table for the decimal display decoder with active low outputs.

Table 1 shows a partially filled in truth table for the decimal display decoder with active low outputs. When the BCD value is 0 through 9 and the lamp test input is inactive, the BCD value should be displayed on the 7-segment display. When the **Lamp Test (LT)** input is active, signal S4 is 1, all of the segments for the 7-segment display are turned on to test if the segments are good—that is, not burned out. When a BCD value is not entered (binary values 10 through 15 are entered) and the lamp test is inactive, the display should be blank—that is, all segments turned off.

SEL	AX(3:0)
0	
1	

Table 2. Template for a compressed truth table for the array of MUXs.

Table 2 shows a template for a compressed truth table for the array of MUXs. When the input signal SEL is 0 and 1, you must fill in the outputs provided by the array of MUXs (module 2). Keep in mind that the output of the array of MUXs is a bus that has the data type std_logic_vector in VHDL.

Tasks:

1. Fill in the rest of the truth table in Table 1 for the decimal display decoder.
2. Create a new project and write complete VHDL code for the decimal display decoder (module 1). Using a any design style with a conditional signal assignment ***(when-else statement) (if, else statement) or case statement***. Simulate the design to verify that it follows the truth table that you created just for the inputs S(4) S(3) S(2) S(1) S(0) = 00000 to 1XXXX. If the outputs of the decimal display decoder do not follow the truth table, then you know that the VHDL code for module 1 has an error. You must find the error or errors and fix them.
3. Fill in the truth table in Table 2 for the array of MUXs.
4. Create a new project and write complete VHDL code for the array of MUXs (module 2). Use any design style with a conditional signal assignment ***(when-else statement) (if, else statement) or case statement***. Simulate the design to verify that it follows the truth table that you created. If the outputs of the array of MUXs do not follow the truth table, then you know that the VHDL code for module 2 has an error. You must find the error or errors and fix them.
5. Create a new project and write complete VHDL code for the **custom decimal display decoder system** using a **structural design approach**. Use the designs for modules 1 and 2 and include them in a single architecture as components; then add the VHDL code for modules 3 and 4. Be sure that all external input and output signals are declared in the entity and that all internal signals are declared between architecture and the first begin. Use Boolean equations or any VHDL code that we have discussed in the class to write the assignments for DP, D(7), and SINGLE_LED. (Hint: To form the bus for D, use either an aggregate or the concatenation operator.)
6. Complete the design cycle for your custom decimal display decoder system by doing the following:
   a. Assign package pins for all the port signals in the entity for your design. Note: Bus signals such as S(0), S(1), etc, must be listed in the Edit Constraints (Text) file as S<0>, S<1>, etc.
   b. Generate a programming file, and then check to see if your VHDL code needs to be corrected based on reported errors and warnings; if so, correct your VHDL code, then rerun Generate Programming File.
   c. Download the programming file into the FPGA board.
7. Check to see if your custom decimal display decoder system works in hardware. Verify that each binary coded decimal (BCD) number entered by the four slide switches is displayed in decimal on the 7-segment display and non-BCD numbers will result in a blank display. When the lamp test push button is pressed (lamp test input is 1), all of the segments of the 7-segment display, the decimal point of the 7-segment display, and the single LED LD0 turns on. When slide switch SW7 is pulled back, DSP0 is enabled and when slide switch SW7 is pushed forward, DSP3 is enabled. If your design does not perform in this manner, you have made a mistake that you must find and correct. Demo your work to the lab TA

For part 2 of this lab, please follow the same lab requirements as (part 1) above.
HAVE FUN!