

## Lab # 5

# Implementing Combinational Logic Circuits using FPGAs

### Lab Format

- This is a **Individual Lab** so each student must design and test their own circuits.
- Students are free to assist each other in all labs.
- Each student must complete the Preliminary Work Section **before** lab begins. Preliminary Work will be checked in lab and will be part of the lab report grade.
- Each student must submit his or her own lab report.
- Lab reports will not be accepted until all required circuits have been demonstrated to the instructor.

### A. Objective

The objectives of this laboratory are to:

- introduce the student to Field Programmable Gate Arrays (FPGAs)
- introduce the student to Aldec Active-HDL to generate VHDL code to design and simulate combinational logic circuits
- introduce the student to Xilinx ISE to synthesize a VHDL design into a specific FPGA
- introduce the student to the Digilent BASYS2 FPGA board
- introduce the student to Digilent Adept to download synthesized designs to the BASYS2 FPGA board
- give the student an opportunity to design, program, and test a simple combinational logic circuit using an FPGA

### B. Materials

Digilent BASYS2 FPGA board

Aldec Active-HDL software

Xilinx ISE software

Digilent Adept software

Reference materials (available on course Blackboard site):

*Combinational Logic Circuits using Aldec Active-HDL 9.2 and Xilinx ISE 14.2*

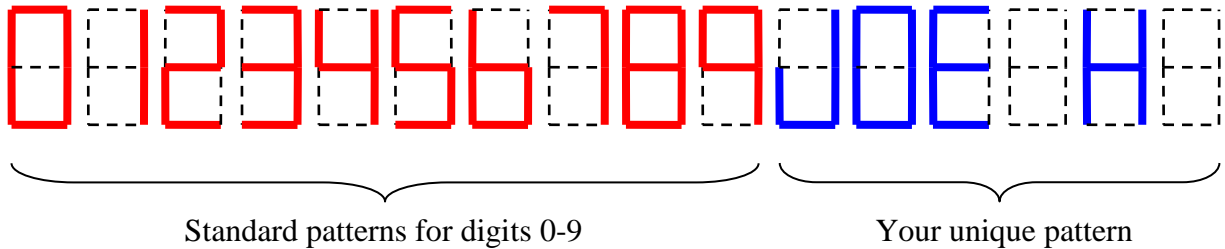
*Digilent BASYS2 Reference Manual*

### C. Introduction

Refer to class lectures and the reference materials listed above.

## D. Preliminary Work

1. Design a custom BCD-to-7-segment decoder circuit as pictured below that will have the following features:
  - The output will be connected to a common-anode 7-segment display, so remember that a 0 (LOW) lights each segment.
  - Inputs 0-9 should light the usual segments for those digits.
  - Inputs 10-15 should light your name or initials as you would like them to appear (each student must have a unique pattern). Do not use any don't cares. An example is shown below. Clearly show which segments you will light for all BCD input combinations 0-15.
  - Include a truth table.
  - Include Karnaugh maps for the 7 outputs and determine a minimal SOP expression for each output.



2. Draw a logic diagram (by hand or using PSPICE) for the SOP circuit above using only AND gates, OR gates, and inverters. Assume that ANDs and ORs are available with any number of inputs. How many gates are required?

## E. Laboratory Work

**Note: For all filenames and folder names in this lab, use only letters, numbers, and underscores. Also begin them with a letter.**

1. Use the Aldec Active-HDL software to create a VHDL design for your 7-segment decoder circuit:
  - Use VHDL to define your custom BCD-to-7-segment decoder circuit. In particular,
    - Define 12 outputs, including:
      - Cathodes a,b,c,d,e,f,g
      - DP (decimal point)
      - AN0, AN1, AN2, AN3 used to turn on or off each of the four 7-segment displays
      - Recall that:
        - AN0 <= '0' will "assert" or turn on display 0
        - AN3 <= '1' will turn off display 3
        - Use similar instructions for AN1 and AN2
    - Define 4 inputs for the BCD input
  - In the architecture section of the VHDL file, add the following expressions:
    - The SOP expressions developed in the Preliminary Work section for cathodes a-g
    - DP <= '1' to turn off the decimal point
    - AN0 <= '0' will "assert" or turn on display 0
    - AN3 <= '1' will turn off display 3
    - Use similar expressions for displays 1 and 2
    - Modify comments at the beginning of the VHDL program (name, course, description, etc)
  - Generate a testbench to test all 12 outputs for the 16 input combinations.
    - Use the testbench to generate a truth table (print it once you verify that it is correct)
    - Use the testbench to generate the input and output waveforms (and print them)
  - If the truth table was correct, print the VHDL file and the testbench file.

Use the **Xilinx ISE** software to synthesize your design for the FPGA to be used in lab. Be sure to assign pin numbers to each input and output. Print the following items from Xilinx (or PlanAhead):

- Package View of the FPGA – highlight and label each input and output
- Design Summary
- Pinout Report arranged by signal
- User Configuration File
- Static Timing Report
- RTL Schematic

2. Connect the Digilent BASYS2 FPGA board to the computer using the USB cable provided.

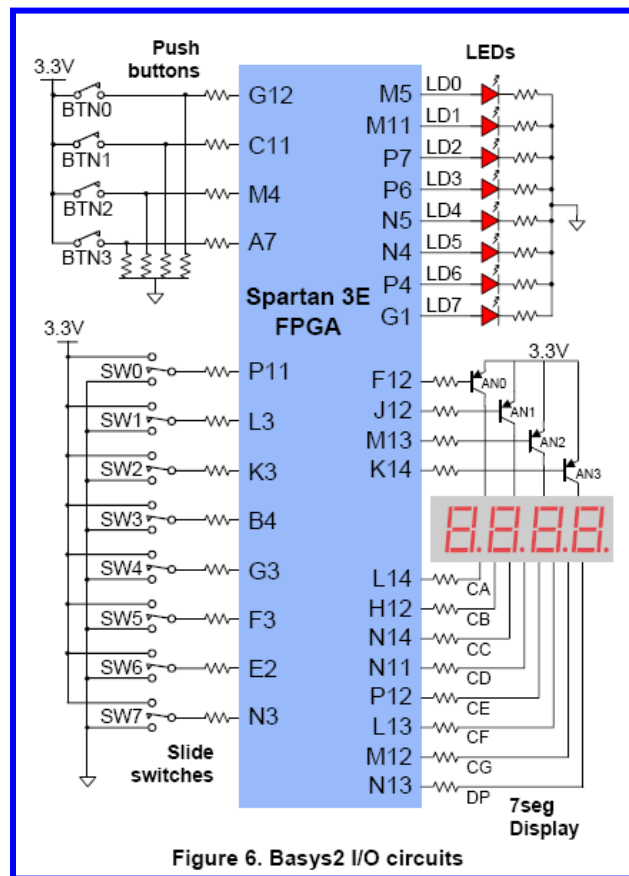
- Move the JP3 jumper to the PC (left) position
- Turn ON the board (slide switch in lower left position). Red Status LED should turn ON.

3. Use the **Digilent Adept** software to download your design to the BASYS2.

- Test your design and verify that it is correct for each of the 16 possible switch combinations. Record the results (the truth table).
- Turn off the BASYS2 board and turn it back on again. Your design should have been lost.
- Move the JP3 jumper to the ROM position and use the Digilent Adept software to program the PROM on the BASYS2 board.
- Test your design again and verify that it is correct.
- Turn off the BASYS2 board and turn it back on again. Your design should still work properly.

4. **Demonstration**

- Demonstrate the proper operation of your design to the instructor.
- Demonstrate that the design has been programmed into PROM by turning off the BASYS2 board and turning it on again and showing that the design still functions properly.



## F. Report

Remember that each lab report should have the following four sections. Also see additional notes below.

### Title Page

### Preliminary Work

- Include instructions

### Lab Results

- Include all measured results (truth tables).
- Include step numbers and titles or headings that make it clear what is being shown.
- Include printouts of the following files from Aldec software (add a title to each item):
  - **VHDL code for your 7-segment display circuit.** Add title section information. Highlight all information that you entered.
  - **VHDL testbench code.** Add title section information. Highlight all information that you entered.
  - **Truth table.** Write your functions (as sums of minterms) on the sheet and verify if they are correct.
  - **Waveform File.** Write your functions (as sums of minterms) on the sheet and verify if they are correct.
- Include printouts of the following files from Xilinx (including PlanAhead) software. (add a title to each item):
  - **Design Summary**
  - **Pinout Report** – arranged by used signal name
  - **User Configuration File** (use Notepad and look for file with .ucf extension)
  - **RTL Schematic**
  - **Static Timing Report.** Highlight and identify the longest path
  - **Package View of Xilinx Spartan-3E.** Highlight and label all used inputs and outputs.

### Discussion/Conclusion

- Answer the following questions about your design:
  - How many Look Up Tables (LUTs) were used? (See Design Summary)
  - What is the total number of LUTs available?
  - What percentage of the LUTs did your design use?
  - How many inputs/outputs did your design use (unbounded IOB's). (See Design Summary) List all inputs and outputs for your design and show that the total number agrees.
  - How many total IOBs were available?
  - What is the longest signal path through your design? (See Static Timing Summary)
  - What is the purpose of the UCF file?
- Discuss the use of discrete gates (Labs 1-4) with the use of an FPGA.
- How many gates did you calculate in step 2 of the Preliminary Work that would have been required to implement your circuit using only AND gates, OR gates, and inverters? Compare this to the number of gates shown in the RTL schematic. Are the circuits identical? Are they equivalent?