EGR 270
Fundamentals of Computer Engineering
File: N270L5

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 Vivado to implement a VHDL design into a specific FPGA
- introduce the student to the Digilent BASYS3 FPGA board
- give the student an opportunity to design, program, and test a combinational logic circuit using an FPGA

B. Materials

Digilent BASYS3 FPGA board
Aldec Active-HDL software
Xilinx Vivado software
Reference materials (available on course Blackboard site):

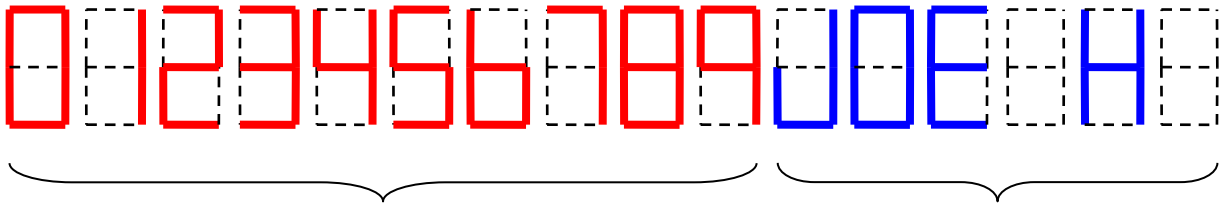
Tutorial 1 - Combinational Logic Circuits using Aldec Active-HDL and Xilinx Vivado
Digilent BASYS3 Reference Manual

C. Introduction

Refer to class lectures and the reference materials listed above.

D. Preliminary Work

1. ***Custom decoder design***: Design a custom BCD-to-7-segment decoder circuit as pictured below that will have the following features:
   - Use the inputs A,B,C,D and the outputs ca,cb,cc,cd,ce,cf,cg, where ca indicates cathode a. Let A be the MSB.
   - 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.



Standard patterns for digits 0-9 Your unique pattern

2. ***Logic Diagram***: 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?
3. ***Pinout***: Show the BASYS3 pinout.
4. ***Pin Number Table***: Complete the table below by adding the BASYS3 pin numbers. Refer to the BASYS3 pinout.

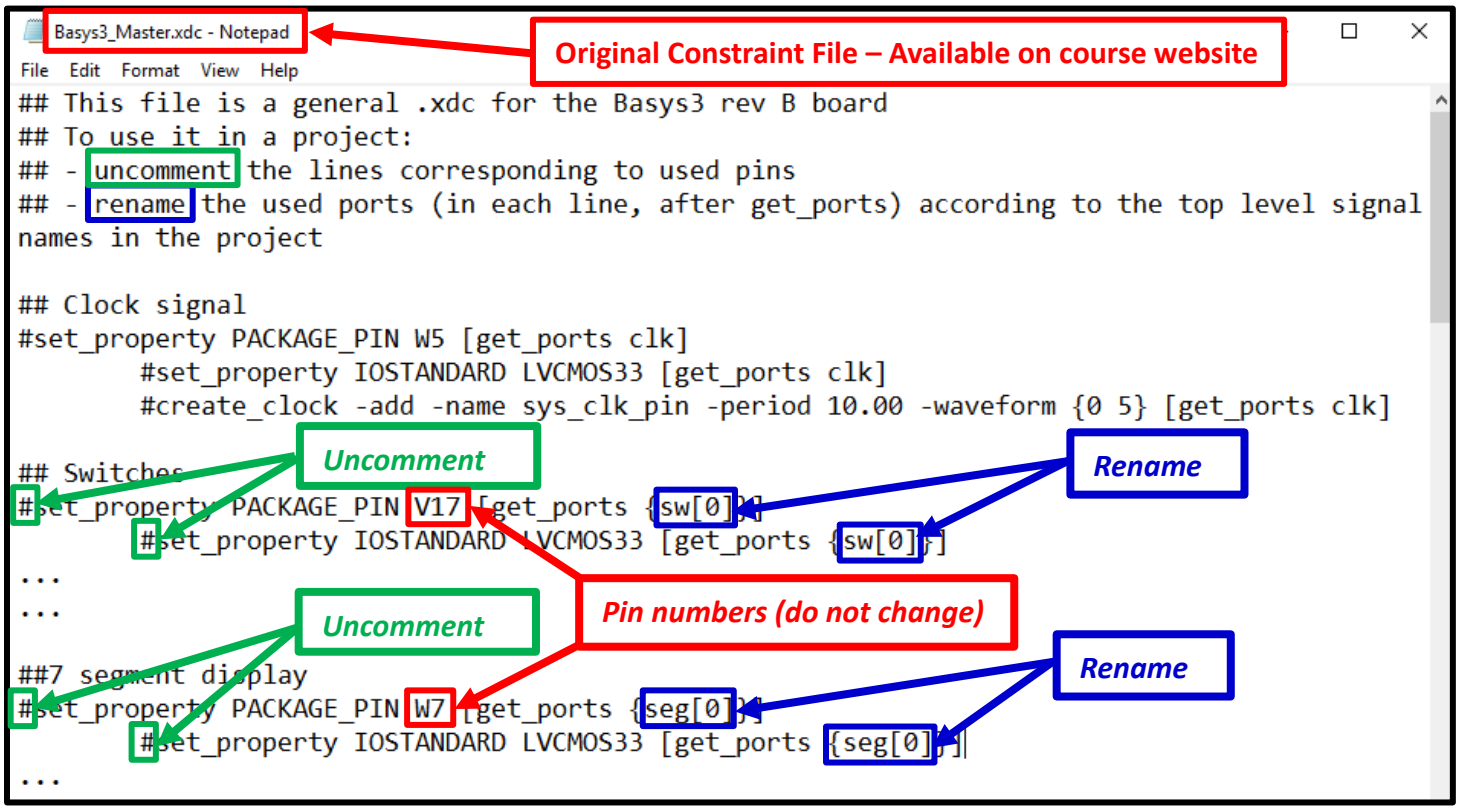
Signal	BASYS3 input/ouput	BASYS3 pin number
A	SW3	
B	SW2	
C	SW1	
D	SW0	
AN3	AN3	
AN2	AN2	
AN1	AN1	
AN0	AN0	

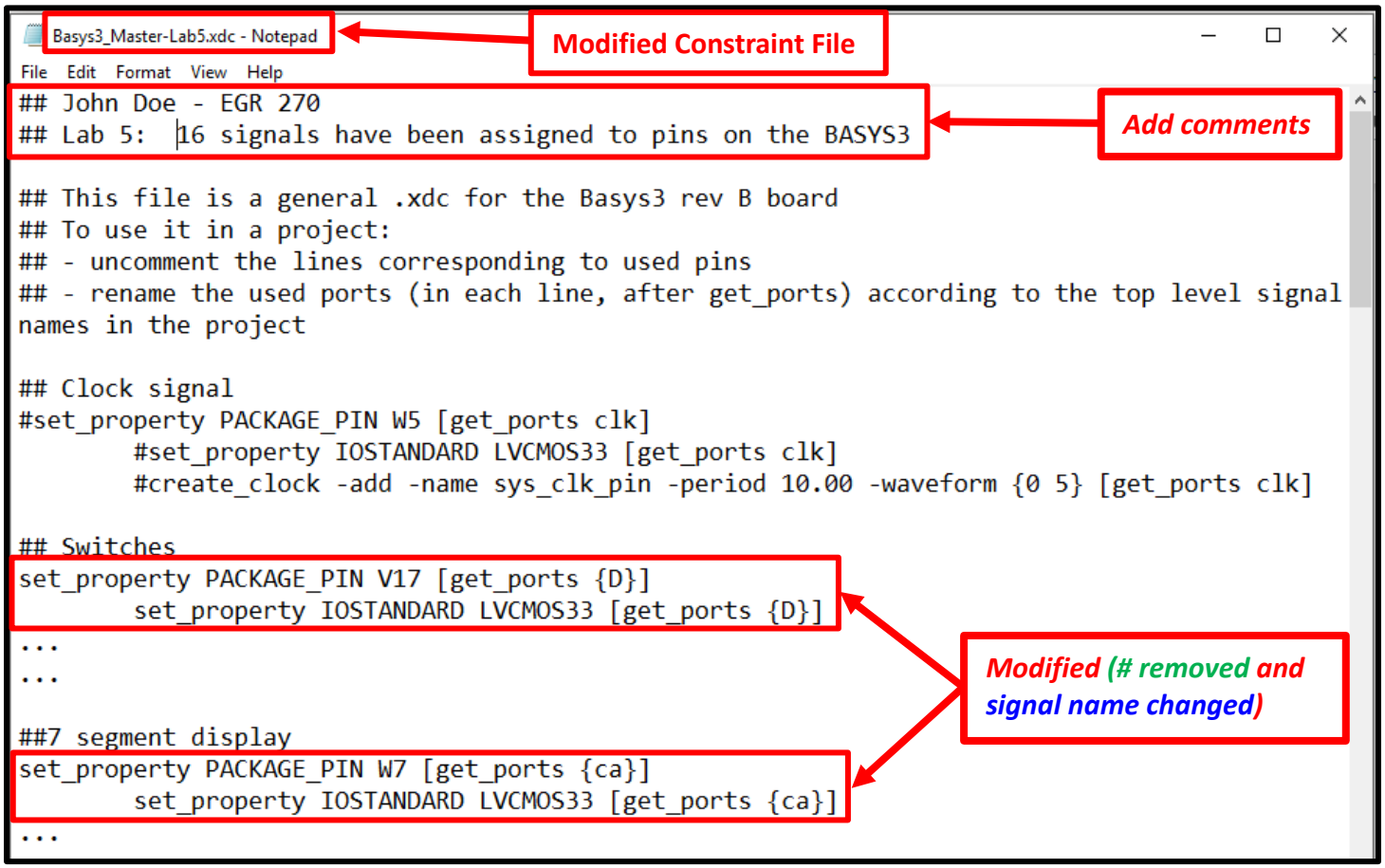
Signal	BASYS3 input/ouput	BASYS3 pin number
ca	CA	
cb	CB	
cc	CC	
cd	CD	
ce	CE	
cf	CF	
cg	CG	
dp	DP	

5. ***Configuration File***: Create a configuration file containing the pin assignments for input and output signal in the vhdl file. Specifically:
   A. Download the configuration file **Basys3_Master.xdc** from the course website.
   B. Copy and paste the file and then rename the newly created file (**Basys3_Master_Lab5.xdc)**
   C. Open the configuration file using ***Notepad***. Modify it to remove comments (#) and assign pins for each of the 16 signals listed in the table above. See the example on the following page.
   D. Save the new file.

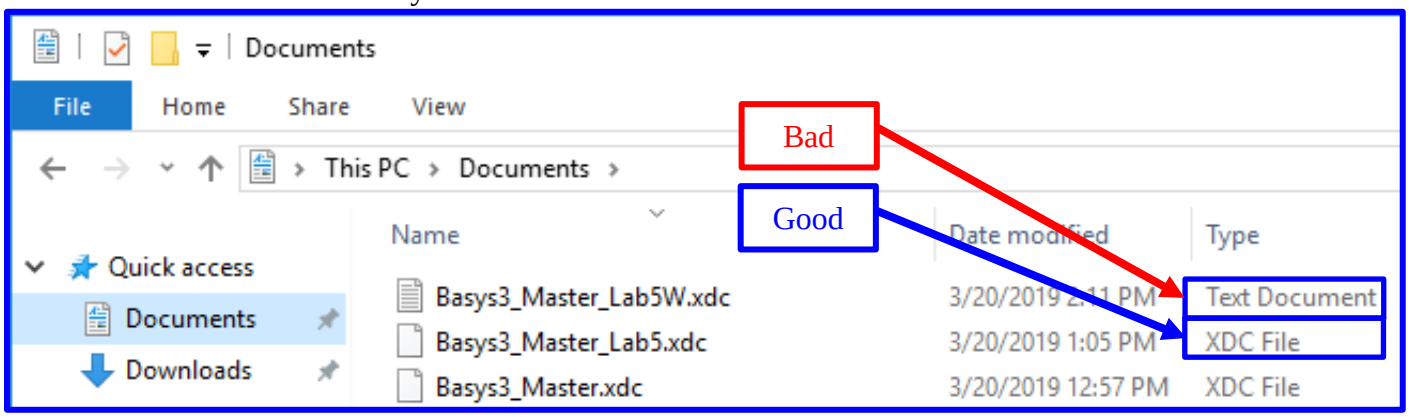
Warning: Signal names in the xdc file are case sensitive. If your signal is named CA in the VHDL file, it must be CA (not ca) in the xdc file.

Example: Modifying the Configuration File (2 of the 16 signals to be modified are shown)

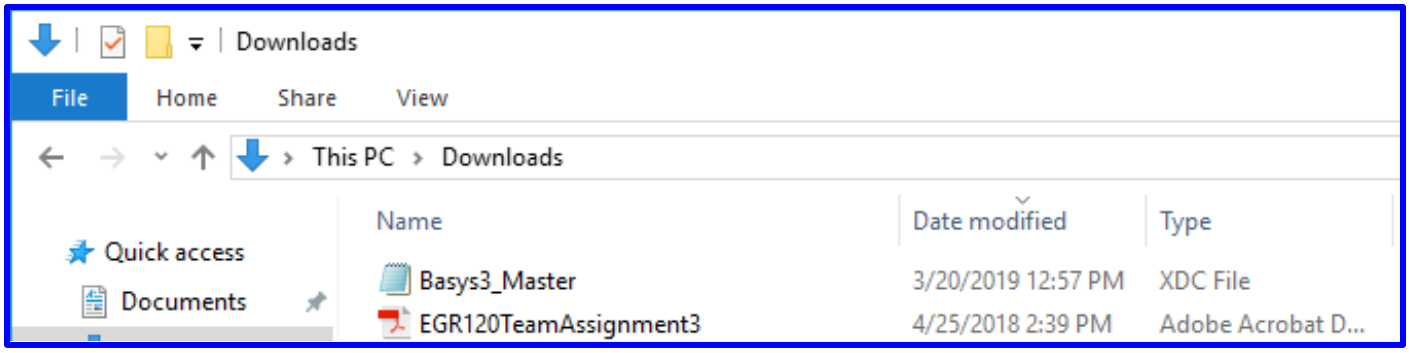




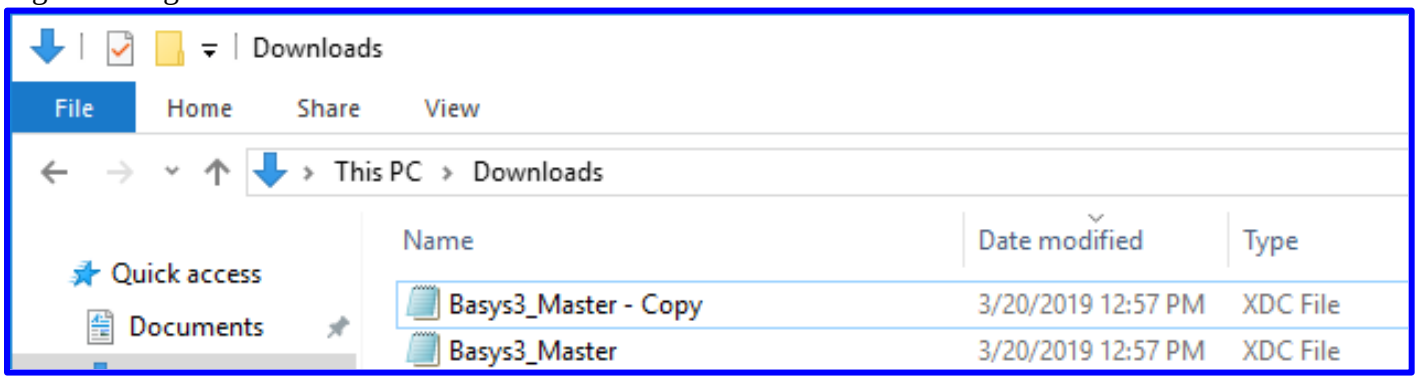
Note: Be sure that the XDC file you create is saved as an XDC file and not a TXT file. See below:



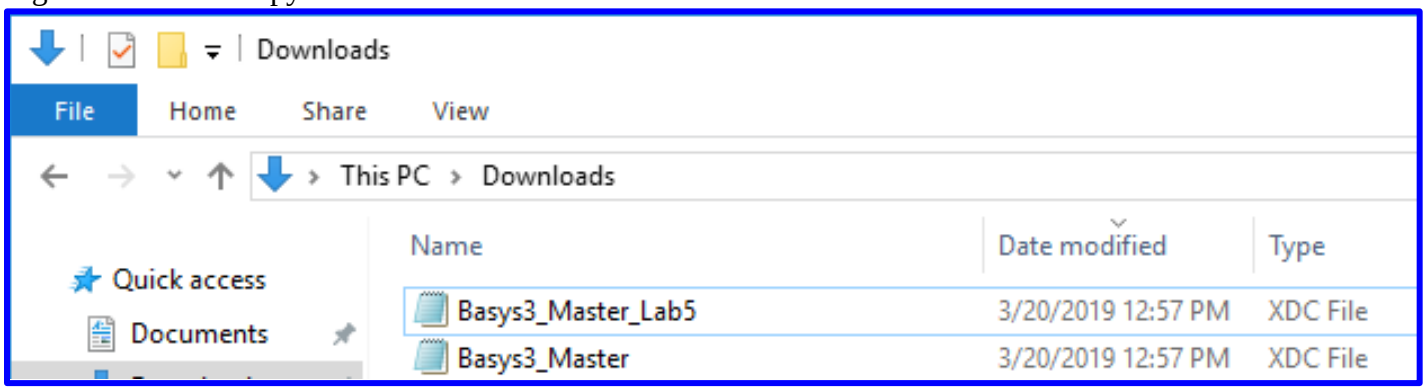
If you open the original file with NotePad and save it, NotePad may convert it to a Text file.
One way to get around this problem is to copy and past the original file and then edit it with NotePad.
This is illustrated below.



Right-click on the XDC file and select COPY.
Right-click again and select PASTE.



Right-click on the Copy above and select RENAME.



Now open the new file (Basys3_Master_Lab5) using Notepad and make the required changes. Save the file.

E. **Laboratory Work**

Note: For all filenames and folder names in this lab, use only letters, numbers, and underscores (no spaces or special symbols). Also begin them with a letter.

1. ***Aldec:*** Use the **Aldec Active-HDL** software to create a VHDL design for your 7-segment decoder circuit (refer to the tutorial):
   - Use VHDL to define your custom BCD-to-7-segment decoder circuit. In particular,
     - Define 4 inputs for the BCD input: A,B,C,D
     - Define 12 outputs, including:
       - Cathodes ca,cb,cc,cd,ce,cf,cg
       - 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. AN0 is the rightmost display.
         - AN3 <= '1' will turn off display 3
         - Use similar instructions for AN1 and AN2
   - In the architecture section of the VHDL file, add the following expressions:
     - The SOP expressions developed in the Preliminary Work section for cathodes ca-cg
     - 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 the 7 outputs to the display (ca, cb, etc.) for the16 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.
   - Summary of items printed in this section:
     - ***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 (list file).*** Write your functions for ca, cb, etc., (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.
2. ***Xilinx:*** Use the **Xilinx Vivado** to implement your design into the FPGA on the BASYS3 (refer to the tutorial):

   Print the following items from Xilinx: (If you have trouble printing, you might try the Snipping Tool in Windows)
   - ***Constraints File (xdc)*** – Highlight all changes made to the file
   - ***Project Summary*** – Shows project and FPGA information. Be sure to select the ***Table*** option (not Graph) showing the ***Utilization***. Highlight the number of LUTs and number of I/Os used.
   - ***Schematic (for implemented design)*** – Shows used LUTs, input buffers, output buffers, etc.
   - ***Package View*** – Shows the bottom view of the FPGA showing which of the 250 pins on the Artix-7 FPGA have been used.
     - ***Print the entire package***

- **_Print the parts of the package showing your signal names:_** Zoom in on areas where your 16 inputs and outputs are listed. You might capture them using the Snipping tool in Windows. Highlight all 16 signals.

- **_IO Ports_** – Shows the used inputs/outputs and the assigned pins. Be sure that all 16 signals are shown.
- **_Schematic (for RTL simulation)_** – Shows schematic using logic gates.

3. **_Testing_**: Download your design into the BASYS3 using Method A (download **_bit file_** into the FPGA):
   - Test your design and verify that it is correct for each of the 16 possible switch combinations.
   - Turn off the BASYS3 board and turn it back on again. Your design should have been lost.
   - Download your design again and press the Reset button on the BASYS3. Your design should have been lost.
   - Download your design again and demonstrate proper operation to the instructor.
   - ***Record the results (the truth table), including a description of what happened when you pressed the Reset button on turned the BASYS3 off and on again.***
4. **_Testing_**: Download your design into the BASYS3 using Method B (download **_bin file_** into the flash memory on the BASYS3):
   - Test your design and verify that it is correct for each of the 16 possible switch combinations.
   - Turn off the BASYS3 board and turn it back on again. Your design should still work.
   - Press the Reset button on the BASYS3 board. Your design should still work.
   - Demonstrate proper operation to the instructor.
   - ***Record the results (the truth table), including a description of what happened when you pressed the Reset button on turned the BASYS3 off and on again.***

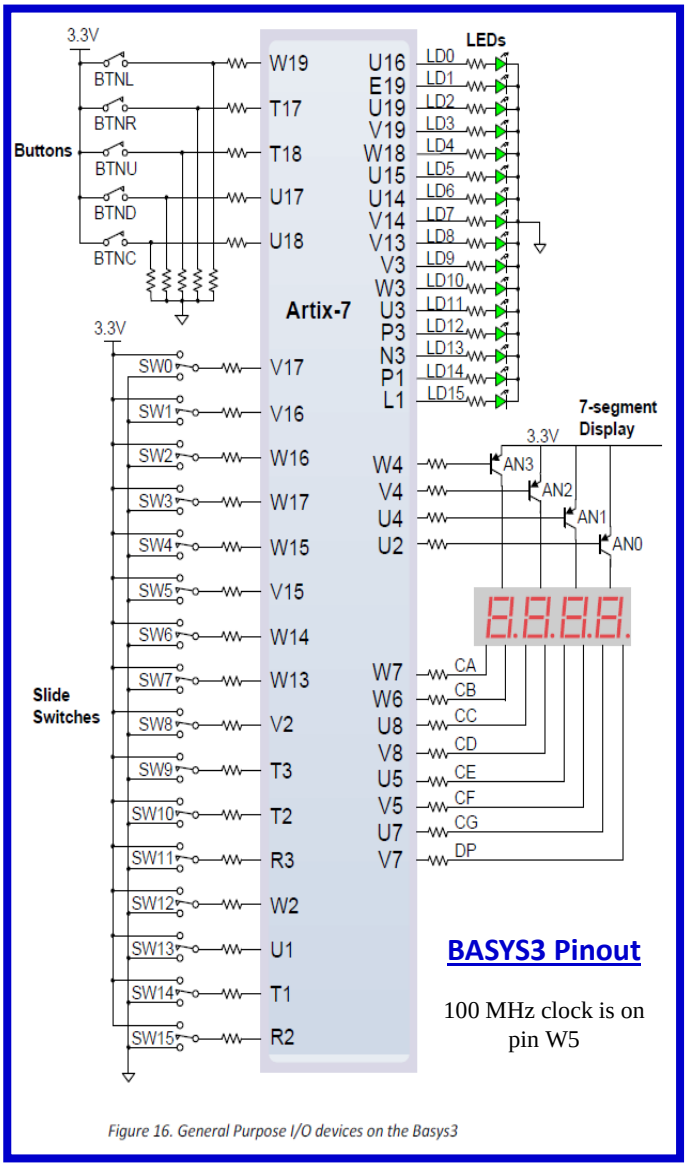


Figure 16. General Purpose I/O devices on the Basys3

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

- Programming the FPGA directly (Method A): Include all measured results (truth tables) and describe what happened when the Reset button was pressed and when the BASYS3 was turned off and on again.
- Programming the FPGA using onboard flash memory (Method B): Include all measured results (truth tables) and describe what happened when the Reset button was pressed and when the BASYS3 was turned off and on again.
- 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 the Xilinx Vivado software (add a title to each item):
  - ***Constraints File (xdc)*** – Highlight all changes made to the file
  - ***Project Summary*** – Shows project and FPGA information. Be sure to select the Table option (not Graph) showning the Utilization. Highlight the number of LUTs and number of I/Os used.
  - ***Schematic (for implemented design)*** – Shows used LUTs, input buffers, output buffers, etc.
  - ***Package View*** – Shows the bottom view of the FPGA showing which of the 106 pins have been used.
    - ***Print the entire package***
    - ***Print the parts of the package showing your signal names:*** Zoom in on areas where your 16 inputs and outputs are listed. You might capture them using the Snipping tool in Windows. Highlight all 16 signals.
  - ***IO Ports*** – Shows the used inputs/outputs and the assigned pins. Be sure that all 20 signals are shown.
  - ***Schematic (for RTL simulation)*** – Shows schematic using logic gates.

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 (IOs) did your design use. (See Design Summary) List all inputs and outputs for your design and show that the total number agrees.
  - How many total IOs were available?
  - What is the purpose of the constraints file?
- Discuss the use of discrete gates (Labs 1-4) with the use of an FPGA (Lab 5).
- Discuss the difference between Method A and Method B for programming the FPGA.
- Compare the number of gates used in the RTL schematic to the number of gates you calculated in the Preliminary Work. Are the circuits identical? Are they equivalent?