EC605: Computer Engineering Fundamentals Fall 2016

Vivado Block Design Tutorial

Goals

- Introduction to Vivado design environment and Vivado Block Design.
- Create a simple inverter circuit and program Xilinx FPGA board.

Overview

This tutorial will give a step-by-step walk through of the Vivado design environment. Once completed, using Vivado should be second nature. This tutorial starts with creating a new project from scratch and completes with a simple block diagram circuit running on a Xilinx FPGA board.

Task 1: Creating a new project for a specific board

1. Open the Vivado design environment:

   In a new terminal run the following commands to setup and launch Vivado.

   > source /ad/eng/opt/64/Xilinx/Vivado/2014.4/settings64.sh
   > vivado

   OR

   Copy the *start_vivado* script into your home directory from the following location */ad/eng/courses/ec/ec605/start_vivado* , and then run the command

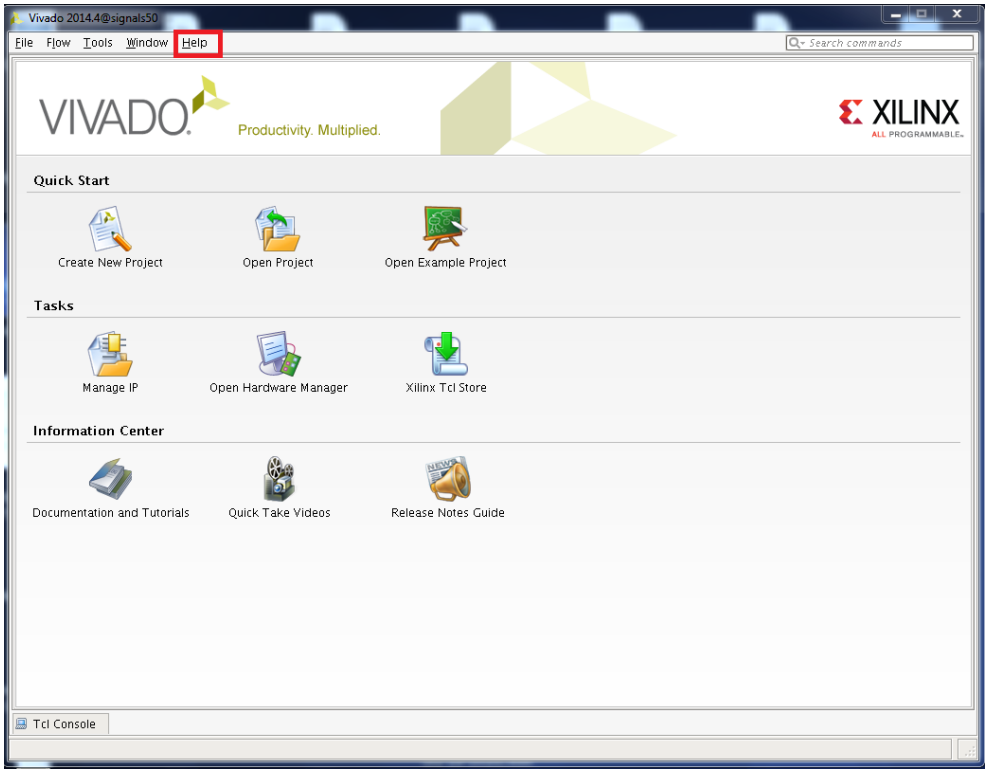
   > ./start_vivado

   You should get the following result:

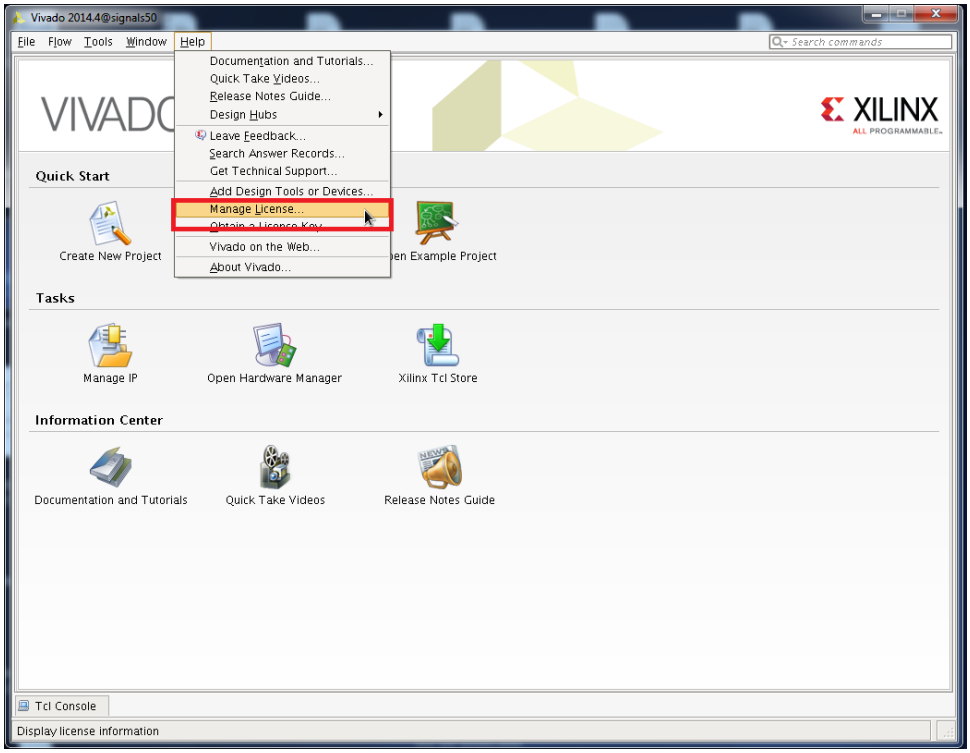
   ```
   ****** Vivado v2014.4 (64-bit)
     **** SW Build 1071353 on Tue Nov 18 16:48:31 MST 2014
     **** IP Build 1070531 on Tue Nov 18 01:10:18 MST 2014
       ** Copyright 1986-2014 Xilinx, Inc. All Rights Reserved.

   start_gui
   ```

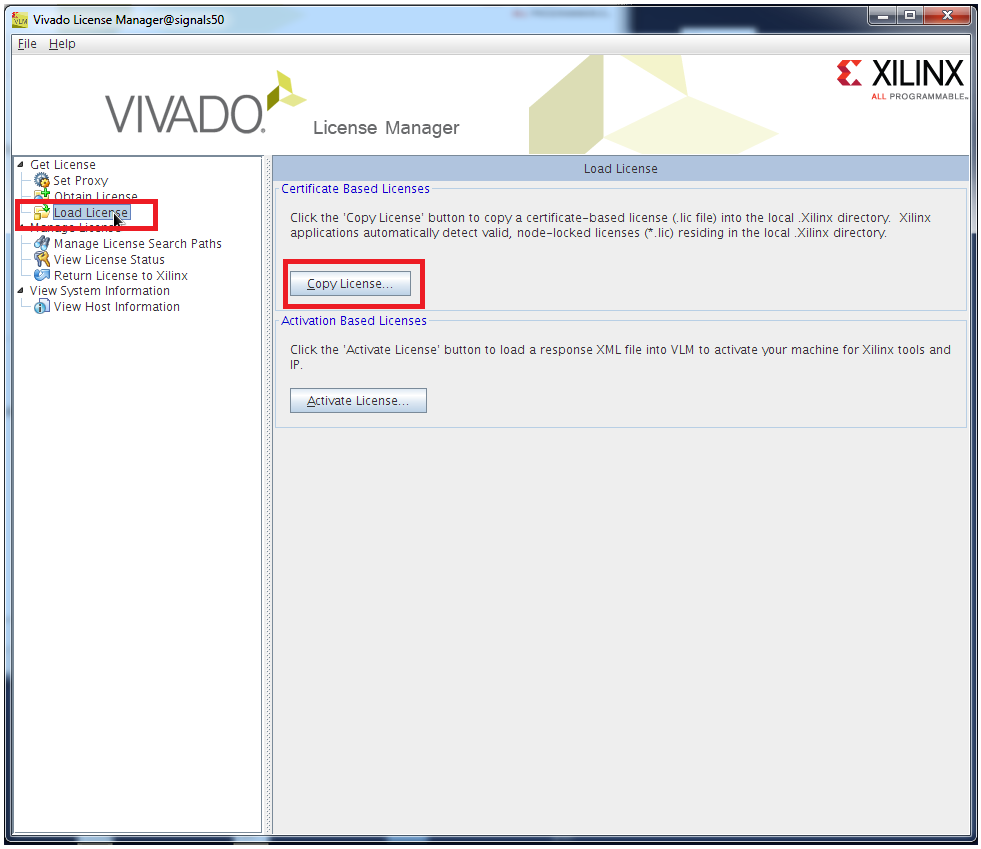
2. Set the license (this should only be done the first time you run Vivado in your account):
   a. Navigate to the *Help* tab in the Vivado Window



   b. From the *Help* tab select the "*Manage License*..." Option to open the License Manager Window

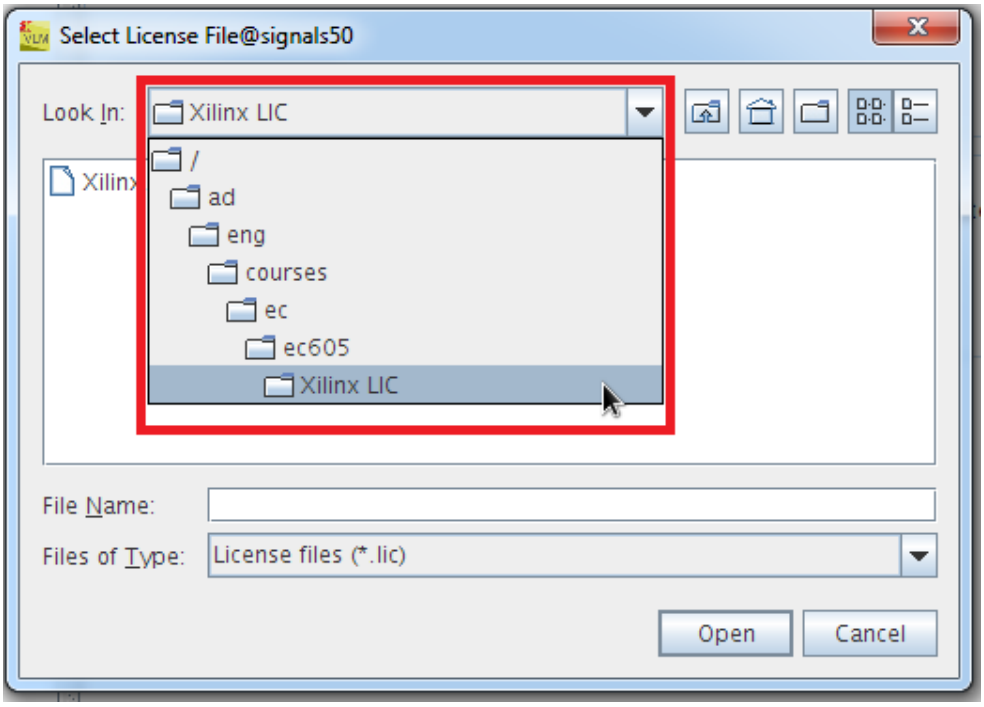


c. From the *License Manager* Window select the "*Load License*" option and click on the "*Copy License…*" button, which should open a Select License File window.

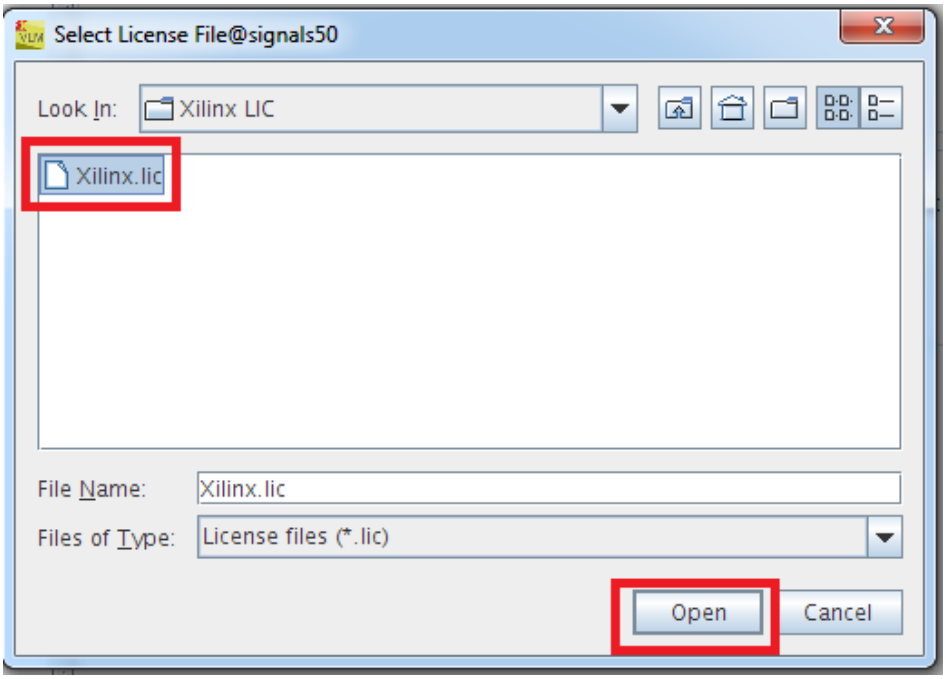


d. From the *Select License File* window navigate to the *Xilinx LIC folder* in the ec605 course

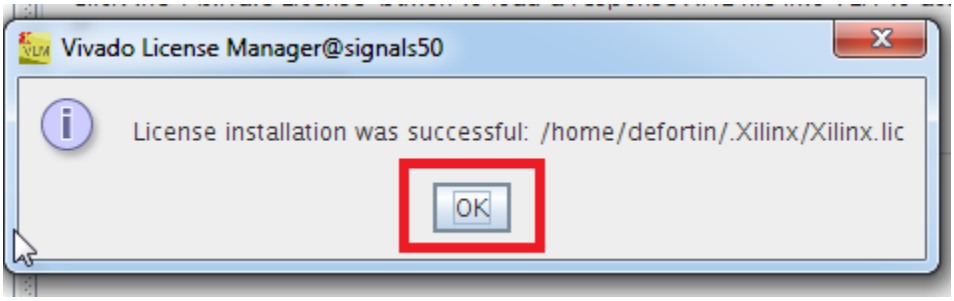
directory (See Screenshot for path):



e. Once in the *Xilinx LIC* directory, select the *Xilinx.lic* file and hit the *Open* button.



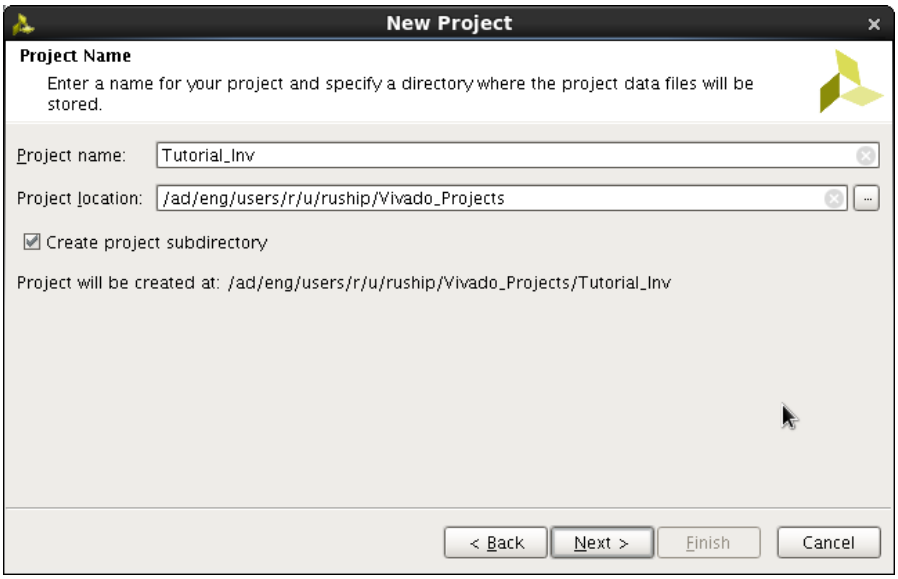
f. Once you hit *Open* you should receive the following message:



g. If you do not, try again or email David Fortin at defortin@bu.edu.

3. Click *Create New Project*, and then click *Next.*

4. Give your new project a name and location to save, and click *Next.*

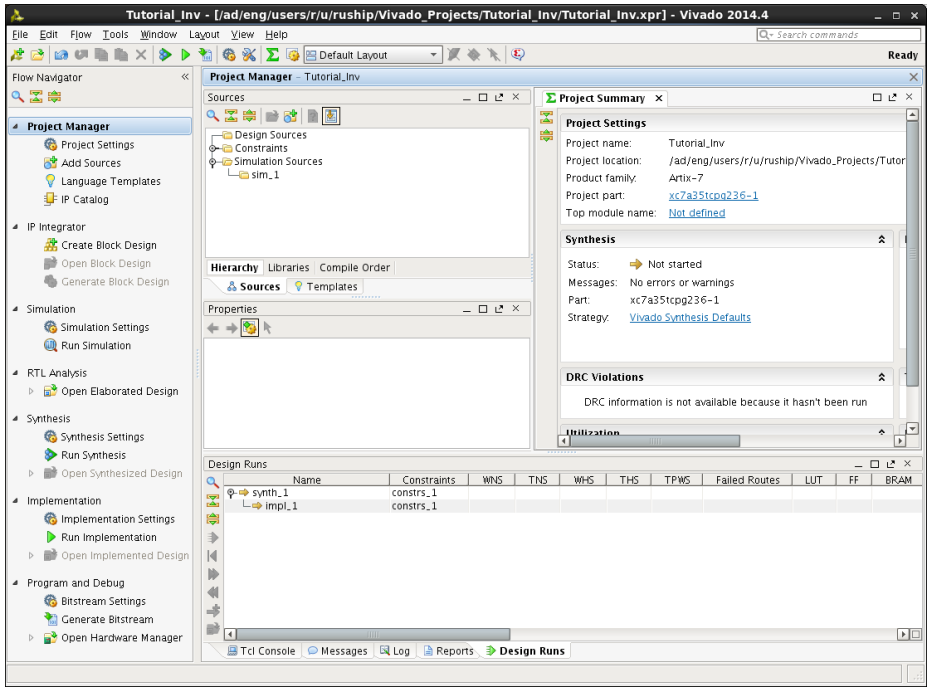


5. Choose *RTL Project* and select the "*Do not specific sources at this time*" check box. Click *Next.*

6. Find the specific FPGA board that we will be using in class:

   Product category: *All*
   Family: *Artix-7*
   Sub-Family: *Artix-7*
   Package: *cpg236*
   Speed Grade: -1
   Temp grade: All
   *Part: xc7a**35**tcpg236-1*

7. Click *Finish*. Your window should look as follows:

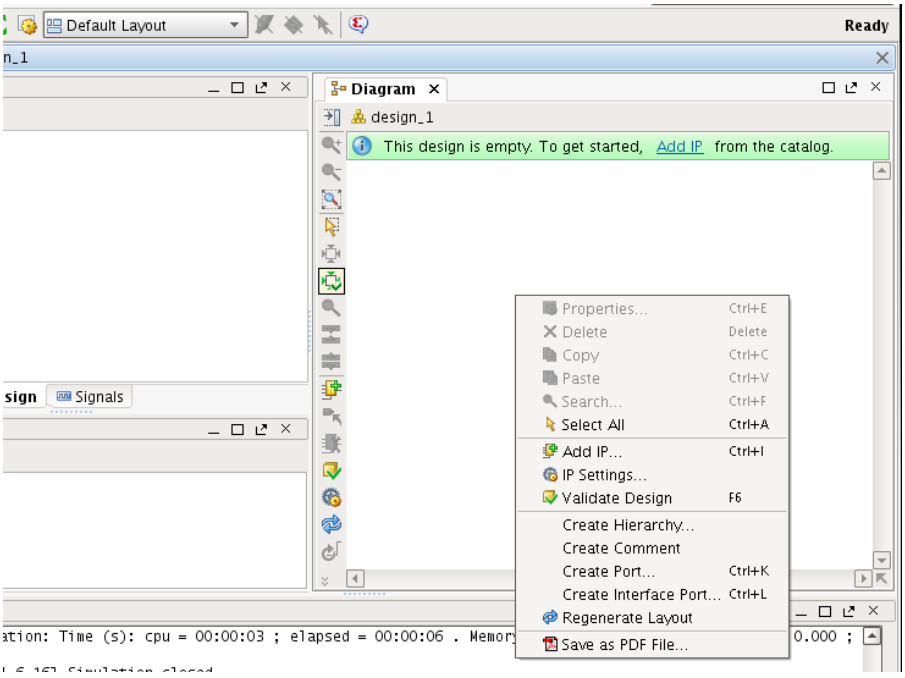


Task 2: Add a source block diagram and IP repository

1. On the *Flow Navigator* on the left select "*Create Block Design*".

2. Give the new design a name (we'll leave it as '*design_1*' for now) and press *OK*.

3. Next we will add the IP repository that will be used for our assignment. Right Click inside the Block Diagram and choose IP Settings:



4. Add an IP library:

   Click *Add Repository* and navigate to the IP library *XUP Lib.*
   Navigate to */ad/eng/courses/ec/ec605/XUP_Lib.*
   Click Select on the *XUP_Lib* folder.
   Once Vivado adds all IPs to the project click *OK*.

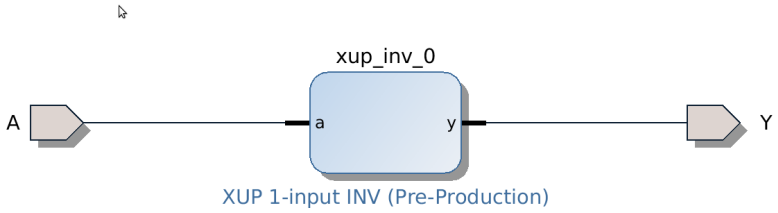
5. Add a new IP block from the newly added repository:

   Right Click inside Block Diagram and choose *Add IP*.
   Type *XUP* to narrow the results to the new repository.
   Select "*XUP 1-input INV*" by double clicking.

6. Now add 2 ports one for input and one for output:

   Right click and select *Create Port*.
   Give your input port the name "*A*" and verify the direction is "*input*".
   Click *OK* and connect the port to the inverter by dragging a line.
   Do the same for output port "*Y*" and verify the direction is "*output*".
   Click *OK* and connect the port to the inverter by dragging a line.

7. Your final design should look like the following:



Task 3: Generate an HDL wrapper and compile for the board.

1. On the *Flow Navigator* click on *Project Manager.*

2. Right click on the name of your block diagram design ('*design_1*') under "*Design Sources*".

3. Click on "*Create HDL Wrapper*".

4. Select "*Let Vivado manage wrapper and auto-update*" and click *OK*.

5. Synthesize your design:

   On the *Flow Navigator*, under "*Synthesis*" select "*Run Synthesis*".
   You may be asked to save the project – Click *Save.*
   When successful, the *Synthesis Complete* dialog will open.
   Press *Cancel* in the *Synthesis Complete* dialog.

Task 4: Add a constraints file

A constraints file is provided for the first part of Lab 1. This file will specify how the input and output ports are connected to the Xilinx FPGA board.

1. Select *Project Manager* from the *Flow Navigator.*

2. Select *Add Sources*.

3. Choose "*Add or create constraints*", then click *Next*.

4. Select "*Add Files*" and select the provided Lab1_Task1_Inv_Port_List.xdc, and check the "*Copy constraints file into project*" box.

5. Click *Finish*.

6. Double click on the file inside the Constraints folder to inspect its formatting. For each input and output, we specify the pin number on the board that we connect it to. For example, the following line:

   set_property PACKAGE_PIN V17 [**get_ports** A]

   Connects input A to pin number V17, which is the rightmost switch on the board. Pin numbers are specified on the board itself, as well as on the pin layout figure in the appendix of this tutorial.

   In addition, for each connected pin, we must specify the source voltage, 3.3V as follows:

   set_property IOSTANDARD LVCMOS33 [**get_ports** A]

   You will extend and modify this file for future projects.

Task 5: Run Implementation, Generate Bitstream

1. In the Flow Navigator select "*Run Implementation*" under *Implementation*.
   This will generate a design file based on the hardware board and block diagram. If successful, an *Implementation Complete* dialog box will appear. Click *OK* to close this dialog box.

2. To generate a bitstream for the design select "*Generate Bitstream*" from the bottom of the *Flow Navigator,* under "*Program and Debug*". Click *OK* to close this dialog box. This file is necessary to program the FPGA.

Task 5: Program the FPGA Board

1. Connect the board to the computer via the USB cable, and turn the ON switch on the board, right next to the cable. A red LED labeled "POWER" should light up.

2. Select "*Open Hardware Manager*" from the bottom of the *Flow Navigator,* under "*Program and Debug*".

3. Click "*Open target*" on the top green bar, and select "*Auto Connect*".

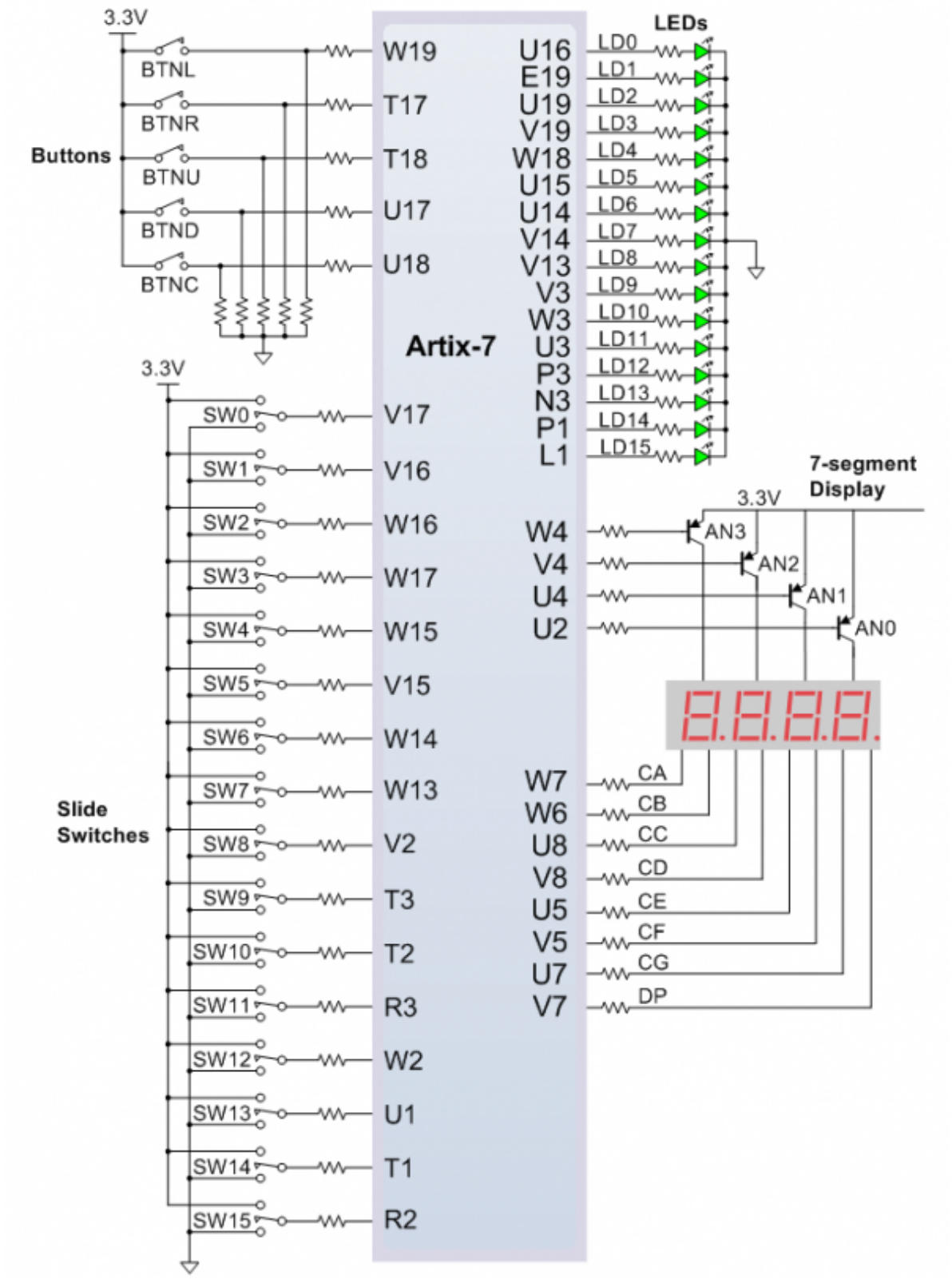
4. Click "*Program device*" on the top green bar. A green LED labeled "DONE" should light up.

5. Test the functionality of your design by toggling the rightmost switch and inspecting the response in the rightmost LED. The LED should respond in reverse to the switch, behaving according to the output of the inverter gate.

Appendix A: Basys 3 Reference Manual

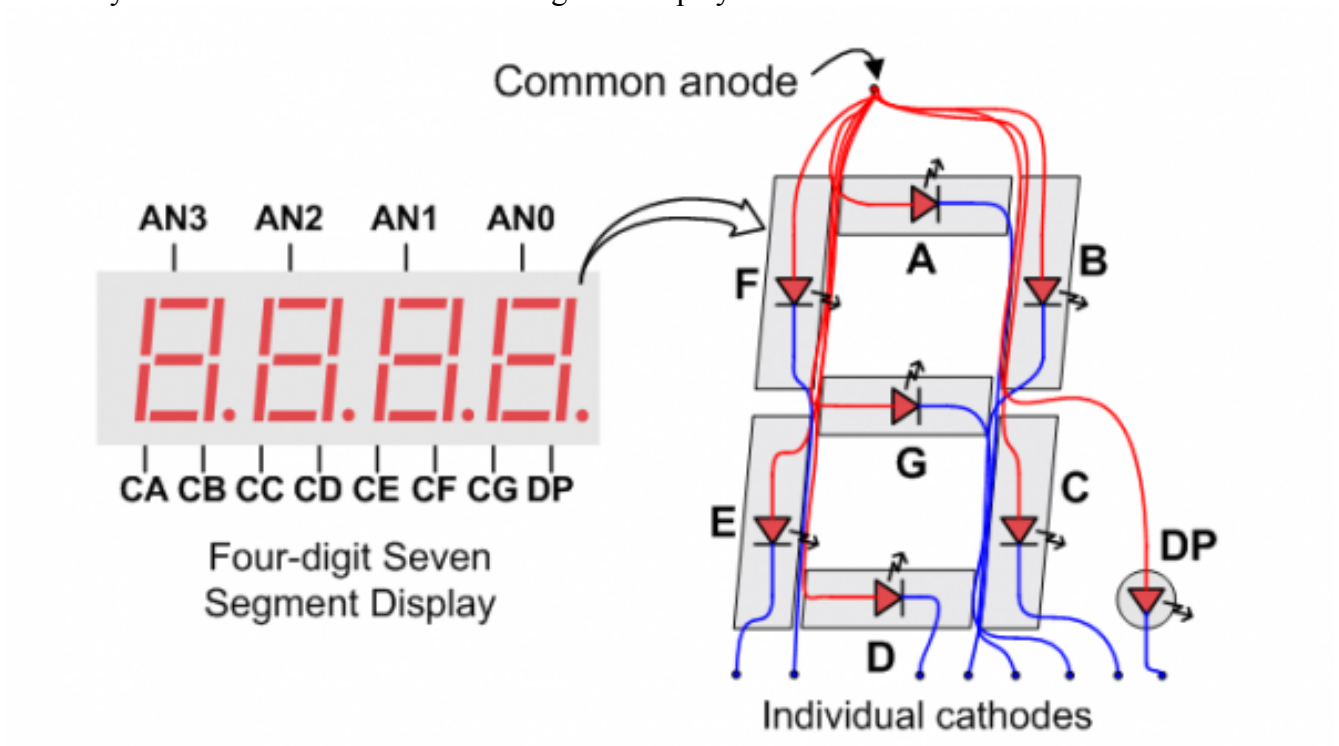
1. List of input / output pins

The Basys3 board includes sixteen slide switches, five push buttons, sixteen individual LEDs, and a four-digit seven-segment display, as shown in the below diagram. The five pushbuttons, arranged in a plus-sign configuration, are "momentary" switches that normally generate a low output when they are at rest, and a high output only when they are pressed. Slide switches generate constant high or low inputs depending on their position.


3.3V
BTNL
BTNR
BTNU
BTND
BTNC
Buttons
W19
T17
T18
U17
U18
Artix-7
LEDs
U16
E19
U19
V19
W18
U15
U14
V14
V13
V3
W3
U3
P3
N3
P1
L1
LD0
LD1
LD2
LD3
LD4
LD5
LD6
LD7
LD8
LD9
LD10
LD11
LD12
LD13
LD14
LD15
3.3V
SW0
SW1
SW2
SW3
SW4
SW5
SW6
SW7
SW8
SW9
SW10
SW11
SW12
SW13
SW14
SW15
Slide Switches
V17
V16
W16
W17
W15
V15
W14
W13
V2
T3
T2
R3
W2
U1
T1
R2
7-segment Display
3.3V
AN3
AN2
AN1
AN0
W4
V4
U4
U2
CA
CB
CC
CD
CE
CF
CG
DP
W7
W6
U8
V8
U5
V5
U7
V7

2. Seven Segment Display

The Basys3 board also includes 4 seven-segment displays.



To illuminate a segment, the anode should be driven high while the cathode is driven low. However, since the Basys3 uses transistors to drive enough current into the common anode point, the anode enables are inverted. Therefore, both the AN0..3 and the CA..G/DP signals are driven low when active.

3. Oscillators / Clocks

The Basys3 board includes a single 100MHz oscillator (clock) connected to pin W5 (W5 is a MRCC input on bank 34).

Additional information can be found here: https://reference.digilentinc.com/basys3/refmanual