Downloading to the FPGA Board

PURPOSE -- This lab will present a brief overview of the Digilent Basys 3 FPGA Board and how to download a bitstream (the result of the implementation of a design) file to the Xilinx Artix-7 FPGA or the ROM on this board.

Introduction to the Artix-7 FPGA and Basys 2 Board.

In this lab, you will become familiar with the Digilent Basys 3 FPGA Board to which you will download the bitstream for the 8-bit full adder synthesized and implemented in the previous two labs.

The Basys 3 is richer in hardware than the earlier Digilent Basys 2 or Basys boards. It can be programmed and powered via a USB to micro-USB cable and uses an Artix-7 FPGA. Digilent designed this board with the intent of providing a entry level design platform for Xilinx series 7 FPGAs. The Artix-7 device on this board has significantly more capability that the Spartan-3E FPGA devices used on the earlier Basys 2 and Basys boards. Moreover, the number of switches, pushbuttons, and leds on the Basys 3 is increased. Like the earlier Basys 2 board the Basys 3 has The Artix-7 on this board requires Vivado and cannot be used with the older ISE design tool.

In addition to the on-board The four on-board 12-pin connectors can accommodate any of Digilient's low-cost "PMOD" accessory circuits making it easy to add A/D and D/A converters, motors, sensors and a variety of
other devices and circuits.

Features:

- Features the Xilinx Artix-7 FPGA (XC7A35T-ICPG236C)
  - 33,280 logic cells in 5200 slices
  - 1,800 Kbits of fast block RAM
- Internal clock speeds exceeding 450 MHz
- Serial Quad SPI FLASH (Spansion S2FL032)
- USB-UART bridge
- Four 12-pin Pmod connectors
- 12-bit VGA output
- Digilent USB-JTAG port for FPGA programming & communication

NOTE: Take a few minutes and read the reference manual for the Basys 3. It can be found as a pdf document on the class webpage or at the Digilent website (www.digilentinc.com).

For the following make sure that the power supply source is set to USB. Connect the Basys 3 board and power it ON. Generate the bitstream, open a hardware session, and program the FPGA

The power supply source jumper on the Basys 3 should be set to USB and a micro-USB cable should be connected between the Basys 3 board and your PC. This is the only cable needed, the board can be powered and configured with just this cable. These are all located near the corner of the board where the DIGILENT logo is located.

Power the board by setting the power switch on the board to ON. The red POWER led should light.

Programming Methods

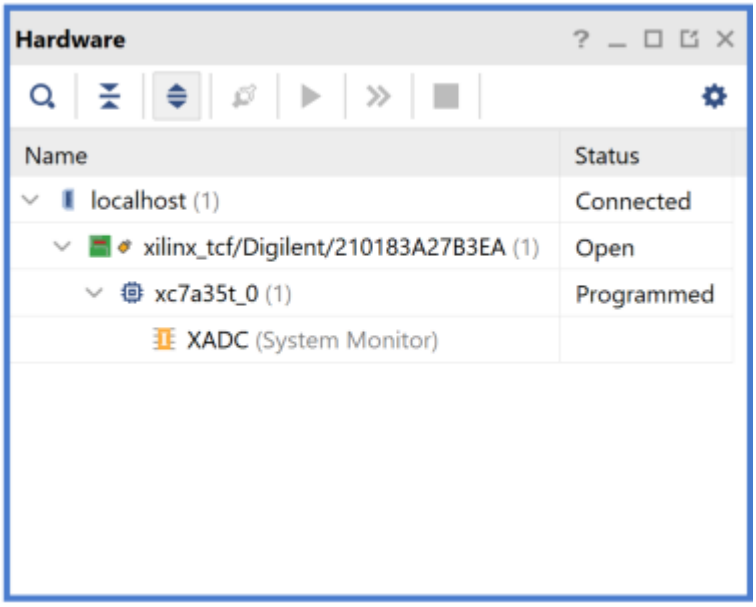
The device can be programmed in various ways, two of which are described below:
1. Jumper (JP1) set to JTAG and using the Bitstream (.bit file) – Program is volatile in this case
2. Jumper set to QSPI and using the .bin file

1. Creating the bitstream for JTAG method

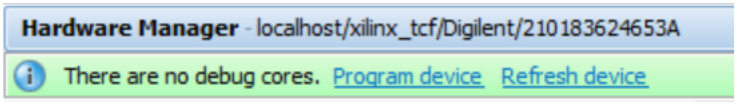
Click on **Generate Bitstream** under the **Program and Debug** tasks in the **Flow Navigator** pane. The bitstream will be generated . A pop-up menu Bitstream Generation Completed will appear when this is done. Click on **Open Hardware Manager**.

Downloading to the Basys 3

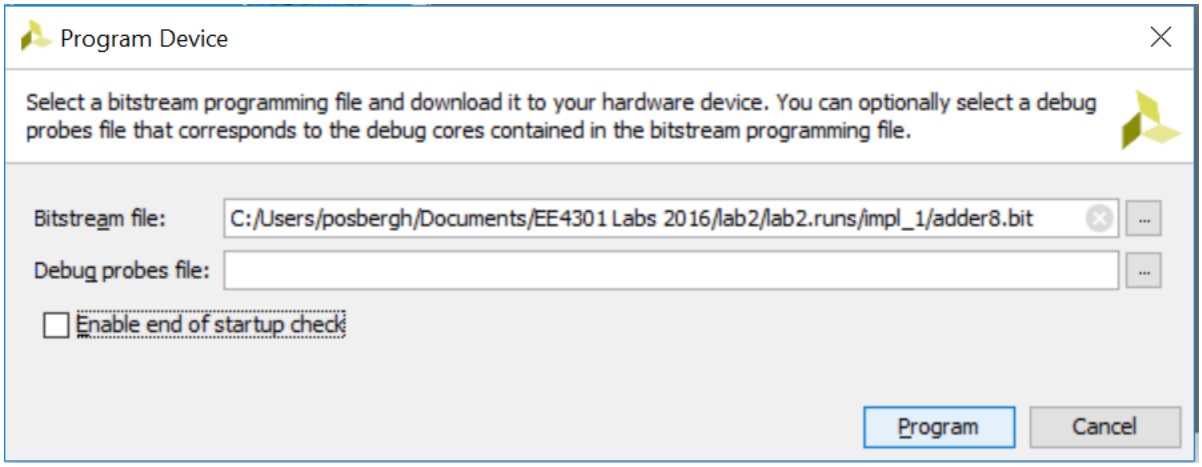
When the hardware manager is open you will see the message Hardware Manager – unconnected. And beneath this No hardware target is open. **Open target**. Click on the Open target link. From the dropdown menu that appears select **Auto Connect**. You should now see



Above this you should see



Click on **Program Device** and then on the device you want to program, xc7a35t_0. The following pop-up window appears.



Check that you have the correct bit file, *adder8.bit*, and click on Program. Vivado will now program the FPGA.

Verify the functionality of your design by flipping switches. Demonstrate to your TA.

Turn off power to your Basys 3, now turn it back on. Your design is gone. The SRAM cells in the FPGA which were programmed with your bitsteam are volatile. If power to the FPGA is lost so is your configuration. For a SRAM based FPGA the configuration file is typically stored in non-volatile ROM which is automatically read to initialize the FPGA when it is powered-up. In fact, the FPGA on the Basys 3 board can be programmed three different ways, by JTAG, from the onboard quad SPI flash, or from a USB flash drive. The last of these two methods store the configuration file in non-volatile memory which is read by the FPGA on power up.

2. Programming the QSPI Flash ROM – QSPI method

To program the FPGA so that it retains after subsequent power-ups, we need to program the QSPI serial flash on the Basys 3 board with a configuration file. For this we need to create the **.bin** file with the configuration.

Steps:

1. Click on the Bitstream Settings in the Program and Debug section of the Project manager Select the bin_file option Click OK Click on Generate Bitstream (it is generated in the lab3/lab3.runs/impl_1 directory if the project name in lab3 and current_run is impl_1)

2. In the Hardware Manager follow open target steps until device xc7a35t_0 is visible Right-click on xc7a35t_0 **Add Configuration Memory Device** select the *Spansion s25fl032p* (this is the QSPI flash memory chip on the Basys 3) Click OK In the "Do you want to program the configuration memory device now" window, click **OK** Select the Configuration File (adder8.bin) in the impl_1 directory Click **OK** (this will erase any existing design in the QSPI, including the configuration file shipped with the Basys3 board)

3. Make sure the Jumper is set to QSPI the next time you power up.

Modify the Design

The design for this lab and the two previous labs was an 8-bit adder. Now that you are familiar with the design process modify your design to be an 8-bit adder/subtractor. For this design you will continue to input two 8-bit numbers via the 16 switches and display the result with 8 leds for the sum and a 9^{th} for the carry out. However, you should modify the design so the pushbutton used to generate the carry input is repurposed. When this button is open the sum of the two 8-bit numbers is displayed. When this button is pressed it should display the difference between the two 8-bit numbers.

When you have completed this modified design demonstrate it to your TA

Lab Notebook deliverables:

- Brief description of what you did in the lab, what steps were followed
- Discussion of results:
    a. What were the test cases you used and discuss why you chose the set of test cases, what changes did you do to test the new adder/subtractor?
    b. If you had to modify your design to meet your design discuss what methodology you followed, etc. (i.e. for the adder/subtractor)
    c. Attach snippet of the reports of the tool.
        i. Table and Simulation waveforms showing all the testcases used above.
        ii. Schematic screenshot of synthesized design.
        iii. Utilization report from synthesis
        iv. Timing report
        v. Utilization report from implementation
        vi. Power report

    NOTE: Only attach relevant sections, not the entire report.
- Summary or Conclusion:
    a. Were all design goals met?
    b. What were the difficulties encountered?
    c. Any improvement you would like to make time permitting? Etc.…
- Verilog Source code:
  (Font: Courier, size 8pt, single line spacing, no paragraph spacing)
- All files must be labeled (ex. Figure 1: Simulation of eight-bit adder…). You should also highlight more interesting parts of your attachments.

SUMMARY -- This lab was about creating and downloading a configuration file (*.bit* or *.bin*) to the FPGA mounted on the Basys 3 board. You should now be familiar with the Basys 3 board and how it works.

REFERENCES

[1] Digilent, *Digilent Basys 3 Board Reference Manual*, (Available at www.digilentinc.com)

[2] Xilinx, *Vivado Design Suite User Guide: Programming and Debugging*, (UG908, June 8, 2016