

Computer Assignment 3

ECGR 2181 - Fall 2015

Assignment Overview

In this assignment you will use Vivado 15.2 Webpack to write, simulate and program an FPGA. You'll use the encoder.vhd created in the last CAD and connect it up to switches and the seven segment displays on the FPGA board. You'll also need to turn in a single PDF to Moodle (instructions below).

Background

This assignment is pretty simple. You need to declare your encoder.vhd component, then instantiate it 4 times, and connect all of the signals. I find it useful to tie the switches to the LEDs to make it easier to read what the seven segment display should read.

Structural VHDL is simple. You have to declare a component before you instantiate it. Once you instantiate it, you simply connect signals to it. Sort of like working with a schematic but with text. The VHDL slides offer information on structural syntax. If you want to

Drivers for the FPGA Board

The FPGA board requires some drivers for it to be recognized by Vivado. You need to download and install Digilent's Adept System software which can be located at <https://www.digilentinc.com/Products/Detail.cfm?NavPath=2,66,828&Prod=ADEPT2>

Getting started

1. Create a directory to store all of the assignment's files.
2. Open Vivado 2015.2 and create a new project called **computer_assignment_3**.
3. It will be an RTL project
4. Add your encoder.vhd file from CAD 2 and the supplied `ssd_muxer.vhd` file for Dr Conrad's website.
5. No existing IP.
6. Add the constraints/xdc file.
 - a) DL https://www.digilentinc.com/Data/Products/BASYS3/basys3_master.zip .
 - b) Extract the zip file.
 - c) Add the extracted file as a constraints/xdc file.
7. The Basis3 uses an **XC7A35TCPG236C-1** FPGA
 - Artix 35T
 - Speed grade -1
 - CPG236 packaging
 - C temp grade

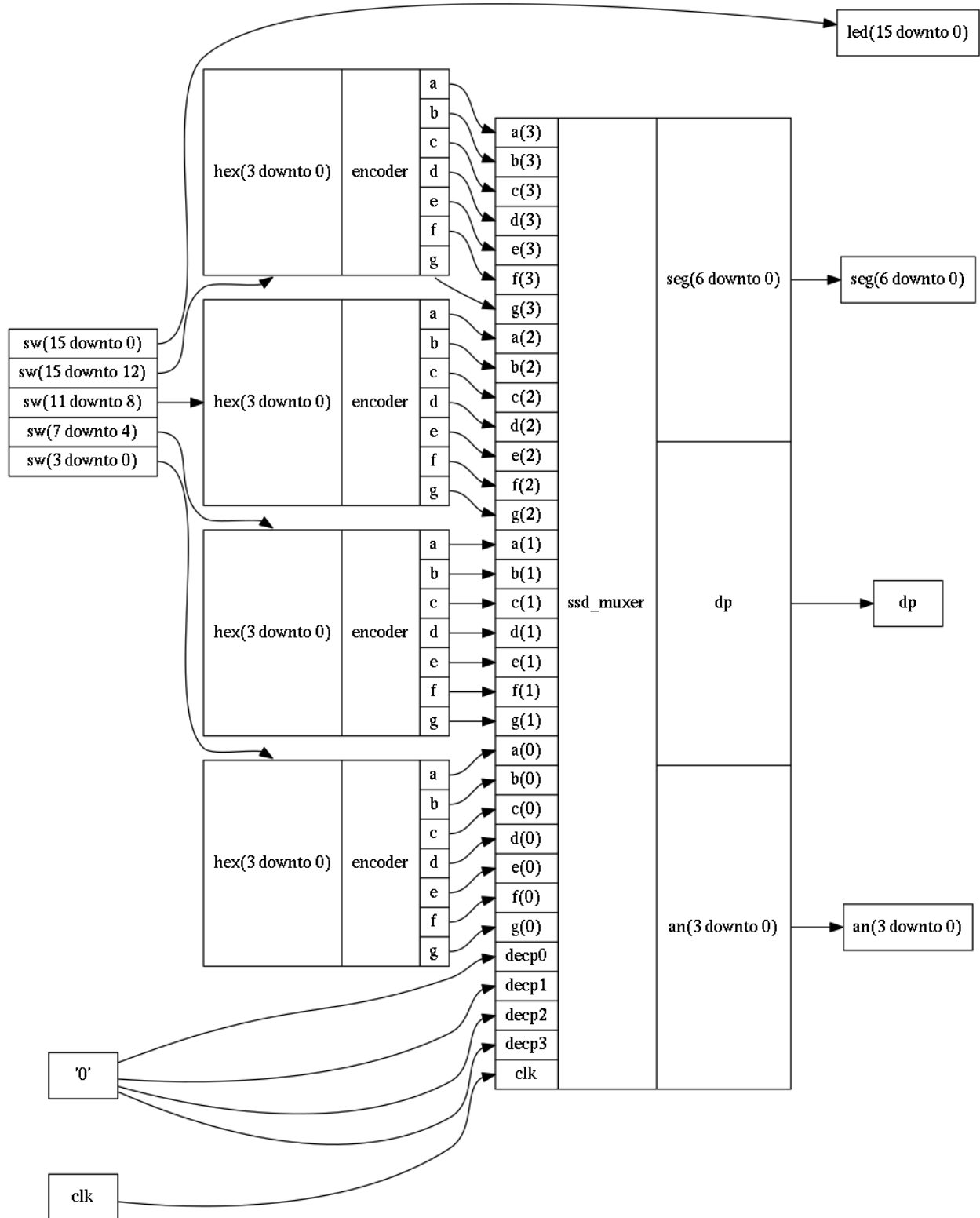


Figure 1: Overview of the CAD

Task

This task is to implement an encoder which input is a 1 4-bit hex signal and the output is 7 1-bit signals for each display segment (A-G).

1. Edit the constraints file and uncomment the lines corresponding with:
 - sw
 - clk
 - seg
 - dp
 - an
 - led
2. Create the top.vhd file:
 - a) Under “Flow Navigator” click “Add sources.”
 - b) Select “Create or add design sources.”
 - c) Create top.vhd and then click “Finish.”
 - d) Create inputs:
 - sw ~ (15 downto 0)
 - clk
 - e) Create outputs:
 - seg ~ (6 downto 0)
 - dp
 - an ~ (3 downto 0)
 - led ~ (15 downto 0)
3. Once you have your vhdl source file (top.vhd), edit it to implement 4 encoder entities and connect the switches to the LEDs.
 - Since you already have the encoder.vhd, you’ll just be using **structural VHDL** to connect multiple encoder entities to the `sdd_muxer` entity.
 - If you’re feeling adventurous you can use a *for generator* statement or you can manually place four encoder entities.
4. Make sure it’s syntax error free and can be synthesized.
5. Add the top_tb.vhd file:
 - a) Under “Flow Navigator” click “Add sources.”
 - b) Select “Create or add simulation sources.”
 - c) Add top_tb.vhd.
 - d) Make sure simulation set is “sim_1.”
 - e) Click “Finish.”
6. Edit `ssd_muxer.vhd`:
 - a) Go to line 57-58, there will be two `counter_max` constants.
 - b) Make sure the one that says `fpga` is commented out and the one that says `sim` is uncommented.
7. Select the right simulation set:
 - a) Under “Flow Navigator” click “Simulation Settings.”
 - b) Under “Simulation set” input “sim_1.”
 - c) Under “Simulation top module name” input “top_tb.”
 - d) Click “Ok.”
8. Run the simulation by and check your results. The expected output is on the last page of the assignment.
9. Once you’ve simulated the circuit correctly, you need to edit `ssd_muxer.vhd` again.

- a) Go to line 57-58, there will be two counter_max constants
 - b) Make sure the one that says fpga is uncommented out and the one that says sim is commented.
10. Create the file you need to load onto the FPGA (the bitstream). Click “Generate Bitstream” under the “Flow Navigator.”
11. Program the FPGAG board
- a) “Hardware Manager” in the Flow Navigator.
 - b) “Open Target” and then “Auto Connect.”
 - c) “Program Device” and then “xc7a35t_0.”
 - d) The defaults of the pop up are ok, just press program.

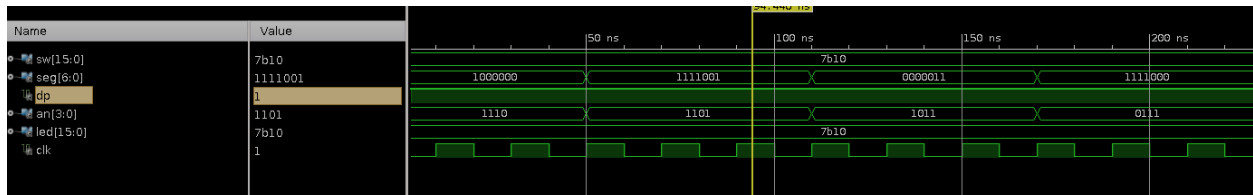


Figure 2: Expected Waveform of Simulation

Turn in

The write up is very simple, just include things like: the basic design process, anything you had trouble with, an explanation of anything that doesn't work, etc. It isn't meant to be extensive, but a brief overview.

Make sure everything is in a single PDF on Moodle. You can join PDFs with Adobe Acrobat if you have multiple PDFs. Name the assignment CAD3-xxxxxx-yyyyyy, where xxxxxx is the last name of one lab partner and yyyyyy is the last name of the other lab partner. Turn in only one pdf under one of the lab partner's Moodle account. Make sure everything is in a single PDF on moodle.

Grading

Write up	/4 Points
Printout and correctness of code (top.vhd and encoder.vhd)	/6 Points
Demonstration of FPGA	/10 Points
	/Total of 20 Points

Demonstration

The demonstration of the code is simple. Brian will be the TA you need to demonstrate the VHDL running on the FPGA. All that is needed is to go through some cycles of each set of 4 switches and show that the correct output is displayed on the seven segment displays. **Print off the demonstration page (see the website) when you go to demo the code on the FPGA.**

Demonstration Grading:

Each seven segment display works correctly	/2 points x 4
The switches are tied to the LEDs.	/2 points
	/Total of 10 points