Computer Assignment 4
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 CAD2. Connect it up to switches to a **signed** accumulator and display the output to the seven segment displays on the FPGA board. For the seven segment displays, you need to display a negative sign and the absolute value on the display. For example if you have X"FE", then on the display it needs to show "-02". You'll also need to turn in a single PDF to Moodle (instructions below).

Background

You need to declare the encoder.vhd, ssd_muxer and debounce components, then instantiate them like in the previous lab. In this lab you'll only use 8 of the switches. The IEEE numeric package will be useful in this lab. Simulation for this lab isn't required since you have to demo it, it's only to help you debug the code.

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
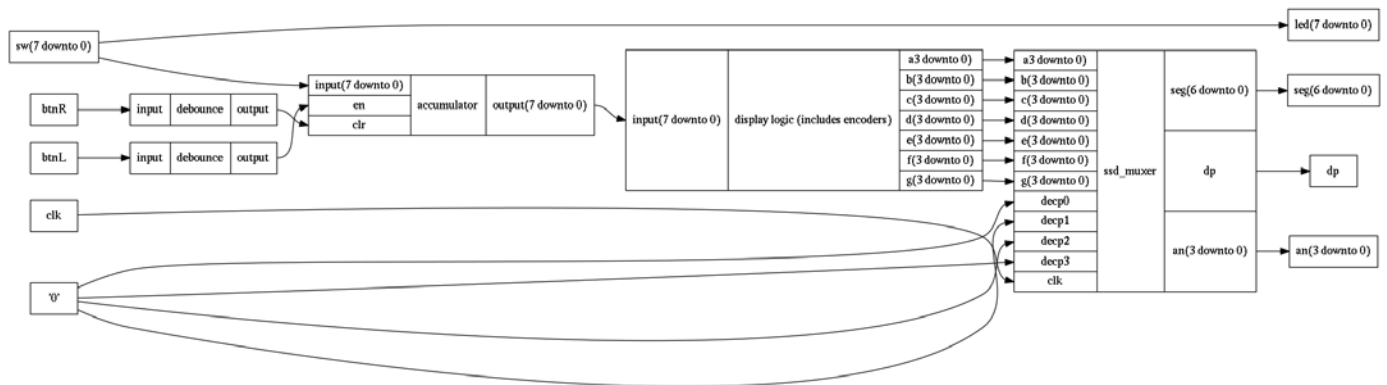https://www.digilentinc.com/Products/Detail.cfm?NavPath=2,66,828&Prod=ADEPT2

Figure 1: Overview of the CAD (note, accumulator and display logic aren't necessarily components, but just parts of the project.

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_4**.
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

Task

This task is to implement a 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 (7 down to 0)
   - btnR
   - btnL
   - clk
   - seg
   - dp
   - an
   - led (7 downto 0)
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 ~ (7 downto 0)
      - btnR
      - btnL
      - clk
   e) Create outputs:
      - seg ~ (6 downto 0)
      - dp
      - an ~ (3 downto 0)

- led ~ (7 downto 0)

3. Once you have your vhdl source file (top.vhd), edit it to implement the encoder, ssd_muxer and debounce instantiations.
    - Since you already have the encoder.vhd, you'll just be using **structural VHDL** to use logic inside of it.
    - .Likewise with ssd_muxer and debounce, you will use structural VHDL to use these components.
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. Remove the debounces and connect the btrR and btrL signals to their corresponding signals,
8. 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."
9. Run the simulation by and check your results. The expected output is on the last page of the assignment.
10. 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.
11. Edit top.vhd and add your debouncers back in.
12. Create the file you need to load onto the FPGA (the bitstream). Click "Generate Bitstream" under the "Flow Navigator."
13. 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."
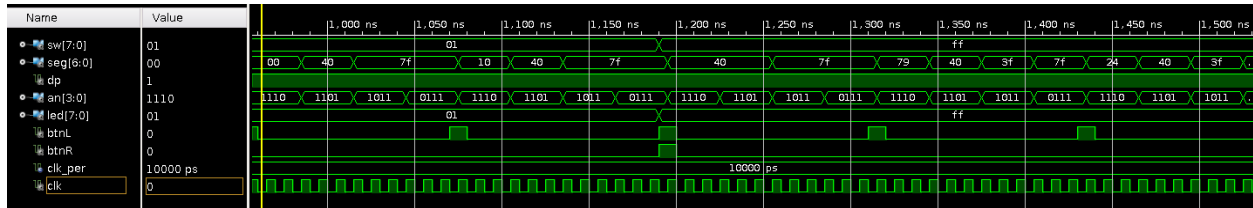   d) The defaults of the pop up are ok, just press program.

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.

Include top.vhd, encoder.vhd, or any code you wrote in the PDF.

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 CAD4-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	/8 Points
Printout and correctness of code	/12 Points
Demonstration of FPGA	/20 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 to. Fill out the demonstration sheet, print it and fill out the self-review before the TA demo. You need to be able to accumulate positive and negative numbers. For the seven segment displays, you need to display a negative sign and the absolute value on the display. For example if your accumulated value is X"FE", then on the display it needs to show "-02".

Demonstration Grading:

Accumulate positive numbers correctly	/6 points
Accumulate negative numbers correctly	/6 points
Seven segment display shows number correctly	/6 points
LEDs are connected to the SWs	/2 point
	/Total of 20 point