Computer Assignment 5

ECGR 2181 - Fall 2009

Assignment Overview

In this assignment you will use Xilinx ISE 7.1i and Diligent Adept ExPort to create a digital circuit and program it to the Nexys2 Spartan 3E FPGA. This uses some of the components built in Computer Assignment 4. If you have not completed CAD 4 you will need to refer back to its assignment for instructions. In CAD 5 you will be creating a game called "Binary Sequence". This is a two player game where Player 1 begins by setting the sliding switches up and down to generate a "sequence" and presses the "set_sw" switch. Player 1 then returns all the slide switches back to the down position and hands the board to Player 2. Player 2 then tries to guess the sequence by sliding the slide switches into position and pressing the "guess_sw" switch. Player 2 will then see between 0 and 8 LEDs light up to denote the number of switches in the correct position. For example, if 4 LEDs are lit, Player 2 will know that of the 8 slide switches, 4 are in the correct position. The LEDs do not indicate which slide switches are correct (obviously if it did, Player 2 would only ever have to make 2 guesses). Once Player 2 successfully guesses the sequence, 8 LEDs will remain lit until the "rst_sw" switch is pressed. Once the "rst_sw" is pressed, the game resets and is ready for Player 1 to set a new sequence.

Since this game is intended for 2 players, you are **STRONGLY ENCOURAGED** to partner with someone from the **SAME SECTION** (it can be someone you have worked with in the past).

Three additional VHDL Files (debounce.vhd, sequence_compare_vhdl.vhd, sequence_top_vhdl.vhd) have been provided which are available on the class website under the "Computer Labs" section. Demo the final circuit on the FPGA to one of the TAs during their Office Hours (in Woodward 200). Since there are a limited number of FPGAs we request that you only check out an FPGA right before you need to demo the working circuit. Use ModelSim to test the functionality of the circuit (saving you lots of hardware debugging time) then once the circuit works check out the FPGA from the ECE Shop. **START EARLY! There are fewer boards available than groups, so if you wait until the due date to start you might not get a board to test and demo with.**

In addition you will need to upload a **single PDF** of the listed files along with printing out and handing in the Grade Rubric. Print each page to a PDF file and use Adobe Acrobat Pro to combine the pages to a single PDF. Name the PDF <lastname_lastname_cad5>.pdf (schmidt_conrad_cad5.pdf). Upload the PDF to Moodle under Computer Assignment 5.

Computer Assignment 5 is due Wednesday, December 9th by 3 PM.

Getting Started

1. Create a new directory to store all of the Computer Assignment Files
    a. Make sure the directory does not contain any spaces!
2. Open **Xilinx ISE 7.1i** and Create a New Project called: **cad_5**
3. Create it for the **Nexys2 Board** (Spartan 3E, xc3s500e, fg320, -4)
4. Add 1 New Source File of type Schematic: **cad_5_top_sch**
    a. This will be the top level schematic for your design.
5. Under Add Existing Sources add the following sources from CAD 4:
    a. debounce.vhd
    b. d_latch_sch.sch
    c. d_flip_flop_sch.sch

d. register_1_bit_vhdl.vhd
e. register_8_bit_vhdl.vhd

6. Click “VHDL Design File” for all VHDL files imported in the previous step when prompted
7. Click Next and Finish to exit the New Project Wizard
8. Generate all Schematic Symbols for the newly added files
9. Download the “sequence_compare_vhd.vhd” and “sequence_top_vhdl.vhd” files from the Class Website and Add Copies of these files to your project. Each file contains a majority of the code; however, you will need to fill in the missing code. Read all of the comments carefully as they give hints as to how to complete the VHDL.

sequence_compare_vhdl

1. The Sequence Compare component consists of 3 functional pieces:
    i. Compare each bit of the *sequence input* to the *guess input*
    ii. Add the number of correct matches between the sequence and guess inputs
    iii. Output a ‘1’ for 0 to 8 LEDs based on the number of correct matches
2. You must figure out what kind of GATE is necessary to perform two bit equality comparison.
3. Once each bit has been compared the “eq_compare” register will contain a series of ones indicating whether that bit position of the guess input matches the sequence. The next step is to add all of the 1’s to find out how many bits are correct. This requires a VHDL function called “conv_integer” to convert each bit of eq_compare to an integer to be added together.
4. Based on the “sum_ones” signal use a case statement to output the equivalent number of ones to light up to corresponding LEDs. For example, if sum_ones = 4 the num_correct output should be “00001111” and if sum_ones = 6 the num_correct output should be “00111111”.

sequence_top_vhdl

1. The Sequence Top VHDL file contains the Finite State Machine (FSM) code to switch between Player 1 setting the sequence and Player 2 guessing the sequence. There are five state in the FSM (refer to notes taken during the recitation hour on the construction of the State Diagram):
    i. idle
    ii. wait_for_sequence
    iii. wait_for_guess
    iv. compare_sequences
    v. done
2. The file also contains the instantiation of the sequence_compare_vhdl.
3. The file also contains the 3 8-bit registers (seq_reg, guess_reg, led_reg). The registers are only allowed to be set under the following conditions:
    i. seq_reg: FSM is in the wait_for_sequence state and the set_sw is pressed.
    ii. guress_reg: FSM is in the wait_for_guess state and the guess_sw is pressed.
    iii. led_reg: FSM is in the compare_sequences state.
4. When Player 2 successfully guesses the sequence the FSM should stay in the done state until the reset switch is pressed.

cad_5_top_sch

1. The Top Level Component will have the following inputs and outputs:

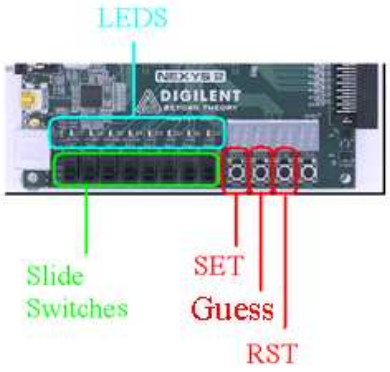
   a. Inputs: **set_sw**, **rst_sw, guess_sw, clk, d(7:0)**
      i. **d** will be the 8 input slide switches on the Nexys2 Spartan 3E board
   b. Outputs: **q(7:0)**
      i. **q** will be the 8 LEDs above the slide switches on the Nexys2 Spartan 3E board
2. Add the **sequence_top_sch** to the schematic
3. Create the Schematic Symbol for the debounce-behavorial (debounce.vhd) component
4. Add 3 debounce components and connect the 3 push button inputs (**set_sw**, **rst_sw**, **guess_sw)** to each debounce's **sw** input. Connect the **clk** to the debounce's **clk** input. Connect the **sig** output from each of the three debouncer's to the correct input on the sequence_top_sch. Leave **sglPulse** output from the debouncers unconnected.
5. Under the Process View Expand User Constraints and double click on Edit Constraints (Text)
6. Copy and Paste the following constraints. This is connecting the physical pins for the LEDs, Switches and Clock on the FPGA to the top level component's input and outputs. (You can copy CAD 4's User Constraints File (UCF) and edit it:

```
NET "clk"  LOC = "b8"  ;
NET "d<0>"  LOC = "G18"  ;
NET "d<1>"  LOC = "H18"  ;
NET "d<2>"  LOC = "K18"  ;
NET "d<3>"  LOC = "K17"  ;
NET "d<4>"  LOC = "L14"  ;
NET "d<5>"  LOC = "L13"  ;
NET "d<6>"  LOC = "N17"  ;
NET "d<7>"  LOC = "R17"  ;
NET "guess_sw"  LOC = "E18"  ;
NET "q<0>"  LOC = "J14"  ;
NET "q<1>"  LOC = "J15"  ;
NET "q<2>"  LOC = "K15"  ;
NET "q<3>"  LOC = "K14"  ;
NET "q<4>"  LOC = "E17"  ;
NET "q<5>"  LOC = "P15"  ;
NET "q<6>"  LOC = "F4"  ;
NET "q<7>"  LOC = "R4"  ;
NET "rst_sw"  LOC = "D18"  ;
NET "set_sw"  LOC = "H13"  ;
```

7. Double click "Synthesis – XST" - Synthesize the top level component and check for errors.
8. Double click "Implement Design" – The Tools determine where to put the logic on the FPGA
9. Double click "Generate Programming File" – Creates the bitstream needed to program the FPGA with for your design. The generated .bit file will be used by ExPort in the next step.

Programming the FPGA – Downloading the Design to the FPGA

1. Open Diligent Adept ExPort from the Start Menu (Refer to the Computer Assignment 0 Tutorial)
2. Plug in the Nexys2 Spartan 3E FPGA board into the USB port of the Computer (you will need to check out this board from the ECE Shop at this point).
3. Turn on the board if it isn't already on (there is a slide switch in upper left corner of the board)
4. In the program ExPort click the button "Initialize Chain" (PC identifies which FPGA is present)
5. Browse to your **.bit** file generated in the previous Step 9
6. Click "Yes" to the question of using "CCLK" instead of "JTAG CLK"
7. Click the Check Box next to the XCF04S ROM (Do not browse to find it)
8. Click the "Program Chain" Button to program the FPGA and Test Circuit



Demo

1. When you are ready to show the TA your code is correctly functioning on the FPGA find the TA to take the demo. The TA will test the design for various input and outputs.
2. The TA will ask questions regarding the Computer Assignment prior to accepting the grading rubric and signing off on it.

Required Electronic Submission (uploaded to Moodle)

Print the Grading Rubric (the last page of this hangout) and write both names and section number at the top. Then print the following schematics and VHDL files as individual PDFs. Combine the PDFs into a single PDF with both your last names as the filename (I.E. schmidt_conrad_cad4.pdf). Upload the single PDF (one upload per group) to Moodle. Finally, find one of the TAs **during their Office Hours** and demo.

1. VHDL of sequence_compare_vhdl
2. VHDL of sequence_top_vhdl
3. Schematic of cad_5_top_sch

Computer Assignment #5 - Grading Rubric

Student 1 Name: _____ Section Number: _____

Student 2 Name: _____

Uploaded correctly functioning schematics and VHDL to Moddle:

- VHDL of sequence_compare_vhd _____ / 5 Points
- VHDL of sequence_top_vhdl _____ / 5 Points
- Schematic of cad_5_top_sch _____ / 5 Points

Demo on the FPGA:

- Demo of Circuit on FPGA _____ / 15 Points

Total _____ **/ 30 Points**