Computer Assignment 3

ECGR 2181 - Fall 2009

Assignment Overview

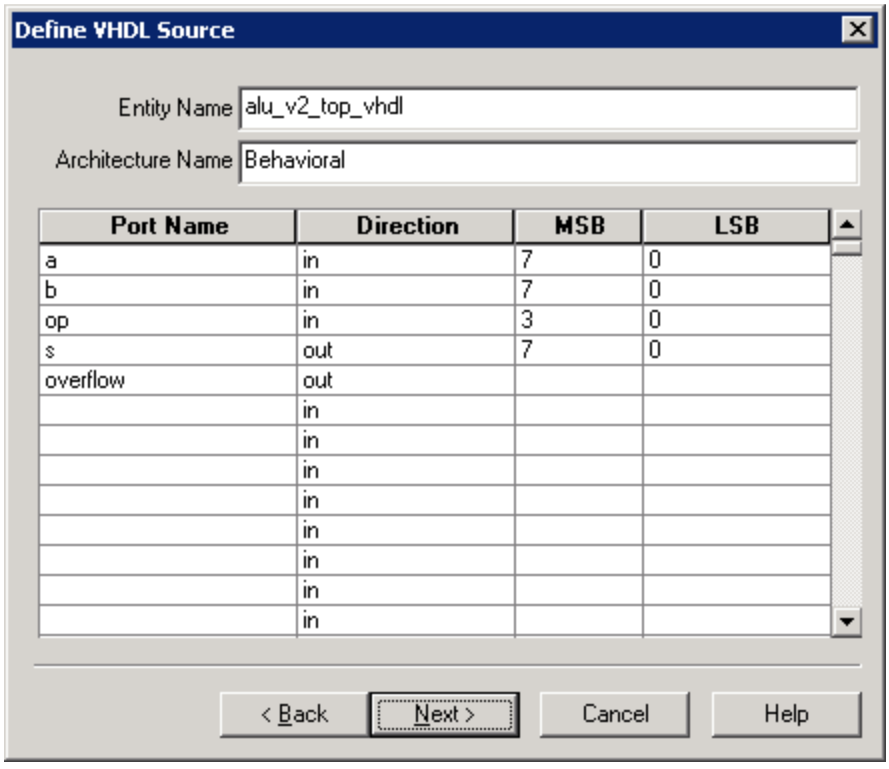
In this assignment you will use Xilinx ISE 7.1i and ModelSim to augment the ALU created in Computer Assignment 2. The modifications will be based on homework assignment 9. Refer to Homework 9 for the operations the ALU must be able to perform. Demo the final simulation to one of the TAs during their Office Hours (in Woodward 200). **You are allowed to work in groups of 2 as long as you are both in the same section.** Make sure you both understand everything, when you demo the final solution, the TA will ask you both questions regarding the system to make sure you both did the work.

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 five pages to a single PDF. Name the PDF <lastname_lastname_cad3>.pdf (schmidt_conrad_cad3.pdf). Upload the PDF to Moodle under Computer Assignment 3.

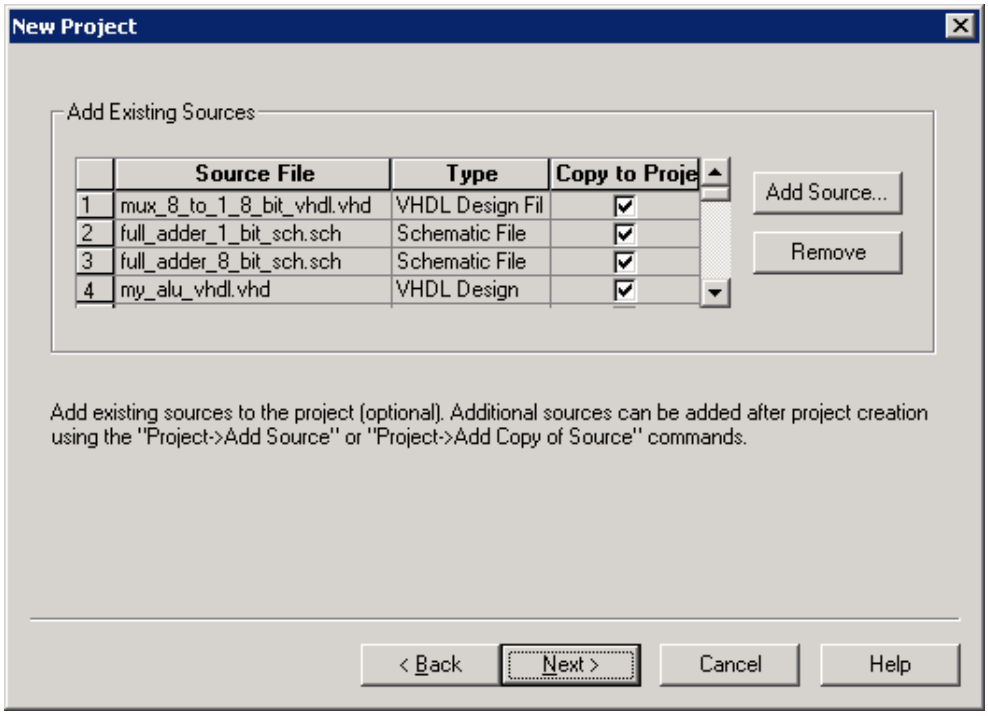
Computer Assignment 3 is due Wednesday, November 11th by 3 PM.

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_3**
3. Create it for the **Nexys2 Board** (Spartan 3E, xc3s500e, fg320, -4)
4. Add 1 New Source File of type VHDL Module: **alu_v2_top_vhdl**
    a. This will be the top level design for your new Version 2.0 Arithmetic Logic Unit. At the end we will describe what code is necessary in this file. For now simply set the Input and Output ports to the following:



5. Click Next then Finish to complete the New VHDL Wizard
6. Under Add Existing Sources add the four source files from Computer Assignment 2:
    i. my_alu_vhdl.vhd
    ii. mux_8_to_1_8_bit_vhdl.vhd
    iii. full_adder_1_bit_sch.sch
    iv. full_adder_8_bit_sch.sch



7. Click Next and Finish to complete the New Project Wizard
8. Module View click on: **full_adder_1_bit_sch** then run **Create Schematic Symbol**
    a. This regenerates the Schematic Symbol for the 1-bit Full Adder
9. In the Module View click on: **xc3s500e-4fg320** then run **Update All Schematic Files**
    a. This updates all files to use the latest schematic symbols (I.E. the 8-bit Full Adder)
10. Add 3 new VHDL files, one for each of the following components:
    a. max_comparator_8_bit_vhdl
        i. a : in std_logic_vector(7 downto 0)
        ii. b : in std_logic_vector(7 downto 0)
        iii. max : out std_logic_vector(7 downto 0)
    b. min_comparator_8_bit_vhdl
        i. a : in std_logic_vector(7 downto 0)
        ii. b : in std_logic_vector(7 downto 0)
        iii. min : out std_logic_vector(7 downto 0)
    c. mux_4_to_1_8_bit_vhdl
        i. i0 – i3 : in std_logic_vector(7 downto 0)
        ii. s : in std_logic_vector(1 downto 0)
        iii. f : out std_logic_vector(7 downto 0)

mux_4_to_1_8_bit_vhdl

1. Using the 8 to 1 8-bit multiplexer as a template create the 4 to 1 multiplexer
    a. (Hint you can copy and paste a lot of the existing code from the 8 to 1 multiplexer, just remember what changes between an 8 to 1 and 4 to 1 multiplexer)
2. Synthesize the **mux_4_to_1_8_bit_vhdl** component, fixing all syntax errors

max_comparator_8_bit_vhdl

1. Create a process called: **max_output** with the sensitivity list of **(a, b)** in the **max_comparator_8_bit_vhdl** file.
2. Add the following code to the **max_output** process:

```
if (signed(a) > signed(b)) then
        max <= a;
else
        max <= b;
end if;
```

3. The code above compares the signed A input (which is a 2's Complement binary number) to the signed B input and sets the max output to A when the condition is true (A is greater than B). This is the power of VHDL, representing the expression of a Magnitude Comparator and a Mux in 5 lines of code!
4. Synthesize the **max_comparator_8_bit_vhdl** component, fixing all syntax errors

min_comparator_8_bit_vhdl

1. Follow the 4-steps above for the **max_comparator_8_bit_vhdl**, replacing the max logic with the correct min logic.

alu_v2_top_vhdl

1. Add the following Signal and Component Declarations to the **alu_v2_top_vhdl**. Do not re-add what you see at line 38 and 74 below; those two lines should already exist. You add the signal and component declarations between the architecture and the begin sections of the VHDL file.

```
38  architecture Behavioral of alu_v2_top_vhdl is
39     -- Internal Signals used to connect the Min/Max outputs and the
40     -- Computer Assignment 2 ALU (my_alu_vhdl) output to the 4:1 Mux
41     signal max_out : std_logic_vector(7 downto 0);
42     signal min_out : std_logic_vector(7 downto 0);
43     signal alu_out : std_logic_vector(7 downto 0);
44     -- Declaration my_alu_vhdl (Computer Assignment 2 ALU)
45     component my_alu_vhdl is
46     port ( a        : in std_logic_vector(7 downto 0);
47            b        : in std_logic_vector(7 downto 0);
48            op       : in std_logic_vector(2 downto 0);
49            sum      : out std_logic_vector(7 downto 0);
50            overflow : out std_logic);
51     end component my_alu_vhdl;
52     -- Declaration of max_comparator_8_bit_vhdl Component
53     component max_comparator_8_bit_vhdl is
54     port ( a        : in std_logic_vector(7 downto 0);
55            b        : in std_logic_vector(7 downto 0);
56            max      : out std_logic_vector(7 downto 0));
57     end component max_comparator_8_bit_vhdl;
58     -- Declaration of max_comparator_8_bit_vhdl Component
59     component min_comparator_8_bit_vhdl is
60     port ( a        : in std_logic_vector(7 downto 0);
61            b        : in std_logic_vector(7 downto 0);
62            min      : out std_logic_vector(7 downto 0));
63     end component min_comparator_8_bit_vhdl;
64     -- Declaration of mux_4_to_1_8_bit_vhdl Component
65     component mux_4_to_1_8_bit_vhdl is
66      port ( i0 : in std_logic_vector(7 downto 0);
67             i1 : in std_logic_vector(7 downto 0);
68             i2 : in std_logic_vector(7 downto 0);
69             i3 : in std_logic_vector(7 downto 0);
70             s  : in std_logic_vector(1 downto 0);
71             f  : out std_logic_vector(7 downto 0));
72     end component mux_4_to_1_8_bit_vhdl;
73
74  begin
```

2. Add the following code for each component, filling in the correct signals.

```
74  begin
75     -- Instantiate 4 to 1 Mux
76     mux_4_to_1_8_bit_vhdl_i : mux_4_to_1_8_bit_vhdl
77     port map (
78        i0 => ,
79        i1 => ,
80        i2 => ,
81        i3 => ,
82        s  => ,
83        f  => );
84     -- Instantiate ALU from Computer Assignment 2
85     my_alu_vhdl_i : my_alu_vhdl
86     port map (
87        a => ,
88        b => ,
89        op => ,
90        sum => alu_out,
91        overflow => );
92     -- Instanitate Max Comparator
93     max_comparator_8_bit_vhdl_i : max_comparator_8_bit_vhdl
94     port map (
95        a => ,
96        b => ,
97        max => max_out);
98     -- Instanitate Min Comparator
99     min_comparator_8_bit_vhdl_i : min_comparator_8_bit_vhdl
100    port map (
101       a => ,
102       b => ,
103       min => min_out);
104
105 end Behavioral;
```

Simulation and Demo

1. Add a New Source of type Test Bench Waveform to the project: **alu_v2_tb**
2. Set the Test Bench to show the Functionality of the V2 ALU. During the Demo you should show the TA a Test Bench that at least covers the Computer Assignment 2 Test Bench PLUS testing your Min and Max Comparators. There is no sample Test Bench or Waveform provided, it is up to your group to prove during your Demo that your ALU is fully functional.

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 four new VHDL files and the waveform of your schematic (from ModelSim) as individual PDFs. Combine the five PDFs into a single PDF with both your last names as the filename (I.E. schmidt_conrad_cad3.pdf). Upload the single PDF (one upload per group) to Moodle. Finally, find one of the TAs **during their Office Hours** and demo the outputs of both ModelSim simulations.

1. VHDL of ALU V2
2. VHDL of 4 to 1 8-bit Multiplexer
3. VHDL of Max Comparator
4. VHDL of Min Comparator
5. Waveform from ModelSim of Test Bench

Notes

You do NOT need to checkout a Nexys2 Board for this Computer Assignment. You are only using the software tools to build and test the designs. Future assignments will rely upon these components so make sure to keep a copy of your project's directory.

If you run into a License Error during Simulation save all files and close Xilinx ISE 7.1i. The open Xilinx ISE 7.1i from the Start Menu:

Start -> Mosaic XP -> Engineering -> Electrical -> Xilinx Webpack -> Xilinx ISE 7.1i

If you run into a ModelSim "read only" error on your Test Bench VHW file:

a. To avoid getting any errors with ModelSim in "read mode" you MUST create a new project directory WITHOUT any spaces! Make sure the path to the directory also does not have any spaces. If you are not sure where to save this project to, create the following directory on your desktop:
    iv. ecgr2181
    v. Then within this directory add a new directory for each project
    vi. The Path to this directory will now be: **U:\pc\win_data\Desktop\ecgr2181**

Computer Assignment #3 - Grading Rubric

Student 1 Name: _____ Section Number: _____

Student 2 Name: _____

Uploaded schematics and VHDL as a Single PDF to Moddle:

- VHDL of ALU V2 _____ / 10 Points
- VHDL of 4 to 1 8-bit Multiplexer _____ / 2 Points
- VHDL of Max Comparator _____ / 2 Points
- VHDL of Min Comparator _____ / 2 Points
- Waveform from ModelSim of Test Bench _____ / 4 Points

Demo of the simulations:

- Simulation of ALU V3 in ModelSim _____ / 10 Points

Total _____ **/ 30 Points**