EE 2700
Digital Circuits

Lab 7 – Programming Logic Devices from VHDL

Objective: To gain experience designing with CPLDs using VHDL code.

Discussion: It this lab, we will use ISE to synthesize a circuit design for a programmable logic device (in this case, a Xilinx XC2C64A CPLD). We will start with a description of the design in VHDL and simulate it functionally. Once the simulation yields the correct results, we will synthesize the design. In this case, synthesis is a process that reduces the VHDL code into a JEDEC file containing all the information necessary to program the XC2C64A. We will then use Adept (another application) to program the CPLD.

Preparation: Write the title and a short description of this lab in your lab book. Make sure the page is numbered and make an entry in the table of contents for this lab.

Design a 4-bit full adder in using VHDL. A full adder has a carry-in and a carry-out. Call the carry-in *ci*, the buses for the inputs to the adder *a* and *b* (of type std_logic_vector(3 downto 0)), the carry-out *co*, and the sum *s* (also of type std_logic_vector(3 downto 0)). Write a behavioral model for the adder.

Test your design by simulating it for all possible input combinations (there are 512 in all). Design your test fixture to automatically test all the outputs for each input combination. Use something like the code below in your test fixture:

```
process
   variable test: unsigned(8 downto 0);
   variable sum: unsigned(4 downto 0);
begin
   for i in 0 to 511 loop
      test := to_unsigned(i,9);
      A <= std_logic_vector(test(3 downto 0));
      B <= std_logic_vector(test(7 downto 4));
      ci <= test(8);
      wait for 2 ns;
      sum := to_unsigned(i mod 16 + (i/16) mod 16 + i/256, 5);
      assert S = std_logic_vector(sum(3 downto 0)) report "bad sum";
      assert co = sum(4) report "bad carry-out";
   end loop;
   wait;
end process;
```

Manually check at least 10 input combinations. (Include cases with large and small input values and cases with and without carry in.) Record these test cases in your lab book. Also affix a copy of your VHDL code and an excerpt from your simulation (not all of it).

Once you have a working simulation, you will want to synthesize your design. To do this, you will need to create a constraints file to assign signals to pins. Select the “Implementation” radio button then create an “Implementation Constraints” file using the “New Source” menu option. Associate pins to the inputs and outputs by adding the following lines to the constraints file:

```
NET "CI" LOC = "P18" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;

NET "A<0>" LOC = "P20" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "A<1>" LOC = "P21" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "A<2>" LOC = "P22" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "A<3>" LOC = "P23" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;

NET "B<0>" LOC = "P27" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "B<1>" LOC = "P28" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "B<2>" LOC = "P29" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "B<3>" LOC = "P30" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;

NET "S<0>" LOC = "P31" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "S<1>" LOC = "P32" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "S<2>" LOC = "P33" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
NET "S<3>" LOC = "P34" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;

NET "CO" LOC = "P36" | IOSTANDARD = LVCMOS33 | SLEW = SLOW ;
```

In the process window, double-click “Generate Programming File (this will cause ISE to synthesize the design. Address any errors and repeat if necessary.

If you are using your own laptop, it is recommended that you download and install Adept. This program works with the programming cables we will be using in the labs. The application can be downloaded from:

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

Parts: 1-XC2C64A Breakout Board
Input switches from previous labs
5-LEDs with current-limiting resistors (e.g. 220 ohms)

Procedure: Solder pins into the pads on the breakout board for V+, ground, and the inputs/outputs you are using. (Put the pins in your prototype board then set the breakout board on top of them so that the pins are straight when you solder them.)

Attach 5V to the V+ input (and ground to ground). **IMPORTANT: DO NOT attach 5V to any pin other than V+, otherwise you will almost certainly destroy the CPLD.**

Connect the JTAG HS-1 cable to the breakout board (make sure the labels line up). Connect the USB end of the cable to a computer that has Adept installed.

Use Adept to program the CPLD on the breakout board. Remove power from the board. Attach nine input switches to the inputs (the button on the breakout board can serve as carry-in if, in ISE, you right-click on the “Fit” process, select “Process Properties” and set options –unused and –terminate both to Pullup.). Attach 5 LEDs (with resistors) to the outputs.

Turn the power back on and use the test cases you recorded in your lab book to verify the 4-bit full adder functions properly. Demonstrate your circuit to your lab instructor.

Draw a schematic diagram that shows the CPLD. Label each line with a net name (e.g. A_3 or Ci) and a pin number. Label the whole CPLD with a part number and designator.

Caution: **The XC2C64A is sensitive to static electricity which can be generated by improper handling. Keep it in a conductive plastic bag when it is not in use. Unfortunately, it is not only sensitive but expensive.**

Signoff: A lab score can only be given if the circuit is functional.

Rubric (10 points total)

- Lab book is clearly legible and has a title and description. (1 point)
- Lab book contains VHDL code for the adder (1 point)
- Lab book contains simulation results (1 points)
- Lab book contains test results (1 point)
- Lab book contains a schematic with pins labeled (1 point)
- Lab book contains no obliterations. (1 point)
- Lab book contains a signed, dated summary (1 point)
- Each used page has a number and is initialed* and dated* (1 point)
- The circuit is functional before the end of the lab period. (2 points)

Note: If the circuit is working at the end of the lab period but the lab book is not yet complete, the lab can be signed off as “working”, and no late penalty will be assessed if it is graded on or before the next lab period.

* It is not necessary to initial and date a page that contains a signature and date unless the dates are different.