Chapter 5

Modular Combinational Logic Using HDL

5.1 Section Overview

The schematic capture feature of Xilinx is frustrating. Using a HDL (hardware description language), such as Verilog, avoids lots of the small bugs that make Xilinx frustrating, but the key to using HDL is creating a solid design before even starting to use Xilinx.

5.2 Objectives

In this section, the following items will be covered:

1. Step by step requirements for drafting a complete digital logic design on paper
2. An introduction to using Verilog
3. Using the schematic viewer to inspect the synthesized Verilog description

5.3 Materials

1. Xilinx ISE 12 software (Currently installed on the lab computers)
2. Digital Logic Board (d_logic.2 board)
3. Small breadboard and resistors
4. A working TekBot with the analog control board
5. The ECE 271 textbook, Digital Design and Computer Architecture by David and Sarah Harris

5.4 Procedure

There are 5 steps to digital logic design:

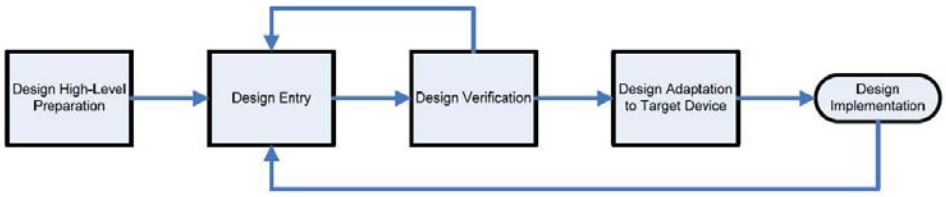

Figure 5.1: Use this process for designing an intelligent remote.

1. *High Level Preparation:* The custom remote control, Section 3, gave the TekBot an ability to understand commands. This section will use the analog control board to override the commands from the buttons and force the TekBot to automatically back up and turn when it runs into an object. Essentially the remote control should only work when the analog control board is outputting a 'forward' signal.

   Make a block diagram

   Label all Port and Port pins that are used for this project on the block diagram.

Label which inputs on the CPLD are connected to the analog control board LeftDirection and RightDirection, Al and Ar. Look at page 3 of the CPLD data sheet on the lab website to find the tolerances (V_{IL}, V_{IH}, and $V_{INAbsoluteMax}$ for the voltages going into Al and Ar.

Label which outputs on the CPLD are connected to Le, Ld, Re, and Rd.

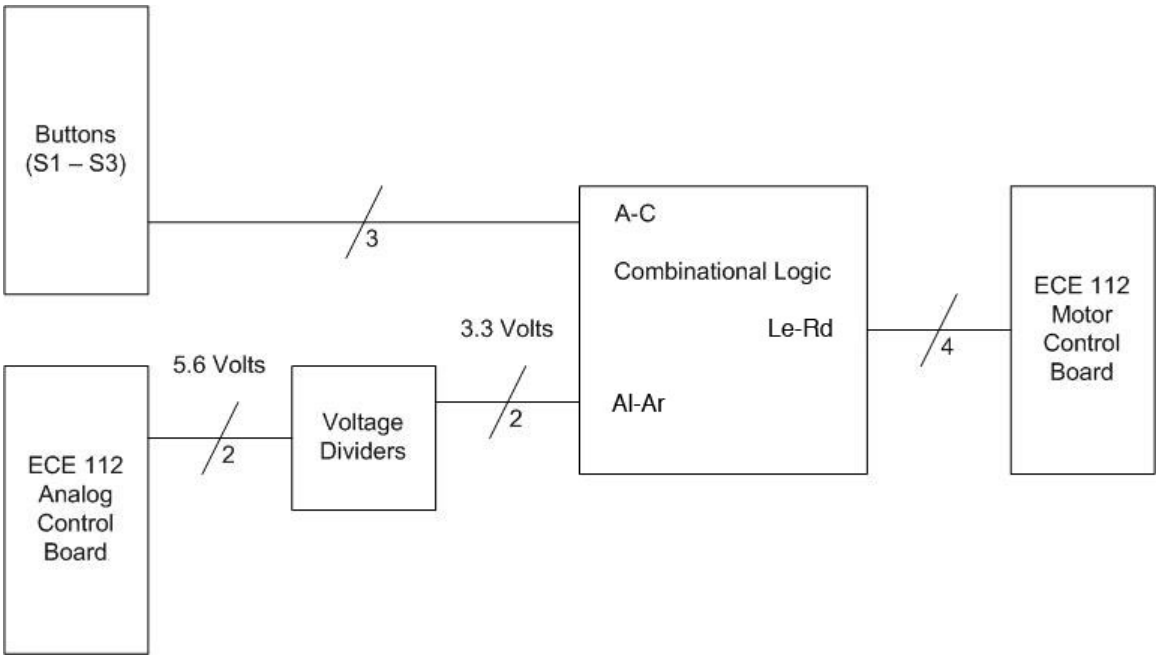

Figure 5.2: This is the block diagram for the smart remote control

Make a functional truth table

The TekBot should operate according to the functional table below:

Controller	Inputs: Al, Ar, A, B, C	$TekBotAction$	Outputs: Le, Ld, Re, Rd
AnalogControl	$00xxx$	Automatic Reverse	
AnalogControl	$01xxx$	Automatic Turn	
AnalogControl	$10xxx$	Automatic Turn	
Section3	11000	Section3 Command	
Section3	11001	Section3 Command	
Section3	11010	Section3 Command	
Section3	11011	Section3 Command	
Section3	11100	Section3 Command	
Section3	11101	Section3 Command	
Section3	11110	Section3 Command	
Section3	11111	Section3 Command	

Plan the design:
Figure 5.3 has the design needed for this section. There are three types of modules, Section 3, an AND gate, and 4 multiplexers. There are six instances that all need unique names that obey the Verilog naming conventions [1]. There are 14 nets used inside of Section 5. Assign each net a unique name.

[1] http://www.xilinx.com/itp/xilinx4/data/docs/dev/ngd2ver10.html

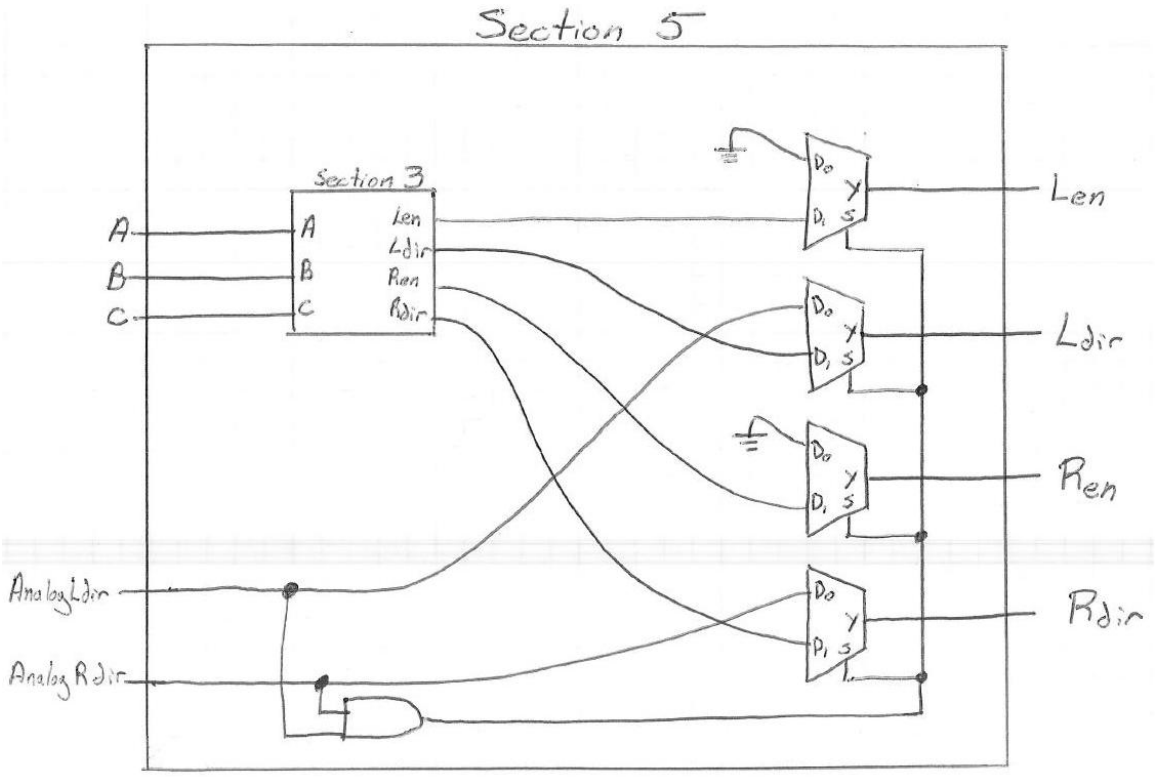

Figure 5.3: This is the schematic for the smart remote control

2. *Design Entry:*

   Create a new project. Create a new source with Verilog module as the source type. After selecting the Verilog module a new prompt will appear. Add the 5 inputs and 4 outputs for Section 5 in this prompt. This is the same as setting up the I/O markers in the schematic capture view.

   Add the source from Section 3 to the project.

   Double click on the .v source and fill in the comments completely. Make up your own company and put your name on the engineer field. This source depends on the Section 3 source, so add that as a dependent. The module inputs and outputs for Section 5 are defined in the ( )'s.

   Click on the Section 3 source, open up the Design Utilities option, and double click on View HDL Instantiation Template.

   Copy and paste the template into Section A of Figure 5.4. Change UUT (unit under test) into the planned instance name for this instance. Write the name of the net that connects to each port of the Section 3 instance into the ( )'s.

   Turn to page 175 of the textbook. There is a model for a mux that has two 4 bit buses entering as inputs and one 4 bit bus as an output. Modify this module and place the module definition in Section B of Figure 5.4.

   Save the file and note that a new source (the recently defined mux) has appeared in the sources window.

   Add the line *supply0 GND;* into Section A. This will create a GND net to be used for grounding the two inputs of the multiplexers.

   Copy and paste four of the instantiation templates for the mux into Section A of Figure 5.4 and change the instance name to the one you chose for each mux.

   Look at page 173 and figure out how to make an AND gate in section A of Figure 5.4.

   (Hint: This will not use a separate module.)

Select the TopLevel source for this project and click Implement Design ⇒ Synthesize ⇒ View RTL Schematic.

Open the signals folder, select all of the signals, and then click Create Schematic.

If changes are made to the design after viewing the RTL Schematic, they may not be seen after the RTL Schematic is re-created. The RTL Schematic might not refresh itself without Xilinx being closed and re-opened.

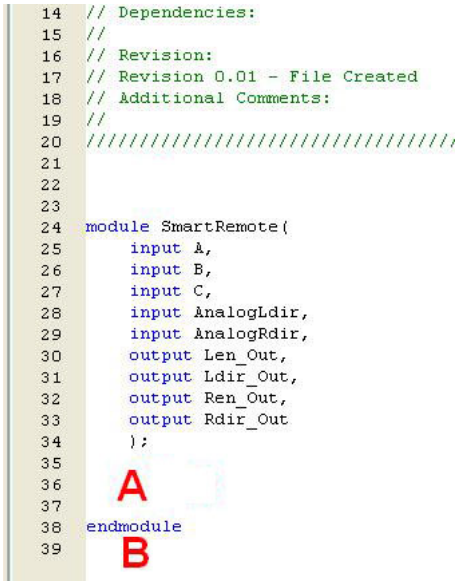

Figure 5.4: This is the screen for a Verilog source in Xilinx Webpack.

3. *Design Verification:* Follow the same process for using the simulator as in Section 3.
4. *Design Adaptation to Target Device:* Follow the same process for using PACE and iMPACT as in Section 2.
5. *Design Implementation:* Program the Xilinx CPLD with the Universal Programmer software.

TA Signature: _____
(TekBot functions properly & work is displayed)

5.5 Study Questions

1. Include a detailed block diagram of Section 5, the RTL Schematic with the blocks compressed, the RTL Schematic with the blocks expanded, and a copy of the Verilog source.
2. Discuss two advantages and two disadvantages to using HDL versus schematic capture from previous sections.
3. A standard 2 input multiplexer is built using 3 gates, why are two of the multiplexers in this design built using only 1 gate?
4. Look at page 198 of the textbook. Write the Verilog equivalent of Section 4, the seven-segment display encoder. Include A - F outputs. Explain the benefit of using HDL instead of schematic capture for this application.

5.6 Challenge - Extra Credit

Make a self-checking test bench to verify correct operation of Section 5. Look at page 215 of the textbook for more information. For full credit turn in a copy of the verilog source with a written explanation of how it works.

