Chapter 6

Finite State Machine Using HDL

6.1 Section Overview

Section 5 used an analog controller to control the timing of a TekBot that automatically backs up, and spins away from the object that bumped it. This section uses a finite state machine to control the timing of backing up and spinning away from the obstacle.

6.2 Objectives

In this section, the following items will be covered:

1. Step by step requirements for drafting a complete finite state machine design on paper.
2. Using HDL examples from 4.32 in the textbook to describe a finite state machine.

6.3 Materials

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

6.4 Procedure

There are 5 steps to digital logic design:

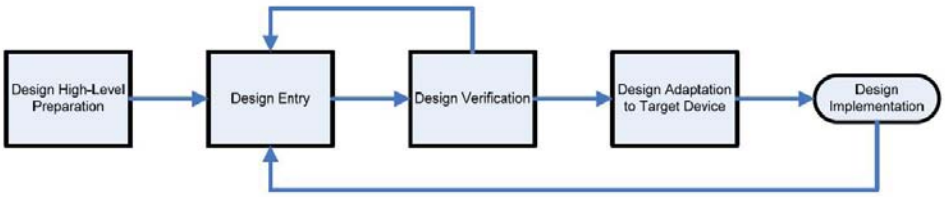

Figure 6.1: Use this process for designing an intelligent remote.

1. *High Level Preparation:* Section 6 is going to digitally reproduce the analog control board.

   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 right and left bumper of the sensor board. Are these going to be active high or active low? Do they need pull-up or pull-down resistors?

   Label which outputs on the CPLD are connected to right enable, right direction, left enable, and left direction of the motor control board.

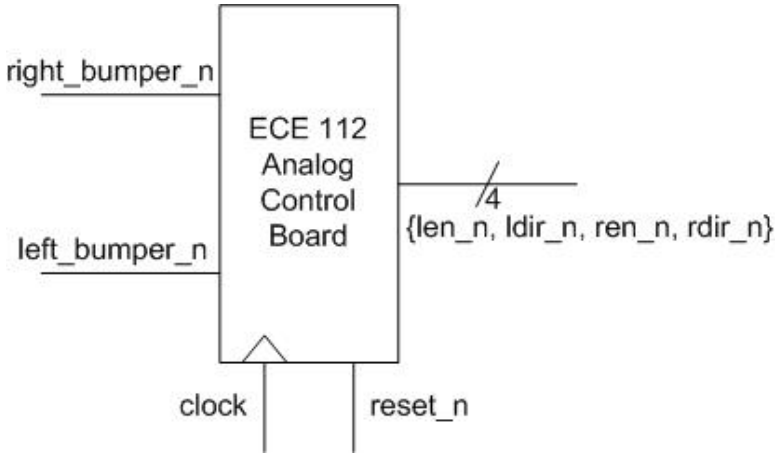

Figure 6.2: This digital block will replace the analog control board

Make a state diagram on paper

The state diagram for this project is shown in Figure 6.3.

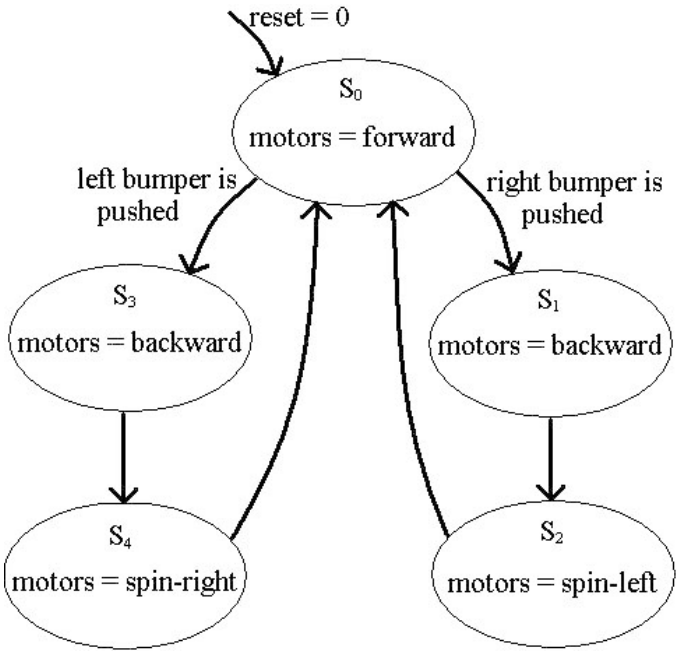

Figure 6.3: This is the state diagram for the digital version of the analog control board

Label each state with a digital encoding. How many bits are needed for 5 seperate states?

Label the outputs for each state. This is going to be a Moore State Machine, so the outputs depend only on the current state.

2. *Design Entry:*

   Find some templates: Use HDL Example 4.32 on page 209 of the ECE 271 textbook as a guide for Section 6. Use a decoder (on page 199 of the text) to decode the current state and assign the motor controller outputs.

Add a clock divider: Download the clock_divider.v file from the lab website. Add it as a source, instantiate it, and connect it to the state machine to correctly set an appropriate clock frequency to drive the state machine.

3. *Design Verification:* Follow the same process for simulation as in Section 3.

Because of the extra time that the Tekbot is given to back up, you will need to change the clock divider frequency back to its original state for simulations. This change will allow you to view all possible states in the simulation timeframe. Be sure to change to frequency back once the design is verified.

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.

6.5 Study Questions

1. Include a detailed block diagram of Section 6, the RTL Schematic with the blocks compressed, the RTL Schematic with the blocks expanded, and a copy of the Verilog source.
2. Explain two advantages of using a digital equivalent of the analog control board.
3. Explain two disadvantages of using a digital equivalent of the analog control board.

6.6 Challenge - Extra Credit

Make a state machine that doesn't use the same timing for every action. Make it back up for half a second and turn for a quarter second. There are lots of ways to do this, so think of several options and select the easiest. For full points implement this on your tekbot, print out the verilog source, and explain the method you used.

