

Chapter 7

Final Design Project

7.1 Section Overview

It is very important to be able to actually use the concepts learned in this course to create real systems and designs. This lab presents a design problem that will use the information and selected aspects gathered from previous labs.

7.2 Objective

Build a remote control robot that will protect itself and provide the user with information about what it is doing. Combine these previous designs together.

1. Section 6, A state machine to enable the TekBot to back up and turn after bumping into an object.
2. Section 5, An intelligent remote that chooses between a remote control and a state machine.
3. Section 4, A seven segment display to indicate the direction the motors are going.
(Example: If the output for going forward is 0101, the seven segment will display a 5.)

7.3 Materials

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

7.4 Procedure

There are 5 steps to digital logic design:

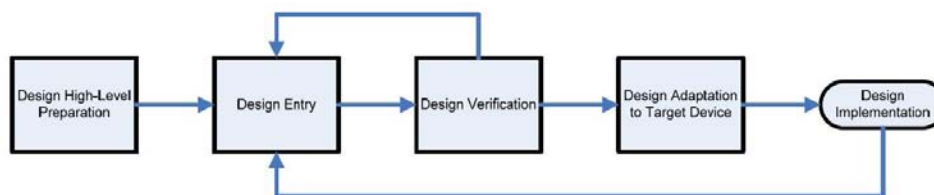


Figure 7.1: Use this process for designing the final project.

1. *High Level Preparation:* Section 7 combines the previous designs of the remote control robot, seven segment display, and the bumpbot. The robot will respond to the remote control until it runs into an obstacle, in which case its onboard obstacle avoidance system will take over (in order to keep the robot from being damaged). Once the robot has avoided the object successfully, it should return to doing what it was doing before the emergency systems were activated. The remote will then be controllable again. The seven segment display will display a number for the direction the robot is traveling. Perform this task using only d.logic boards. No analog controllers are allowed for this project.

A more difficult method: A decoder could be made to enable the seven segment display to show the direction of each motor instead of numbers. One example of this would be to have the middle segment turn on

when the TekBot is stopped, the front segment turn on when it going forwards, and the other segments light up depending on the other directions the TekBot can go.

Make a block diagram: Do this on a separate piece of paper to be turned in.

2. *Design Entry:* No new modules should need to be created for this project. Connect Sections 4, 5, and 6 together to finish Section 7.

If any modules have a bus as an input or output that needs to be separated into wires for each bit or several wires that need to be combined into a bus for another input, complete the following steps.

- (a) A new bus must be instantiated with the same number of bits as the bus in question.
- (b) This bus can be separated by making a wire for each bit and setting that wire equal to the bus at a certain bit.

Example: There is a bus named bus, the wire named wire, and you want bit 0.

```
wire = bus[0];
```

- (c) The wires created from a single bus can be used as an input for another module.
- (d) To combine separate wires into one bus is the opposite.

Example: To store a wire named wire into the 0th bit of a bus named bus.

```
bus[0] = wire;
```

- (e) The bus created from separate wires can be used as an input for another module.

3. *Design Verification:* Follow the same process for simulation 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 & lab work is displayed)

7.5 Study Questions

1. Include a detailed block diagram of Section 7, the RTL Schematic with the blocks compressed, the RTL Schematic with the blocks expanded, and a copy of the Verilog source.
2. What was the toughest aspect of ECE 272? What should be changed or added to the ECE 272 manual to make this course better?
3. What would you like to explore further about Xilinx or Digital Logic Design?
4. What section of ECE 272 did you dislike the most? Why?
5. What was your favorite section of ECE 272? Why?

7.6 Challenge - Extra Credit

Enjoy your break!