Lab 6: SMART HOME PROJECT

The University of Washington | The Remote Hub Lab | Last Revised: March 2022

Summary

This lab is the culmination of the Intro to Digital Logic Lab series. In this lab, you will have the opportunity to apply what you've learned in a self-guided project to gain experience with creating working digital systems on an FPGA.

Table of Contents

RHL-BEADLE © 2022 by Rania Hussein is licensed under CC BY 4.0. This project is funded by Intel Corporation.

Putting it All Together

Labs 0-5 have covered many different topics, and connecting these topics together will help solidify your understanding of digital logic. This lab consists of two tasks which require you to apply all that you have learned in this introductory lab series.

Remember the five recommended steps (from Lab 5) when approaching any digital logic problem statement:

1. Design an FSM to understand system behavior.
2. Describe inputs and outputs with a State Table.
3. Determine Boolean expressions for output signals using K-Maps.
4. Develop the system with SystemVerilog (Quartus) and simulate it (ModelSim).
5. Demonstrate the system on an FPGA (LabsLand).

The Premise

In the era of high-speed data and the Internet of Things, there is a growing demand to leverage new technologies in smart homes.

In this lab, you are tasked to help create a smart home by developing the following:

1. A better automatic door system using motion detection, and
2. A security system using camera-based motion detection.

RHL-BEADLE © 2022 by Rania Hussein is licensed under CC BY 4.0. This project is funded by Intel Corporation.

Task 1

Recall the automatic door system covered in Lab 5. That system was simplistic because it assumed that doors open and close instantaneously. However movement usually triggers a closed door to go to an “opening” state and then an “open” state. Likewise, a lack of movement usually would trigger an open door to go to a “closing” state and then to a “closed” state.

Implement and demonstrate this improved automatic door system, assuming it also checks for movement every 5 seconds. The door can only transition when a timer transmits a signal that 5 seconds has elapsed. For example, an open door should transition to the “closing” state and then to the “closed” state after 10 seconds of inactivity (that is, no movement is detected and the timer has transmitted the elapsed signal twice in a row).

Requirements

- Leverage Boole-Web as a verification tool. Include screenshots of the circuit schematic in your report.
- Include screenshots of your SystemVerilog code on Quartus and your waveform on Modelsim in your report.
- Demonstrate this modified automatic door system on LabsLand. (Remember to use switches and LEDs as your inputs and outputs respectively. It is left up to you to choose which switches and LEDs to use.)
- Reflect: How can you tell the FPGA is demonstrating the right behavior for this system?

Hints

- Consider the “5-seconds-elapsed” signal. Should it be an input or an output signal?
- Consider the FSM for the improved automatic door system. How many states should there be in total?
- Consider the output signal DoorOpen. What happens when movement is detected in the “opening” or “closing” state?

RHL-BEADLE © 2022 by Rania Hussein is licensed under CC BY 4.0. This project is funded by Intel Corporation.

Task 2

The smart home we are creating has security cameras strategically placed throughout the residence in both indoor and outdoor locations. These cameras are capable of detecting motion and identifying whether that motion originated from a human. When the homeowners are on vacation, they want a security system with specific requirements.

Requirements

- If the outdoor cameras detect human motion, they will do the following:
    a. Record footage for 5 minutes.
    b. Stop recording after 5 minutes have elapsed.
- If the indoor cameras detect human motion, they will do the following:
    a. Notify the homeowner by visually displaying the word “siren” on an app.
    b. Record footage until the alarm is turned off in the app.
- Assume there are two main factors which will cause the cameras to detect non-human motion: cars and animals. We want to avoid any alarms for non-human motion.

Transferrable Requirements

The same requirements from Task 1 also apply to Task 2.

- Leverage Boole-Web as a verification tool. Include screenshots of the circuit schematic in your report.
- Include screenshots of your SystemVerilog code on Quartus and your waveform on Modelsim in your report.
- Demonstrate this security system on LabsLand.
- Reflect: How can you tell the FPGA is demonstrating the right behavior for this system?

Hints

- Remember that the security camera system is independent of Task 1’s automatic doors; each system can use different timers.
- Use the 7-segment displays on the DE1_SoC board to display the word “siren.”
    - The 7-segment display connections are shown in Figure 1. The segments are “active low” which means that a value of 0 is needed to turn ON any segment.
    - For example, to make HEX5 display the letter ‘A’, all the segments need to be turned ON except segment 3 (HEX5[3]).
    - Accordingly, to display the letter ‘A’ on HEX5, the following statement can be used in SystemVerilog:

```
assign HEX5 = 7'b0001000;      //displays 'A' where all segments except 3 are turned on.
```

RHL-BEADLE © 2022 by Rania Hussein is licensed under CC BY 4.0. This project is funded by Intel Corporation.

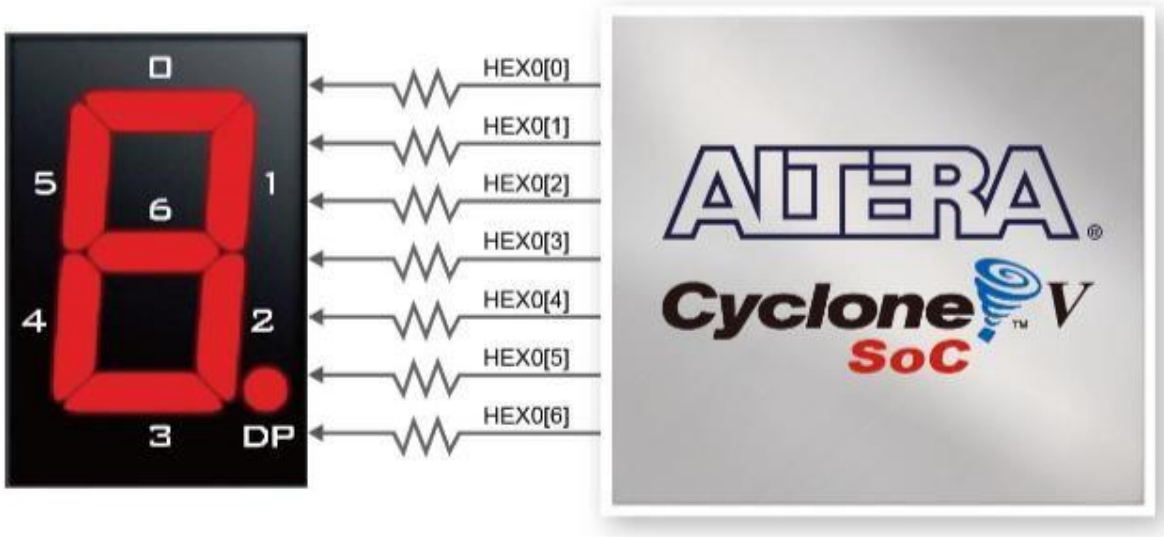

Figure 1: HEX0 connections on the DE1_SoC board

(Source: DE1_SoC Datasheet)

RHL-BEADLE © 2022 by Rania Hussein is licensed under CC BY 4.0. This project is funded by Intel Corporation.

Reflection and Learnings

Congratulations! You have just completed your first digital systems which you created on your own. Take a moment to reflect on all that you have learned in this lab series. What is your takeaway from each of the following topics? Write down what stood out to you, or something you still wonder about.

- Operators
- Logic Gates
- Truth Tables
- Binary Numbers
- Boolean Algebra
- Karnaugh Maps (K-Maps)
- Finite State Machines (FSMs)
- System Verilog
- Field Programmable Gate Arrays (FPGAs)
- Quartus
- ModelSim
- LabsLand

Lastly, now that you have created your first digital system independently, what do you think about digital logic as a whole? Write down your thoughts below.

Thank you for completing this Intro to Digital Logic Lab Series. You're well on your way in the electrical and computer engineering field!

RHL-BEADLE © 2022 by Rania Hussein is licensed under CC BY 4.0. This project is funded by Intel Corporation.