Connor Aksama
EE 371
March 13, 2023
Lab 6 Report

Procedure

This lab is comprised of two tasks. The first task involved repurposing code from Lab 1 to interface with a different GPIO interface. The second task consisted of implementing a parking lot tracker for a 3D simulator of a parking lot.

Task 1

I approached this task by first updating the name of the GPIO interface to V_GPIO and changing the port assignments for the logical controls in my Lab 1 top-level module.

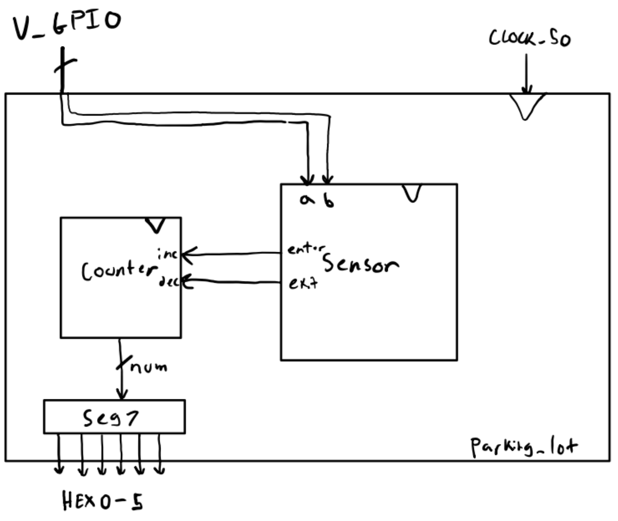

Figure 1. Top-Level Block Diagram for Lab 6 Task 1.

Task 2

I approached this task by first designing the ASMD for the rush hour tracker portion of the task. I determined the necessary states for the system, what control signals would be necessary for the data path, and defined how the output would be reported.

Next, I implemented the car tracking portion of the task, which consisted of instantiating a RAM module, which would read and write data about the total number of cars that have entered at each hour.

After testing both of these modules, I created the top-level module that would handle the I/O connections and the logic for what information would be displayed on the HEX displays as time progressed.

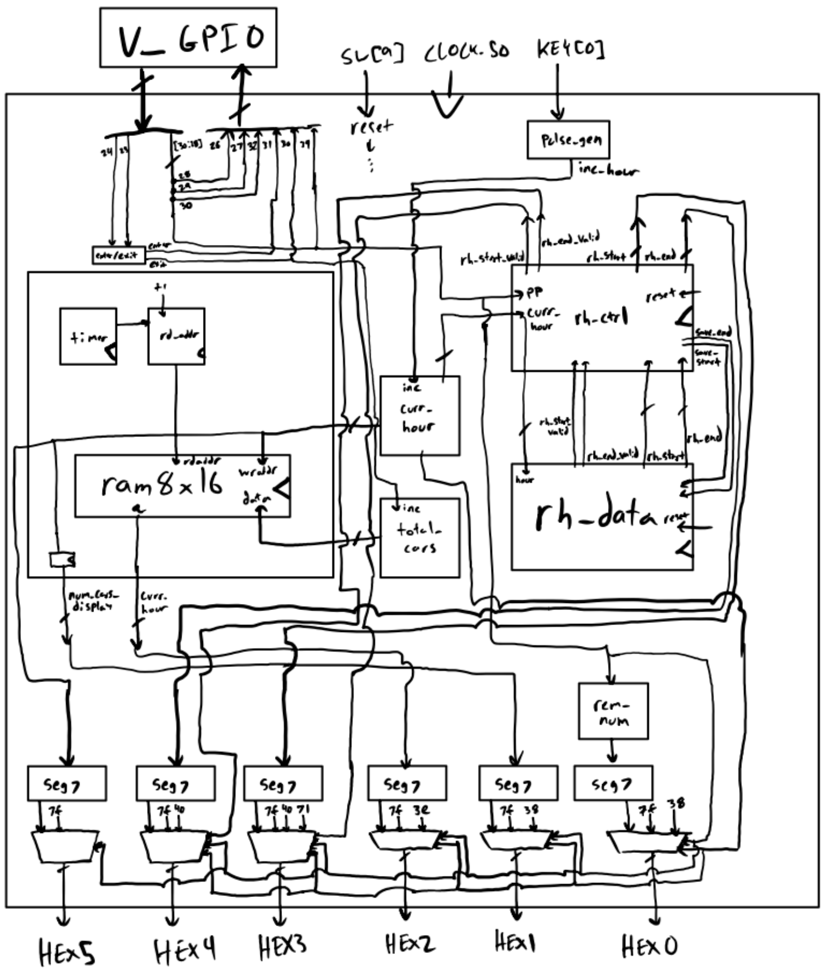

Figure 2. Top-Level Block Diagram for Lab 6 Task 2.

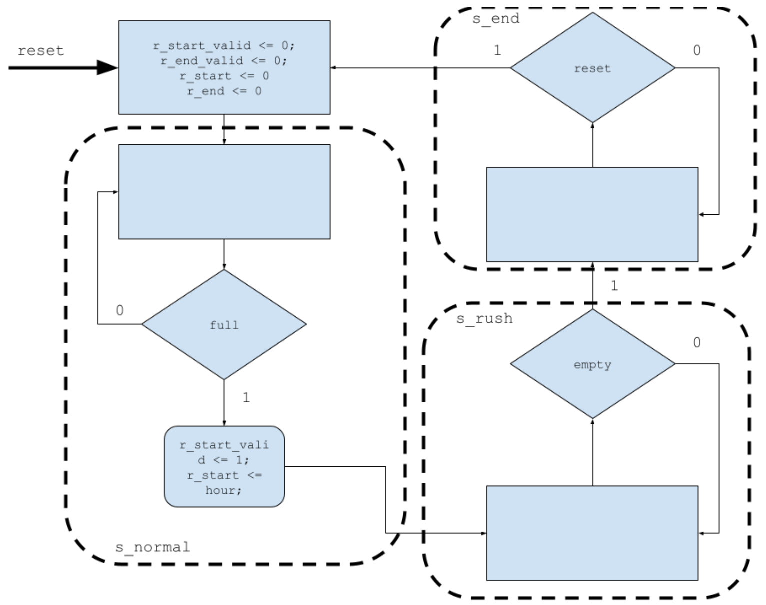

Figure 3. ASMD Chart for the Rush Hour Tracking Module

Results

Task 1

No simulations were required for Task 1 of this lab.

Task 2

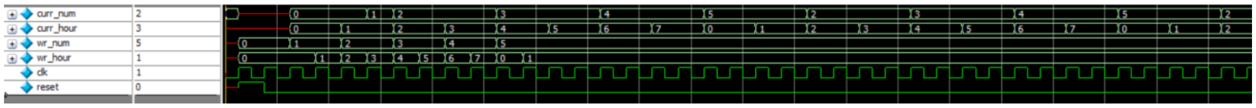

Figure 4. ModelSim Waveforms for the Car Tracking module.

This simulation shows different numbers of cars being written at different hours into the tracking module. Then after the first 8 hours, the simulation shows the cycling output, displaying the record of the total number of cars at each hour.

ModelSim Hexadecimal Code	Seven Segment Display Character
40	0
79	1
24	2
30	3
19	4
12	5
02	6
78	7
00	8
10	9
08	A
03	b
27	c
21	d
06	E
0e	F

Table 1. Conversion between hexadecimal values shown in ModelSim and the character shown on a seven segment display.

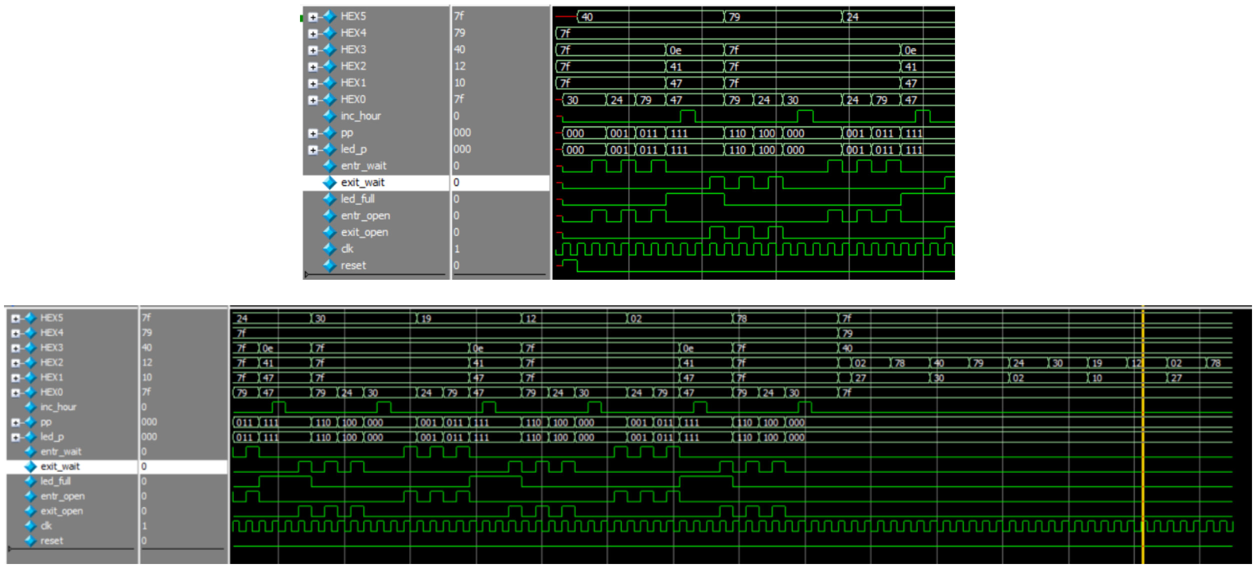

Figure 5. ModelSim Waveforms for the Task 2 top-level module.

This simulation demonstrates filling the parking lot with 3 cars one hour, then removing them the next hour, and repeating until the end of the workday. The simulation demonstrates that the correct numbers are displayed showing the hour and remaining spots, as well as the appropriate messages and external LEDs.

At the end of the 8th hour, the simulation then demonstrates displaying the start and end of rush hour, as well as the number of cars at each hour.

See Table 1 for translations between input numbers and hexadecimal values for the HEX output.

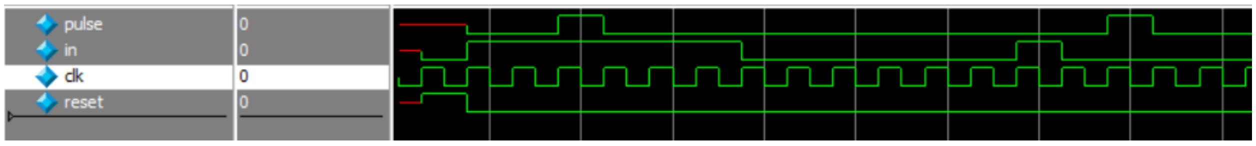

Figure 6. ModelSim Waveform for the Pulse Generator module.

This simluation demonstrates that a one cycle length signal is generated for an arbitrary length continuous input.

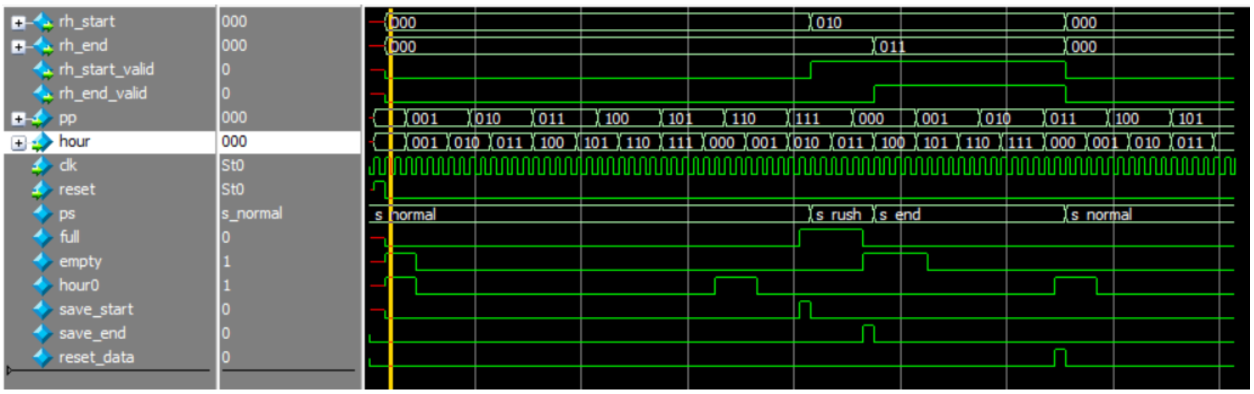

Figure 7. ModelSim Waveform for the Rush Hour Control Path module.

This simulation demonstrates a work day with no rush hour, and a work day with a start and end of a rush hour. The results show that the correct control signals to save the start/end hour are generated, and that the start and end hours are output only if they are set valid.

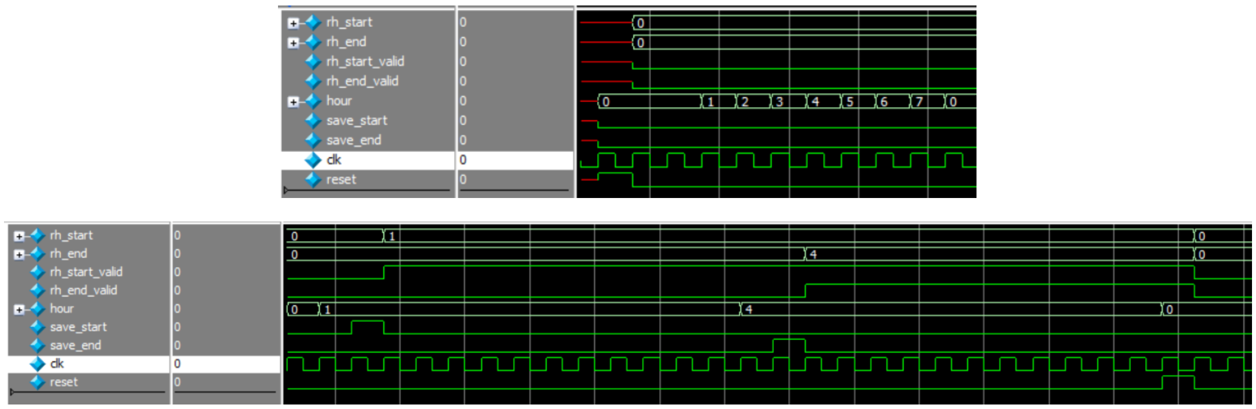

Figure 8. ModelSim Waveform for the Rush Hour Datapath module.

This simulation demonstrates that the start and ending hours for rush hour are only saved by the data path when the corresponding control signals are received by the module.

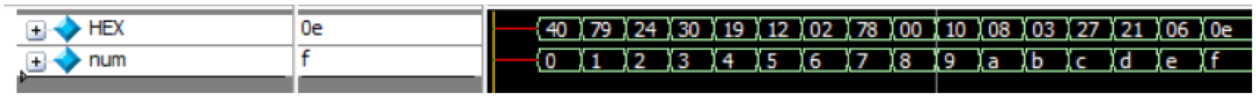
HEX 0e
num f
40 79 24 30 19 12 02 78 00 10 08 03 27 21 06 0e
0 1 2 3 4 5 6 7 8 9 a b c d e f

Figure 9. ModelSim Waveform for the single seven segment display module.

This module shows that the correct HEX patterns are generated for corresponding 4-bit inputs. See Table 1 for translations between input numbers and hexadecimal values for the HEX output.

Final Product

The overarching goal of this lab was to gain experience interfacing with a different GPIO interface from previous labs. In task 1, the same product from Lab 1 was produced, except now being able to interface with the V_GPIO bus. In task 2, I implemented a parking lot simulator/tracker that interfaced with a virtual 3D parking lot simulator. This system was able to control the virtual LEDs based on the detected positions and number of cars, allow them to enter and exit, and display different information about the parking lot using the board HEX displays. The system contains the ability to increment the current work hour, keep track of the number of cars that have entered in each hour, control the enter and exit gates, and keep track of when a rush hour starts and end if applicable. At the end of the workday, the system displays the number of cars that have entered each hour on the HEX display along with the detected start and end of rush hour if applicable.

Appendix: SystemVerilog Code

(See next page)