Khoa Tran

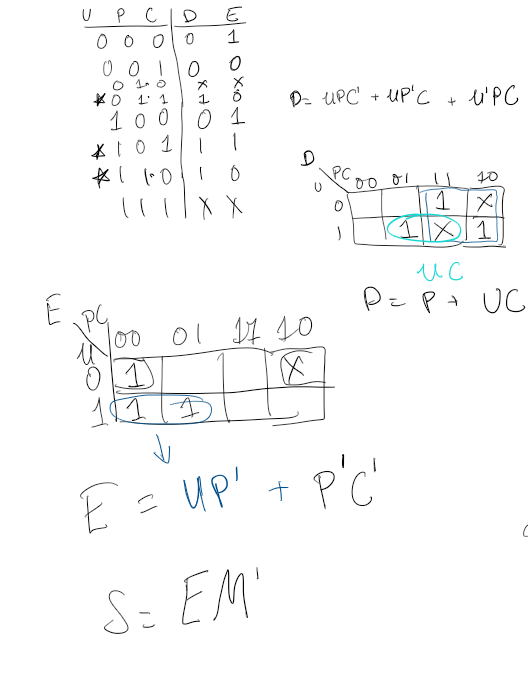
07/10/2020

EE 271

Lab 2 Report

**Procedure**

Task 1: Designing the UPC module



Approaching this problem, I first drew up the K-maps in order to figure out the equations for each specific output. For the stolen output, it depends on the expensive. So, I found out the equation for the expensive and then solved for the stolen equation. To the left is my work. Afterwards, I started to implement the equations on system Verilog by making a variable expensive and using the inputs for that equation. Afterwards, I used the expensive variable to build an equation for the Stolen output. Then, I was build the test bench by testing all the possible combinations with the U,P,C, and M inputs.

After all of this, I wrote another module to connect the switches on the board to the inputs and the LEDs to the outputs.

Task 2: Seven-segment display

Approaching this problem, I first watched the lecture to understand more about how instantiating works and what I had to do. After using the seg7 code, I then instantiated it by making an inseg7 module that matches the HEX0 output to the leds and the switches 6-9 with the bcd input. Afterwards, I combined the task 1 and task 2 by building upon my original DE1\_SoC by instantiating the seg7 with associated SW[8:6] with the bcd input and the HEX0 with the leds. I then build a testbench for both modules in order to test the inputs

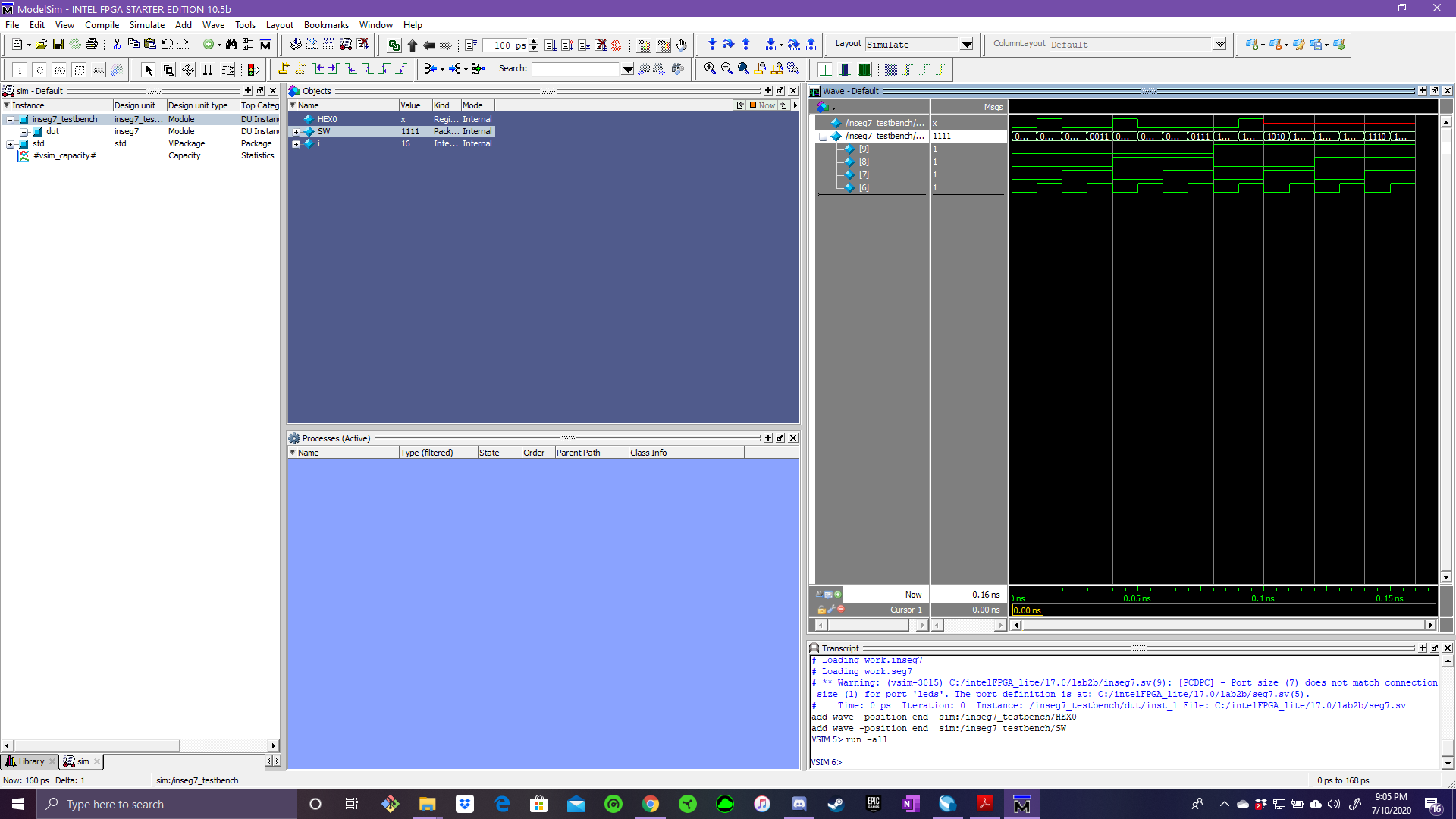
Task 3: Extra Credit

Approaching this problem, I first looked at the seg7 code and the inputs given from my professor in class. I saw that I was able to have different leds outputs for a specific case. As a result, I made variables leds1 through leds5 to represent the HEX5-0. I used mainly the HEX5-2 to display the item name and HEX0 for the item number. I wasn’t able to figure out what to do for HEX1, so I left it to not be displayed. Lastly, I instantiated the new seg7 called nseg7 on the DE1\_SoC in order to match the switches with the bcd input and the HEX board with the leds1-5

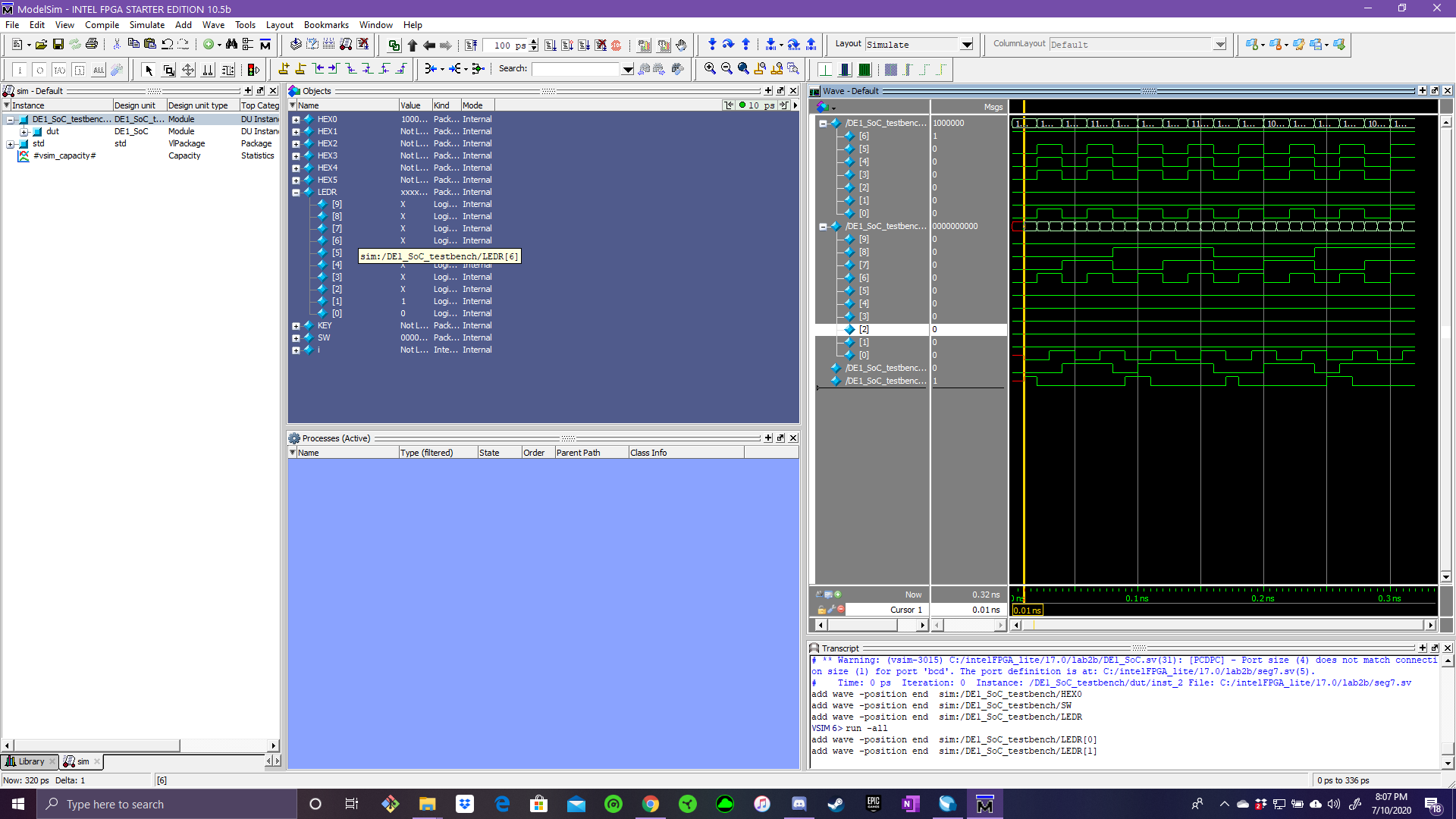
**Results**

1. Waveform from first part of Task 2

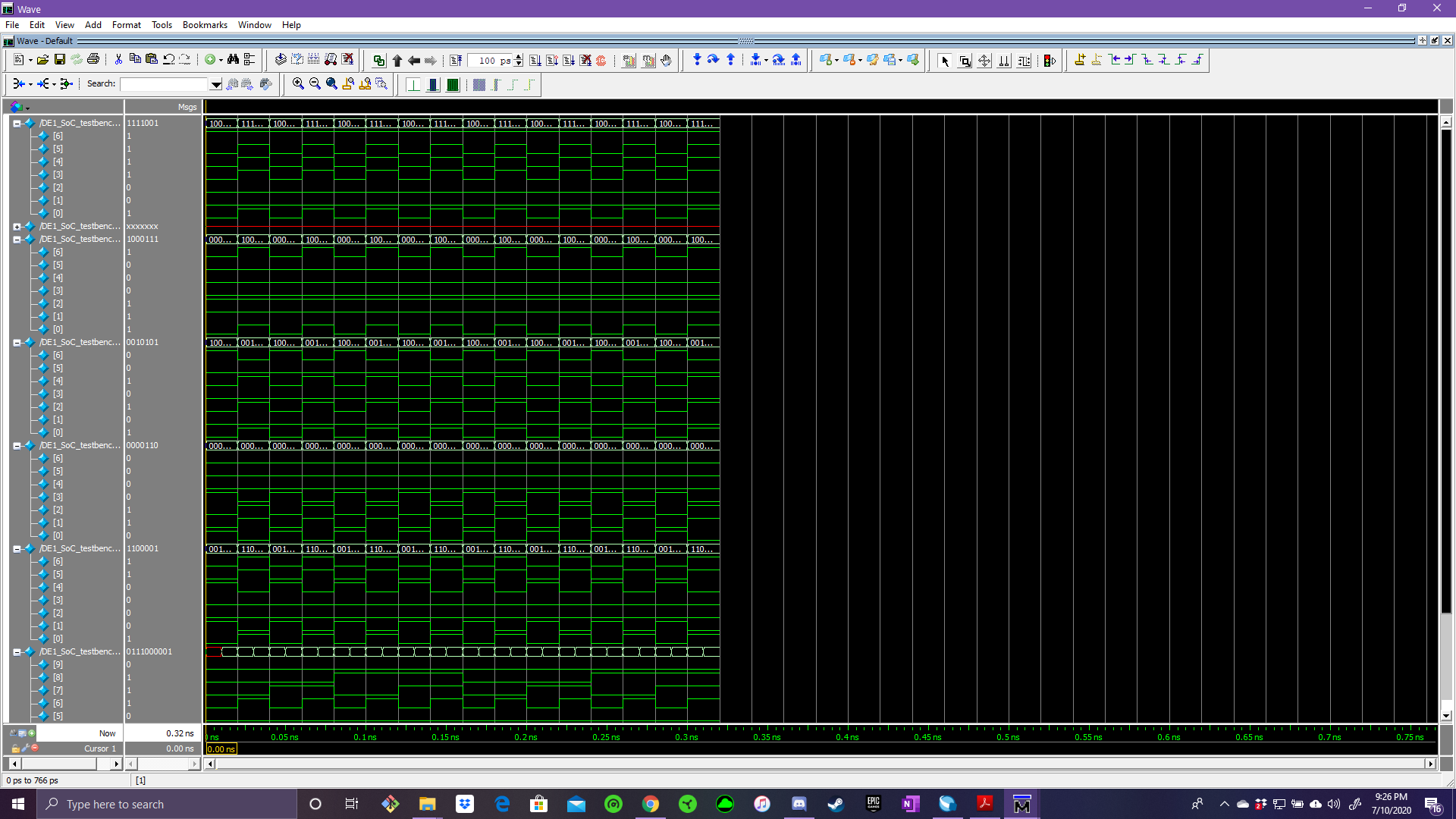
Tested the inputs of the SW and the display of HEX0. The results show that for each specific item that is associated with a UPC code that is connected with a combination of switches would display the correct item number, starting from shoes and ending with socks. The screenshot shows how the input matches with the HEX0 output and the correct item number.



1. Second part of Task 2

This simulation tested the outputs of the SW, LEDS, and the HEX0. This tested all the possible inputs of SW6-8 for the associated UPC code and SW0 for the Mark in order to match with the associated outputs of LEDR1 and LEDR0 to indicate if the item was stolen or discounted. Along with the LEDR outputs, also test for the HEX0 output and the number associated with the item. The screenshot shows how a specific input for an item has the correct output with the HEX0 and the LEDR.

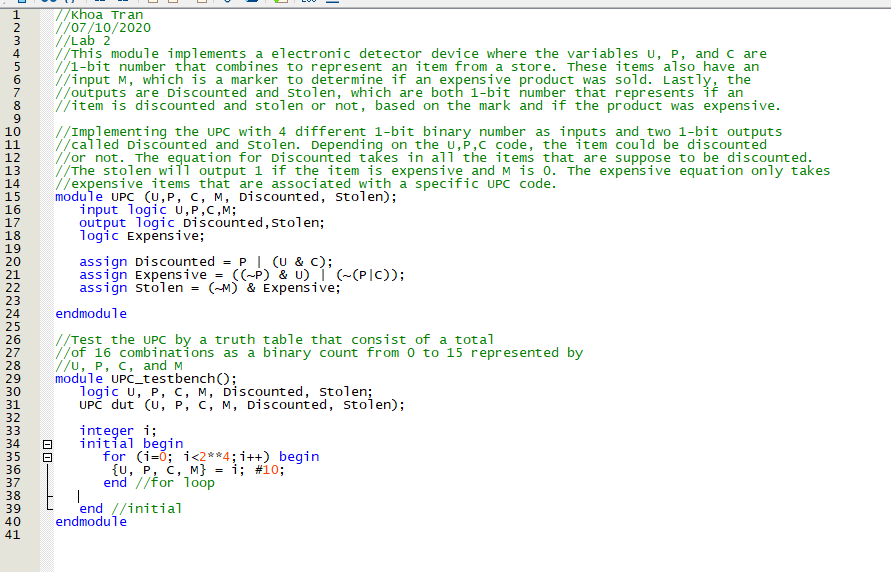
1. Extra Credit

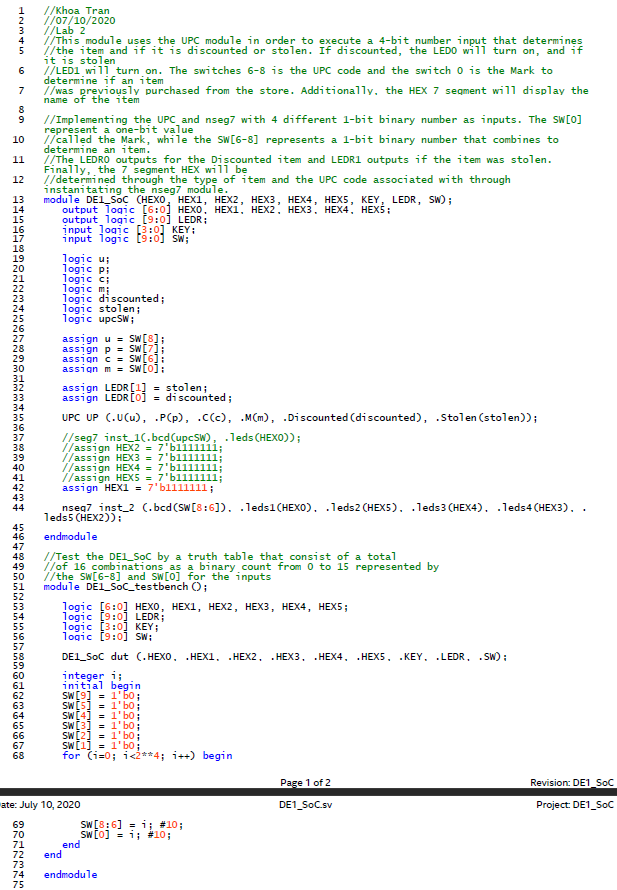
This simulation tested the outputs of the SW, LEDS, and the HEX0-5. This tested all the possible inputs of SW6-8 for the associated UPC code and SW0 for the Mark in order to match with the associated outputs of LEDR1 and LEDR0 to indicate if the item was stolen or discounted. Along with the LEDR outputs, also test for the HEX0-5 output for the name of the item along with the number associated with the item. The screenshot shows how a specific input for an item and the mark has the correct output with the HEX0-5 for the name and number of the item and the LEDR for either discounted or stolen.

Overview:

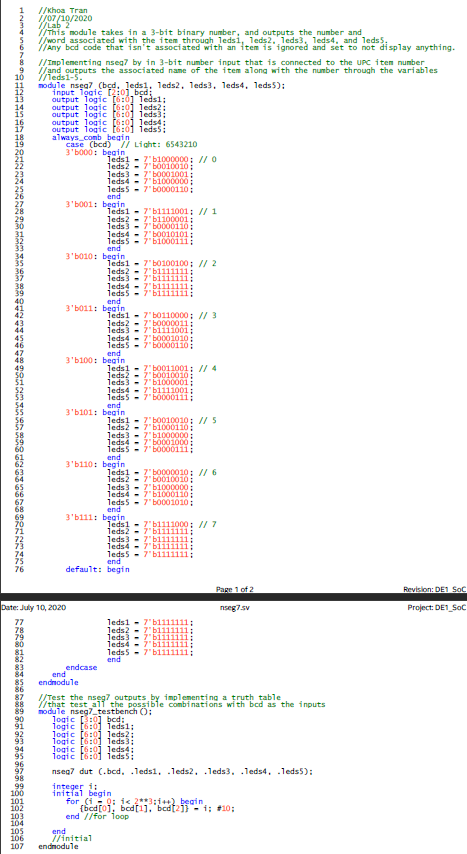
Overall, this project was designed to learn how to develop system Verilog code through a given problem by designing the equations and matching the inputs and outputs to the FPGA board correctly. Along with designing the equation, the purpose was to model the different outputs of the HEX board with different case and situation. Through all that, I was able to develop a project that was similar to what was asked by displaying the correct outputs for the associated inputs of an item number along with the mark for the expensive and stolen outputs. On top of that, I was also able to instantiate a new version of seg7 that I wrote that includes the name of the item that is associated with the UPC code. This allows for the user to check the UPC number with the word on the HEX display to see if they are matching. Along with this, I was able to display the item number as well. Overall, I did exactly what was asked along with the extra part in order to make the process seamless and running without any issues.

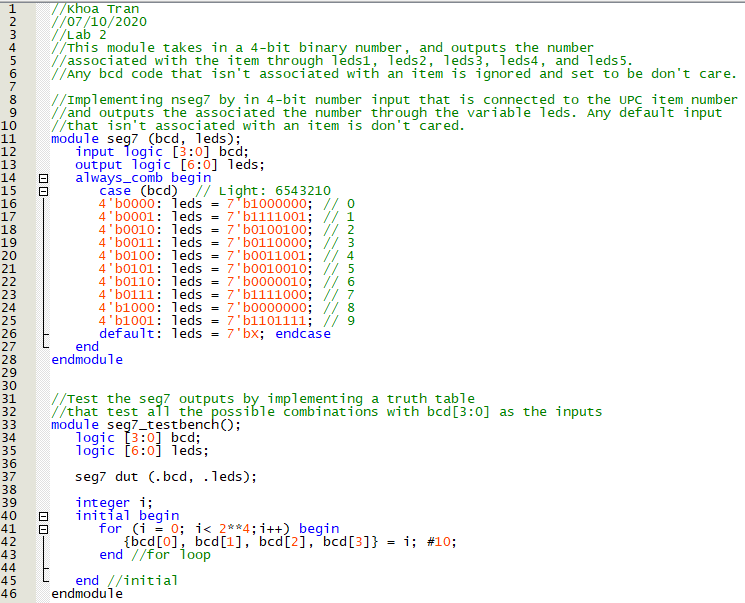
**Appendix**

1. UPC module
2. DE1\_SoC module



1. nseg7 module



1. seg7 module
2. Inseg7 instantiating seg7 module

