Shiraz University	**Shiraz University** **Department of Computer Science & Engineering**
	Professor: *Dr. Javad Dogani*
	TAs: ***A. Rostami***, *O. Asali*, *M. Mollaee*, *S. Anvari*, *M. Naziri*
	Course: Logic Circuits **Semester:** Spring, 2023
	Due Date: July. 6th (Thu.)
	Verilog - Project

Further Details:

1- This is you final project, totalling 150 of 700 gradery points.

2- Pay attention to the desired specifications.

3- In case of unmentioned details, adhere to the standards you have learned in the course and/or the Verilog classes.

4- Do not hesitate to ask questions. A. Rostami will answer your questions and doubts.

5- Any cheating will be retaliated.

6- Submit your project at Quera.

Introduction

Your final project is as follows: you are tasked to design and implement a vending machine. In this project, we are going to simulate a simple vending machine using Verilog HDL.

Description

This machine is a refrigerator that has some items each with a specific price and customers could obtain their desired items after paying the respective price. It has a keyboard, which takes the type of material as input and after receiving the price, delivers it to the customer. It stores customers' money and the received money. It can also display the remaining money of any customer. It also has an LED that indicates the status of the machine.

It has 8 types of products along with their other specifications in a file called `stuff.txt`. The general format of this file is as follows:

Type of the product (3 bits)	Supply of the product (4 bits)	Price of the product (4 bits)

TABLE 1 – General format of the `stuff.txt` file

Each type has its respective entry in the `stuff.txt` file. An example of this file looks like this:

00001100110
00100001111
01011110001
01110110010
10010001000
10101100110
11001100110
11101100110

TABLE 2 – Exemplative format of the `stuff.txt` file

The machine works in three modes; two modes for the owner and one mode for the customer.

1 Customer

The customer has some initial money and can buy goods with that money by specifying the type of product and the desired quantity. Each sale must be processed first: whether the customer has enough money and the desired quantity of the products is less than the total supply should be checked.

If the amount of money and the number of products were sufficient, this step will be completed by deducting the respective amount of money from the customer's account and depositing it into the machine's account, and updating the number of products of the machine. Should we encounter any problem, the red light should be turned on by the error signal (i.e., the red light is sensitive to errors.)

2 Owner

The owner can charge the machine's supplies (i.e., they can add to the supply of each product.) The owner can also retrieve the money stored in the machine. Again, it must be checked first whether the new supply exceeds the capacity of 4-bit numbers and whether there is any amount of money saved. Needless to say, encountering any error will trigger the red light.

Specifications

1. Your implementation must contain three modules, each for each mode, and each with its respective test cases. The `main` module is where these quasi-independent modules are connected.

2. Your implementation must also contain a separate module for displaying the system's results, including a 7-segment that displays the remaining money of the customer or saved money of the machine depending on the current mode of the system, and red light which is sensitive to errors.

3. Your implementation must be clock-based.

4. A 4-bit variable would be sufficient for each amount of money.