Emre Can Kucukoglu
eckucukoglu@gmail.com
https://github.com/eckucukoglu/basic-vending-machine

Basic Vending Machine

This project aims to implement basic functionality of a vending machine as a Verilog module.
Money can be deposited into the vending machine, multiple products may be ordered, and the machine will give back the remainder amount in coins.

Although the vending machine works synchronously, there will be an **asynchronous reset** function.

Problem Definition

Your basic vending machine (BVM) module will have the following inputs and outputs:

Name	Type	Size
Keypad	Input	3-bits
Mode	Input	2-bits
CLK	Input	1-bit
RESET	Input	1-bit
Credits	Output	8-bits
Coins	Output	6-bits

BVM can operate in one the following four modes of operation:

1. Do-nothing mode (DN)
2. Money-deposit mode (MD)
3. Product-receive mode (PR)
4. Return remainder of money (RR)

MD, PR and RR modes will be synchronized to clock input (CLK) and input will take effect with the rising edge of clock. In DN mode, changes in CLK input won't have any effect.

BVM will work according to the following mechanism:

1. Initially the machine will be in DN mode, waiting for a customer. *Credits* and *Coins* outputs are equal to zero.
2. When a customer deposits a money into the vending machine, *Mode* will be set to MD, and *Keypad* will reflect the type of money that is inserted. With the rising edge of clock (*CLK* input), BVM will read and deposit the amount of money represented by *Keypad* input. *Credits* output should be incremented by the amount represented by *Keypad* input.
3. The customer can continue depositing money into the vending machine. Each money bill will be deposited in a separate clock cycle.
4. After depositing of money is done, *Mode* will be set to PR, and customer will enter the code of the desired product by setting *Keypad*. With the rising edge of *CLK*, BVM will decrement the amount of *Credits* by selected product's price.
5. The customer can receive multiple products from the vending machine. Each product will be dispensed in a separate clock cycle.
6. After customer receives all products she wants, she switches *Mode* to RR to receive remainder of money in coins. Each bit of *Coins* output corresponds to one type of coin. Only a single coin can be dispensed in a clock cycle. Remainder of money is returned by turning corresponding coin bit to 1.
7. Until *Credits* reaches to 0, BVM will continue to dispense coins in each clock cycle.

Formats of inputs and outputs in each mode are explained in more detail below:

Do nothing mode (DN – 00)

BVM can be switched to this mode by setting *Mode* to 00. In this mode, no output shall change when the clock input comes.

Money deposit mode (MD – 01)

BVM can be switched to this mode by setting *Mode* to 01. This mode allows customer to deposit money to the machine. The amount of money represented by *Keypad* input is determined according to the following table:

Value of *Keypad* input	**Amount represented**
000	1 TL
001	2 TL
010	5 TL
011	10 TL
100	20 TL
101	50 TL
110	100 TL
111	200 TL

With the rising edge of *CLK,* the amount represented by *Keypad* will be deposited to the machine and *Credits* output shall be incremented by that amount.

Credits will always represent the amount of money available in the machine in the binary form. Rightmost bit will be the least significant.

Product receive mode (PR – 10)

BVM can be switched to this mode by setting *Mode* to 10. This mode will decrement amount of money deposited by the amount of desired product's price. Desired product is selected by setting product code using *Keypad* input. Product codes, product names and their prices are given in the table below:

Product code	Product Name	Price
000	Sandwich	220
001	Soda	120
010	Soda (Diet)	180
011	Biscuit	55
100	Water	13
101	Candy	25
110	Chocolate	85

Product code	Product Name	Price
111	Cracker	75

After the product code is set, with the rising edge of *CLK*, the price of product is subtracted from the available *Credits*. If available credits is less than the price of product, the product is not dispensed and there should be no change in credits.

Return remainder of money (RR – 11)

BVM can be switched to this mode by setting *Mode* to 11. When the customer is done with purchasing products, this mode will be used to dispense the remainder credits. Each bit of *Coins* output corresponds to a specific amount of coin. With each rising edge of *CLK* the machine will dispense a single coin (i.e. one bit of *Coins* will be 1 at a time which corresponds to the coin returned to the customer), starting from the coin with highest possible value. Each time a coin is dispensed to the customer, *Credits* will be decremented by the value of returned coin. Bits of *Coins* and corresponding coins are given in the following table:

Bit of *Remainder*	5^{th}	4^{th}	3^{rd}	2^{nd}	1^{st}	0^{th} (LSB)
Value of coin	50 TL	20 TL	10 TL	5 TL	2 TL	1 TL

RESET input

The machine has an **asynchronous *RESET*** input which will take effect immediately (without waiting for change in *CLK*) and reset all outputs to zero.

Specifications

1. Maximum amount of money that can be available in the machine is 255 (when all bits of *Credits* are one).
2. There should not be overflow in *Credits*. If amount of money will exceed 255 when customer tries to deposit more money, the machine won't accept the money and value Credits won't change. (if *Credits*=70 and customer tries to deposit 200TL, $Credits_{new}$ will still be 70)
3. Minimum amount of money that can be available in the machine is zero (when all bits of *Credits* is zero).
4. There should not be underflow in *Credits* in PR and RR modes.
5. In PR mode, if price of the desired product is larger than available amount of money, there should not be any change in *Credits* with the rising edge of *CLK*.
6. In RR mode, if *Credits* is 0 when rising edge of *CLK* is received, no new coins shall be dispensed (all *Coins* outputs are set to 0).
7. When machine is switched to RR mode from another mode, all bits of *Remainder* should be zero until rising edge of *CLK* – dispensing of coins start at the receipt of rising CLK edge.
8. When machine is not in RR mode, all bits of *Coins* should be 0.

Use the following Verilog definition for the module:

```
module BasicVendingMachine(input [2:0] Keypad, input [1:0] Mode, input CLK,
input RESET, output reg [7:0] Credit, output reg [5:0] Coins)
```

Simulation

A sample testbench will be provided to you. This testbench is only a starting point, which you should extend for proper testing of all operations of the vending machine.

Test your Verilog file in simulation mode first to find any problems, before moving on to FPGA implementation part.

FPGA Implementation

You will be provided with a Board232.v file (and a ready-to-use Xilinx project), which will bind inputs and outputs of the FPGA board with your Verilog module. You are required to test your Verilog module on the FPGA boards. After the submission date, you will make a demo to course assistants.

Name	FPGA Board	Description
Keypad	SW2, SW1, SW0	Right-most 3 switches (A)
Mode	SW7, SW6	Left-most 2 switches (B)
CLK	BTN3	Left-most button (C)
RESET	BTN0	Right-most button (D)
Credit	7-segment displays	
Coins	LD5, LD4, LD3, LD2, LD1, LD0	Right-most 6 leds (F)