Lab 5: Vending Machine

Submission Due Dates:

Demo:	2023/10/31 17:20
Source Code:	2023/10/31 18:30
Report:	2022/11/05 23:59

Objective

1. Getting familiar with the modeling of finite state machines in Verilog.
2. Getting familiar with the FPGA design flow and the keyboard control of the demo board.

Description

Design a vending machine with the FPGA board to sell a single product, which quantity and price can be configured by the keyboard.

- The basic concept of how this vending machine works:
  1. Initially, the 7-segment display shows “----”, and all the LEDs are off.
  2. BTNL is used to change to the SET state. In the SET state, we can set the stock and price of the products. Use the keyboard to choose which one you want to modify, and then input the new value.
  3. BTNR is used to change to the PAYMENT state. In the PAYMENT state, we can use the keyboard to input the amount of money you have, and then press 'enter' to proceed with the payment.
  4. If the money we pay is less than the price, we will change to the CHANGE state. The 7-segment display will show the number of items you bought and the amount of change. After 3 seconds, we will return to the IDLE state.
  5. If the money we pay is equal to or more than the price, we will change to the BUY state. The 7-segment display will show the number of items you bought and the amount of money you spent. After 'flashing' 3 times, it will change to the CHANGE state to show the change you will receive.
  6. After buying the products, change to the SET state, and we can see that the stock has been updated.

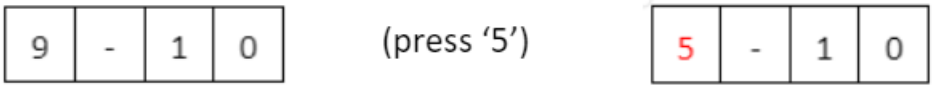
I/O signal specification

- ***clk***: the clock signal with the frequency of 100MHz (connected to pin W5)

- ***rst***: the asynchronous positive reset (connected to BTNC).
- ***BTNR, BTNC, BTNL***: pushbuttons.
  - ★ Signals from these pushbuttons should be processed by debouncing and one-pulse converters properly.
- ***LED[15:0]***: signals to control LEDs (connected to **LD15 ~ LD0**).
- ***digit[3:0]***: signals to enable one of the 7-segment digits.
- ***display[6:0]***: signals to control the digits on the 7-segment display.

Detail

1. IDLE state:
   a. After being reset, the machine goes to the IDLE state.
   b. The 7-segment display shows “----”.
   c. All the LEDs are off.
   d. Press BTN_L: go to the SET state.
   e. Press BTN_R: go to the PAYMENT state.
2. SET state:
   a. Initially, the 7-segment display shows “9-10”. It means that there are 9 items in total and 10 dollars for each one.
   b. Initially, the LED[15:8] are on. When we press the keyboard numbers 0 to 9, the number of items displayed on the 7-segment changes accordingly.



   c. When we press the ‘space’ key, the LED[7:0] will turn on, and LED[15:8] will turn off. If we press the keyboard numbers 0 to 9, the price on the 7-segment (i.e., the two digits on the right) will change by shifting the new numbers from the right into the display. After the new price is done, you can press the ‘space’ key to go back to the status of 2.a, in which you can set the number of items again.

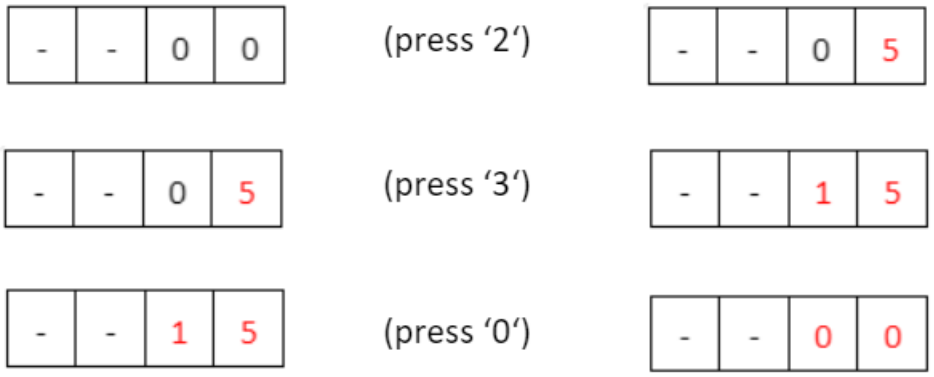
Note that in this example, we press '583' and the final price will be 83.

d. Press the "enter" key to change the IDLE state.

3. PAYMENT state:

a. Initially, the 7-segment display shows "--00", and all the LEDs are off.

b. Press the keyboard to put the money into the machine: "1" denotes 1 dollar; "2" denotes 5 dollars; "3" denotes 10 dollars; "4" denotes 50 dollars; "0" denotes that you want to cancel and get the money return (in this case, the 7-segment will show "--00" afterward).



c. The maximum money we can pay is 99 (i.e., "--99").

d. Press the "enter" key to proceed the payment. If the money you pay is equal to or greater than the price, the state machine will go to the BUY state. If the money you pay is less than the price, the state machine will go to the CHANGE state.

4. BUY state:

a. The 7-segment display shows the amount you have purchased and the money you should pay. For example, there are 5 products, and each one costs 15 dollars. If we pay 55 dollars in the PAYMENT state, the 7-segment display will show "3-45". It means that you will receive 3 products and pay 45 dollars.

b. Then, the 7-segment display and LEDs will flash three times. One 'flash' means the LEDs will be on for 0.5 seconds and off for 0.5 seconds. After three ‘flash’ (i.e., a total of 3 seconds), the state machine will go to the CHANGE state.

c. After buying the products successfully, the number of products should be reduced and updated on the 7-segment display.

5. CHANGE state:

a. The 7-segment display shows the amount you have purchased and the money you can get back. Assume there are 5 products, and each one costs 15 dollars. If we pay 55 dollars in the PAYMENT state, the 7-segment display will show “3-10” in this state. It means that you will receive 3 products and get 10 dollars in change.

b. All the LEDs will be on, and the 7-segment display will show the number of products you will receive and the dollars you will get back for 3 seconds (without flashing). Then the system will return to the IDLE state.

- Note:

1. The flashing period in the BUY state should be exactly 1 second, instead of the approximation by using clk/2^{27} or clk/2^{28}.
2. When you press and hold a key, another key should not become active. For example, after you press and hold the key ‘2’, pressing the key ‘3’ will be invalid until you release the key ‘2’. If you still hold the key ‘3’ after releasing the key ‘2’, it is acceptable whether the key ‘3’ becomes active or not.

Questions and Discussion

Please answer the following questions in your report.

A. Regarding Note 2, we only need to handle the first key press and ignore the subsequent ones. How can this be achieved? How can you prevent continuous detection of a positive signal when a key is pressed and held down? E.g., in the SET state, pressing and holding '2' will add 5 dollars only once.

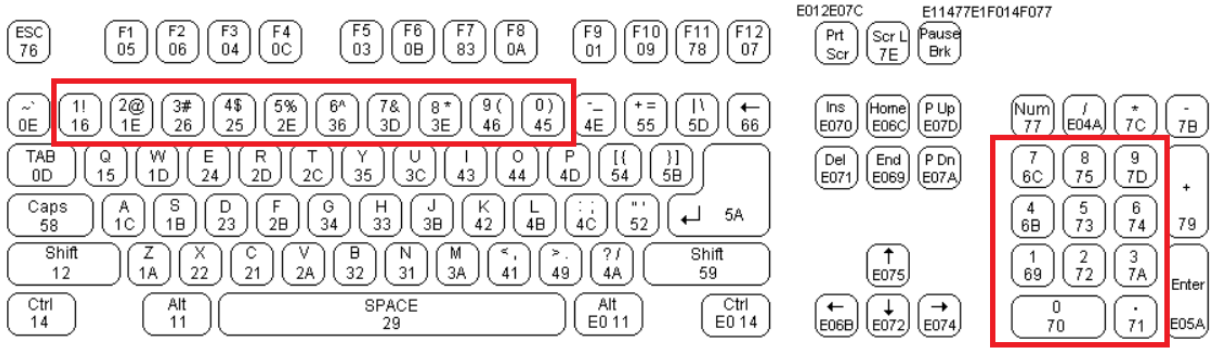
B. Regarding Question A, what can we do if we ignore Note 2? In this case, when a key is pressed first, another key can still become active (e.g., pressing two keys simultaneously.) You can explain your thoughts or use part of the code to illustrate.

HINT

1. You must design at least one finite state machine (FSM). There should be at least five states. More states or multiple FSMs are acceptable. But remember to explain your design in the report.
2. You have to use the following template for your design:

```
module lab5 (
                input wire clk,
                input wire rst,
                input wire btnL,
                input wire btnR,
                inout wire PS2_DATA,
                inout wire PS2_CLK,
                output reg [15:0] LED,
                output reg [3:0] digit,
                output reg [6:0] display
  );
    /* Note that output ports can be either reg or wire.
    * It depends on how you design your module. */
    // add your design here
 endmodule
```

3. PS2_CLK and PS2_DATA are inout signals, that is, bidirectional signals. Don't change them to input or output.
4. If you are not familiar with the usage of KeyboardDecoder module, try running SampleDisplay.v first. Remember to add the keyboard control IP if needed.
5. Both parts of the number keys will be tested in the demo.



6. Demo video: https://drive.google.com/file/d/16-f29eW-Gyqbqm1mWpNSgtKeTM0oW_wz/view?usp=drive_link

Attention

- DO NOT include the KeyboardDecoder, debounce, and one-pulse modules in the file you hand in. Please do not integrate them into lab5.v

- If you have two or more modules used for any specific lab, merge them into one Verilog file before the submission.
- You must submit one source file, **lab5.v**. DO NOT hand in any compressed ZIP files, which will be considered an incorrect format.
- You should also hand in your report as **lab5_report_StudentID.pdf** (e.g., lab5_report_111456789.pdf).
- You should be able to answer the questions for this lab from TA during the demo.
- You need to prepare the bitstream files before the lab demo to make the demo process smooth.
- Feel free to ask questions about the specification on the EECLASS forum.