Lab 5: The Fancy Ticket Machine

Submission Due Dates:

Demo: 2021/11/09 17:20
Source Code: 2021/11/09 18:30
Report: 2021/11/14 23:59

Objective

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

Description

The ticket machine behaves as follows:

- The basic concept of how this ticket machine works:

  1 BTNU and BTND act as OK and Cancel buttons, respectively.
  2 Initially, the machine will show "----" on the 7-segment display. The 7-segment display and LEDs will be flashing.
  3 When the customer presses BTNL (child), BTNC (student), or BTNR (adult), the corresponding type of ticket will be determined. The two rightmost 7-segment digits will show the price. The leftmost one digit will show the corresponding alphabet.
  4 After selecting the type of ticket, the customer can adjust the number of tickets to purchase.
  5 After adjusting the number of tickets, the customer can deposit coins by pressing BTNL ($1), BTNC ($5), or BTNR ($10).
  6 When the customer presses the Cancel button, the machine will return the deposited money.
  7 Once the customer deposits enough money to purchase the tickets, the ticket will come out. At the same time, the leftmost 7-segment digit will show the alphabet representing the type of ticket. The two rightmost ones will show the number of tickets, and all LEDs will be flashing to emulate the tickets coming out.
  8 After the tickets are released, the change will be returned. The 7-segment display will show the decreasing balance to indicate that the change is returning coin by coin.

- The detailed specification:
  - Any unspecified input (buttons, switches) should do nothing.
  - Any unspecified output should be turned off (i.e., the LEDs and 7-segment display

are turned off).

- The flashing frequency of LEDs and 7-segment display is 0.5Hz.

1 IDLE state:

a. After being reset, the machine goes to the IDLE state.

b. The 7-segment display will be flashing “----”.

c. All the LEDs will be flashing.

d. When the customer presses BTNL (child), BTNC (student), or BTNR (adult), the corresponding type of ticket will be selected. Then the machine goes to the TYPE state.

2 TYPE state:

a. When BTNL (child), BTNC (student), or BTNR (adult) is pressed, the corresponding type of ticket will be selected.

b. The price for adult, student, and child tickets are $15, $10, and $5, respectively.

c. The two rightmost 7-segment digits will show the price. The leftmost digit will show the corresponding alphabet (adult: A, student: S, Child: C).

Alpha-bet		Price	Price

d. When BTNU (OK) is pressed, the machine goes to the AMOUNT state.

e. When BTND (cancel) is pressed, the machine goes to the IDLE state.

3 AMOUNT state:

a. The number of tickets is one, initially.

b. When BTNL (-) (or BTNR (+)) is pressed, the number of tickets will decrease (or increase) by one.

c. The minimum ticket number is one; the maximum ticket number is three.

d. The rightmost 7-segment digit will show the amount. The leftmost digit will show the corresponding alphabet (adult: A, student: S, Child: C).

Alaph-bet			Amount

e. When BTNU (OK) is pressed, the machine goes to the PAYMENT state.

f. When BTND (cancel) is pressed, the machine goes to the IDLE state.

4 PAYMENT state:

a. When BTNL ($1), BTNC ($5), or BTNR ($10) is pressed, the corresponding amount of money will be deposited.

b. When the amount of money is equal to or greater than the price of tickets, the machine goes to RELEASE state.

c. The two rightmost 7-segment digits will show the price of tickets (The amount multiply the price for the selected type of ticket), and the two leftmost one digits will show the amount of money deposited.

Money	Money	Price	Price

d. When BTND (cancel) is pressed, the machine goes to the CHANGE state.

5 RELEASE state:

a. The rightmost 7-segment digit will show the number of tickets; the leftmost digit will show the corresponding alphabet (adult: A, student: S, Child: C).

Alpha-bet			Amount

b. All LEDs will be flashing.

c. The machine goes to the CHANGE state after 5 seconds.

6 CHANGE state:

a. The two rightmost 7-segment digits will show the change to be returned.

		Change	Change

b. If there are $5 or more to return, the change will be decreased by $5 (to simulate dropping a $5 coin) every second. When there is less than $5 to return, the change will be decreased by $1 (to simulate dropping a $1 coin).

c. After all the change is returned, the machine will return to the IDLE state.

I/O signal specification

- ***clk***: clock signal with the frequency of 100MHz (connected to pin **W5**).
- ***rst***: active-high reset (connected to **SW0**).
- ***BTNL, BTNR, BTNU, BTND, BTNC***: pushbuttons.
  - ✧ Each signal from the pushbutton in the following 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 show the digits on the 7-segment display.

Note

1 The operating frequency of each debouncing or one-pulse converter should be the same as your FSM's.

2 Demo video: https://youtu.be/5Of4IDI9Ics

Hint

1 You must design at least one finite state machine (FSM).

   d. There should be at least six states. More states or multiple FSMs are acceptable. But remember to explain your design in the report.

2 You can use the following template for your design:

```
module lab5 (
    input clk,
    input rst,
    input BTNL,
    input BTNR,
    input BTNU,
    input BTND,
    input BTNC,
    output [15:0] LED,
    output [3:0] DIGIT,
    output [6:0] DISPLAY
    // you can declare the output data type by yourself:)
    // e.g., output (reg/wire) [15:0] LED;
);
  // add you design here
endmodule
```

3 It will help if you drew the state diagram(s) before coding. You should draw your

state diagram(s) in the report.

Attention

- ✓ You should hand in only one Verilog file, lab5.v. If you have several modules in your design, integrate them in lab5.v. You don't need to include the debounce and one-pulse modules in the file you hand in. Please do not integrate them in lab5.v)
- ✓ You should also hand in your report as lab5_report_StudentID.pdf (e.g., lab5_report_109062666.pdf).
- ✓ Please do not hand in any compressed files, which will be considered as an incorrect format.
- ✓ You should be able to answer the questions of this lab from TA during the demo.
- ✓ You need to generate the bitstream before the demo.
- ✓ If you have any questions about the specification, feel free to ask on the EECLASS forum.