Lab 3 Advanced: Snake

Submission Due Dates:

Demo: 2024/10/15 17:20
Source Code: 2024/10/15 18:30
Report: 2024/10/20 23:59

Objective

1. Getting familiar with the 7-segment display and pushbuttons on the FPGA board. Note: Signals from pushbuttons should be processed by debouncing and one-pulse converters properly.
2. Getting familiar with finite-state machines in Verilog.

Action Items

1 lab3_advanced.v (100%)

In this lab, you are required to design the controller of the snake movement and see the result on the seven-segment display.

a. I/O list:

I/O	Connected to	Definition
clk	W5	Clock signal with the frequency of 100MHz.
rst	SW0	Asynchronous positive reset signal; reset the counter machine back to the **INITIAL** state.
right	BTNR	To turn the snake to the right.
left	BTNL	To turn the snake to the left.
up	BTNU	To move the snake forward.
down	BTND	To switch the state from **MOVING** to **FILLING**.
DISPLAY[6:0]		Showing the current position of the snake.
DIGIT[3:0]		Enabling one of the 7-segment digits.

b. **Finite-state machine (FSM)**

The FSM consists of an **INITIAL** state to display the snake, a **MOVING** state to move the snake and a **FILLING** state to record the location visited.

✓ **INITIAL** state

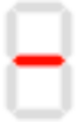
- State transition:

i. After 3 seconds, change to the MOVING state.

ii. In this lab, you can consider **one cycle of 100MHz/(2^27) as approximately 1 second.** Additionally, 0.5 seconds is approximately one cycle of 100MHz/(2^26).

- Seven-segment display:

  The seven-segment display shows the original position of the snake, which is in the middle of the rightmost digit of the seven-segment display.
  The snake's head is facing to the left initially.



The 7-segment display in the **INITIAL** state

- NOTE: The BTNR, BTNL and BTNU control buttons don't work in this state.

✓ **MOVING** state

- State transition:

  After pressing the **down** button, change to the **FILLING** state.

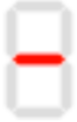
- Seven-segment display:

  i. Continue at the position in the INITIAL state.

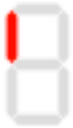
  ii. There are only three possible actions. BTNR is to turn the snake to the right, BTNL is to turn the snake to the left, BTNU is to move the snake forward.

  iii. The snake can only move on the **rightmost digit** of the seven-segment display.

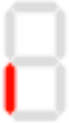
  iv. The initial position of the snake is in the middle, facing left, which is the same as that in the INITIAL state.



a. If the BTNR is pressed (i.e., turning right), the 7-segment display changes to the following:

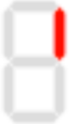


b. However, If the BTNL rather than the BTNR is pressed (i.e., turning left), the 7-segment display changes to the following:

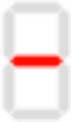


c. Intuitively, **the BTNU button won't do anything in this situation.**

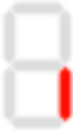
v. For another example, if the snake is in the following position facing down:



a. If the BTNR is pressed (i.e., turning right), the 7-segment display changes to the following:



b. If the BTNU rather than BTNR is pressed (i.e., moving forward), the 7-segment display changes to the following:



d. Intuitively, **the BTNL button won't do anything in this situation.**

- NOTE: **The snake will flash every second (on for 0.5 seconds and then off for 0.5 seconds).**

✔ **FILLING** state

- State transition:

  A counter will be triggered when **the snake has visited every position at least once.** After one second has elapsed, the FSM will return to the INITIAL state. You can still move the snake anywhere within this second. Note that the snake keeps flashing as well.

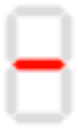
- Seven-segment display:

  i. Continue at the position in the MOVING state.

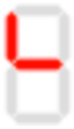
  ii. All the moving policy is the same as in the MOVING state. However, in the

FILLING state you must record the location visited and keep the visited location displayed.

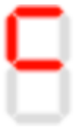
iii. For example, assume that the snake is in the middle while facing left:



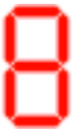
a. If the BTNR is pressed (i.e., turning right), the 7-segment display changes to the following. Note that the middle segment will keep on.



b. After pressing the BTNR again (i.e., turning right), the 7-segment display change to the following:



iv. The FSM will return to the **INITIAL** state once every place has been visited:



- NOTE: **The snake will flash every second (on for 0.5 seconds and then off for 0.5 seconds) so that it can be identified easily.**

c. DEMO video

URL: https://youtu.be/4EZgb8naGQE?si=E9SPTles5AMYDZg0

d. You have to use the following template for your design:

```
    module lab3_advanced (
      input wire clk,
      input wire rst,
      input wire right,
      input wire left,
      input wire up,
      input wire down,
      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
```

2 Questions and Discussion

Please answer the following questions in your report.

A. In this lab, our reset signal is an asynchronous reset. What if it is a synchronous reset? And how to modify your design to implement a synchronous reset?

B. Why do we need the debounce and one-pulse modules?

3 Guidelines for the report

Refer to the guidelines in the report template (or in the previous lab assignments).

Grading policy (subject to change): Part (A): **35%**; Part (B): **50%**; Part (C): **10%**; (D): **5%**

Attention

- DO NOT copy-and-paste code segments from the PDF materials. It may also paste invisible non-ASCII characters, leading to hard-to-debug syntax errors.
- In this lab, we provide you **onepulse.v**, **debounce.v** and **clock_divider.v**. Do not include them in your submissions. We will take care of them when verifying your source code.
- If you have two or more modules used for any specific lab, merge them into one Verilog file before the submission.
- You should submit one source file, **lab3_advanced.v**. DO NOT hand in any compressed ZIP files, which will be considered an incorrect format.
- You should also hand in your report as **lab3_report_StudentID.pdf** (i.e., lab3_report_110062320.pdf).
- You should be able to answer questions about 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 any questions about the specification on the EECLASS forum.