

Lab 3 Advanced: One-stroke Puzzle (一筆畫)

Submission Due Dates:

Demo: 2025/10/07 17:20
Source Code: 2025/10/07 18:30
Report: 2025/10/12 23:59

Objective

This lab introduces you to pushbutton input handling and seven-segment display control on the FPGA. You will:

- Design a finite-state machine (FSM) to implement the one-stroke puzzle game.
- Correctly handle pushbutton input signals using debouncing and one-pulse converters.
- Manage real-time display updates and timing records on the FPGA board.

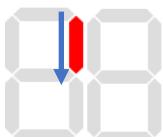
Action Items

1. **lab3_advanced.v (100%)**

In this lab, you are required to design the one-stroke controller and display progress and timing records on the seven-segment display.

a. Game explanation:

- In this game, player is asked to solve a puzzle by traverse through all the lit seven-segment display parts.
- For all the puzzles, the first stroke(第一筆) should be start at the top-right corner of the left-most digit and go downward.

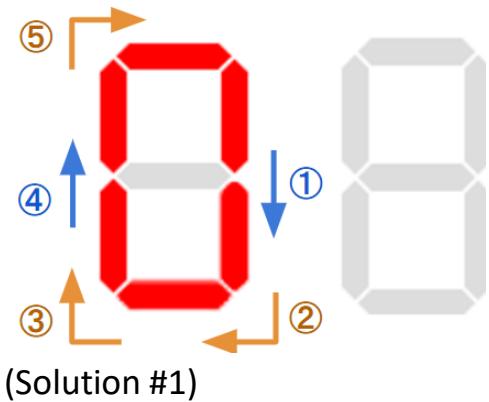


- Puzzle #1 is illustrated in Picture A.



(Picture A, illustrating puzzle #1)

To solve puzzle #1, player should follow Solution #1. Here, the numbers indicate the operation order. The color blue indicates pressing forward and the color orange indicates pressing right.

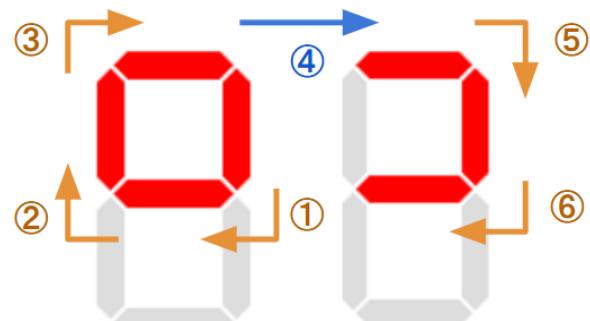


- Puzzle #2 is illustrated in Picture B.



(Picture B, illustrating puzzle #2)

To solve puzzle #2, player should follow Solution #2.



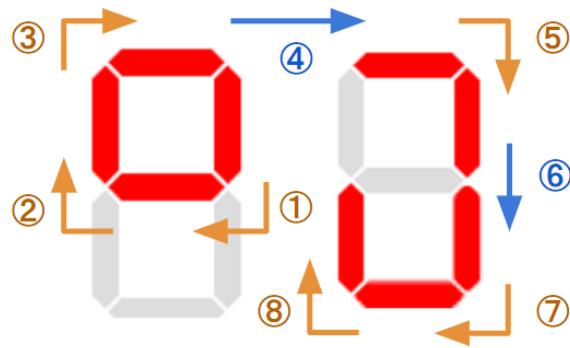
(Solution #2)

- Puzzle #3 is illustrated in Picture C.



(Picture C, illustrating puzzle #3)

To solve puzzle #3, player should follow Solution #3.



(Solution #3)

b. I/O list

I/O	Connected to	Definition
clk	W5	Clock signal with the frequency of 100MHz.
rst	SW0	Asynchronous active-high reset signal; reset the machine to the INITIAL state.
mode	SW1	Decide whether the FSM goes to the PRACTICE or TIMING state when the play button is pushed.
play	BTNC	Start the game (transition to the PRACTICE or TIMING state, depending on mode).
right	BTNR	In INITIAL : select next puzzle In PRACTICE or TIMING : turn the pen to the right.
left	BTNL	In INITIAL : select the previous puzzle.
forward	BTNU	Move the pen forward in the PRACTICE and TIMING state.
LED	LD15-LD0	Indicates result of the game.
DISPLAY[6:0]		Shows pen position or scores.
DIGIT[3:0]		Selects active digit.

c. FSM Specification

The FSM has four states: of an **INITIAL** state to select a one-stroke puzzle to be solved, a **PRACTICE** state to finish the one-stroke puzzle without recording time, a **TIMING** state to finish the one-stroke puzzle with recording time, and a **FINAL** state to display the record.

✓ **INITIAL** state

- Seven-segment display:
 - For the left two digits, show different puzzle by pressing the **left** and

right button. Initially, it shows puzzle #1. For example, if the current puzzle is puzzle #1, pressing right will change the current puzzle into puzzle #2 and pressing left will change the current puzzle into puzzle #3.

- For the right two digits, they show the best score (the least time spent) for the selected puzzle. If it is the first time solving the puzzle, show --.
- For example, if the best score for puzzle#2 is 20, the seven-segment display should be:

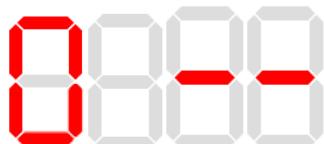


- LED display: LD15~LD0 stay off.
- State transition:

Use left/right buttons to select a puzzle.

Press the **play** button to start:

- If **mode== 0**, go to the **PRACTICE** state.
- If **mode== 1**, go to the **TIMING** state.
- When returning from **FINAL** state, the puzzle will show the one selected here.
- Notice that in any state, pulling **rst** makes FSM jump into **INITIAL** state and seven-segment display should show as follows.

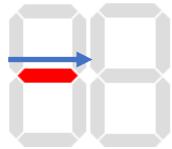


✓ PRACTICE state

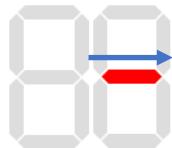
- Seven-segment display:
 - Here we have a pen. The pen is indicated by the flashing seven-segment display part. The flash follows the rule: on for 0.5 second and then off for 0.5 second.
 - The two 7-segment display digits on the left indicate how far the current puzzle has been solved, and the pen can only move on these two seven-segment display digits.
 - At the beginning, the two left seven-segment display digits show the selected puzzle.

There are only two possible actions. The **right** button is to turn the pen

to the right, and the **forward** button is to move the pen forward. Note that every press makes the pen move only 1 segment. That is, if the pen is facing right and the position is as Picture D, the player should press forward button again to move the pen to the position in Picture E.



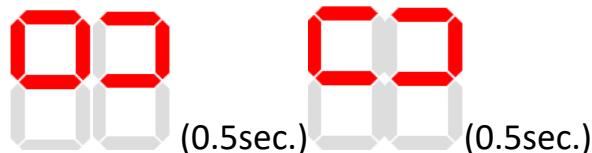
(Picture D)



(Picture E)

- For example, for puzzle#2,

- ◆ initial



- ◆ press **right** button



- ◆ press **right** button



- ◆ press **right** button



- ◆ press **up** button



- ◆ press **right** button



- ◆ press the **right** button and the puzzle should be solved
change to the **FINAL** state in the next cycle

- The two seven-segment displays on the right remain off in this state.

- LED display:
LD15~LD0 stay off.
- State transition:

When the one-stroke puzzle is solved, change to the **FINAL** state in the next cycle.

✓ **TIMING** state

- Seven-segment display:
 - The two seven-segment display digits on the left follow the rules in **PRACTICE** state.
 - The two 7-segment displays on the right show a decimal number representing the number of seconds elapsed since entering this state. The count starts from 00 and increments every second. If the count exceeds 99, it remains at 99.

- LED display:
LD15~LD0 stay off.
- State transition:

When the one-stroke puzzle is solved, change to the **FINAL** state in the next cycle.

✓ **FINAL** state

- seven-segment display:
 - If the previous state is **PRACTICE** state, show the following picture:
(PrAC)



- If the previous state is **TIMING** state, the two seven-segment display digits on the left shows the number of the puzzle followed by a “-”, and the two seven-segment displays on the right show the timing record. For example, the timer value is 20 for puzzle#1,



- LED display:
 - If the previous state is **PRACTICE** state, LD15~LD0 will flash every second (on for 0.5 second and then off for 0.5 second). The LEDs will be on->off->on->off->on->off in this state.
 - If the previous state is **TIMING** state, and the time taken to solve the puzzle is shorter than the current record for that puzzle, then LD15~LD0 will flash every second (on for 0.5 second and then off for 0.5 second). The flashing LEDs mean a new best time was achieved. Make sure to **update the best score** when returning to the **INITIAL** state. If the time taken did not set a record, keep LD15~LD0 lit.
- State transition:
After staying in the FINAL state for three seconds or **rst** is set to 1, the FSM will go back to **INITIAL** state.

d. Demo Video

<https://reurl.cc/6qedgO>

e. Verilog Design Template:

You must use the following template for your design

```
module lab3_basic (
    input wire clk,
    input wire rst,
    input wire mode,
    input wire play,
    input wire right,
    input wire left,
```

```
    input wire forward,  
    output reg [15:0] LED,  
    output reg [3:0] DIGIT,  
    output reg [6:0] DISPLAY  
);  
  
/* Note that output port can be either reg or wire.  
 * It depends on how you design your module. */  
  
// add your design here  
  
endmodule
```

2 Questions and Discussion

- A. Why are debounce and one-pulse modules required? How does the FSM's clock rate relate to the one-pulse module's clock rate? Explain your reason. What issues occur if the wrong clock rate is used? (Illustrate with waveforms to explain.)
- B. How do you control and record the pen's position in the puzzle? Explain clearly and provide partial code examples.

Attention

- ✓ Display the digits according to lecture material:



- ✓ DO NOT copy-and-paste code segments from PDFs (hidden characters may cause hard-to-debug syntax errors).
- ✓ Use provided **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 create several modules, merge them into a single Verilog file.
- ✓ Submit only:
 1. **One** Verilog source file: **lab3_advance.v**
 2. Lab report: **lab3_report_StudentID.pdf** (i.e, lab3_report_123456789.pdf)
- ✓ **DO NOT** submit ZIP files, which will be considered as incorrect format.
- ✓ You should be ready to answer questions about this lab to TA during the demo.
- ✓ 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.