EECS2070 02 Exam 2_2

15:30 ~ 17:30, December 14, 2021

➢ Instructions

1. There are **one (1) design problem** in this exam, with the PDF specification of **four (4) pages** in total.
   - You also have the hardcopy of the first page to sign and hand back.
2. Download the ***.cpp** files from OJ. Change them to ***.zip** and decompress them. (Skip the *.h file, which the OJ system enforces to have.)
3. Submit each Verilog code to OJ **immediately** after it is done.
   a. You have the responsibility to check if the submission is successful. The incorrect submission results in a zero score.
   b. The module names should be **exam2_2**.
   c. The first line of each Verilog code should be a comment with your student ID and name as follows:

```
// 107123456 王小明
module exam2_2 (...
```

   d. The **exam2_2.v** should be able to be compiled by Vivado, generating the bit file.
   e. **The submission is due at 17:30**!
4. Please **take the OJ password slip with you** when leaving your seat. Do not litter!
5. The score will get deducted if you fail to follow the rules.
6. Hand back this problem sheet **with all the following items checked**. Also, **sign your name with your student ID**.

☐ I confirm that I read the instructions carefully and understand that my score will get deducted if failing to follow the rules.

☐ I confirm that I follow the naming rule of the modules. And the first line of each answer code shows my student ID and name.

☐ I confirm that all my answers were submitted successfully.
- ✓ I've submitted the module **exam2_2** and the complete design in a single file. And that single file can be synthesized, generating the bit file.
- ✓ I understand that the generated bit file will be scored. And the incorrect submission will result in a zero score.

☐ I hereby state that all my answers are done on my own.

ID: ____________________ Name: _____________________

➢ Design Problems

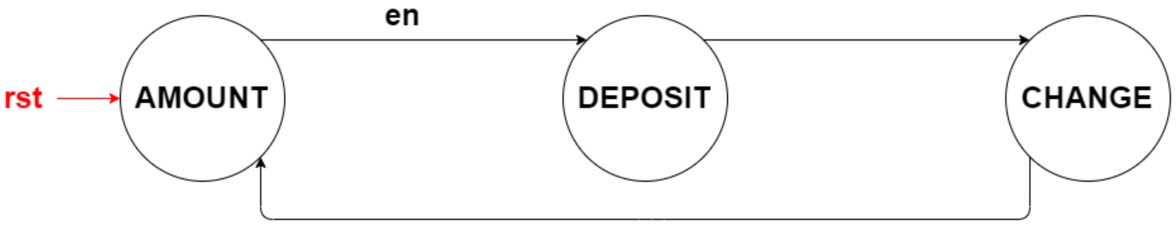
[100%] [FPGA Implementation]

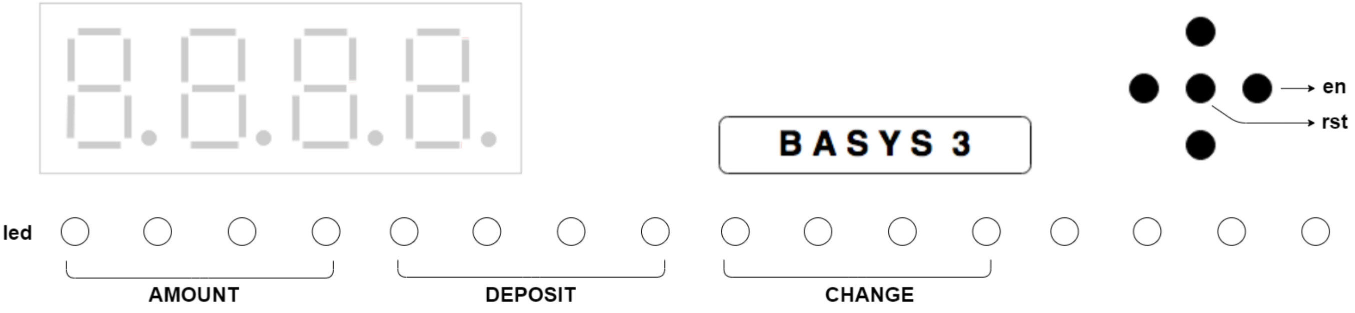
1. Use the Verilog template **exam2_2.v** to implement a counter. DO NOT modify I/O signals of **exam2_2**.
2. The state machine is synchronous with the clock rate of 100MHz / 2^{25}
3. The seven-segment display is synchronous with the clock rate of 100MHz / 2^{15}
4. Your design should be reset **synchronously**.
5. Your design should change its value at the **positive edge** of each clock cycle.
6. Here is the table showing the function with the I/O connection:

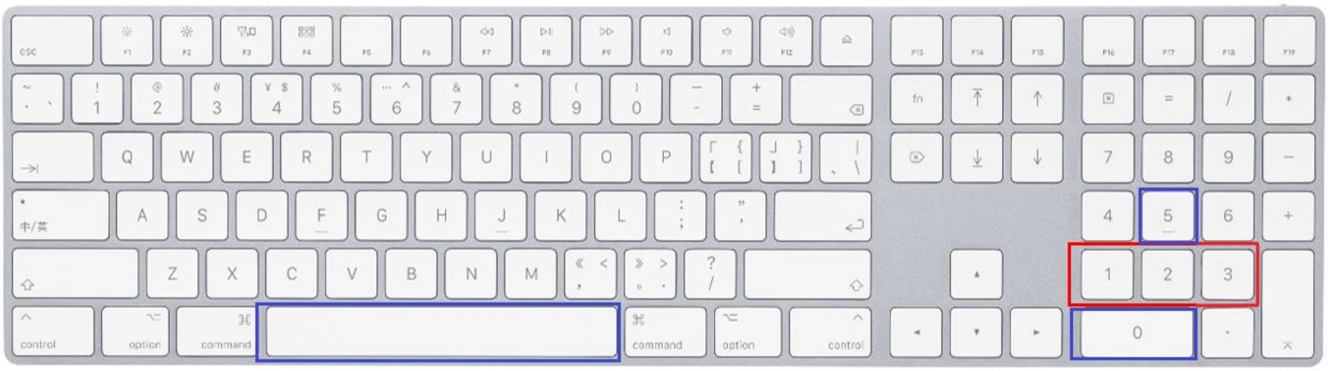
Name	**I/O**	**Pin**	**Description**
clk	Input	W5	100MHz clock signal
rst	Input	BTNC	Active-high reset
en	Input	BTNR	Control the states
PS2_CLK	Inout	C17	Signal used for the keyboard
PS2_DATA	Inout	B17	Signal used for the keyboard
DIGIT [3:0]	Output	4-Digits Pin	Display counter
DISPLAY [6:0]	Output	7-Segment Pin	Display counter
led [15:0]	Output	16-LEDs	Display state

7. Please refer to the demo video (**exam2_2 _DEMO.mp4**) for further details.

8. The design concept to help you understand:







9. Function description

   (1) LED is used to indicate which state it is.

   (2) Implement a ticket machine.

   (3) There are three states: **AMOUNT**, **DEPOSIT**, and **CHANGE**.

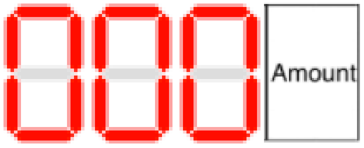
   - AMOUNT

     Everything should be initiated in the beginning. **The number of tickets is one, initially.** The price for ticket is $15. Choose the amount of tickets wanting to buy with keyboard “numpad 1”, “numpad 2” and “numpad 3”, and show it on the rightmost 7-segment digit.

     (a) State transition:
     After pressing the en button, change to the **DEPOSIT** state.

     (b) LED
     led [12:15] (P3~L1) are on; others are off.

     (c) Seven-segments



   - DEPOSIT

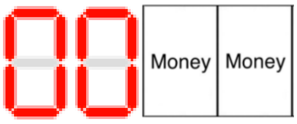
     The two rightmost 7-segment digits will show the ***Money*** you deposit. When keyboard “numpad 0” ($10) or keyboard “numpad 5” ($5) is pressed, the corresponding amount of money will be deposited.

     The two leftmost 7-segment digits will show the ***Price*** you need to pay (the amount multiply the ticket price) **only when you press and hold the Space key.** Otherwise, show “00”.
     **Note: assume that the number key and Space key will not be pressed at the same time.

     (a) State transition
     When the ***Money*** you deposit is equal to or greater than the ***Price*** you need to pay, change to the **CHANGE** mode.

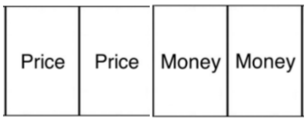
     (b) LED
     led [8:11] (V13~U3) are on, others are off.

     (c) Seven-segment display:



(without pressing the Space key)

Or

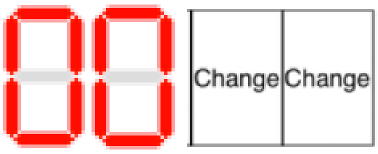


(press and hold the Space key)

   - CHANGE

The two rightmost 7-segment digits will show the ***Change*** to be returned. ***Change*** Start from the remaining money (***Money*** - ***Price***), and it will be decreased by $1 (to simulate dropping a $1 coin) with the frequency of clk / (2^25).
Note: when the ***Change* decreases to zero in the end, you still need to show “0000” for a clock.

(a) State transition
After showing “0000”, return to the **AMOUNT** mode.
(b) LED
led [4:7] (W18~V14) are on, others are off.
(c) Seven-segment display:



10. Refer to the XDC constraint file (**exam2_2.xdc**) for the pin connection. Also, DO NOT modify the constraint file. Otherwise, your design will fail the synthesis then get a ZERO score.
11. There are already several modules in the template, including the clock divider, seven-segments, debounce, one-pulse, and KeyboardDecoder modules. You can modify them if necessary.
NOTE: The submitted exam2_2.v should be **self-contained**. That is, you must include **all the modules** you need in the single file and submit it, or you will get a 5 points penalty.
12. Grading

Function	Score
LED	10%
AMOUNT	20%
Money in DEPOSIT state	25%
Price in DEPOSIT state	20%
CHANGE	25%

Happy Designing and Good luck!