EECS2070 02 Exam 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**.
   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 (...
```

   d. The **exam2.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** 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

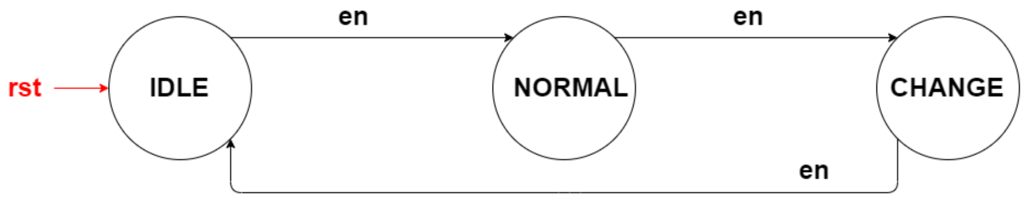
[100%] [FPGA Implementation]

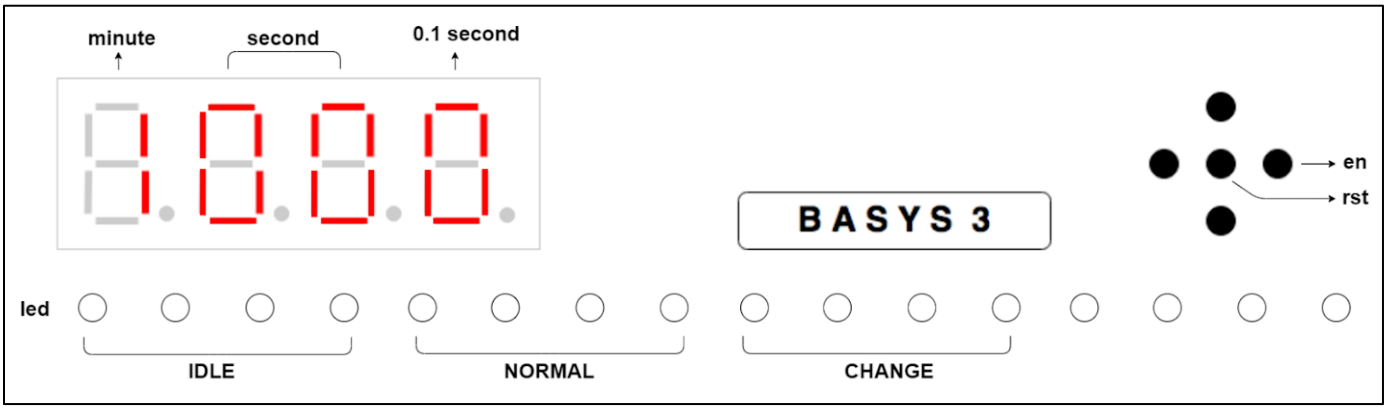
1. Use the Verilog template **exam2.v** to implement a counter. DO NOT modify I/O signals of **exam2**.
2. The state machine is synchronous with the clock rate of 100MHz / 2^{21}, 100MHz / 2^{25} , 100MHz / 2^{27}
3. The seven-segment display is synchronous with the clock rate of 100MHz / 2^{15}
4. Your 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:

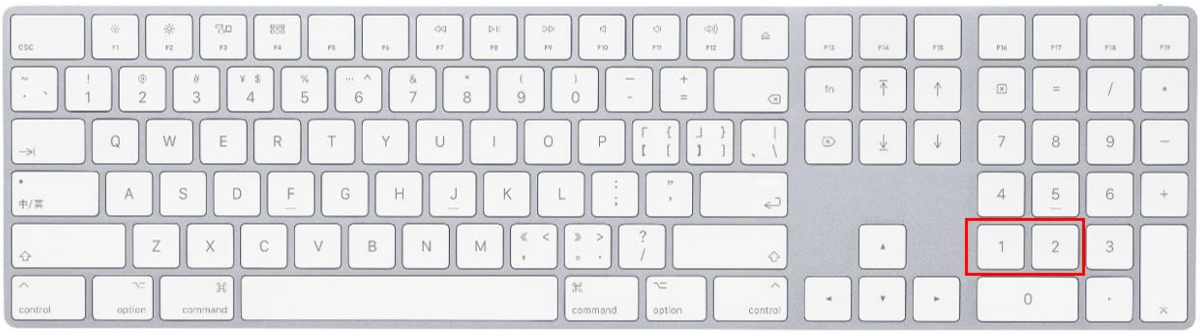
Name	I/O	Pin	Description
clk	Input	W5	100MHz clock signal
rst	Input	BTNC	Active-high reset
en	Input	BTNR	Control the states
PS2_CLK	Inout	C17	Signal used for the keyboard
PS2_DATA	Inout	B17	Signal used for the keyboard
DIGIT [3:0]	Output	4-Digits Pin	Display counter
DISPLAY [6:0]	Output	7-Segment Pin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

7. Please refer to the demo video (**exam2 _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 counter whose counting rate can be changed with the keyboard.

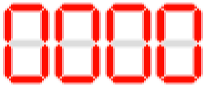
(3) There are three states: **IDLE**, **NORMAL**, and **CHANGE**.

- IDLE

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

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

  (c) Seven-segments
  Display “0000”.



- NORMAL

  **Note: The time unit “sec” is just a convenient way to describe. Follow the frequency of the questions. You don’t need to implement an accurate 0.1 sec time.

  Count from **0.00.0 to 1.00.0 (60sec)** with the clock rate of 100MHz / 2^{25} for an increment of 0.1 sec. Show it on the 7-segment display. When the counter counts to 1.00.0, stop counting.

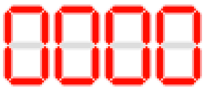
  The user can change the counting speed with the keyboard during the counting. When pressing the numerical Key 1 or Key 2, the counting rate (i.e., the **corresponding clock rate**) will be immediately changed.

Keyboard	Clock Rate
(No Key Pressed)	100MHz / 2^{25}
Key 1 (Speed Up)	100MHz / 2^{21}
Key 2 (Slow Down)	100MHz / 2^{27}

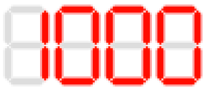
  (a) State transition
  After pressing the **en** button, change to the **CHANGE** mode.

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

  (c) Seven-segment display:
  Note: The counter goes to “1000” after “0599”.



~



- CHANGE

  Almost the same as the NORMAL state! Count from **0.00.0 to 1.00.0**. The only difference is the speed control. The counting rate will **only change when you press and hold the key** and return to the normal speed (100MHz / 2^{25}) after releasing the key.

Keyboard	Clock rate
(No Key Pressed)	100MHz / 2^{25}
Key 1 (speed up)	100MHz / 2^{21}
Key 2 (slow down)	100MHz / 2^{27}

E.g., if Key 1 was pressed at 2.5 sec and released at 5.0 sec, it will count at the rate of 100MHz / 2^{21} from 2.5 to 5.0 sec and return to 100MHz / 2^{25} after 5.0 sec.

Note that the counter starts from **0.00.0 and stops counting when reaching 1.00.0.

(a) State transition

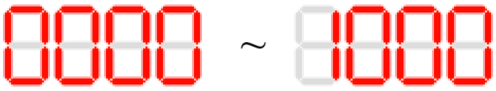
After pressing the **en** button, return to the **IDLE** mode.

(b) LED

led [4:7] (W18~V14) are on, others are off.

(c) Seven-segment display:

Note: The counter goes to “1000” after “0599”.



10. Refer to the XDC constraint file (**exam2.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. You must include **all the modules** you need in the same file and submit it, or you will get a 5 points penalty.
12. Grading

Function	Score
LED	9%
IDLE	6%
A counter (without changing speed) in NORMAL	35%
A counter (without changing speed) in CHANGE	10%
Change speed in NORMAL state	20%
Change speed in CHANGE state	20%

Happy Designing and Good luck!

(If you have too much time left, there is always a joke for you.)

A professor experienced a severe headache in the classroom. So, he went to see a doctor. After medical examination, the doctor told him:

"Mr. professor, after my detailed examination, we found out that your brain has two parts: one is left, and the other is right."

"Your left brain has nothing right. And your right brain has nothing left!"