CS210402 Exam 2

15:30~17:30, December 5, 2023

、

I. Instructions

1. There are **two design problems** in this exam, with the PDF specification of **five 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_A or exam2_B**.
   c. The first line of each Verilog code should be a comment with your student ID and name as follows:

   ```
   // 111012345 王小明
   module exam2_A (...
   ```

   d. The **exam2_A .v** and **exam2_B.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: _____________________

II. Design Problem

Hint: You may want to do Problem B first if it looks more familiar.

A. [40%] [FPGA Implementation]

1. This design problem requires the **FPGA board**.
2. Use the Verilog template **exam2_A.v** to implement the **optimization problem**.
3. DO NOT modify I/O signals. (You can define I/O signals as either reg or wire type.)
4. 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
DIGIT [3:0]	Output	4-Digits Pin	To control the 7-segment display
DISPLAY [6:0]	Output	7-Segment Pin	To control the 7-segment display

5. Your design should be reset to the 0 **synchronously**.
6. Your design should change its value at the **positive edge** of each clock cycle.
7. Function description

   In this problem, you need to **maximize**

   $f(x) = (\sum_{i=0}^{i=15} \quad i * x_i) \quad - (\sum_{i=0}^{i=7} \quad i * x_i - \sum_{i=8}^{i=15} \quad i * x_i)^2$,

   where x_i is **either 0 or 1**, $i \in \{0,1,2,3,\ldots,15\}$.

   You need to output your answer on the Seven-segments display in **decimal** format.
   Remember to take care of the leading zero problem.
   For example, if the answer is 23, you need to display "0023":

   Hard-code the result is strictly prohibited. You must compute all possible solutions and figure out the maximal one in your Verilog design.

8. Refer to the XDC constraint file (**exam2_A.xdc**) for the pin connection. Also, DO NOT modify the constraint file. Otherwise, your design will fail to synthesize and get a ZERO score.
9. There are already several modules in the template. You are free to modify them if necessary. Remember to include **all the modules** for FPGA synthesis and implementation in the same file to submit. Otherwise, you will get a 5-point penalty for any missing template module. In other words, the submitted file must be ready for synthesis.

10. In case you are not lucky to solve the problem. You may solve the simplified version instead:

    Try to **maximize**

    $f(x) = (\sum_{i=0}^{i=3} \quad i * x_i) \quad - (\sum_{i=0}^{i=1} \quad i * x_i - \sum_{i=2}^{i=3} \quad i * x_i)^2$,

    where x_i is **either 0 or 1**, $i \in \{0,1,2,3\}$.

11. Grading

Function	Score (40%)
Full version (for $i \in \{0,1,2,3,\ldots,15\}$)	40%
Simplified version (for $i \in \{0,1,2,3\}$) (Only when you fail to solve the full version)	10%

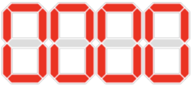
B. [60%] [FPGA Implementation]

1. This design problem requires the **FPGA board** with the **keyboard**.
2. Use the Verilog template **exam2_B.v** to implement the **shooting game** (the Local Co-op Game) which allows you to attack the opponent with the keyboard controls.
3. DO NOT modify I/O signals. (You can define I/O signals as either reg or wire type.)
4. Your design should be reset to the IDLE state **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	To 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	To control the 7-segment display
DISPLAY [6:0]	Output	7-Segment Pin	To control the 7-segment display
led [15:0]	Output	16-LEDs	To display as the problem describes

7. Please refer to the demo video (**exam2_DEMO.mp4**) for further details.
8. Function description
    a. LED is used to indicate the status as the problem describes.
    b. Implement a counter whose counting rate can be changed with the keyboard.
    c. There are three states: **IDLE**, **COUNTDOWN**, and **BATTLE**.

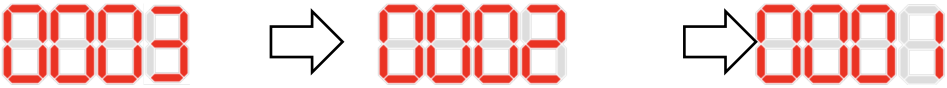
        1. IDLE
            a. State transition:
            After pressing the **en** button, change to the **COUNTDOWN** state.
            b. LED:
            led [0:15] (U16~L1) are on.
            c. Seven-segments display:
            Display “0000”.



        2. COUNTDOWN
        **Note: The time unit “second” is just a convenient way to describe. You can consider one cycle of 100MHz/(2^27) as 1 second.

            a. State transition:
            After **3** seconds, change to the **BATTLE** state.
            b. LED:
            At the beginning, led [0:15] (U16~L1) are off.
            After 0.5 second (one cycle of 100MHz /(2^26)), led [0:15] (U16~L1) are on (reverse all the signals of LED).
            After 0.5 second (one cycle of 100MHz /(2^26)), led [0:15] (U16~L1) are off (reverse all the signals of LED).
            And then two more on-off cycles (make sure to turn off the LED before entering the next state).
            c. Seven-segment display:

You need to countdown **3** seconds. And use the seven-segment to display the current seconds, i.e., 3□2□ 1□ 0. The final ‘0’ may not be easy to observe since the state will be changed at this moment.



after one second after one second

3. BATTLE

We have 2 players, P1 and P2. P1 presses **the left ctrl key** (enclosed by a red box in the figure) to **shoot the bullets**. P2 presses **the right ctrl key** (enclosed by a green box in the figure) to **shoot the bullets**.

Each player has **two lives** in the beginning.

a. State transition:

When any of the player's lives becomes 0, change to the **IDLE** state.

b. LED:

The movement of light represents the trajectory of the bullets.

When **P1 shoots the bullets**, the bullets will start from LD15 as shown in the following figure:

● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

The bullets are heading to the right. Its rate is 100MHz /(2^24).

So after one cycle of 100MHz /(2^24), the state of LED be like:

○ ● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

When the bullet reaches the end (LD0), **P2’s life is decreased by 1**. The bullets also disappear, of course.

When **P2 shoots the bullets**, the bullets will start from LD0 as shown in the following figure:

○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ●

The bullets are heading to the left. Its rate is 100MHz /(2^24).

So after one cycle of 100MHz /(2^24), the state of LED be like:

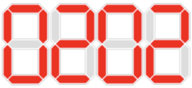
○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ● ○

When the bullet reaches the end (LD15), **P1’s life is decreased by 1**. The bullets also disappear, of course.

There is no limit on the number of bullets that can be used, and you don't have to deal with the problem of collision between bullets from both sides.

c. Seven-segment display:

The **left** two digits represent P1’s life.
The **right** two digits represent P2’s life.
Initial condition:



9. Refer to the XDC constraint file (**exam2_B.xdc**) for the pin connection. Also, DO NOT modify the constraint file. Otherwise, your design will fail to synthesize, and then get a ZERO score.
10. There are already several modules in the template. You are free to modify them if necessary. Remember to include **all the modules** for FPGA synthesis and implementation in the same file to submit. Otherwise, you will get a 5-point penalty for any missing template module. In other words, the submitted file must be ready for synthesis.
11. Grading

Function	Score (60%)
IDLE state	5%
COUNTDOWN state	10%
P1 P2 can shoot the bullets correctly in the BATTLE	20%
P1’s and P2’s life can be deducted correctly in the BATTLE	20%
BATTLE state (with the ability to return to IDLE correctly)	5%

Happy Designing and Good luck!