Lab 2

An Electronic Ping-Pong Game

In this lab, you are going to design an electronic ping-pong game and implement it on Xilinx Spartan-3E Starter Kit. You will use LEDs on the board (i.e. LED7 to LED0 from left to right) to simulate the moving ball. A ball moving from left to right is simulated as LEDs illuminating one after another starting with LED7 (i.e. LED7 illuminates first, then LED6, LED5, etc. It is basically a moving light). You will use switches and push buttons to enter player names and start the game. The push buttons also will be used during the game. You will use LCD display to display game score and the winner.

Your design should start (or after the reset) with IDLE state displaying `PING PONG GAME` on the first line of LCD display as shown below.

```
PING PONG GAME

```

Then, your design should wait for the *BTN North* on the board to start taking 3-digit player names as inputs. For taking player names, you will use switches and *BTN North*. As shown in Fig. 1, each digit is represented with 8 bits. Since your FPGA has 4 switches, you enter each digit by first entering its most significant 4 bits and then its least significant 4 bits. You enter each 4 bits by setting the switches and pressing *BTN North*. You will start with entering the player names with the left-most digit of the first player and end with right-most digit of the second player. After entering the names, your design should display the initial score on the LCD display as shown below where `XXX` on the first and second lines represents the names of the first and second players, respectively.

```
XXX : 0
XXX : 0
```

Then, your design should wait for the *BTN North* to start the game. The game will have 5 rounds where each round starts with *BTN North* and ends when one of the player wins. After each round, current score will be displayed on the LCD display. For example, if round 3 ends and the first player have won 2 of 3 rounds, you should display the score on the LCD as shown below.

```
XXX : 2
XXX : 1
```

Each round is played as follow: the first player (Player1) standing on the left hand side (using *BTN West* on the board) and the second player (Player2) on the right hand side (using *BTN East* on the board). You will use all 8 LEDs to represent the moving ball.

When LED7 is on, it is Player1's turn to serve. The Player1 serves by pushing *BTN West*. The Player1 should not push the button too soon or too late. Otherwise, the light disappears and Player2 wins the round. With pushing *BTN West* on time, the electronic ball starts moving from left to right; in other words LEDs illuminate in the order of LED7, LED6,

Lab 2

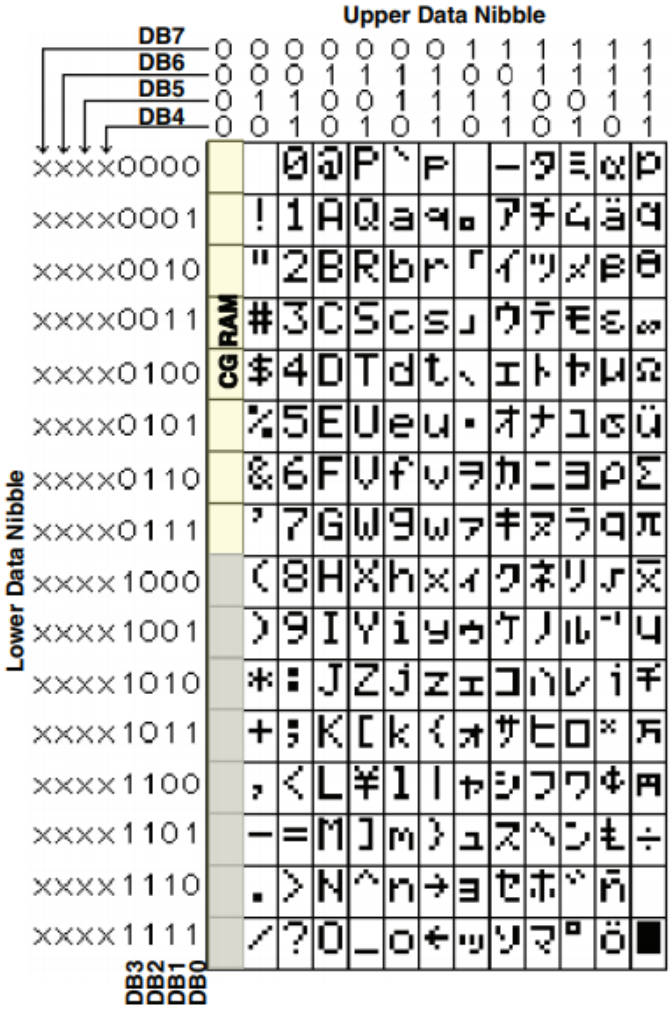

Figure 1: Digits

LED5, LED4, LED3, LED2, LED1 and LED0. As soon as LED0 becomes on, the Player2 should hit the ball to send it back to the Player1. Player2 does this by pushing *BTN East*. However, the Player2 should not push the button too soon or too late. Otherwise, the light disappears and Player2 loses and Player1 wins. A round continues until one of the players wins.

At each round, the ball starts from LED4 position and moves from left to right (towards LED0). In your design, it should take 0.25 seconds for the ball to move from one LED position to another (moving one step). Please note that your FPGA has an on-board clock with 50 MHz frequency (20 ns period). You should implement your 0.25 seconds waiting time based on this information. After the round 5, you should display the final score and the winner as shown below. Then, your design should wait *BTN North* to go to IDLE state.

```
XXX : 2
XXX : 3     Winner
```

The operation steps can be summarized as follows:

Lab 2

- Your design starts in IDLE state displaying `PING PONG GAME` on the LCD. When you press *BTN North*, your design goes to player name entering state.
- In order to get 3-digit player names, you should use switches and *BTN North*. For entering each digit, you should use switches and *BTN North* twice (most-significant 4 bits and least-significant 4 bits). In total, you should set switches and press *BTN North* 12 times for entering the names of Player1 and Player2.
- After entering the player names, your design should display current score on the LCD and wait for *BTN North* to start round 1 of the game.
- After each round, your design should display the current score on the LCD and waits for *BTN North* to start the next round.
- After the round 5, your design should display the final score and the winner on the LCD. Then, it should wait for *BTN North* to go to IDLE state.

Your design should work at 50 MHz. Your hardware design should have the top Verilog module with the following interface. You should also write a UCF file for your design (see Xilinx_Spartan3E_StarterKit_Tutorial.pdf on SUCourse for writing your UCF file).

```
module PingPongTop(clk, rst, BTN_West, BTN_North, BTN_East,
                   SW, LED, Data_Out, LCD_Control);
    input clk, rst;
    input BTN_West, BTN_North, BTN_East; // Push button inputs
    input [3:0] SW;                      // Switch inputs
    output reg [7:0] LED;                // LED outputs
    output reg [3:0] Data_Out;           // LCD control signals
    output reg [3:0] LCD_Control;        // LCD control signals

    // ...

endmodule
```

In your implementation, you will need some extra modules (`debouncer.v` and `LCDI.v`) which we provide you under Resources/Lab Materials/Modules on SUCourse. The debouncer (`debouncer.v`) circuit gets the input from a push button and generates a one clock pulse output. The LCD display interface (`LCDI.v`) module, drives the LCD display (see Xilinx_Spartan3E_StarterKit_Tutorial.pdf on SUCourse for details).

Before synthesizing and implementing your design, we strongly suggest you to write a Verilog testbench and verify the correctness of your finite state machine by simulation (do not use `debouncer.v` in your simulation, it is only for implementation). After verifying the correctness of your design, synthesize and implement your design targeting Xilinx Spartan XC3S500E FG320 FPGA with speed grade 4 using Xilinx ISE WebPack 14.7. Next, generate the FPGA configuration bitstream, download the bitstream into the FPGA and verify your hardware design on the board. Then, you should write a lab report (in .pdf format) with

Lab 2

readable English and good structure (dumping many figures and raw synthesis/implementation reports into a pdf file is not a lab report.). In your report, you should explain your design (your finite state machine, test cases you used for verification if you have any, etc.).

Finally, put your project directory (make sure all *.v files are in that directory) and your report into one ".zip" file named Lab1_username1_username2.zip (e.g. Lab1_berkea_erdinco.zip), and submit this zip file using EE310 SUCourse website. Please note that we will use a plagiarism detection software for your lab assignments.