Electronic Battleship Game

You will design an electronic game of battleship that takes two players to play and implement it on FPGA device. You will be asked to demonstrate your implementation on FPGA device.

The playground is 2-dimensional 4x4 area for each player. You will use LEDs and seven segment display (SSD) to display inputs, turns, score, and ships sinking on FPGA.

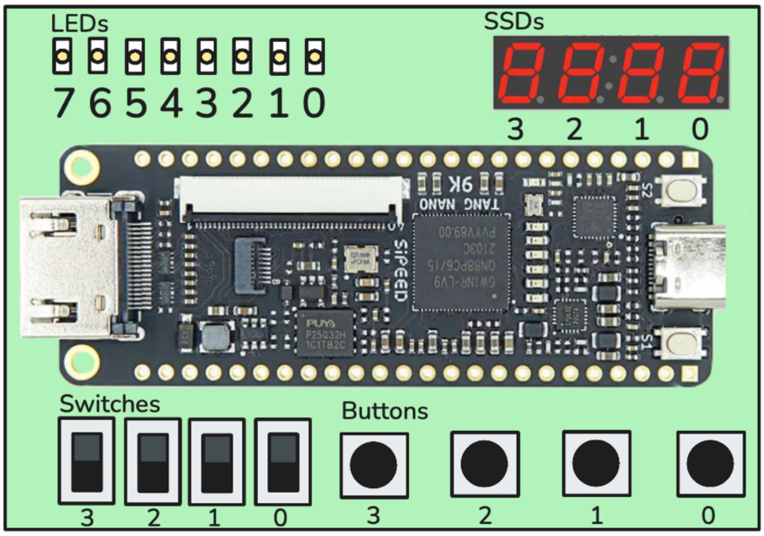

Figure 1 - FPGA Board

Assume that we have two players: A standing on the left-hand side and B on the right-hand side. The players will act using the leftmost and rightmost buttons (BTN3 for A, BTN0 for B), reset the game using “`rst`” (BTN2) and start the game using “`start`” (BTN1). After resetting, the board will display “IDLE” in the SSDs and light up the LEDs 7,4,3,0. After starting, the game will take A’s ship coordinates first. The players will enter their coordinates using switches and pressing their buttons. Switches from 3 to 2 (MSB to LSB) will be the X coordinate, and switches from 1 to 0 will be the Y coordinate. It will take four inputs per player. It will show the player (“A” or “B”) on SSD3 before asking for players’ inputs for one second. When taking the inputs, the leftmost LED will be lit for A, and the rightmost LED will be lit for B. Again while taking the inputs, SSD1 will display the live X coordinate (according to switch positions of switches 3,2) and SSD0 will display the live Y coordinate (according to switch positions of switches 1,0) in real time. The input count will be shown in the LEDs 5 to 4 (MSB to LSB) for A and 3 to 2 for B. The game will show “Erro” on SSDs when the player tries to place a ship on an existing ship, and light up LEDs 7,4,3,0 for one second.

After getting the inputs from both players, the initial score will be shown in the corresponding SSDs (SSD2, SSD1, SSD0) as “0-0” and the LEDs 7,4,3,0 will light up for one second. Then, starting with A, the players will take turns to shoot the ships’ coordinates on the opponent's map using their switches and buttons similar to when they were taking inputs. The X, Y coordinates will be shown on SSDs (SSD1 for X, SSD0 for Y) again in real time. While shooting, the score will be displayed on the LEDs (LED 5-4 for score of A, LED 3-2 for score of B). Also, the leftmost LED will light up indicating the turn for A, and the rightmost LED will light up indicating the turn for B.

When a player shoots at the given coordinate, the SSDs will show the updated score (“A-B” in SSD2, SSD1, SSD0) and light up all LEDs for one second if it sinks a ship. If the player does not sink a ship, all the LEDs will turn off for one second.

When a player’s score reaches four, they win the game. In this case, the score (in score SSDs) and the winner (SSD3) will be displayed. And all LEDs will blink with 1-second period (as a designer, you can change this LED dance pattern to light up certain LEDs and do a different pattern).

Appendix A – Extra Modules Needed for the Implementation on FPGA

In the final implementation of your circuit, you will need some extra modules, with which we provide you under the assignment. These are “clock divider”, “debouncer” and “seven segment driver” modules. There are comments inside the modules about the units the modules implement.

- ➢ The clock divider module (clk_divider.v) generates a clock signal with a frequency of 50 Hz, from a 100 MHz input clock
- ➢ The debouncer (debouncer.v) circuit gets the input from a push button and detects the rising edge of the push button signal. This module will be driven by the divided clock.
- ➢ The seven segment driver (ssd.v) module, drives the segments. This module will be driven by the clock signal, which is generated by the oscillator of the board. Also, your circuit must use the divided clock.

An SSD and its control signals ‘pgfedcba’ are shown below. Using 8-bit ‘pgfedcba’ control signals, you can display different digits and signs on an SSD. For example, ‘pgfedcba’ should be ‘00111111’ to display 0 and ‘01001111’ to display 3. There are four SSDs on the FPGA board. Your Verilog module should have 8-bit output for each SSD.

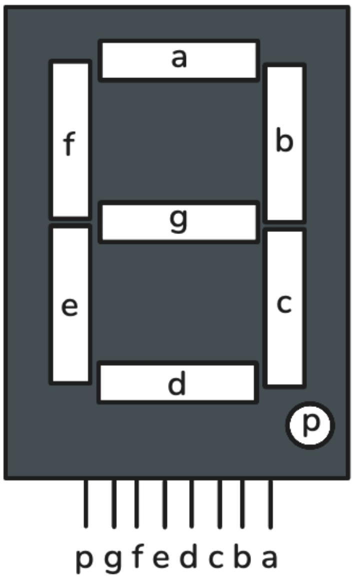

Figure 2 - Seven Segment Display Control Signals