Term Project

Assigned: 19/12/2023
Due: 07/01/2024 23:55

Electronic Air-Hockey Machine

You will design an electronic game of air-hockey 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 5x5 area. You will use LEDs (i.e. LD9 to LD5 from right to left and left to right) and a seven segment display (SSD4) to simulate the moving puck in X and Y axis, respectively, on FPGA. A puck moving from right to left is simulated as LEDs illuminating one after another starting with LD9 (i.e. LD9 illuminates the leftmost column, then LD8, LD7, etc. It is basically a moving light). SSD5 will display the Y-coordinate of the puck in decimal form.

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 left and right buttons (BTNL and BTNR). Before the players start playing, a player will take the first turn using his/her button. Then, the initial score will be shown in the corresponding SSDs (SSD2-SSD0) as “0-0” for 2 seconds. Then, the score will disappear. Assume that, player A takes the first turn. After the initial score was displayed, LD15 will turn on to show an input is waited from the player A. Player A will give the Y-coordinate and the direction which s/he wants to send the puck using his/her sliding switches (SW15-SW13 for Y-coordinate and SW12-SW11 for the direction). Then, s/he will press his/her button to send the puck and the puck will start moving. At the same time, LD15 will turn off (because no input is waited from player A). The speed of the puck will be 2 seconds/step. After 8 seconds, the puck will arrive to the right-most column. When, the puck arrives the right-most column, player B must hit the puck. In order to inform player B, when an input is expected from him/her, LD0 will turn on. When (or before) the puck arrives to the right-most column, player B must adjust the his/her SWs (SW2-SW0 for Y-coordinate and SW4-SW3 for the direction according to the Y-coordinate of the puck and the direction that s/he wants to send the puck. Then, s/he must push his/her button to hit the puck and the puck will start going from right to left. When, s/he hits the puck, LD0 will turn off. Then, the same procedure will be done by player A. And so on.

When a player cannot hit the puck in a second (either s/he does not press his/her button or Y-coordinate is invalid), the other player will win the turn. In this case, the current score will be displayed and LD9-LD5 will turn on for a second. Then, the player who lost the previous turn, will start the next turn.

When a player scores 3 goals, s/he will win the game. In this case, the score (in score SSDs) and the winner (Y-coordinate SSD) will be displayed. And LD9-LD5 will blink with 1 second period.

For the rest of the design details, you may visit office hours.

Make-up Lab
You can do the followings as a make-up lab:

- Your circuit can adjust how fast the light moves. You should demonstrate at least two speed values (0.5 and 1 second period options).
- The number of goals to win the game (between 1-7) and the size of the playground (between 4-7, always square) can be selected by the users.

  You are required to design the procedures by yourself and explain in the demonstration.

Project Plan and Deadlines
Students must follow a project plan and demonstrate that they met a specific deadline by submitting their work. Each work item in the project plan will be graded separately. The project plan and grade of each work item is as follows:

1. <u>REQUIREMENTS:</u> Project requirements are given to the students.
   Dec 19, 2023. (not graded)
2. <u>STATE_DIAGRAMS</u>: State diagrams are submitted in hard copy as instructed.
   9 am, Dec 26, 2023; (10 %)
3. <u>SIMULATION</u>: Simulation is demonstrated to lab assistants after the deadline of the simulation phase, Jan 1, 2024. 23:55 (30%)
4. <u>DEMO:</u> The circuit is realized on FPGA and a demonstration must be made.
   Jan 7, 2024. (60%) An announcement will be posted for demo sessions later.

<u>Note that these deadlines are hard, and there will be no additional time.</u>

Notes:

- You work with your lab partner in this project.
- You can use Verilog language. (Recommended)
- If you want to demonstrate your design before the deadlines, you can do so by scheduling an appointment with your TA.
- In your design you are **<u>NOT</u>** allowed to use latch.

The instructor has the right to have an oral interview for the term project (until the end of the final exam period.)

- Students who will have the oral interview may be selected randomly or according to a suspicious situation observed by TAs or the instructor.
- Performance of the student in an oral examination may affect their grades of the term project they have been called upon.
- If a student fails to show up at an oral exam in person, (s)he will automatically get 0 (zero).

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 in 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 generates a one clock pulse output. 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 should also use the divided clock.

An SSD and its control signals 'abcdefg' are shown below. Using 7-bit 'abcdefg' control signals, you can display different digits and signs on an SSD. For example, 'abcdefg' should be '0000001' in order to display 0 and '0000110' in order to display 3. There are eight SSDs on the BASYS board. Your Verilog module should have 7-bit output for each SSD.

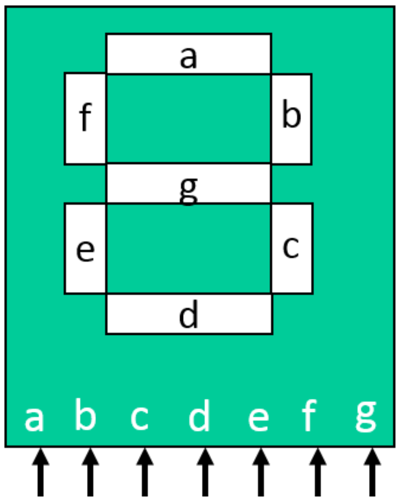

Fig. SSD control signals