Shikha Mody (604922996)

Brendon Ng (304925492)

Armaan Singh (404932331)

COM SCI M152A Lab 5, TA: Logan Kuo

19 February 2020

Lab 4 Proposal

**What We Want To Do**

For our Lab 4 project, we would like to implement a Whack-a-Mole game on the FPGA board. We will use the switches, LED lights, seven segment display, buttons, and Verilog modules to implement this design. The mechanism for this game is as follows:

Each individual LED represents a “mole.” The switches correspond to the “whack.” Points are earned by quickly flipping the switch corresponding to which LED light is on. By flipping the switch (either from OFF to ON or ON to OFF) within a certain amount of time, the mole is effectively “whacked,” meaning that the LED now turns off and another random LED turns on. The score will be displayed on the seven-segment display of the FPGA board. Each mole that is whacked within the time limit will add to the point total. If a mole gets missed (i.e. not whacked within the time limit), or incorrectly “whacked,” the game is over. Buttons will be enabled for “play again” functionality (essentially reset the game), and choosing different levels of difficulty (easy, medium, hard). Each level will have a different amount of time before the mole expires and the whack is missed, effectively ending the game.

This project combines skills we have learned in previous labs. We will have multiple internal clocks, similar to the functionality we implemented during Lab 3, different Verilog modules for playing the game, and a state machine for the seven segment display as we did in Lab 3.

**Preliminary Research**

In Lab 4, our project will require us to use multiple libraries and functions in order to design our game. One such library is std::randomize(), which will allow us to generate random numbers. Since Whack-a-Mole is a game designed around reaction time, it is important that we randomize each game instance, so that patterns cannot be memorized.

In addition, in order to see how well a player is doing, the score must be reported to them. This will be done using the seven segment display similar to how it was used in Lab 3. Hoeverer, instead of displaying the time, the value displayed will only change when a mole is whacked. The seven-segment display manual that we used for Lab 3 will be useful to implement the appropriate state machines and output assignments for the display. In addition, we would like to display the level on the 7 segment display. This involves a more complex state machine to be able to determine the level when editing the specific "level" digit on the display. New shapes will be needed to display an "E" for easy, "N" for normal/medium, and "H" for hard.

Also, it is important for the player to be able to reset the game at any time, and select the difficulty before starting a new game. All of these buttons will require debouncing algorithms, as seen in Lab 2 and Lab 3, which will require setting up a sampling clock at an appropriate frequency.

Finally, the game will also need a timer, which will require an implementation of a countdown similar to that in Lab 3. This will provide the time limit for the player, within which they must whack a mole depending on the difficulty selected. This is not a timer for the general game; rather, it is the time taken to actually whack the mole (flip the switch). This timer will need to be running in the background. Also, for different levels, this timer will need to be modified (i.e. for a difficult level, the timer should be shorter than the easiest level). We will need to implement a straightforward method to set this limit depending on the level, and then use it to determine if the user wins or loses.

**Functions/Specifications/Grading**

Because we are implementing a game, the functions and grading should be based on being able to actually play the game.

Functions to implement and be graded upon:

* 25%: Correctly registering a “whack” of a “mole”
  + LED lights up → when the corresponding switch is flipped within the time limit, the light turns off; if the switch is not flipped within the time limit, game is over
  + Synchronization of LEDs and switches (no perceivable delay)
* 10%: Incrementing score on display
  + Each correct “whack” increments the score on the seven-segment display
* 20%: Timer capabilities
  + Each level has a different (unknown to the player) time limit for registering a “whack” to the “mole.” If the user does not successfully whack a mole within this limit, the game is over
  + See/feel a difference between the different levels (i.e. the timer differences are noticeable)
* 10%: Button functionality: restart, choose level
  + Successfully choose level and be able to play and replay the game using buttons (debounced)
* 15%: Randomization of LEDs lighting up
  + Use relevant library functions to randomize LED such that there is no pattern
* 20%: Game over functionality
  + If the timer for whacking a mole is exceeded, indicate that the game has ended. To replay, press a button. Otherwise, the game stays in this state (no game being played)
* *5% extra credit depending on Logan's Whack-A-Mole score on "Hard" :)*