Armaan Kohli, Ivan Chowdhury
ECE150: Digital Logic Design
Final Project: Speed Racer
Professor Risbud

Final Project Design Proposal

Abstract

For our final project, we will create a game modeling Speed Racer, the popular 2-D arcade game. An 10x8 LED matrix display consisting of 80 LEDs will be used, wherein the player controls the bottom row of 8 LEDs. Two pushbuttons will be used to control the movement of a single light within the row of LEDs. There will also be three score displays - representing player score, death score, and in-game level - in the form of four 7 segment displays. As the game begins, barrier-esque patterns of lights with gaps within them will descend down the LED matrix display, with the objective of the game being to maneuver the player-controlled light through the gaps in the barriers. As the game extends, the player score will increment, as shown on the corresponding display. The score will be based on the duration of time the player can stay alive. If the player fails to maneuver through a barrier, the death score will increment. As the player score reaches certain thresholds, the speed at which the barriers descend will increase, creating levels for the player to progress through. After reaching a death score of three, the game will end.

The circuit is described in a general sense by the above block diagram. The resistor multiplexer/demux system will alter the frequency of the game clock by changing the resistor values on a 555 timer. The resistor value is selected by counter 2, which increments slowly. This is what allows for the difficulty to increase over time. The game clock is then tied to counter 1, which produces inputs to the random number generator for each pulse. The random number generator will be made of a 555 timer that is clocked to a very high frequency. The random number is then sent to a multiplexer address in the multiplexer array, which controls the LED array. Once the multiplexer sends a signal to the LED array, a decade counter is used to light the LED's down the row using the signal from the game clock as a timer. The player, able to move the bottom row of LEDs, presses a push button, which increments an up/down counter to change the position of the player. And the location of the player is also sent to the multiplexer array to be displayed. The location of the player and the falling blocks are compared by the comparator. If they occupy the same space, then the player loses a life. The comparator is connected to a counter, which tallies the number of times a player and a barrier occupy the same space. When the counter reaches a value of three, all of the clocks are stopped, and the game ends. A one Hertz clock signal is sent through two 4-bit counters to two seven segment displays for score keeping purposes; the score will reset if a player's death count reaches 3.

Player Experience

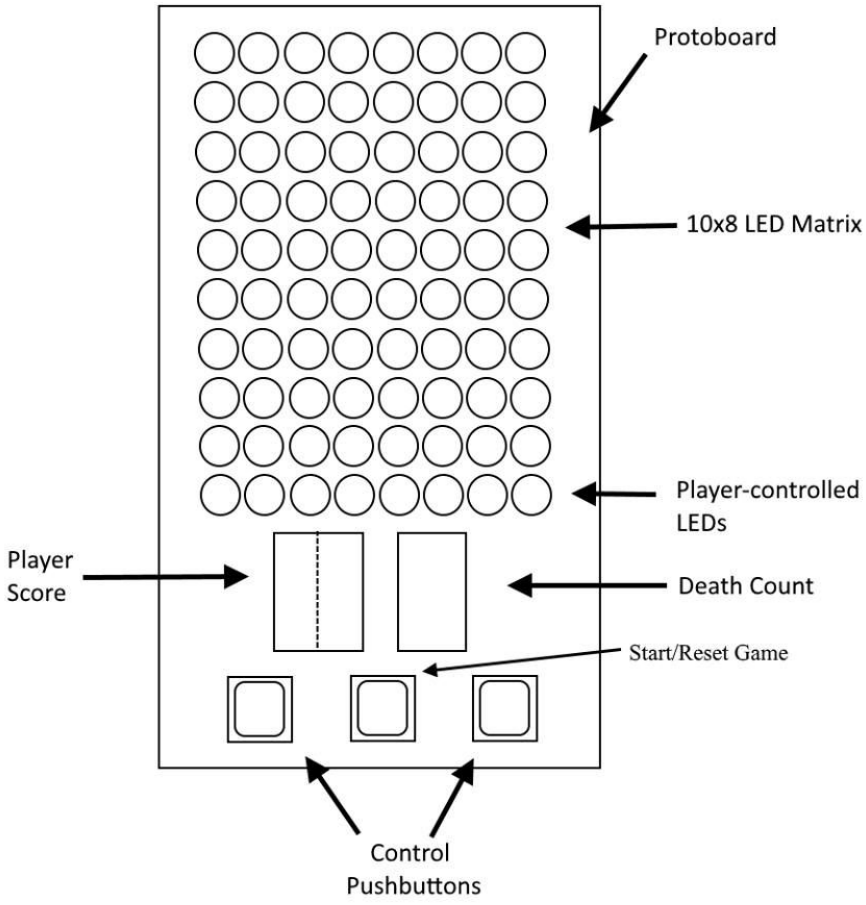

Figure 1

The above is an image of the game board. If the player hits the control buttons, the car will move left or right respectively. When the player hits a barrier, the player's death count increases. The player's score keeps increasing the longer they avoid crashing. When the player has crashed three times, the game ends. As the game progresses, the speed at which the barriers appear increases, thus increasing the difficulty as time continues to provide a more rewarding player experience.

Inventory

Part	*Use*	*Quantity*
CD4051 Analog Mux/Demux	To controls LEDs, Game Clock	11
LM555 Timer	Game Clock, Timer Clock x2, Fast Clock	4
Push Buttons	Player Controls	4
LEDs	Display	80
Breadboards	To Wire the Circuit	10
7 Segment Displays	Displaying Score, Lives, Levels	4
Protoboard	Display	1
SN74LS85 Comparator	Comparator to test addresses	1
4029 Binary Up Down Counter	To change player position left or right	1
4040 12 Stage Binary Counter	Counters for game clock, fast clock and life counter	3
4009 Hex Buffer	To use both 4000 series and 7400 series chips	1
4013 Dual D-Type Flip Flop	To control the display by storing the LED positions for the Muxes	5
4028 BCD Decoder	To use seven segment displays	4
CD4026 Decade Counter	To generate random numbers and to control the LED array	9
CD4073 3-Input AND Gate	Used to reset the counters	1

Design

Game Clock

The game clock was made using a four bit counter that increments every twenty seconds. This counter is used to control a mux and demux that selected R_2 of another 555 timer. Every twenty seconds, the counter would increment, changing the R_2 value of the resistor, which would increase the frequency of the 555 timer, which is used to control the generation of random numbers. Thus, a leveling system is created. Every twenty seconds, the frequency at which the blocks fall is increased, making levels to the game.

Player Controls

The player controls the location of their LED with two push buttons. Pushing the left button will move the player left, pushing the right button will move the player right. Pressing the buttons increments a binary up/down counter, whose output is sent to a mux that is connected to the bottom row of LEDs. The buttons control not only whether the counter is counter up or down, but also the clock of the counter, so the counter holds its value until another button is pressed. Because we are using a three bit binary up/down counter, the player should be able to wrap around the screen.

Scoring and Life Testing

The score counter is presented to the player in two seven segment displays. This is done using a 1 Hz clock and two 4-bit counters. The score is how many seconds the player stays alive; as the player continues to survive, his score will increase. When the player dies, and the counter

is ended. The death counter increments when the player hits a barrier. This test is made by a comparator. The comparator tests the address of player and the address of the barrier on the row above the player row. If the two address are different, the game continues. If the two addresses are the same, then the output of the comparator will be used to increment a counter which is used to drive a seven segment display, showing how many times the player has lost a life. When the counter becomes three, the game ends.

Display

The display, or user interface, of the project is based around a protoboard. On the protoboard will be an array of LEDs, two seven segment displays for score counting, one seven segment display as a death tally, the two push button player controls, and a push button to reset the game. See figure 1.

Barrier Movement

The random number generator was made using a fast clock and a CMOS4026 display driver. The random number generator block is constantly generating random numbers, and a three bit counter, being incremented by the random number generator, sends an address to an array of muxes, which determine which column the barrier appears in. Once the column has been selected, the barrier progresses down the LED array by a counter that increments based on the 555 timer. This counter controls a mux, and as the counter increments, the barrier will progress down the array.