CSE 316

(Microprocessors, Microcontrollers, and Embedded Systems Sessional)

Name of The Project:

Bounce and Rush

Supervisor	**Dr. Atif Hasan Rahman**
Section	**B2**
Department	**CSE**
Group Members	**1705109** **1705111**
Date Of Submission	**30 July, 2021**

Contents

1 Introduction

In our project we have recreated the offline chrome dinosaur game in our own way. This is a single player game where the player guides a side-scrolling landscape and avoiding obstacles to achieve higher score. The goal is to achieve higher score

2 Components

1. 2-4 Decoder (74HC139)
2. 4-16 Decoder (74HC154)
3. ATMEGA32
4. 2x16 LCD Module (LM016L)
5. 8x8 LED Matrix
6. Resistor 300 ohm
7. SPST Push Button

3 Working Principle

In our project AtMega32 micro-controller is used for interacting with other hardware components. This game is controlled by two SPST push buttons. The player and the obstacles are shown in 6 8x8 LED Matrix. One push button is used to jump and the other one is to restart the game. As long as the player is not hit by any obstacle the score increases in a certain rate and it is shown consistently in the LCD. If the player is hit by the obstacle, then the player can restart the game using one of the buttons.

4 Detailed Circuit Diagram

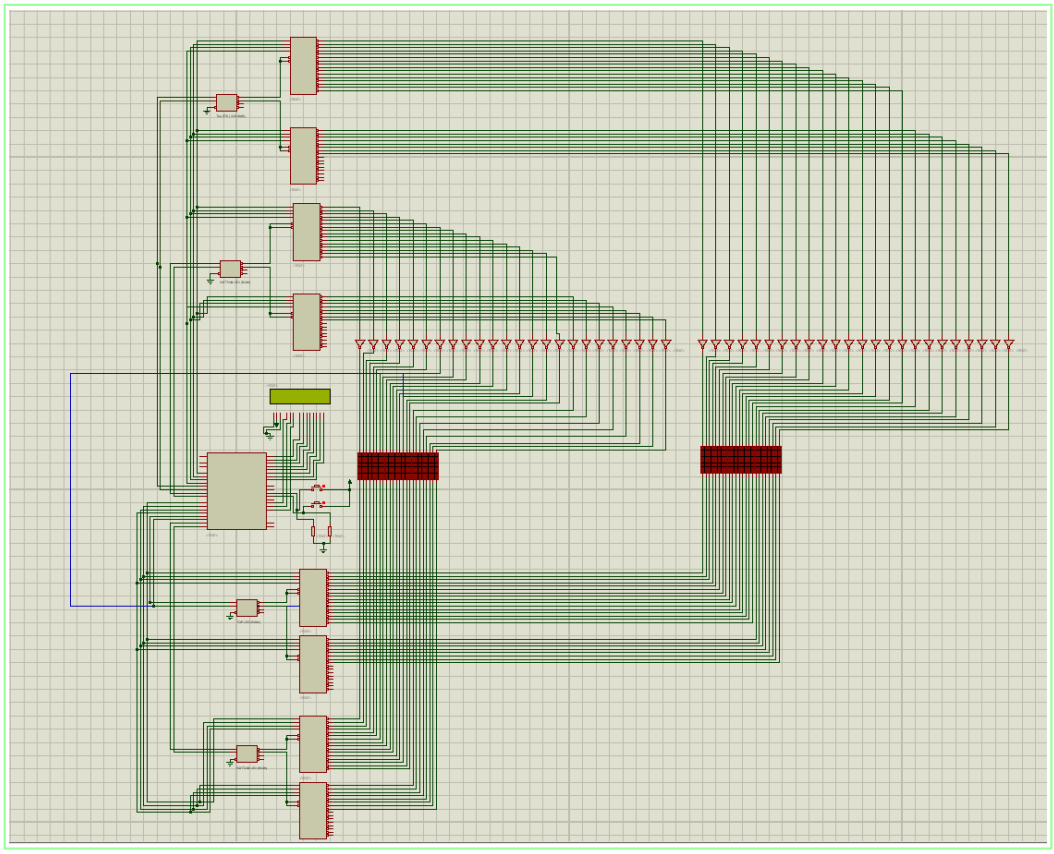

Figure : Full Circuit

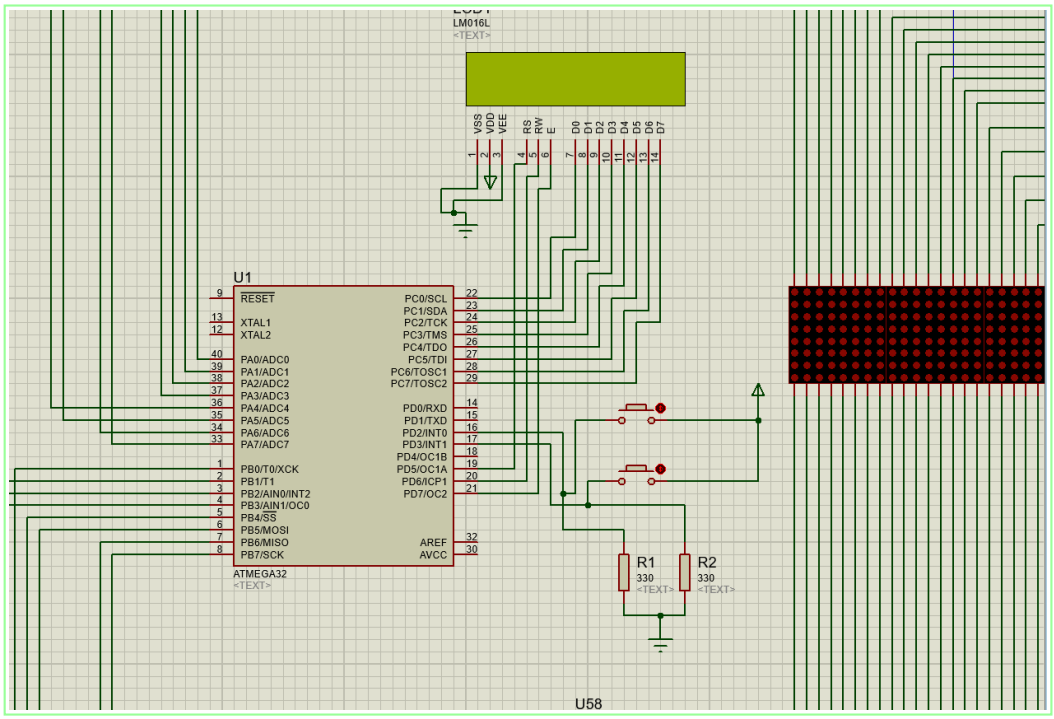

Figure : ATMEGA32 Connections

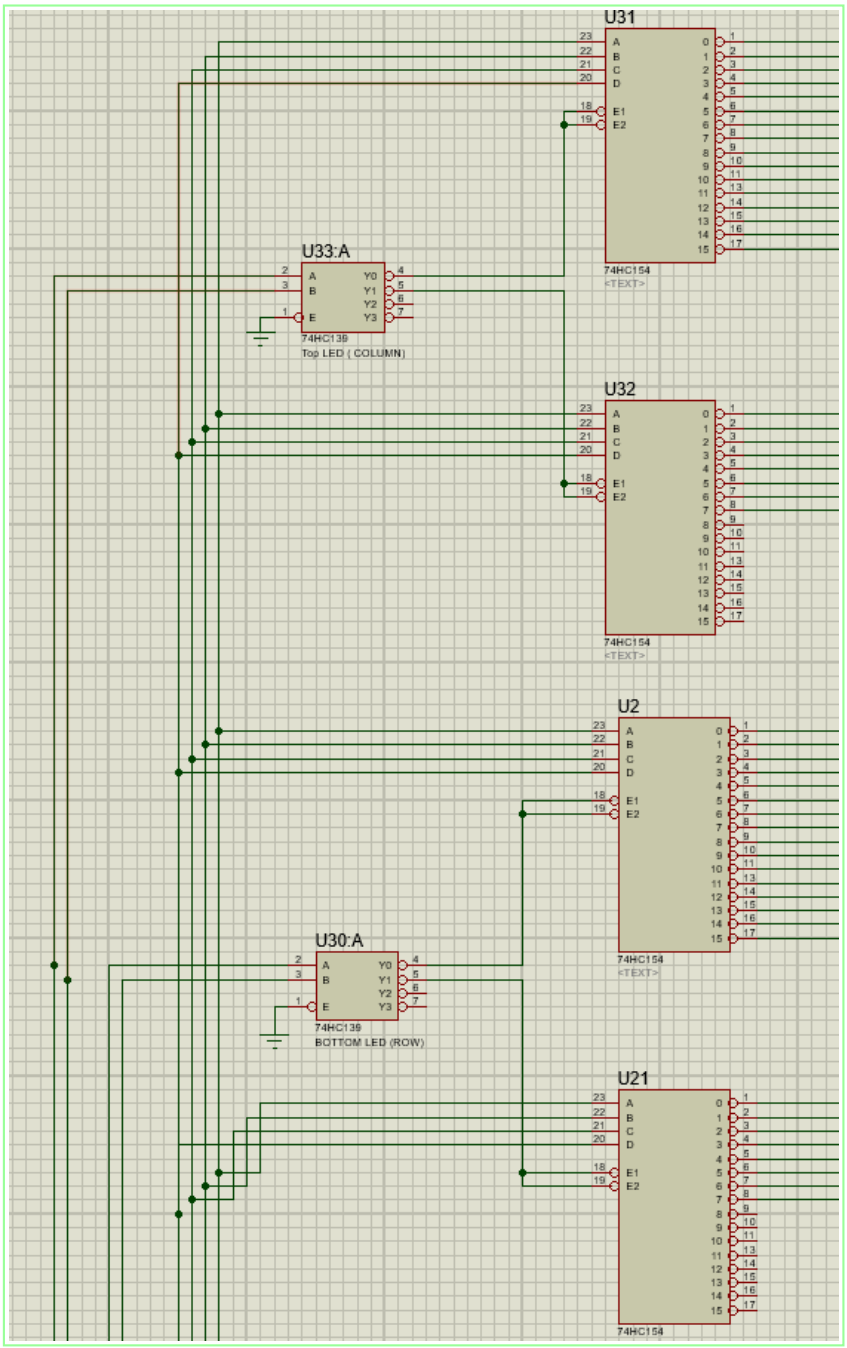

Figure : Decoder Connections

5 Algorithm

1. Start and LED display shows the score
2. If **button.jump** pressed, the dinosaur will jump
3. If score increases the difficulty level increases
4. If player is hit by obstacle, show 'GAME OVER'
5. If **button.restart** pressed, the game will restart

6 Discussion

1. **Handling 6 LED Matrices** : The 6 LED Matrix is connected with ATMEGA32 in such a way that the dinosaur and the obstacles can travel through one to another. At first we tried to implement our code in such a way that we can select a one column and multiple rows to create a obstacle. Initially it didn't make any problem but when we used more than 2 LED Matrix we faced problems to move our object. To counter this problem, we selected only a single dot at a time to create the object and move it through the LED. As there are 6 8x8 LED and we had to make sure we can control each LED. We needed sufficient amount of 4x16 decoders to connect every pins of the 6 LED Matrices. But we again faced problem for connecting the decoders with ATMEGA32. We solved that by using 2x4 decoder which enabled the 4x16 decoders and that 4x16 decoders got connected to the LED Matrices.
2. **LCD and delay** : We used only a single LED at a time, we had to make sure the delay(*_delay_ms()*) is low enough to get smooth transitions of the obstacles and dinosaur. Before connecting the LCD display everything was smooth but after connecting it we faced that the delay got higher. We tried to make our delay (*_delay_ms()*) as low(1ms) as possible to get the LEDs blinking but still we didn't manage to get smooth obstacles and dinosaur.
3. **Collision** : Collision was handled in such a way that if any part of the dinosaur hits the obstacle it is considered as a collision.

4. **LCD** : The score is updated every moment in the LCD. For this we refreshed a certain part of the LCD every time and displayed the score after the previously printed string('SCORE : ').

5. **Decoder** : 4 2x4 decoders were used to handle 8 4x16 decoders. For PORTA and PORTB of ATMEGA32 0-3 ports were used to control the 4x16 decoders and the last 4 bits were used to control the 2x4 decoders and enable them.