

Course Project
1D LED Ping Pong game using Shift Register

ME5323

Course Instructor: Dr. Somashekhar S.

Members:

Adil Mohammed K ED19B041

Suraj Rathi ME19B177

Sai Karthik Brahma ME19B044



Odd Semester July - November 2022

Contents

1	Introduction	2
1.1	Shift Register	2
1.1.1	D Flip Flop	3
2	Build Setup	4
2.1	Components Used	4
2.2	Shift Register	4
2.3	Arduino	5
2.4	Circuit Diagram	6
2.5	Code	7
2.6	Results	9
3	Conclusion	11

Introduction

Pong is a table tennis-themed twitch arcade sports video game, featuring simple two-dimensional graphics, manufactured by Atari and originally released in 1972. It was one of the earliest arcade video games; it was created by Allan Alcorn as a training exercise assigned to him by Atari co-founder Nolan Bushnell, but Bushnell and Atari co-founder Ted Dabney were surprised by the quality of Alcorn's work and decided to manufacture the game. Bushnell based the game's concept on an electronic ping-pong game included in the Magnavox Odyssey, the first home video game console. In response, Magnavox later sued Atari for patent infringement.

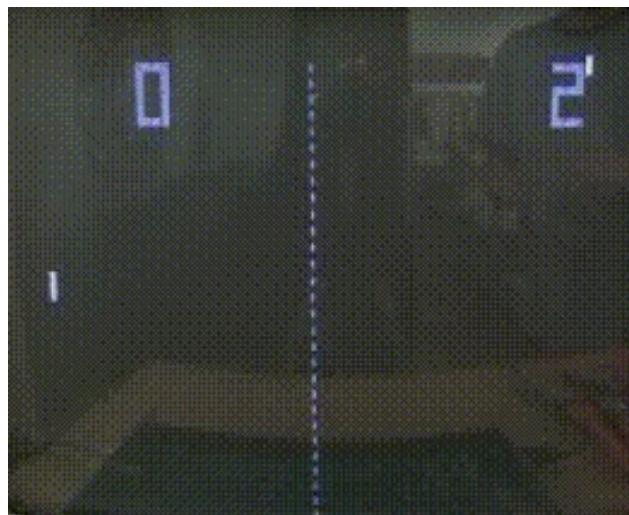


Figure 1.1: Pong Game

To simulate this game, we will be using Shift Register and an Arduino which help multiplex the output of the LED's. The LED's will be arranged in a 1D array and the ball will be moving from one end to the other. The player will have to hit the ball back to the opponent. The player who misses the ball will lose the game. The game will be played for 3 rounds and the player who wins 2 rounds will be the winner.

1.1 Shift Register

Flip flops can be used to store a single bit of binary data (1 or 0). However, in order to store multiple bits of data, we need multiple flip flops. N flip flops are to be connected in an order to store n bits of data. A Register is a device which is used to store such information. It is a group of flip flops connected in series used to store multiple bits of data.

The information stored within these registers can be transferred with the help of shift registers. Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip flop stores a single bit of data.

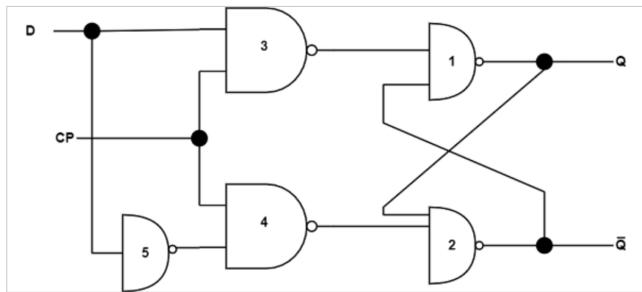
Shift registers are basically of 4 types. These are:

- Serial In, Serial Out (SISO)
- Serial In, Parallel Out (SIPO)
- Parallel In, Serial Out (PISO)
- Parallel In, Parallel Out (PIPO)

1.1.1 D Flip Flop

The D flip-flop is a clocked flip-flop with a single digital input 'D'. Each time a D flip-flop is clocked, its output follows the state of 'D'. The D Flip Flop has only two inputs D and CP. The D inputs go precisely to the S input and its complement is used to the R input.

Considering the pulse input is at 0, the outputs of gates 3 and 4 are at the 1 level and the circuit cannot convert state regardless of the value of D. The D input is sampled when CP = 1. If D is 1, the Q output goes to 1, locating the circuit in the set state. If D is 0, output Q goes to 0, and the circuit switches to a clear state.



(a) D Flip Flop

S	D	Q_{N+1}
0	0	0
0	1	1
1	0	0
1	1	1

(b) Truth Table

Figure 1.2: D Flip Flop

Build Setup

2.1 Components Used

- **Arduino Uno:** Arduino board is a microcontroller kit for building digital devices.
- **IN74HC595A Shift Register:** Shift Register is a group of flip flops used to store multiple bits of data.
- **Breadboard:** A breadboard is a construction base used to build semi-permanent prototypes of electronic circuits
- **Jumper Wires:** A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards
- **LEDs:** A light-emitting diode (LED) is a semiconductor device that emits light when current flows through it
- **Resistors:** A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element

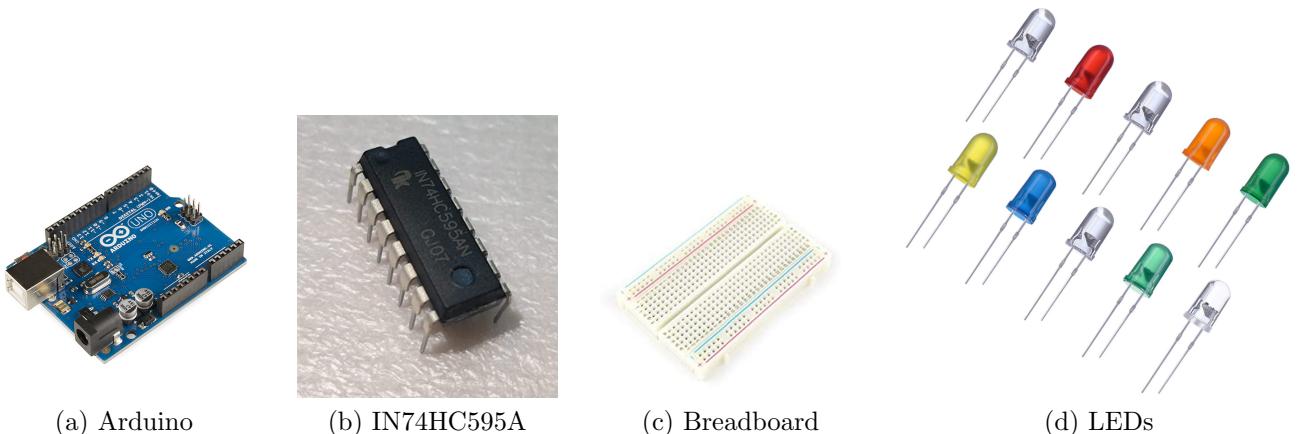


Figure 2.1: Components Used

2.2 Shift Register

IN74HC595A is a 8-bit shift register. It has 8 output pins and 3 input pins. The 3 input pins are used to control the shift register. The 8 output pins are used to output the data stored in the shift register. The 3 input pins are as follows:

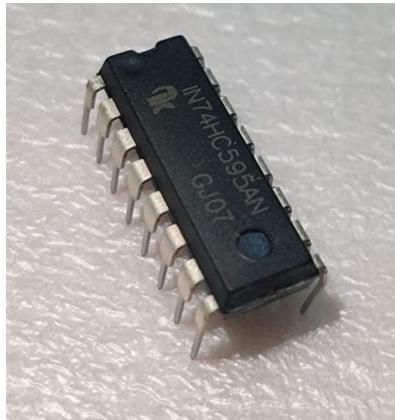
- **Serial Data Input (DS):** This pin is used to input the data to be stored in the shift register.

- **Shift Clock (SHCP):** This pin is used to control the shifting of data in the shift register. When the clock is high, the data is shifted to the next flip flop. When the clock is low, the data is not shifted.
- **Store Clock (STCP):** This pin is used to control the storing of data in the shift register. When the clock is high, the data is stored in the flip flop. When the clock is low, the data is not stored.

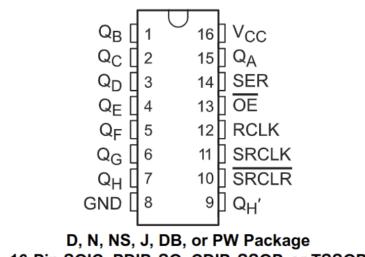
The other pins are:

- **Serial Data Output (Q7):** This pin is used to output the data stored in the last flip flop.
- **Master Reset (MR):** This pin is used to reset the shift register. When the pin is high, the shift register is reset. When the pin is low, the shift register is not reset.
- **Output Enable (OE):** This pin is used to enable the output of the shift register. When the pin is high, the output is enabled. When the pin is low, the output is disabled.
- **VCC:** This pin is used to supply power to the shift register.
- **GND:** This pin is used to supply ground to the shift register.

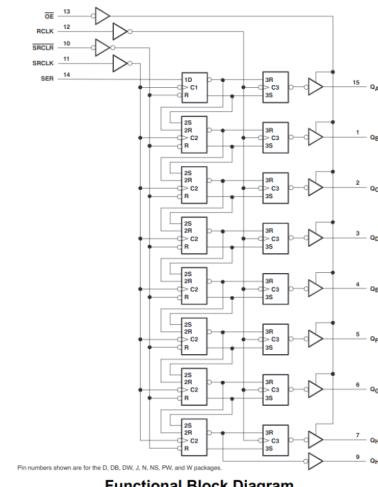
The IC can be used as a SIPO or a SISO shift register. The data is shifted one bit at a time when the shift clock is RISING edge triggered. Subsequently, when the latch clock is RISING edge triggered, the data is stored in the output pins.



(a) IN74HC595A



(b) Pin Diagram



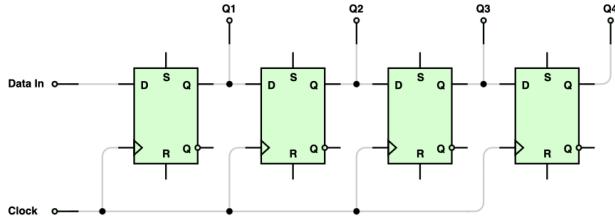
(c) Functional Diagram

Figure 2.2: Shift Register

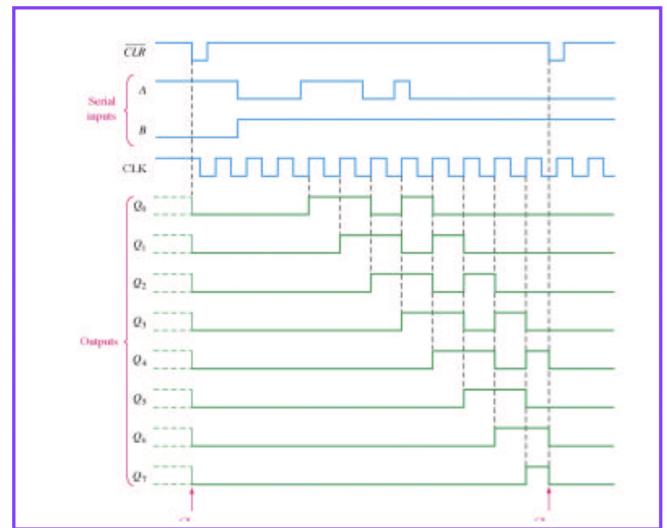
The following is the signal flow and waveforms for the above mentioned pins:

2.3 Arduino

The Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. It is programmed using the Arduino programming language (based on C++).

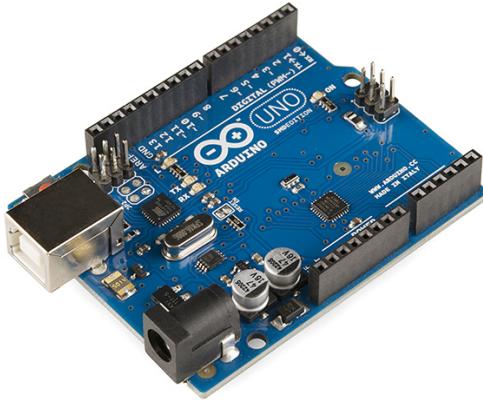


(a) Signal Flow

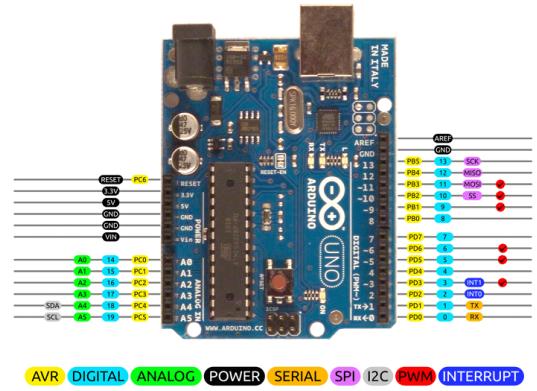


(b) Signal Waveform

Figure 2.3: Signal Flow and Waveforms



(a) Arduino Uno



(b) Pin Diagram

Figure 2.4: Arduino Uno

2.4 Circuit Diagram

The following circuit diagram shows the connections between the components used in the project. The Arduino was connected to the shift register's data, clock and latch pins. The shift register's output pins were connected to the LED's. The LED's were connected in a 1D array.

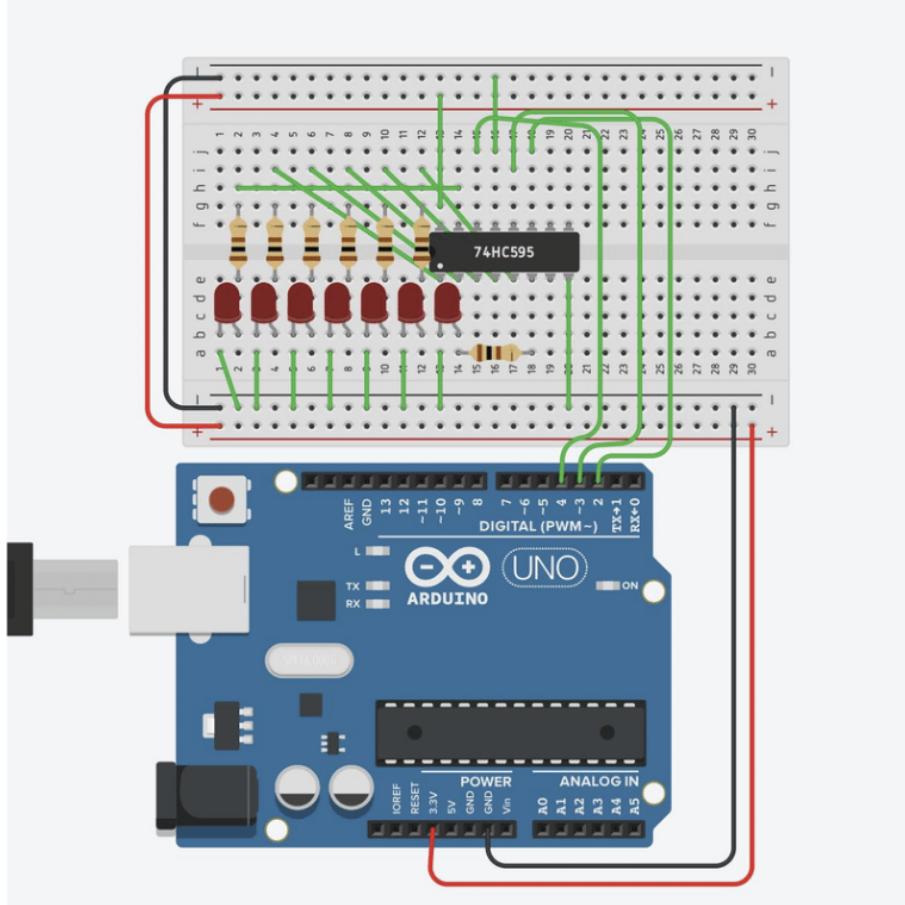


Figure 2.5: Circuit Diagram

2.5 Code

The following code was used to control the shift register. The code was written in the Arduino IDE. The code was uploaded to the Arduino board. The code was written in C++.

```

1 #include <Arduino.h>
2
3
4 const int datapin = 8;
5 const int clockpin = 9;
6 const int latchpin = 10;
7 const int button1_pin = 2;
8 const int button2_pin = 3;
9
10 // We'll also declare a global variable for the data we're // sending to the
11 // shift register:
12 byte data = 0;
13
14 int index = 0;
15 int direction = 1;
16 bool game_over = false;
17
18 volatile bool button1 = false, button2 = false;
19
20 void callback1() { button1 = true; }

```

```

21 void callback2() { button2 = true; };
22
23 void shiftWrite(int desiredPin, boolean desiredState)
24 {
25     bitWrite(data, desiredPin, desiredState);
26     shiftOut(datapin, clockpin, MSBFIRST, data);
27
28     digitalWrite(latchpin, HIGH);
29     digitalWrite(latchpin, LOW);
30 }
31
32 void setup()
33 {
34     Serial.begin(9600);
35
36     pinMode(datapin, OUTPUT);
37     pinMode(clockpin, OUTPUT);
38     pinMode(latchpin, OUTPUT);
39
40     pinMode(button1_pin, INPUT_PULLUP);
41     pinMode(button2_pin, INPUT_PULLUP);
42
43     attachInterrupt(digitalPinToInterrupt(button1_pin), callback1,
44                     FALLING);
44     attachInterrupt(digitalPinToInterrupt(button2_pin), callback2,
45                     FALLING);
45
46     Serial.begin(9600);
47
48     for (int i = 0; i < 8; i++)
49     {
50         shiftWrite(i, LOW);
51     }
52 }
53
54 long cycles = 0;
55 bool cycle_flag = false;
56 long delay_time = 250;
57 void loop()
58 {
59     if (!game_over)
60     {
61         if (index == 1 && button1 == true) {
62             direction = 1;
63             cycles++;
64             cycle_flag = true;
65
66         }
67         if (index == 6 && button2 == true) {
68             direction = -1;
69             cycles++;
70             cycle_flag = true;
71     }

```

```

72
73     button1 = false;
74     button2 = false;
75
76     if (cycle_flag && cycles % 2 == 0)
77     {
78         delay_time = max(51, static_cast<long>(delay_time *
79                         0.85));
80         cycle_flag = false;
81     }
82
83     Serial.print(delay_time);
84     Serial.print(" ");
85     Serial.print(cycles);
86     Serial.print(" ");
87     index += direction;
88     shiftWrite(index, HIGH);
89     shiftWrite((index - direction), LOW);
90
91     if (index == 0 || index == 7)
92     {
93         game_over = true;
94         return;
95     }
96     Serial.println(index);
97     delay(delay_time);
98 }
99 else
100 {
101     for (int i = 0; i < 8; i++)
102         shiftWrite(i, HIGH);
103     delay(100);
104     for (int i = 0; i < 8; i++)
105         shiftWrite(i, LOW);
106     delay(100);
107 }
```

2.6 Results

The circuit was assembled and the code was uploaded to the Arduino board. The following image shows the results of the project. The video of the project can be found at the following link:
https://drive.google.com/file/d/1isGdwd_6Z0JKoO3FJOv7HLgnDQ60ZBAJ/view?usp=share_link

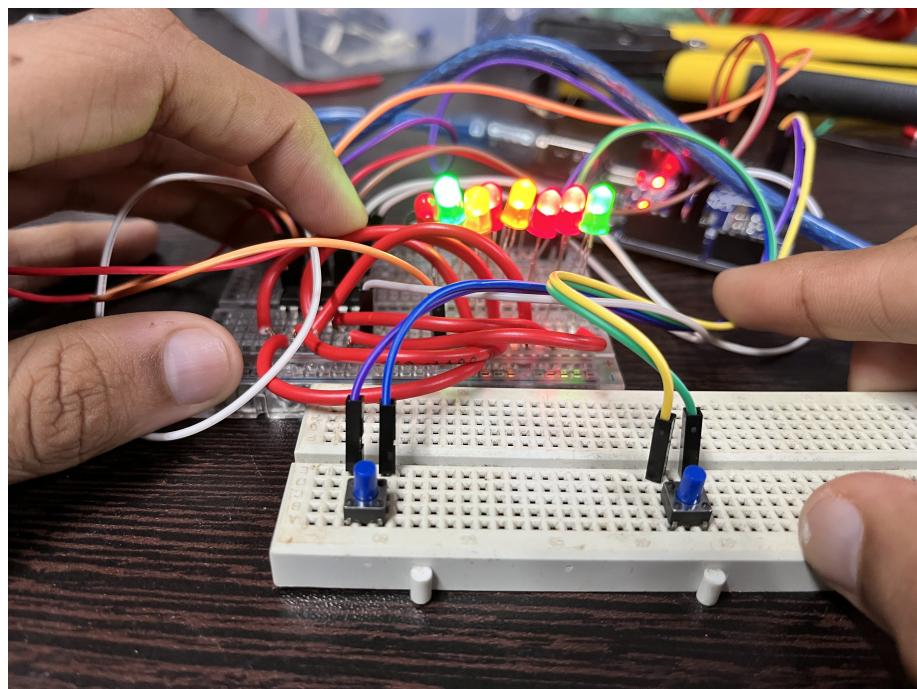


Figure 2.6: Results

Conclusion

We have successfully developed an LED Ping Pong game using a Shift Register and an Arduino.