**Aim:**

Design of a Shift Register Circuit (IC 74HC595) with Manual Pushbutton Control .

**Apparatus:**

74HC595 Shift Register, LEDs, 220Ω resistors, 1KΩ resistors, 10KΩ resistor, Pushbuttons, 47pF capacitor, Tinkercad Simulator.

**Theory:**

In this project, we are going to show how to build a shift register circuit wired to pushbuttons so that we can

manually see how a shift register works. Even though shift registers are made to work with microcontrollers such

as an Arduino, and in the commercial industry, you always see them used with microcontrollers, in this circuit; we

take a manual approach so that you can see exactly how a shift register actually works. Using manual control with

pushbuttons is great for demonstration purposes and sees how shift registers work for those who are beginners

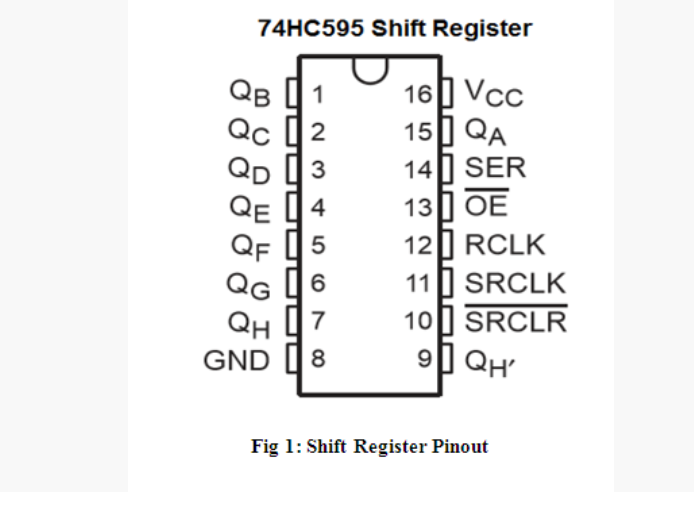
with them. For this circuit, we will have 5 pushbuttons. These pushbuttons will control the serial data input, the

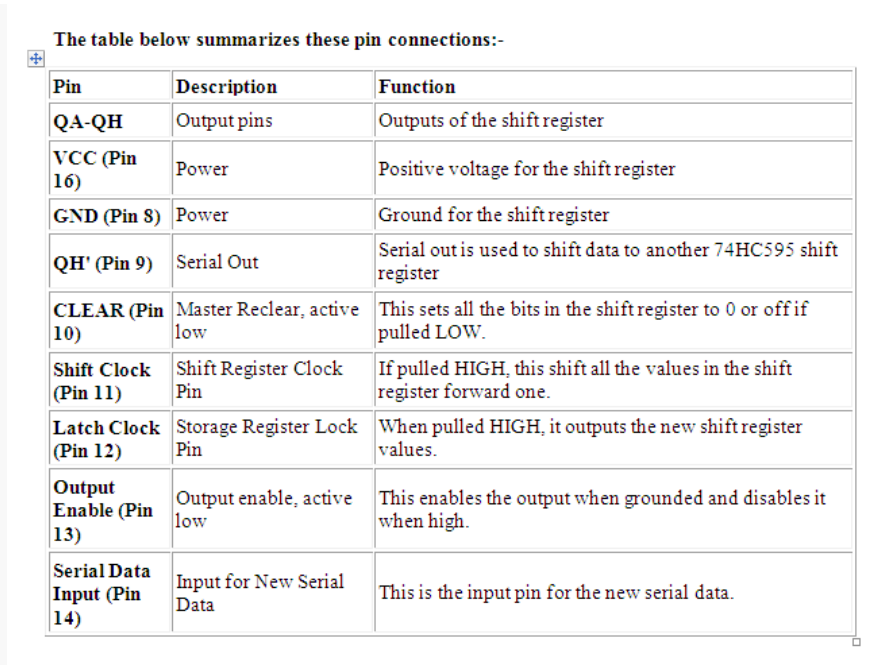
clock line, the latch pin, the output enable pin, and the clear pin. Through manual control of the shift register, you

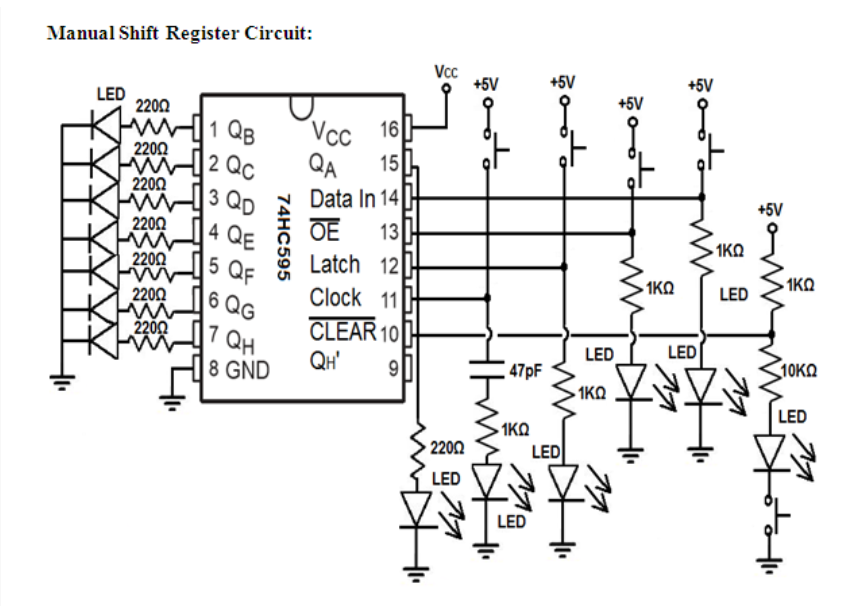
will know exactly how they work in a way that using it with microcontrollers can't teach you, because you're actually

doing the work yourself. The shift register we will use is the popular 74HC595 shift register. This IC can control up

to 8 outputs. In this circuit, our output devices will be LEDs. So one 74HC595 can control 8 LEDs on the output side.







**Working:**

This shift register circuit has pretty basic connections.

First we connect power. So we connect VCC of the shift register to 5V. We connect the ground of the shift register to

the power ground. This establishes sufficient power to the shift register.

Next we connect the clock pin (pin 11) to a pull-down resistor.

We connect the latch pin (pin 12) a pull-down resistor.

We connect the serial data input pin to a pull-down resistor.

Since the Output Enable pin is active low, we want it to be LOW. So we connect it to a pull-down resistor.

The CLEAR pin is an active low pin, so we want it to be HIGH. Therefore, we connect it to the 5V power source,

through a voltage divider. The top resistor is 1KΩ and the bottom resistor is a 10KΩ resistor. This way, the majority

of the voltage falls across the 10KΩ resistor; this ensures that the CLEAR pin is active HIGH. Because the 10KΩ

resistor is HIGH, however, the LED will light dimly, but this is fine for demonstration purposes.

Lastly, we connect the LEDs to the output pins, QA - QH. To each of the LEDs, we connect a current-limiting 220Ω

resistor.

How this circuit works is that we first transfer bits to the storage register. In order to do this, we need the clock to

be HIGH in order to transfer bits to the storage register. If the serial data input line is LOW while we press the clock

pushbutton, a 0 is transferred in to the storage register. If the serial data input line is HIGH while we press the clock

pushbutton, then a 1 is transferred into the storage register. If you press down the clock pushbutton 8 times while

the clock is HIGH, we will transfer 8 1s into the storage register. If we push down 4 times on the serial data input

line while the clock is HIGH and 4 times on the clock pushbutton while the serial data input pushbutton is not

pressed, this will transfer 11110000 into the storage register.

Now, once you have the data transferred into the storage register, the next step is to latch the data into the output.

First, we transferred data into the storage register, but this data does not show up at the output (to the LEDs) until

we latch it to the output. So once we have the data transferred to the storage register, we have to latch it. Once we

press down on the latch pushbutton, it now goes to output. Now we see the LEDs light up. If the data in the storage

register is 11111111, all the LEDs will turn on. If the data is 10101010, every other LED will light up. If the data is

11110000, the first 4 LEDs will light up and the next 4 will be off.

If you want to turn off all the LEDs, you can either press down on the output enable pushbutton or the CLEAR line.

This clears all the 1s in the storage register. Next, you have to press down the latch pin to see all the LEDs turn off.

The CLEAR pin and the output enable pin are not the same. The CLEAR clears all the 1s to 0s in the storage register.

You can then add 1s after. The output enable pin permanently shuts off all outputs.

**Result:**

The Shift register has been designed and implemented using push buttons.

**References**

•[https://www.alldatasheet.com/view.jsp?](https://www.alldatasheet.com/view.jsp?Searchword=74hc595%20datasheet&gclid=CjwKCAjwydP5BRBREiwA-qrCGiNRzfjiL40amvAgQNYZbIGkRnLnV7IXx4mbaMI_Ul9-N4dLgjtbgxoCKDQQAvD_BwE)

•[https://www.ti.com/lit/ds/symlink/cd74hc595.pdfts=1597313125366&ref\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/cd74hc595.pdf?ts=1597313125366&ref_url=https%3A%2F%2Fwww.google.com%2F)

•<http://www.learningaboutelectronics.com/Articles/Shift-register-circuit.php>