**DIGITAL CIRCUITS LABORATORY**

**Experiment-9**

**Universal Shift Register**

Tanish H Talapaneni

200020050

**Aim:**To design a circuit for a 4 bit Universal Shift Register controlled by a 2 bit input.

**Summary:** Implementation of the 4 bit Universal Shift Register controlled by 2 bit input which performs 3 main operations

**Components used:** IC-7474(D Flip-flop), IC-74153(4:1 MUX), DIP Switches, LED Display, Breadboard, Power supply, Arbitrary Function Generator, 1k ohm resistor.

**Procedure:**

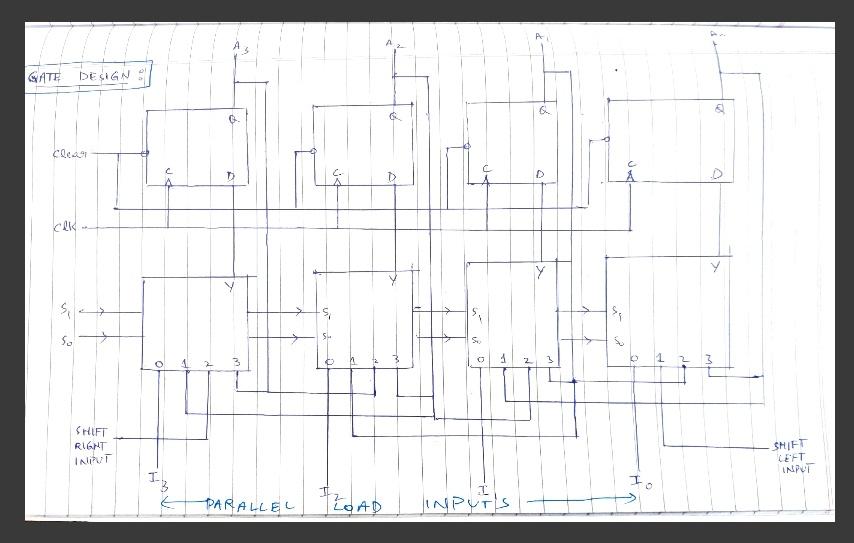
* Connect the circuit as shown in diagrams and give it power input.
* Select inputs from the truth table. Make the changes in the switch accordingly.
* Glowing LED implies that output was 1, otherwise 0.
* Verify if the simulation matches with the expected values.

**Logic Design:**

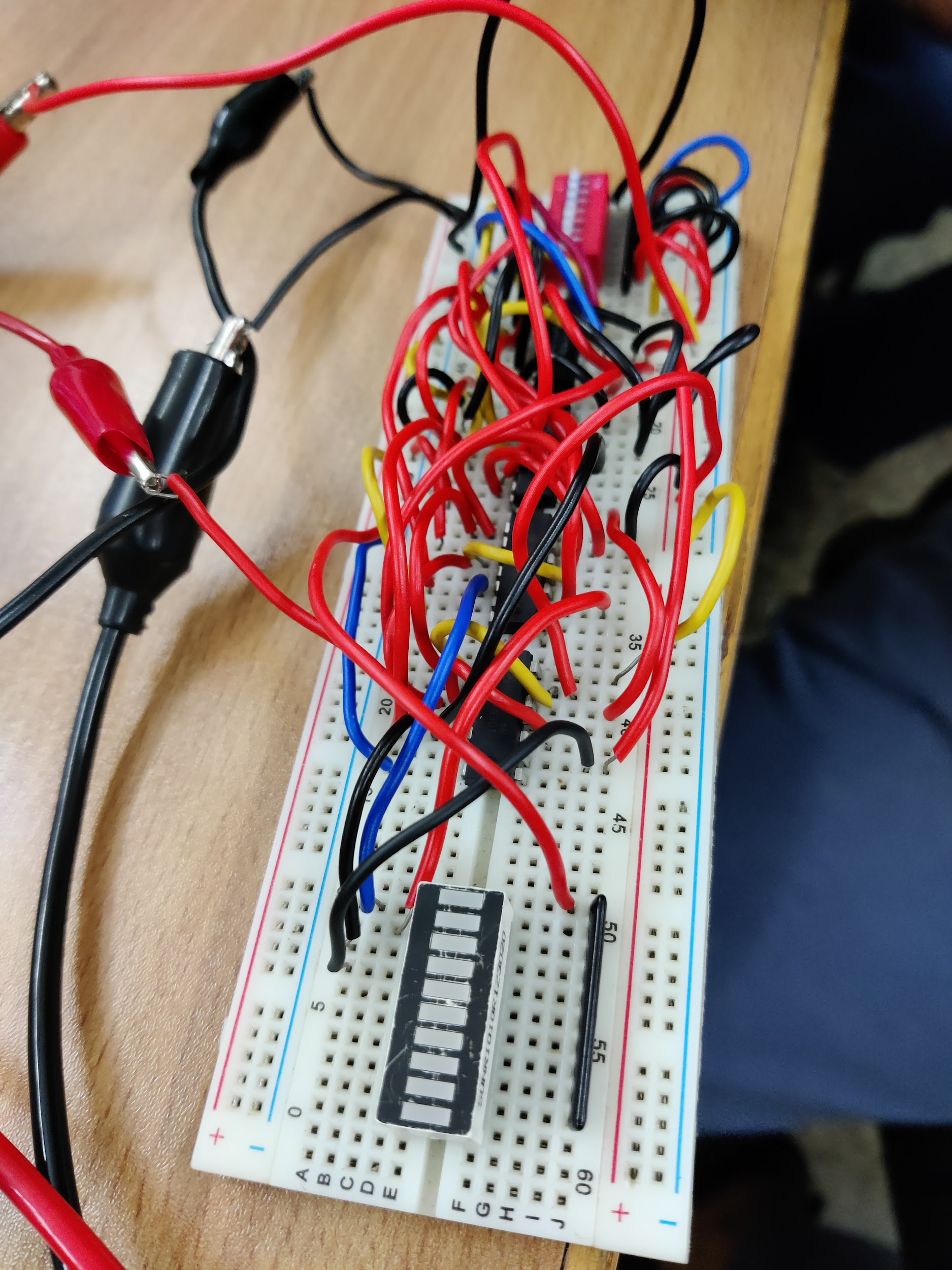
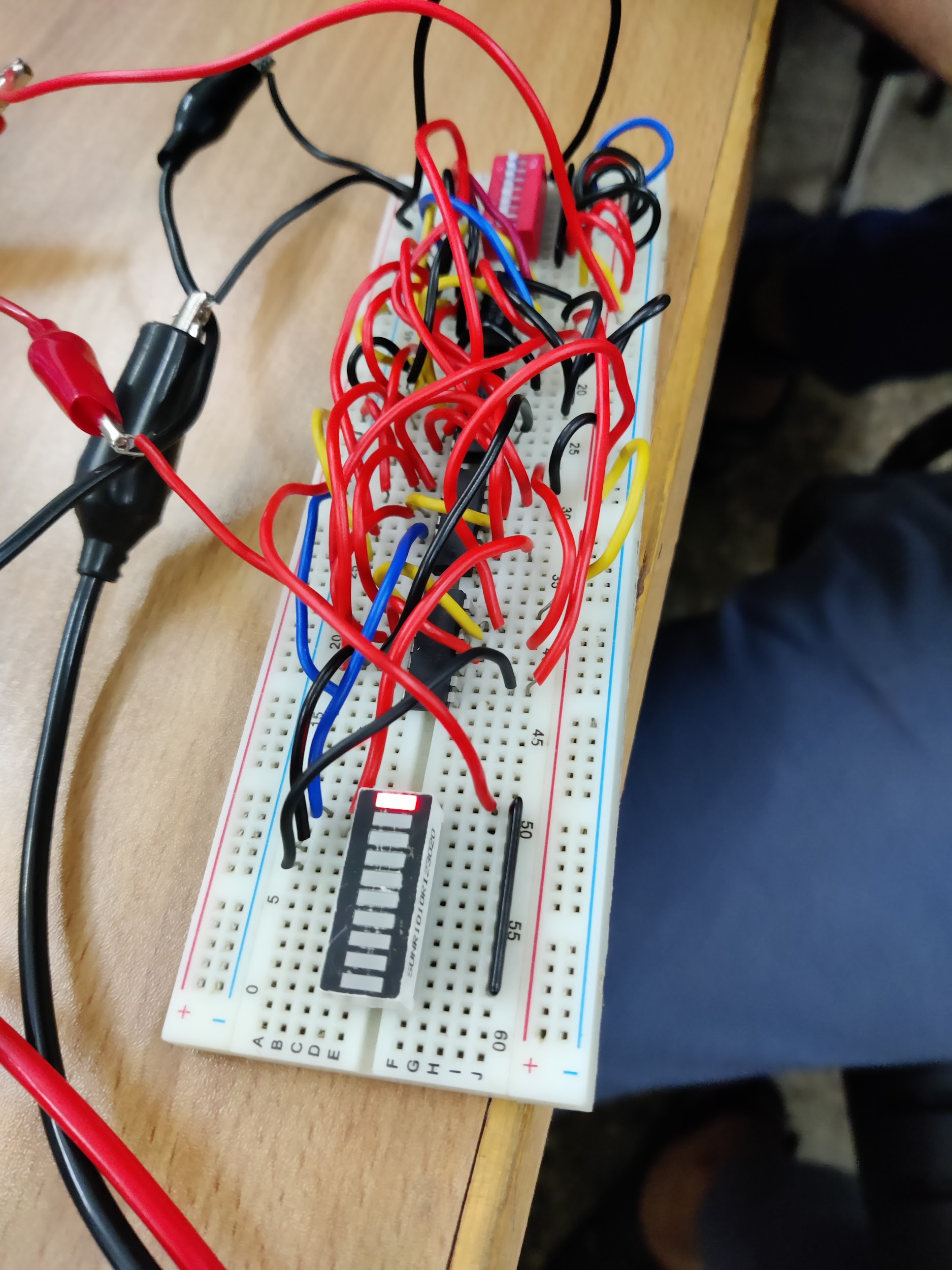
S0and S1 are the 2-bit inputs given to all the multiplexers, which thus decides the operation to be executed

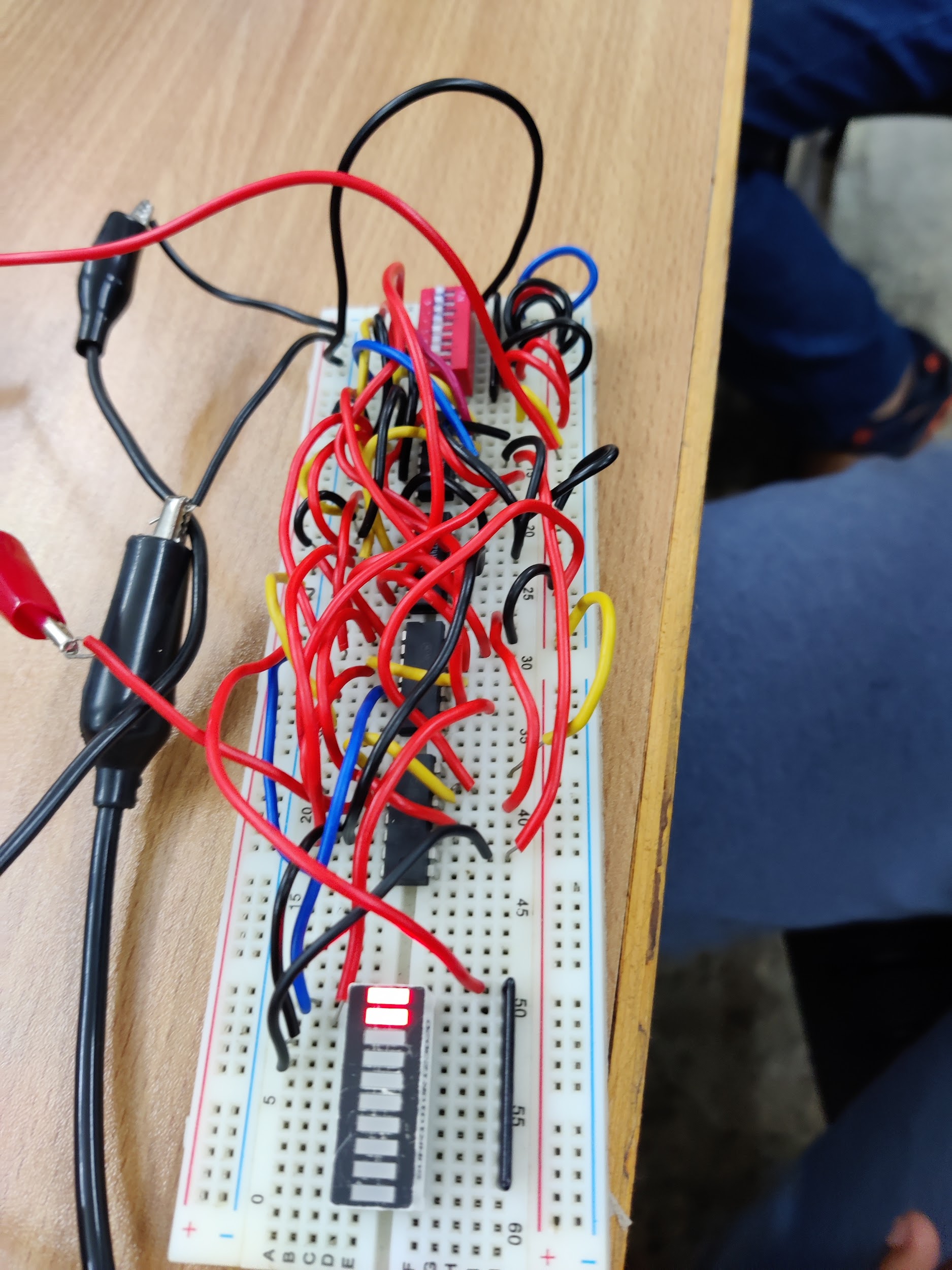
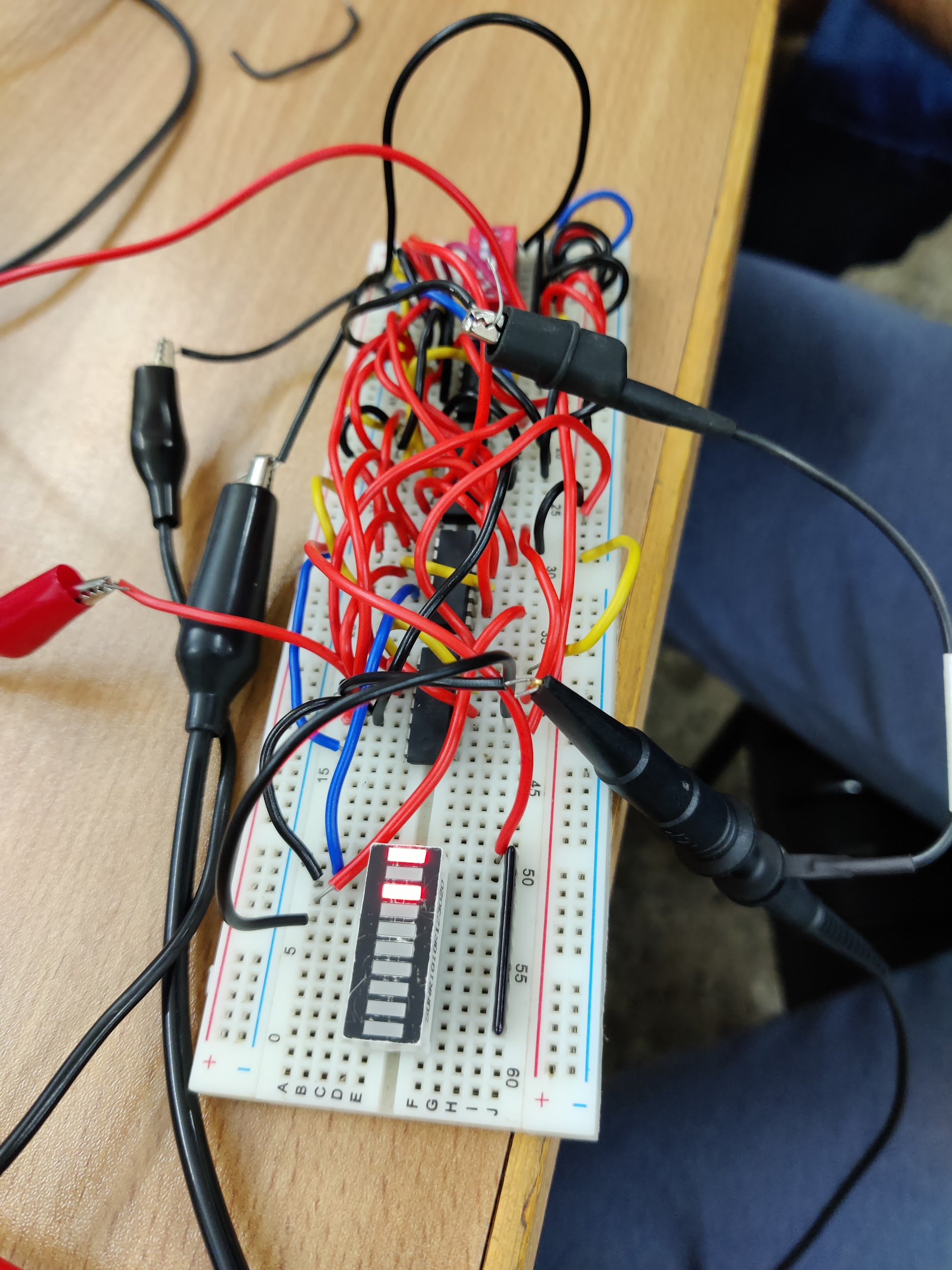
| S0 | S1 | Operation |
| --- | --- | --- |
| 0 | 0 | Parallel Loading |
| 0 | 1 | Left Shift |
| 1 | 0 | Right Shift |
| 1 | 1 | No change |

**Gate Design:**

****

**Hardware:**





**Results and Discussions:** We finally understood how to practically implement the Universal Shift Register. The control inputs used were opposite of the example used in class, so we had to make a few changes to get the gate design. We had to use the Arbitrary Function Generator with appropriate frequency in order to generate the clock signal. It was easier to obtain the parallel load output compared to the shift operation.The circuit connections were pretty complicated with many possibilities of loose connections. Designing the circuits strengthened our understanding of gate-logic of the Universal Shift Register and implementing it on the breadboard was fun.

**Conclusion:** I was able to verify the logic of Universal Shift Register theoretically along with its practical implementation. It involved 3 main operations, with the input to the flip-flops controlled by Multiplexers. When the input-bits are both 0, Parallel Loading is to be done. When both input-bits are both 1, No change is observed. Complementary bit inputs are used to execute shift operations