

LAB REPORT: 6

Name: Arghya Roy

Roll Number: 2021115008

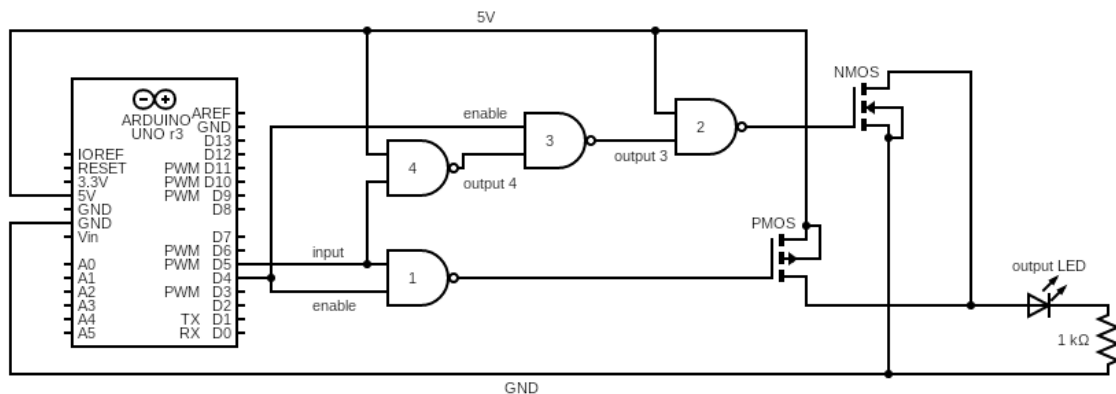
Group: 8

Part A: Working of the Tri-state Buffer

Aim/Objective of the experiment: To verify the working of a tri-state buffer

Electronic components used: 1 Arduino Board, 1 nMOS transistor, 1pMOS transistor, 1 quad NAND gate(74HC00), a 1k Ω resistor, 1 LED, wires.

Reference Circuit:



Procedure:

1. The basic circuit for the implementation of a tri-state buffer was provided.
2. The input and the enable signals were provided to the circuit using Arduino.
3. An appropriate code was written for the purpose
4. To verify its working, a truth table is tabulated by observing the outputs for different combinations of input and enable.

The code:

```
int i,en;

void setup()
{
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);

    Serial.begin(9600);
}

void loop()
{
    if(Serial.available()>0)
    {
        en=Serial.read();
        en=en-'0';
```

```

        digitalWrite(4,en);
    }

    if (Serial.available()>0)
    {
        i=Serial.read();
        i=i-'0';
        digitalWrite(5,i);
    }

    delay(500);
}

```

Conclusion:

We observe that

Enable	Input	Output
0	0	0
0	1	0
1	0	0
1	1	1

This verifies our theoretical understanding of tri-state buffer that

Enable	Input	Output
0	0	Z
0	1	Z
1	0	0
1	1	1

Where Z = impedance state.

So, the working of the tri-state buffer is verified.

TinderCAD simulation:

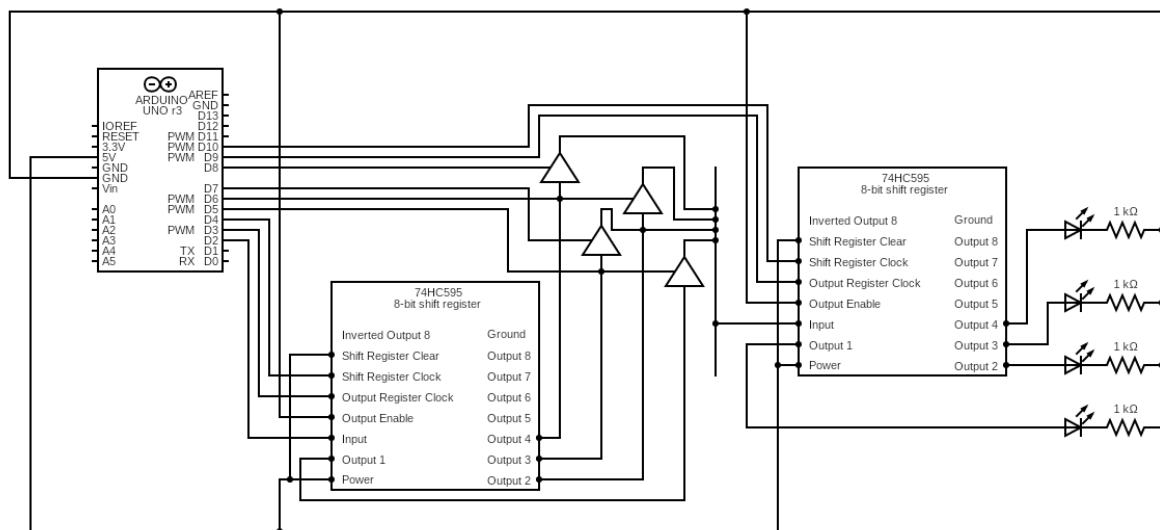
<https://www.tinkercad.com/things/kkWKNLUPWT6-lab-8-part-a-tri-state-buffer/>

Part B: Data flow using Tri-state buffers

Aim/Objective of the experiment: To perform data flow using tri-state buffer

Electronic components used: 1 Arduino board, four 1 k Ω resistors, 4 LEDs, 6 breadboards, two 8-bit shift registers(74HC595), 4 NAND gates(74HC00), 4 nMOS transistors, 4 pMOS transistors, wires

Reference Circuit:



Procedure:

1. The circuit is set up, as shown in the reference figure above, on the breadboards.

2. Nine inputs are provided from the Arduino which include:
 - a. The shift register clock, output register clock and data input for the first shift register
 - b. The 4 enable inputs for the tri-state buffers
 - c. The shift register clock and the output register clock for the second shift register
3. A number between 0-15 is taken as input and sent to the first register.
4. An Arduino code is written to enable the tristate buffers in order and correspondingly apply clock pulses to the second shift register such that the content from the first register is transferred.

The code used:

```
int input=2;
int orc=3;
int src=4;
int tsb1=5;
int tsb2=6;
int tsb3=7;
int tsb4=8;
int orc2=9;
int src2=10;

int num;

void setup()
{
    pinMode(orc, OUTPUT);
```

```

    pinMode(src, OUTPUT);
    pinMode(input, OUTPUT);
    pinMode(orc2, OUTPUT);
    pinMode(src2, OUTPUT);
    pinMode(tsb1, OUTPUT);
    pinMode(tsb2, OUTPUT);
    pinMode(tsb3, OUTPUT);
    pinMode(tsb4, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    if(Serial.available()>0)
    {
        num=Serial.parseInt();

        digitalWrite(orc, LOW);

        int j;
        for(j=0;j<8;j++)
        {
            digitalWrite(input,!!(num&(1<<(7-j))));
            digitalWrite(src, HIGH);
            digitalWrite(src, LOW);
        }

        digitalWrite(orc, HIGH);
    }
}

```

```
digitalWrite(orc2, LOW);  
digitalWrite(tsb4, HIGH);  
digitalWrite(tsb1, LOW);  
digitalWrite(tsb2, LOW);  
digitalWrite(tsb3, LOW);  
digitalWrite(src2, HIGH);  
digitalWrite(src2, LOW);  
digitalWrite(orc2, HIGH);
```

```
digitalWrite(orc2, LOW);  
digitalWrite(tsb3, HIGH);  
digitalWrite(tsb1, LOW);  
digitalWrite(tsb2, LOW);  
digitalWrite(tsb4, LOW);  
digitalWrite(src2, HIGH);  
digitalWrite(src2, LOW);  
digitalWrite(orc2, HIGH);
```

```
digitalWrite(orc2, LOW);  
digitalWrite(tsb2, HIGH);  
digitalWrite(tsb1, LOW);  
digitalWrite(tsb3, LOW);  
digitalWrite(tsb4, LOW);  
digitalWrite(src2, HIGH);  
digitalWrite(src2, LOW);  
digitalWrite(orc2, HIGH);
```



```
        digitalWrite(orc2, LOW);  
        digitalWrite(tsb1, HIGH);  
        digitalWrite(tsb2, LOW);  
        digitalWrite(tsb3, LOW);  
        digitalWrite(tsb4, LOW);  
        digitalWrite(src2, HIGH);  
        digitalWrite(src2, LOW);  
        digitalWrite(orc2, HIGH);  
    }  
}
```

Conclusion:

We see that the 4 LEDs glow corresponding to the bitwise binary representation of the number inputted by the user between 0 and 15, thus verifying data flow through tri-state buffers.

TinkerCAD Simulation:

<https://www.tinkercad.com/things/4esY8Ky4bG8-lab-8-part-b-tristate-buffers-data-flow/>
