Lab Report: 1

Name: Atharv Bhatt
Roll number: 2023101089
Group number: 3

Experiment 1: Familiarization with Digital test Kit and Binary Logic levels
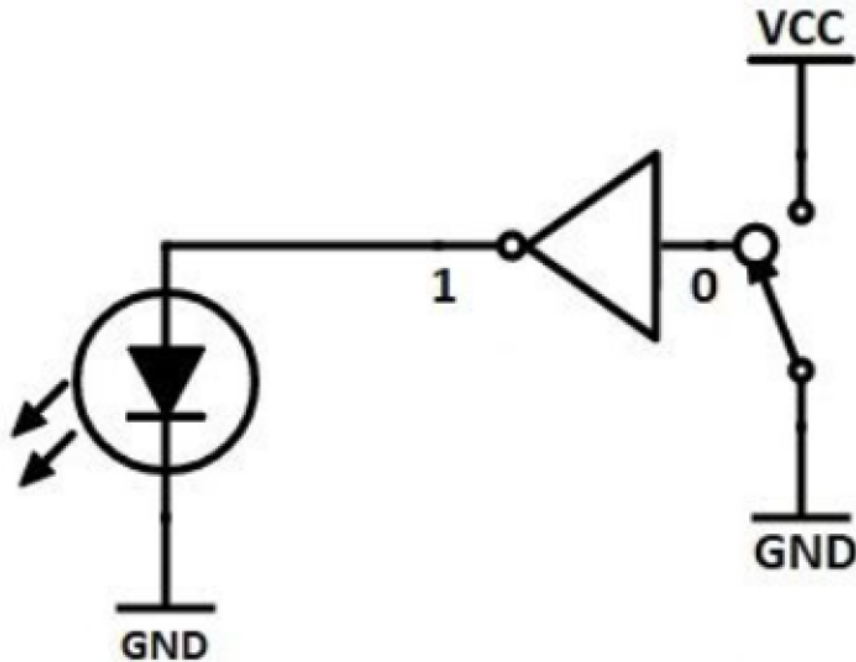
Aim/Objective

To familiarize the students with the Digital Test Kit, understand the binary logic levels concept, and implement a NOT gate using a 7404 IC.

Electronic Components Used:

1. Digital Test Kit
2. 7404 IC (Hex Inverter)
3. Breadboard
4. Various Jumper Wires
5. LED
6. Input Switches

Reference Circuit:

Procedure:

1. Set the clock control switch to the FAST position, and turn on the VCC supply. Then use the Digital Multimeter to measure the voltage between VCC and GND pins.
2. Verify the functionality of all 12 Input Pins (IP1-IP12) by measuring the voltages at these pins for both the ON and OFF positions of the corresponding Input Switches.
3. Confirm the functionality of all 8 LED displays by applying input from an Input pin to DP1-DP8 and observe the LEDs while toggling the input switch.
4. Execute an implementation of a NOT gate using a 7404 IC as follows:
   - Connect the VCC and GND pins of the IC to the relevant lines on the breadboard.
   - Connect an input pin of a gate in the IC to an Input Switch (IP1-IP12).
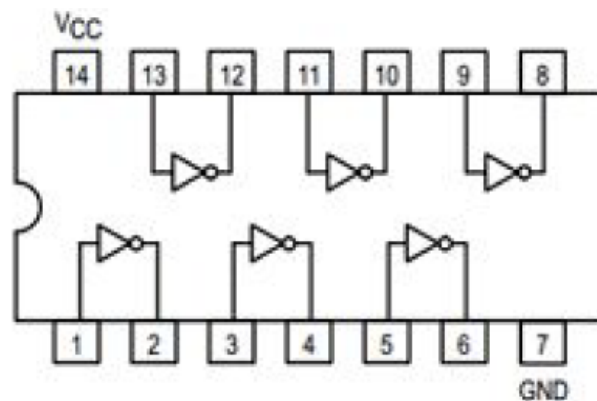   - Connect the corresponding output pin of the IC to a Display Point (DP1-DP8).

Fig.3 NOT Gate IC 7404

Observation:

It was observed that when the input switch was set to Low (0), the output turned Red led to glow.
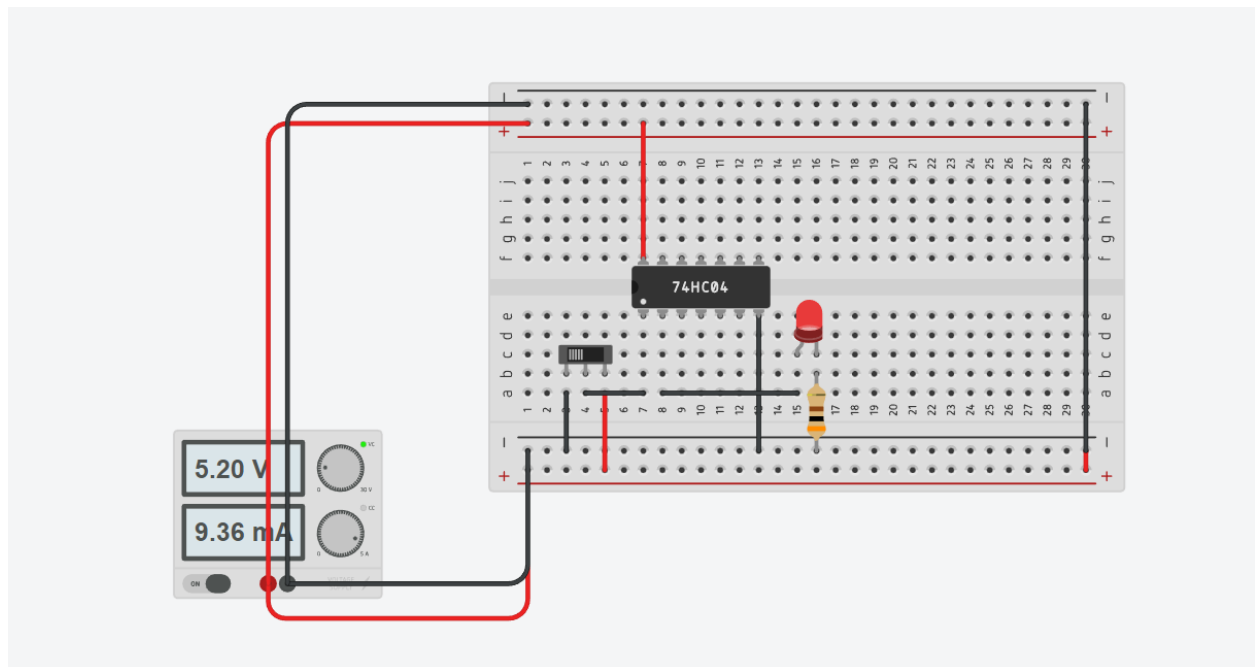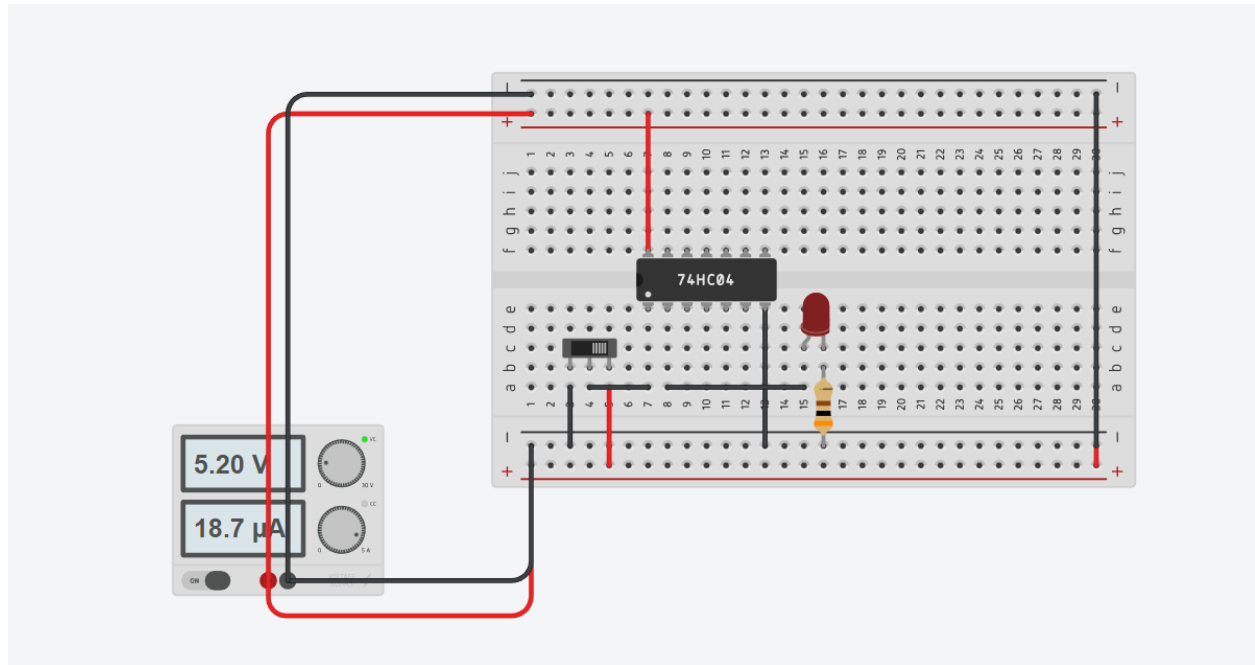
Conclusion

We became familiar with the Digital Test Kit and its components through this lab session. We understood binary logic levels by confirming input switches and observing LED displays. Additionally, we successfully implemented a NOT gate using a 7404 IC, inverting input logic to produce output logic.

Input	Output
0	1
1	0

Link of the TinkerCAD simulation

https://www.tinkercad.com/things/9Ccjgm0IxXC-magnificent-jarv-albar/editel?sharecode=Pqj7CNtheFp6RhLVztecVChAjUB0PJhUbV351uTZHu8

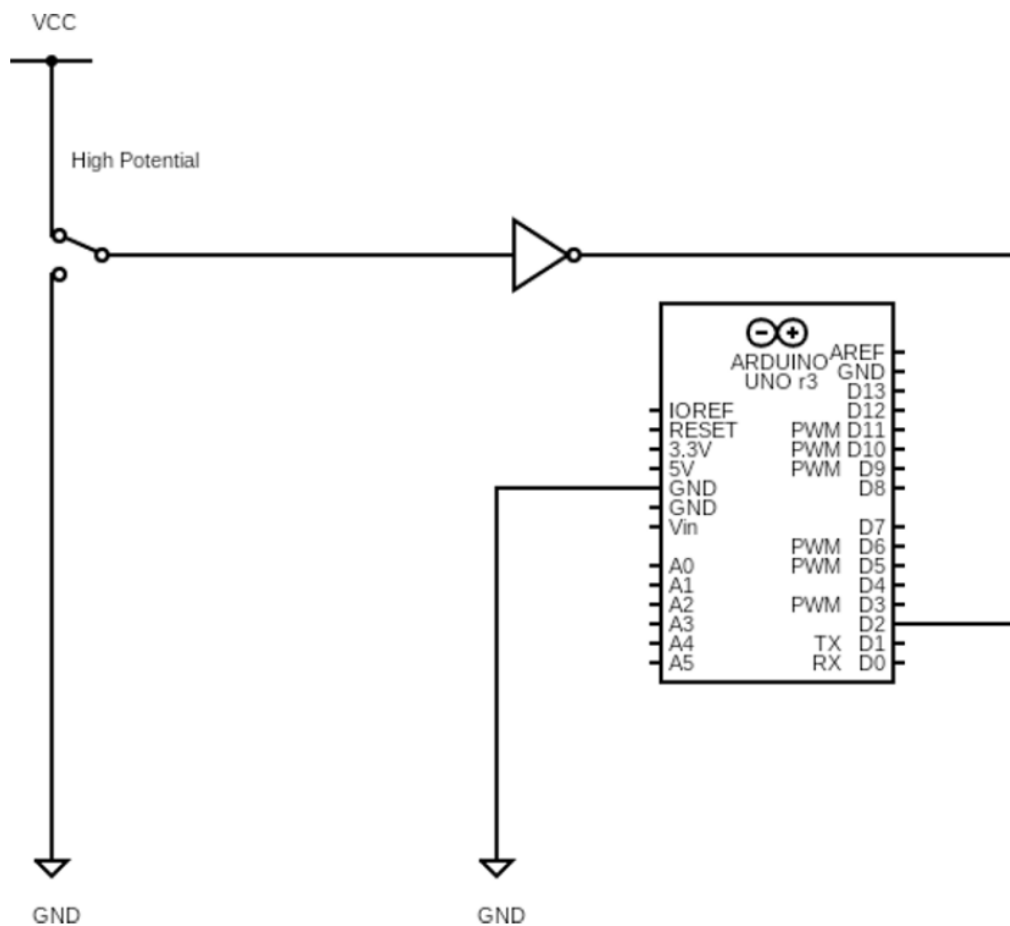**Experiment 2:** Microcontroller Programming using Arduino ISP

Aim/Objective

The objective of this experiment was to learn how to write, compile, and upload C code to an ATMega328P microcontroller using Arduino ISP. We aimed to observe the program's output on the serial monitor of the Arduino IDE based on the output of the NOT gate.
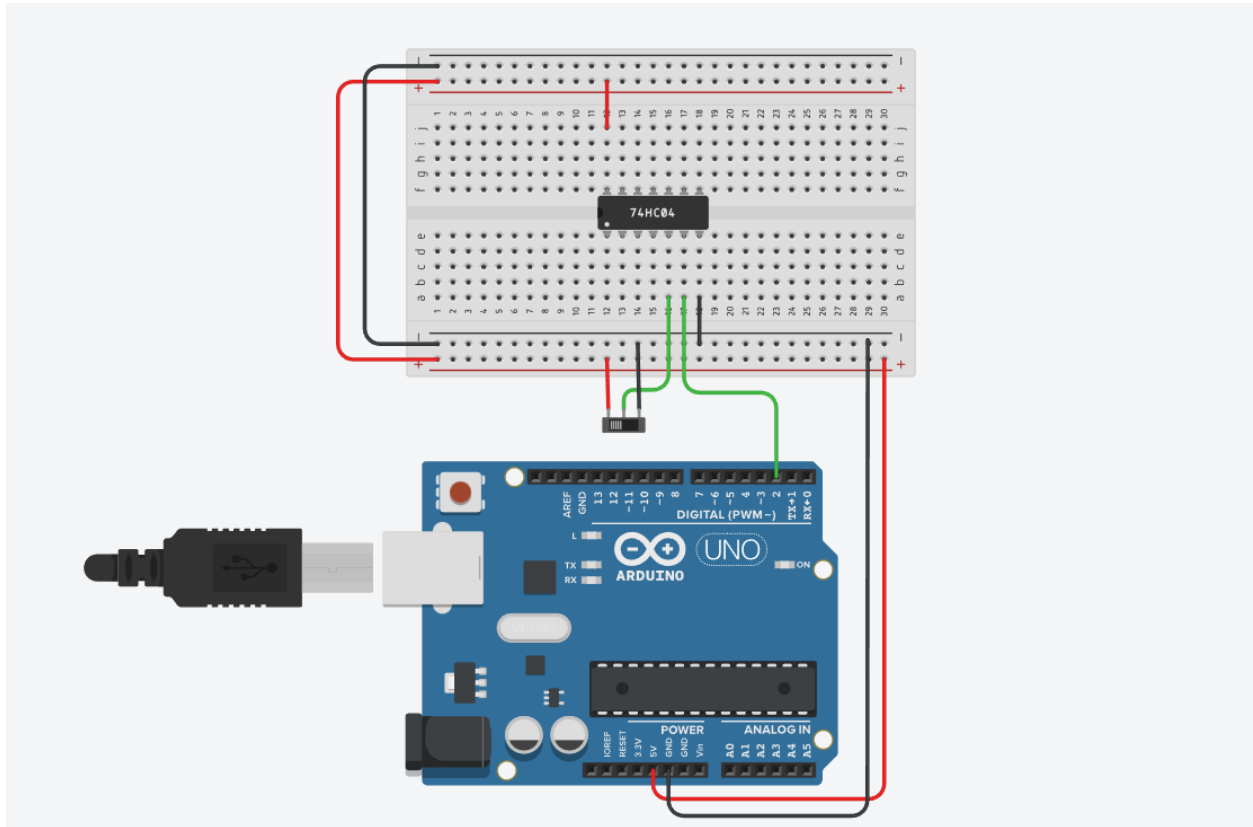
Electronic Components Used:

1. Arduino ISP
2. ATMega328P Microcontroller
3. Laptop with Arduino Software
4. NOT Gate IC 7404
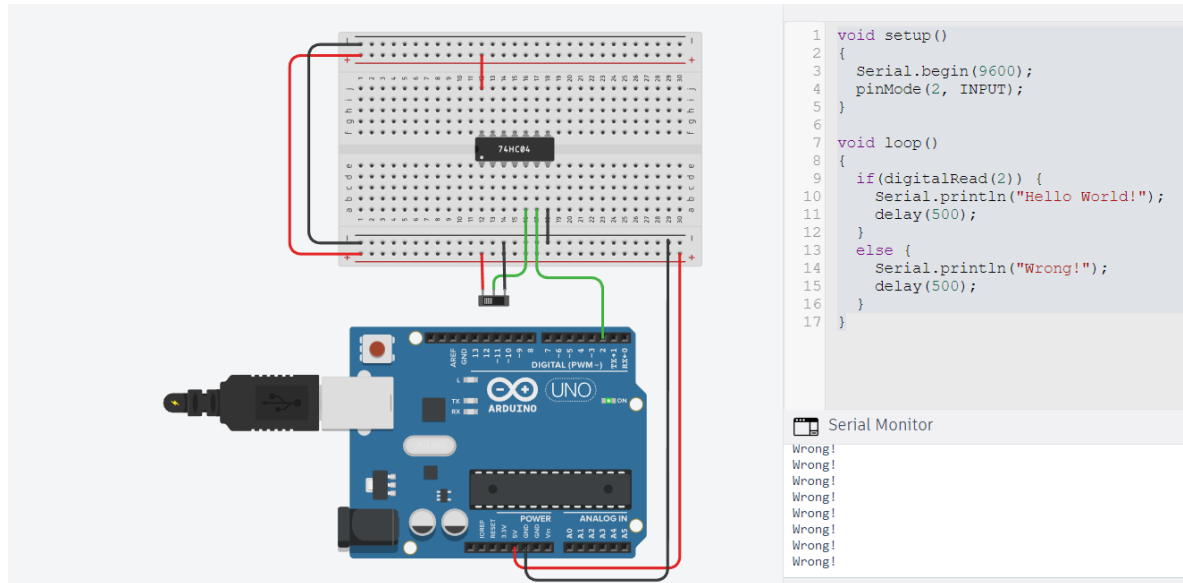5. Connecting Wires

Reference Circuit:

Link of the TinkerCAD simulation

https://www.tinkercad.com/things/ctzhxp4y0Y7/editel?returnTo=%2Fclassrooms%2F65dBoBuN JYW%2Factivities%2F63uB7RDlC42%3Ftype%3Dcircuits&sharecode=uXiWV0d5dxSYIw19x 70r9tBCEPTP9PMt52p3YudQd44

Procedure:

1. Download the provided C code and open it using the Arduino ISP.
2. In the 'Tools' section of the Arduino ISP, select the appropriate 'Board' type and specify the corresponding 'PORT'.
3. Make the necessary connections between the ATMega328P microcontroller and Arduino ISP.
4. Upload the compiled code (pasted below) to the microcontroller using the Arduino ISP.
5. Connect the output of the circuit from "Part A" to any of the 14 digital pins of the Arduino.
6. Observe the output on the Serial Monitor in the Arduino Software.
7. Repeat the process of checking the output on the Serial Monitor for various combinations of inputs to the NOT gate.

```
1   void setup()
2   {
3     Serial.begin(9600);
4     pinMode(2, INPUT);
5   }
6
7   void loop()
8   {
9     if(digitalRead(2)) {
10    Serial.println("Hello World!");
11      delay(500);
12    }
13    else {
14      Serial.println("Wrong!");
15      delay(500);
16    }
17  }
```

Observation

We observed that with a low input (0) it would print "Hello World" and for a high input it would print "Wrong".

```
1   void setup()
2   {
3     Serial.begin(9600);
4     pinMode(2, INPUT);
5   }
6
7   void loop()
8   {
9     if(digitalRead(2)) {
10      Serial.println("Hello World!");
11      delay(500);
12    }
13    else {
14      Serial.println("Wrong!");
15      delay(500);
16    }
17  }
```

Serial Monitor

Wrong!
Wrong!
Wrong!
Wrong!
Wrong!
Wrong!
Wrong!
Wrong!

Conclusion

We got practical experience in writing, compiling, and uploading C code to an ATMega328P microcontroller using the Arduino ISP. When the switch is off, 0 is given as the input and NOT gate returns 1 as the output. Hence "Hello World" is displayed on the serial monitor. When the switch is on, 1 is given as the input and NOT gate returns 0 as the output. Hence "0" is displayed on the serial monitor. NOT gate inverts the given input signal.