IDENTIFICATION OF BASIC LOGIC GATES USING ARDUNIO

NAIDU VENKATA PRAKASH
ECE with spec. in Biomedical Eng.
Vellore Institute of Technology
Vellore, India

SRISHTI CHOPRA
ECE with spec. in Biomedical Eng.
Vellore Institute of Technology
Vellore, India

***Abstract*— The present world is witnessing the tremendous growth in technology and practices. The electronic products have become part of our life and living without them is next to impossible. It is due to following reasons namely, increased efficiency in operation and ease of handling, cost and availability. The increased adoption of special electronic add on often leads to monopoly in business and service causing repurchase because of non-availability of serviceable parts or components for the public. A proper understanding of the individual components of an electronic circuit by the service man and designer can fill the life to dead components by replacing them with the programmable components like Arduino controller. Arduino controller is a very flexible, user friendly and can role itself as a best component for various needs. We work on how to identify basic logic gate ICs viz. NOT, AND, NOR, NAND, XOR using Arduino.**

Keywords- Logic Gates, Arduino, AND, NAND, OR

I. INTRODUCTION (*HEADING 1*)

The logic gates are the building blocks in logic design. The fundamental gates are used in sequence and appropriate manner to develop sequential and combinational circuits for the solution of complex problems. The understanding of functionality of basic gates through simulation using LED and switches controlling them by Arduino control represents a strong learning aid. Teaching learning mechanism is presently transforming into new dimension by making use of latest state of art technologies to present in an easy and lucid manner to the learner. The proposed concept of demonstration of functionality of basic logic gates is the first step towards the same. The Intelligent Arduino controller with its control of input /out puts through program exhibit strong capability and expansion. The current digital world with extensive electronic gadgets provide an opportunity and challenge to the designer and service man to find the suitable substitute of a failed electronic integrated circuit (Logical IC). The unavailability of such special components is a threat and challenge. The designer and repair man if able to reproduce the functionality of failed component (IC) using circuit diagrams and experience, then replacement of such component can be worked out using flexible components like Arduino. The present work focuses the similar activity of replacing the basic logic gate integrated circuits (ICs) by programmed Arduino in the place with suitable retrofit. The work ideally tries to identify the ability of electronic graduate engineer to develop such requirements of the community by making use of programmable peripherals by enhancing the knowledge and adoptability. The literature in this regard has provided sufficient evidences where use of such programmable components in teaching, research and developments to present the ability of soft skills and designers ability to meet the contingent requirements by special design

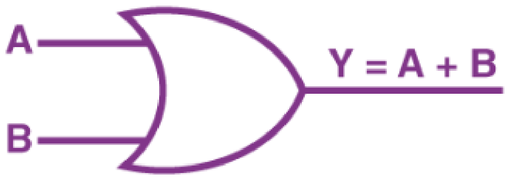
II. LOGIC GATES

Logic gates are an important concept if you are studying electronics. These are important digital devices that are mainly based on the Boolean function. Logic gates are used to carry out logical operations on single or multiple binary inputs and give one binary output. In simple terms, logic gates are the electronic circuits in a digital system.

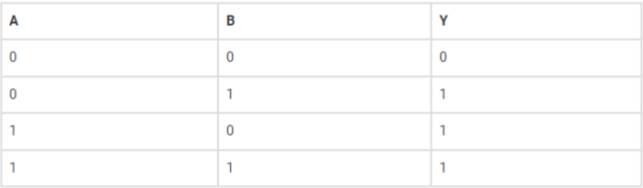
III. BASIC LOGIC GATES

A. OR GATE

In an OR gate, the output of an OR gate attains state 1 if one or more inputs attain state 1.



(Schematic of OR GATE)

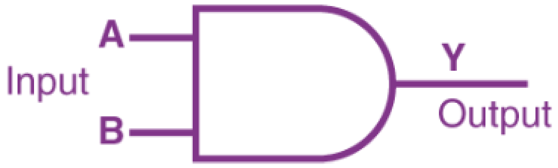


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

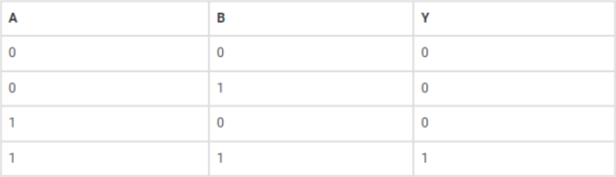
(Truth Table of OR GATE)

B. AND GATE

In the AND gate, the output of an AND gate attains state 1 if and only if all the inputs are in state 1.



(Schematic for AND GATE)

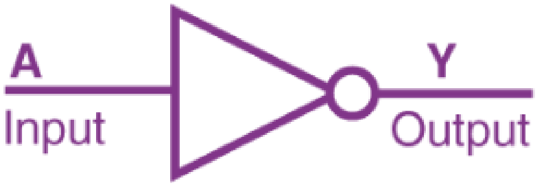


A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

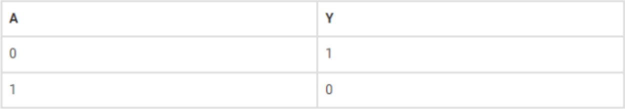
(Truth Table for AND gate)

C. NOT GATE

In a NOT gate, the output of a NOT gate attains state 1 if and only if the input does not attain state 1.



(Schematic for NOR gate)

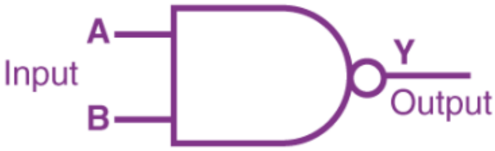


A	Y
0	1
1	0

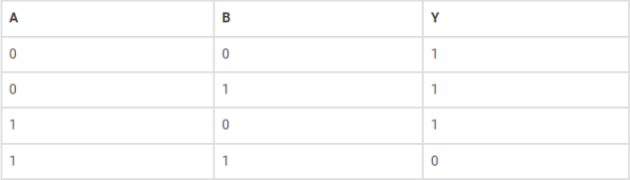
(Truth table for NOR gate)

D. NAND GATE

This basic logic gate is the combination of AND and NOT gates.



(Schematic for NAND gate)

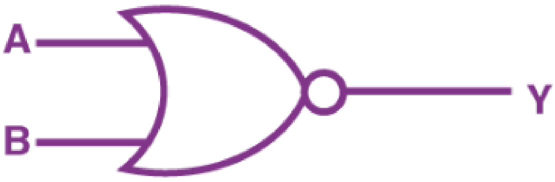


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

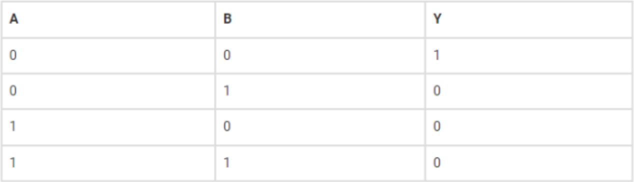
(Truth Table for NAND gate)

E. NOR GATE

This gate is the combination of OR and NOT gate.



(Schematic for NOR gate)



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

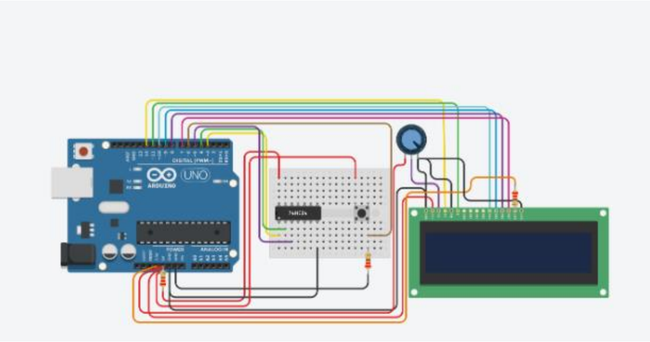
(Truth Table for NOR gate)

IV. LITERATURE REVIEW

No.	AUTHOR/ YEAR	TITLE	INFERENCE
1.)	Z. Tucaković 2016	Technical Diagnosis of Basic Logic Gates	In this paper, entire process of planning and production of electronic device is given. First, theoretical basis of logic gates, causes of malfunction and malfunction diagnosis are explained. After that, thorough analysis and planning software (programming code), as well as hardware (electronic components) is made. Functionality of code was checked, first through simulation, and later by a number of experiments on a real circuit. After numerous verifications, the final version of code is obtained. Only after that, the device could be built, and packed in a box. In that way, the prototype is simpler and safer for usage.
2.)	Logic Gates in a Single Integrated Circuit	Sayandeep Sengupta	This chip will play and important role in field of logic design, because it has all the basic logic gates in a single IC. So it will be an advantage for the beginner, they do not have to get meshed up with lots of IC to design their

			projects. This system is highly portable. Since this is the cluster of logic gates it cannot overcome the FPGA toward its speed and processing capacity. This IC need to be placed on a special PCB in which all of its connections are made through i/o ports. This chip will able to design all logic circuits as per logic design standards.
3.)	Digital Circuit Design Trends	Mark Horowitz, Donald Stark	Digital designs have gone through dramatic changes over the past two decades—moving from chips that contained tens of thousands of devices to today’s chips that may contain over a billion transistors. The job of the digital circuit designer has grown with the chips, moving from optimizing and validating gates, to working on functional units, to now designing complete systems. While the progress in digital design has clearly been tremendous, tackling current and future system issues and power challenges will lead to significant further innovation. We look forward to seeing continued reports of these digital circuit design advances over the next two decades of the conference

V. NETWORK SCHEMATICS

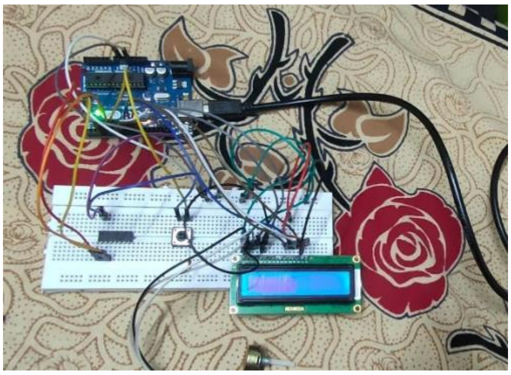


(Tinker-cad circuit-Virtual Simulation)

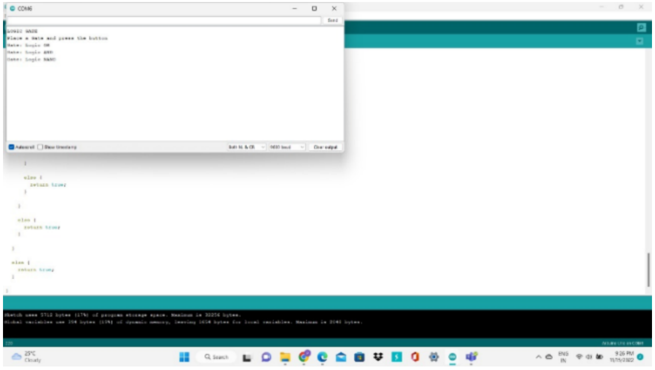
VI. COMPONENTS AND SUPPLIES

a.) Arduino UNO
b.) Serial Enabled LCD
c.) LOGIC GATE IC’s
d.) Push-Button Switch
e.) Resistor (220 ohm)
f.) Solderless breadboard
g.) Jumper wires

VII. HARDWARE SIMULATION



(Real-time circuit diagram following the above schematics)

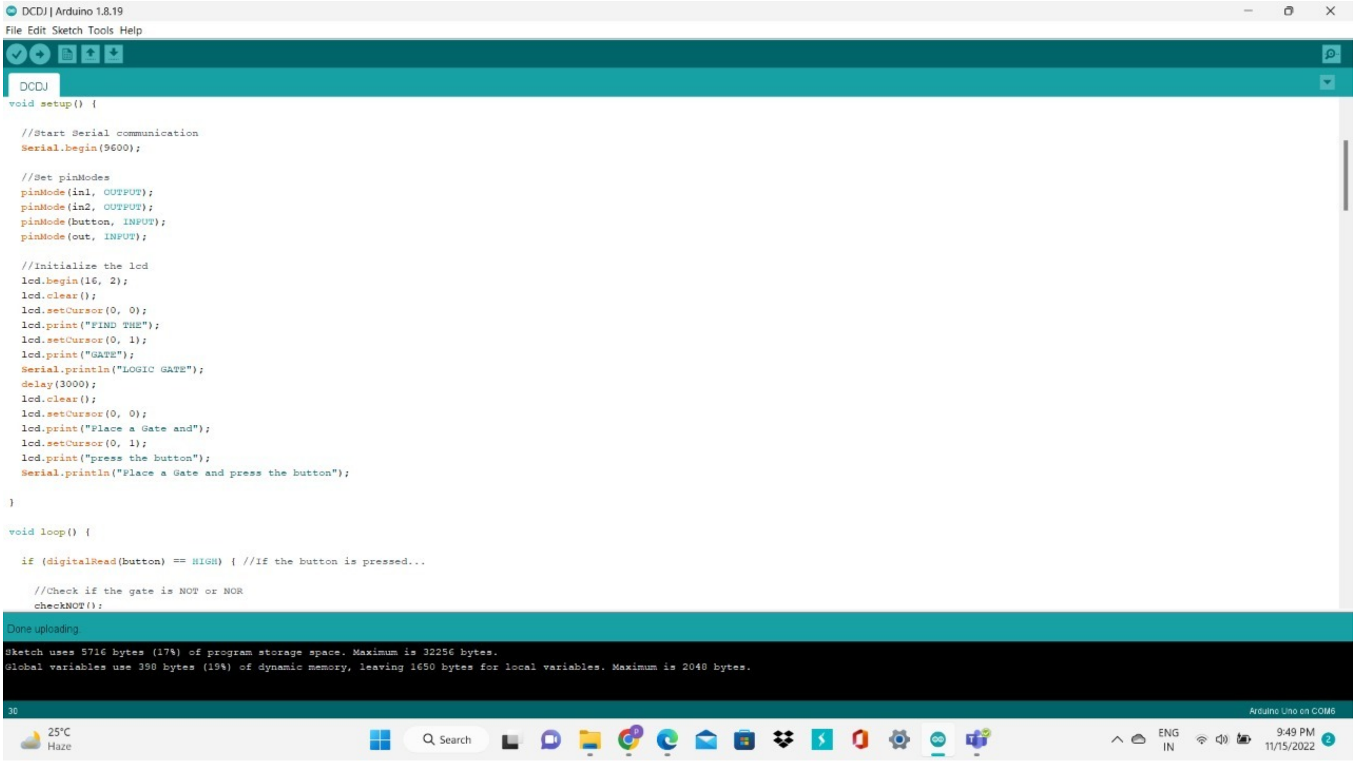


```
LOGIC GATE
Place a Gate and press the button
Gate: Logic OR
Gate: Logic AND
Gate: Logic NAND
```

```
      }

      else {
        return true;
      }

    }

    else {
      return true;
    }

  }

  else {
    return true;
  }

}
```

```
Sketch uses 5712 bytes (17%) of program storage space. Maximum is 32256 bytes.
Global variables use 394 bytes (19%) of dynamic memory, leaving 1654 bytes for local variables. Maximum is 2048 bytes.
```

(Software simulation of above hardware)



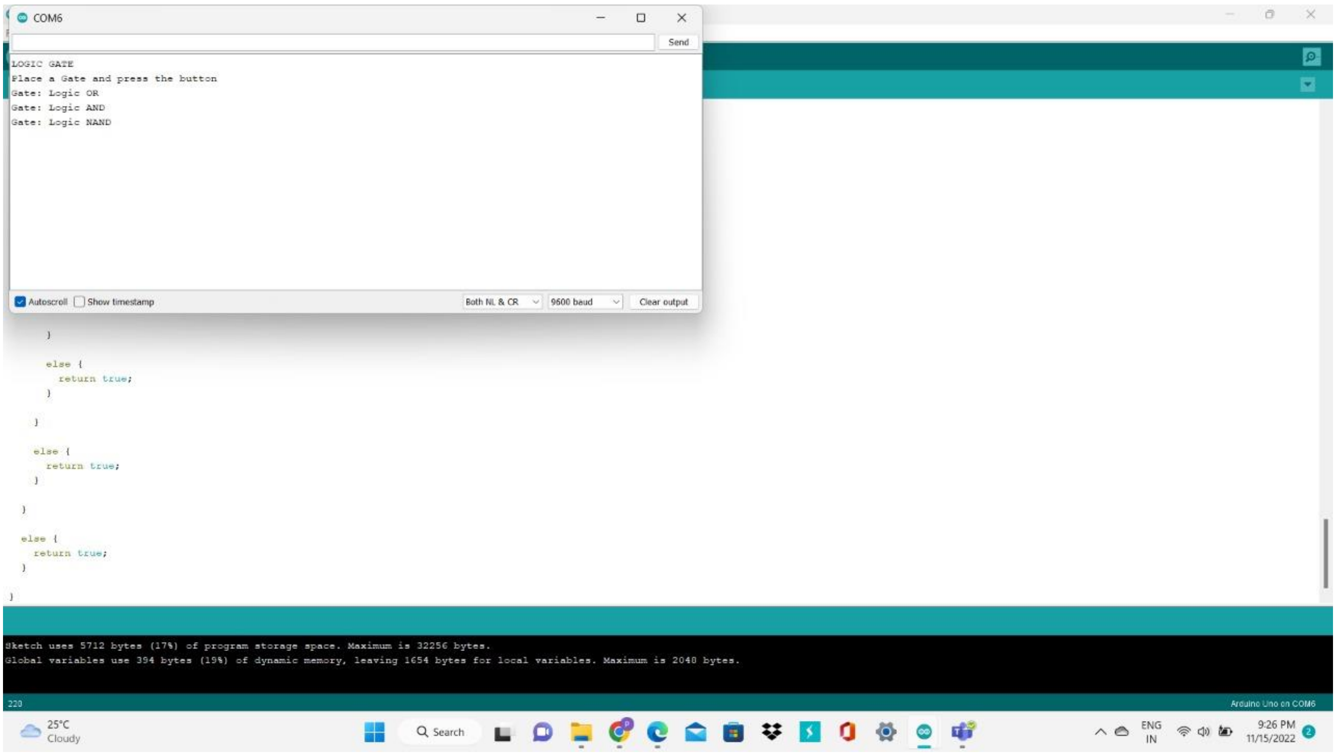
```
void setup() {

  //Start Serial communication
  Serial.begin(9600);

  //Set pinModes
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(button, INPUT);
  pinMode(out, INPUT);

  //Initialize the lcd
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("FIND THE");
  lcd.setCursor(0, 1);
  lcd.print("GATE");
  Serial.println("LOGIC GATE");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Place a Gate and");
  lcd.setCursor(0, 1);
  lcd.print("press the button");
  Serial.println("Place a Gate and press the button");

}

void loop() {

  if (digitalRead(button) == HIGH) { //If the button is pressed...

    //Check if the gate is NOT or NOR
    checkNOT();
```

Done uploading.

```
Sketch uses 5716 bytes (17%) of program storage space. Maximum is 32256 bytes.
Global variables use 398 bytes (19%) of dynamic memory, leaving 1650 bytes for local variables. Maximum is 2048 bytes.
```

(Part of Code: Arduino IDE)

VIII. RESULTS:



```
LOGIC GATE
Place a Gate and press the button
Gate: Logic OR
Gate: Logic AND
Gate: Logic NAND
```

```
      }

      else {
        return true;
      }

    }

    else {
      return true;
    }

  }

  else {
    return true;
  }

}
```

```
Sketch uses 5712 bytes (17%) of program storage space. Maximum is 32256 bytes.
Global variables use 394 bytes (19%) of dynamic memory, leaving 1654 bytes for local variables. Maximum is 2048 bytes.
```

The following result has been obtained on the simulation of the code and the hardware simulation is demonstrated live.

IX. CONCLUSION

The application and use of digital electronics in current context is very great because of the associated efficiency and ease. The complex circuits often cause difficulty in repair and replacement at applicable cost. In line with this the associated difficulty of unavailability of specific replacement component leads to retrofitting at best way. An effort is made to demonstrate the working of logic gates and Boolean algebra using practical based approach using Arduino microcontroller with associated hardware like switches and LEDs. The proposed method is an innovative teaching learning technique as well having advantages like clarity in understanding and reduction in presentation time. The conventional theoretical explanation and added activity found very effective. As a scope for the future work sequential and combinational circuit realization using the Arduino for the effective teaching and learning can be worked out. Then the exploration of the capability of Arduino controller in retrofitting works can be evidenced.

X. REFERENCES

1.] Horowitz, M., Stark, D. and Alon, E., 2008. Digital circuit design trends. *IEEE Journal of Solid-State Circuits*, *43*(4), pp.757-761.

2.] Tucaković, Z., Technical Diagnosis of Basic Logic Gates.

3.] Wadsack, R.L., 1978. Fault modeling and logic simulation of CMOS and MOS integrated circuits. *Bell System Technical Journal*, *57*(5), pp.1449-1474.