LOGIC GATE IC TESTER USING ARDUINO

Than Duc Truong[(1)]
(1) Thu Dau Mot university
Corresponding author: truongtd@tdmu.edu.vn

DOI: 10.37550/tdmu.EJS/2025.02.664

Article Info

Volume: 7
Issue: 2
Jun: 2025
Received: Mar. 25th, 2025
Accepted: Apr. 11th, 2025
Page No: 654-661

Abstract

In electronic circuits that use various integrated circuits (ICs), ICs may malfunction while assembled, used, and repaired. There are numerous ways to verify that ICs are operating, such as by measuring basic current and voltage with a VOM meter. However, many sophisticated operations are hard to measure and test, and the accuracy of the tests is low and takes a long time. Thus, it is crucial to have a tool that can rapidly determine whether or not integrated circuits are operating correctly. The purpose of this article is to develop a tool for testing the functionality of logic gate ICs. By modeling its properties using the truth table of the specific IC, the device employs an Arduino to verify the condition of the gates in a logic gate IC. After successful simulation and testing, they are assembled to form a final device.

Keywords: arduino, ICs, logic gates, testing device, truth table

1. Introduction

Engineers frequently run into numerous issues and mistakes in circuit assembly during the assembly, operation, and repair of electronic circuits. These mistakes can be brought about by wiring, design flaws, or malfunctioning ICs. Engineers can complete more projects and save time by eliminating a step in the error-checking process with the use of a device that can test the functioning of logic gate ICs.

Which gates are defective and which can function can be ascertained by the IC tester. The main purpose of the project is to create a logic gate IC tester that is more practical and significantly less costly than those that are currently on the market. The purpose is to quickly and accurately test the IC and show the results of any defective or excellent ICs right away. The Arduino applies the necessary input signal conditions to the gate inputs, and each gate's output is tracked and compared to the truth table. Based on this comparison, the IC is tested to determine if it is defective or not. Testing the logic functioning of the IC as specified in the truth table is the fundamental duty of a logic gate IC tester. While the microcontroller is being coded, the truth table is kept in the database. The LCD shows the results, which include the name of the defective IC and the identification of the good IC.

There has been and continues to be a lot of research pertaining to the testing of ICs' functionality (Miczo, 2003). A quick overview of tester structures, combinational and sequential test pattern creation, and fault simulation are examples of simulation theories. Design for Test (DFT), Built-In Self Test (BIST), fault tolerance, memory test, behavioral test, and verification are examples of tester architectures. VHDL is used to create and simulate applications. The development of a module and the construction of a laboratory setup utilizing the Field Programmable Gate Array (FPGA) for a variety of potential teaching and learning scenarios that will center on necessary industrial IC test operations are covered (Grout, 2021). A microcontroller-based logic tester was created (Mohammed et al., 2024), especially for testing the majority of 74xx series logic gate integrated circuits using a microcontroller (AT89C52). This project's basic design is to simulate the properties of a logic gate IC using its truth table in order to test the state of its gates. However, this tester does not identify the IC name; it just evaluates whether the IC is good or faulty.

2. Methods

2.1. Proposed System

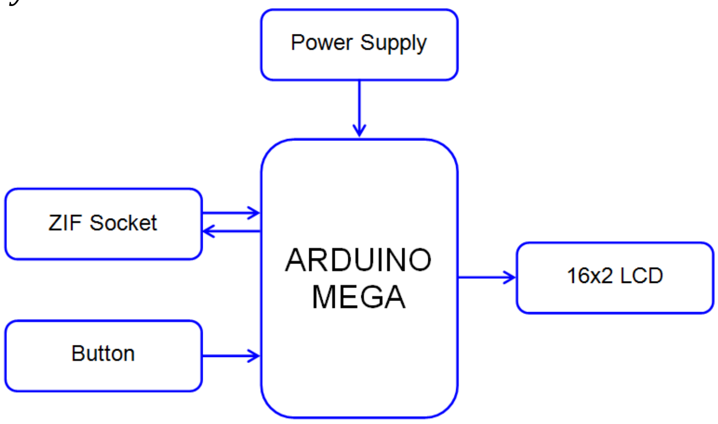


Figure 1. System block diagram

• The system workflow:

• The user inserts an IC into the ZIF socket.

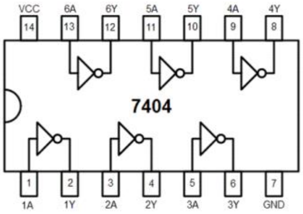
• Arduino applies logic inputs (0 and 1) to the gate inputs.

• Output responses are read and compared to truth tables.

• Results are shown on the LCD, identifying either the IC type or reporting an error.

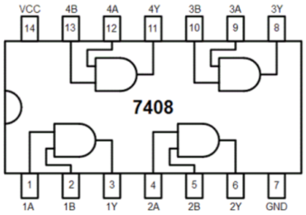
The suggested system mounts the ICs to be tested using a ZIF socket, connects to the microcontroller, and transmits signals to a 16×2 LCD to display the results. The Arduino Mega, a microcontroller board based on the ATmeg2A560, is the one utilized in the system. It features a 16MHz crystal, 54 digital IO pins – 15 of which can be used as PWM outputs – 16 analog IO pins, a power jack, a USB port, and an ICSP reset button. The ICs to be tested are mounted in ZIF sockets, which receive control signals from the Arduino and send them to the appropriate input pins of the IC. The microcontroller then receives the associated output pins and compares the value with the truth table to export the corresponding results to the 16×2 LCD. The testing procedure is initiated by pressing the push button. When the IC is functioning properly, the 16×2 LCD has the output control code in row 1 and the associated IC name in row 2, whereas when the IC is malfunctioning, it displays the result, "Unknown IC."

The appropriate output status in the truth table serves as the foundation for the testing principle of logic gate integrated circuits. Figures 2 to 8 describe the logic gates' truth table and IC connection diagram.



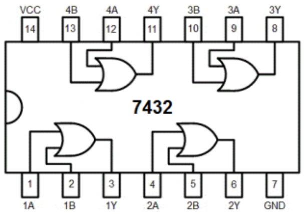
A	Y
0	1
1	0

Figure 2. NOT Gate 7404 Connection Diagram and Truth Table



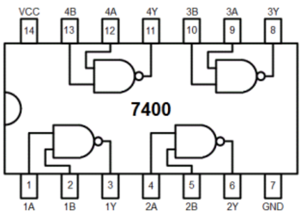
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Figure 3. AND Gate 7408 Connection Diagram and Truth Table



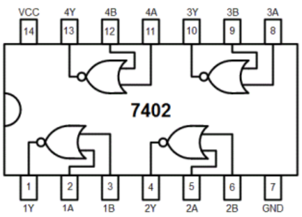
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure 4. OR Gate 7432 Connection Diagram and Truth Table



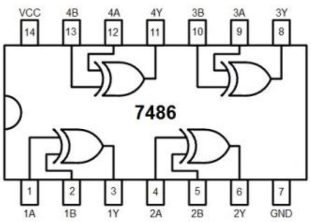
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Figure 5. NAND Gate 7400 Connection Diagram and Truth Table



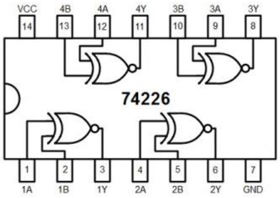
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Figure 6. **N**OR Gate 7402 Connection Diagram and Truth Table



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

Figure 7. X-OR Gate 7486 Connection Diagram and Truth Table



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

Figure 8. X-NOR Gate 74226 Connection Diagram and Truth Table

TABLE 1. Hardware Configuration

Component pin	74xx02 ICs	74xx04 ICs	Other ICs	Arduino Mega
ZIF Socket PIN 1 – 6	1Y - 1A - 1B 2Y - 2A - 2B	1A - 1Y 2A - 2Y 3A - 3Y	1A - 1B - 1Y 2A - 2B - 2Y	PIN D22 – D27
ZIF Socket PIN 8 – 13	3A - 3B - 3Y 4A - 4B - 4B	4Y - 4A 5Y - 5A 6Y - 6A	3Y - 3A - 3B 4Y - 4A – 4B	PIN D28 – D33
ZIF Socket PIN 7	GND	GND	GND	GND
ZIF Socket PIN 14	VCC	VCC	VCC	5V
16x2 LCD I2C (SDA/SCL)				PIN A8 – A9
Button				PIN D8

2.2. Algorithm

2.2.1. General idea:

The logic gate IC testing algorithm assesses whether the operation of a logic gate integrated circuit (such as IC 7400, 7408, 7432...) is still functional or not. Here is a common operating principle for testing a logic gate integrated circuit:

Objective:

Analyze the correct/incorrect operation of each logic gate in the IC.

Methods:

• Apply a specific logic input to each gate in the IC.

• Compare the output to the anticipated value (according to the truth table).

• If all outputs are suitable, the IC is good; otherwise, it is IC error.

Working principle (step-by-step):

The ICs to be tested have several comparable gates. For example, IC 7408 contains four AND gates.

Step 1: Apply input.

• Apply logic inputs 1 and 0 (Vcc and GND) to the IC to confirm that the Vcc voltage is sufficient for delivering a high level; an external power source can be used to power the test circuit.

Step 2: Iterate through each logic gate in the IC

• Example using IC 7408 (with 4 gates 2 inputs AND gates):

Step 3: Run the tests on each gate.

For every logic gate:

• Assign all input value combinations.

For example, the AND gate has four possible combinations: (0,0), (0,1), (1,0), and (1,1).

• Measure the output IC pin value.

• Compare the actual output to the anticipated value based on the truth table.

For example, with an AND gate, the predicted combination (A-B-Y) is (000), (010), (100), (111)

Step 4: Conclusion.

• If all gates are suitable for all input combinations, the IC is functioning properly.

• If one or more gates are incorrect, the IC is defective.

2.2.2. Testing device algorithm:

a. Power on the device → LCD shows "Press Button..."

b. Insert IC → Press the test button.

c. Arduino sends input patterns → Reads outputs.

d. Compares results with truth table.

e. Displays either:

• IC name + pass result.

• "Unknown IC" if mismatch occurs.

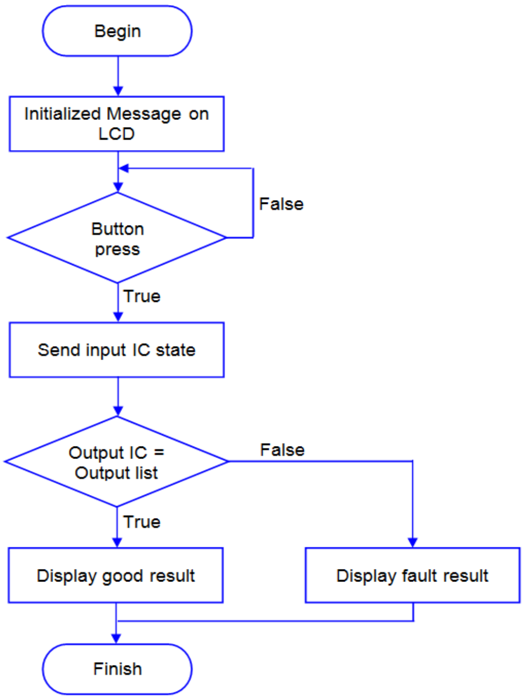


Figure 9. Algorithm Flowchart

Use the power switch on the box body to turn on the system. Once it is powered on, the LCD will initialize and the message "Press button..." will appear. Press the test button once the IC has been inserted into the socket. The input pins A and B will receive values 0 and 1 from the Arduino signal. The outputs Y will be compared with the corresponding values based on the truth table whenever the input values have been received. When the proper results are compared, the LCD will show the excellent IC result, with the appropriate IC name appearing in row 2 and the output control code in row 1. The incorrect IC result will be shown on the LCD with the text "Unknown IC" if the result conflicts with the list.

3. Experimental results

Each component of the circuit will be checked separately by inserting it into a testboard. Following all the components have passed testing, the system will be put together and packaged.

3.1. Test the operation of NAND gate

As illustrated in figure 10, connect the circuit between the Arduino and IC on the test board. Send the appropriate input values, then verify that the outputs are in line with the truth table.

Figure 10. NAND gate operation Testing

The LCD will show the output status code "1110" and the name of the relevant IC Nand gate, 74HC00, when the output reacts appropriately to the truth table, indicating that the IC is operating correctly.

3.2. Test the operation of X-OR gate

As illustrated in figure 11, connect the circuit between the Arduino and IC on the test board. Send the appropriate input values, then verify that the outputs are in line with the truth table.

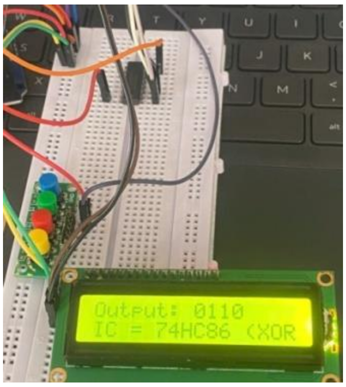


Figure 11. X-OR gate operation Testing

The LCD will show the output status code "0110" and the name of the matching IC Nand gate, 74HC86, when the output reacts appropriately to the truth table, indicating that the IC is operating correctly. The same is true for other ICs.

3.3. Test the operation when IC is faulty

As illustrated in figure 12, connect the circuit between the Arduino and IC on the test board. Send the appropriate input values, then verify that the outputs are in line with the truth table.

Figure 12. Faulty IC gate operation Testing

When the outputs' answer deviates from the truth table due to an IC malfunction, the LCD will indicate this by displaying the output status code "0000" or "1111" along with the words "Unknown IC."

3.4. Assembly and packaging

After every component has been inspected, we will assemble the parts and other system components together to create a complete circuit, box the circuit for protection, and create the final product, which is depicted in figure 13.

Figure 13. Device after packaging

3.5. Operational Assessment and Testing

As illustrated in figure 14, insert each IC to verify its functionality and assess the device's accuracy.

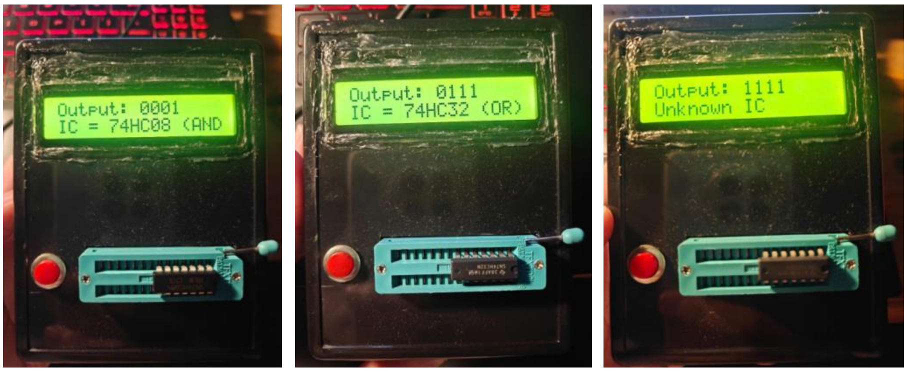


Figure 14. The device's operation testing

We test and identify the ICs 500 times at random using functional ICs to assess the device's accuracy.

TABLE 2. Evaluation results

IC Module	Number of times	Correct	Accuracy
74HC00	41	39	95.12%
74LS00	36	8	22.22%
74HC02	32	31	96.88%
74LS02	28	7	25%
74HC04	42	38	90.48%
74LS04	37	10	27.03%
74HC08	35	33	94.29%
74LS08	34	7	20.59%
74HC32	39	35	89.74%
74LS32	27	4	14.81%
74HC86	30	30	100%
74LS86	43	9	20.93%
74HC226	33	30	90.91%
74LS226	43	5	11.63%

Table 2 summarizes the test findings, which indicate that while the 74LS IC series has inaccurate results, the 74HC IC series has very accurate test results.

4. Conclusion

The project is essentially finished, and its purpose is to construct a tool that can rapidly test logic gate integrated circuits. As a simple tool, the logic gate IC tester may assist students in testing logic gate ICs while they practice and build circuits, which will shorten debugging times and enhance the caliber of useful products. This device is efficient, inexpensive, quick to react, and power-efficient. The product is incredibly versatile due to its small size and low weight. In essence, the system has completed tasks like identifying whether the IC is functioning correctly and recognizing the associated IC name, which is highly helpful in situations when the IC is blurred or missing a label. However, in this model, accurate testing results have only been achieved with the 74HC IC series, while testing with the 74LS IC series has not yet yielded reliable outcomes. Future iterations will incorporate additional features and identify the issue to test the 74LS IC series more precisely.

References

Miczo, A. (2003). *Digital logic testing and simulation*. John Wiley & Sons.

Strunz, B. (2015). Design for Testability in Digital Integrated circuits [Electronic resource]. *Mode of access: http://www. cs. colostate. edu/~ cs530/digital_testing. pdf.–Date of access, 1*.

Darji, C. (2021, March). *Design and Implementation of Microcontroller Based Digital Logic Gate IC Tester*.

Grout, I. (2021, September). Digital IC test development engineering teaching and learning in higher education. In *2021 30th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)* (pp. 1-6). IEEE.

Mohammed, A. A. S., Shakor, P., Abdullah, S. K., Mohammed, M. A., Abbas, S. J., Salman, H. M., ... & Kalmatov, R. K (2024). *Development and Applications of a Microcontroller-Based Logic Tester for Integrated Circuits in the 74xx Series*.

Saw, A., Pradhan, A., Dash, N., Rout, M., & Samal, S. (2025, February). A Novel Method for Testing Digital ICs. In *2025 3rd IEEE International Conference on Industrial Electronics: Developments & Applications (ICIDeA)* (pp. 1-4). IEEE.