A Digital Device for Automatic Checking of Homework Assignments in the Digital Circuits Course

Sergey I. Gavrilenkov[1*], *Elizaveta O.* Petrenko[1], and *Evgeny V.* Arbuzov[1]

[1]Bauman Moscow State Technical University, 2nd Baumanskaya str., 5/1, 105005, Moscow, Russia

Abstract. This paper considers a digital device for automatic checking of homework assignments in the digital circuits course. The assignment is to make a digital circuit corresponding to a given logical expression; the circuit is comprised of elementary logic gates. The process of manual testing the built circuit is very labor-intensive because checking a circuit with N inputs variables requires checking the correctness of the output variable for 2^N cases. We propose automating this pro-cess with a special digital device. The device is comprised of a microcontroller connected to the circuit tested. The microcontroller is connected to a personal computer with an application written in C# for executing the main operations of the testing process. During testing, the student chooses from a database or enters the logical expression corresponding to the circuit tested. For the expression, the software generates truth tables where actual and required responses of the circuit are given. Actual circuit responses are acquired by probing the circuit via the microcontroller, and the expected values are calculated from the logical expression. The truth tables are then presented to the student with a message of whether the circuit works correctly or not. The device was integrated into the process of checking homework assignments in the digital electronics course, and it significantly sped up the process of checking homework assignment circuits, resulting in better education quality.

1 Introduction

Today, education is increasingly dominated by digital technologies, whereby all information used in education (sound, text, video, etc.) is transformed in the digital form [1-2]. The digitalization of education also manifests itself in taking tests using computers, online universities, and online courses [3]. Besides, educational tools are becoming increasingly digitized [4-5]. Another important aspect of the digitalization of education is the automation of routine education tasks, for example, checking homework assignments using digital technology.

However, when a homework assignment is implemented as a piece of hardware, the task of checking it becomes not as straightforward. For example, students of the

[*] Corresponding author: gavrilenkovsergei@bmstu.ru

Department of Computer Systems of Manufacturing Automation of the Bauman Moscow State Technical University have a course in automation equipment, which is also dealing with digital circuits. Within the framework of the course, the students have to complete the homework assignment “Solving logical equation using digital circuits." The homework assignment is done as follows. First, the student must gener-ate a truth table for a given logical expression. The expression may represent the logic of a digital unit controlling an actuator based on boolean variables coming from various sensors. Equation 1 gives an example of the expression.

$$Y = X_1 \wedge X_2 \wedge X_3 \vee X_4 \wedge X_5 \wedge X_1 \quad (1)$$

where $X_1 \ldots X_5$ are the input variables, and Y is the output variable.

Second, the student must simplify the logical expression. Third, the student must make a digital circuit whose truth table corresponds to the truth table of the given logical expression. Then the student must model the circuit in Multisim and check if the circuit is correct in terms of the truth table.

Lastly, the student must find the integrated circuits (ICs), make a PCB for the circuit, then assemble the circuit using a solderless prototyping board. Earlier, the students had to solder the circuit themselves, which took a while.

For brevity, we shall refer to the tested homework assignment circuit as the homework circuit.

There are many ways of testing the homework circuit. Currently, the homework circuit is assembled on a NI ELVIS II board. The board has space for solderless circuit assembly, and such features as virtual instruments (signal generators, scopes, multimeters).

The testing is done as follows. The inputs of the homework circuit are connected to the digital I/O ports of NI ELVIS II, Fig. 1, and the homework circuit output is connected to a built-in NI ELVIS II LED (the device has a dedicated block of LEDs for debugging digital circuits). After that, the Digital Writer virtual instrument is used to set the values of all input variables. The LED indicates the state of the output variable, and this state is compared to the required state determined by the truth table for the given logical expression of the assignment.

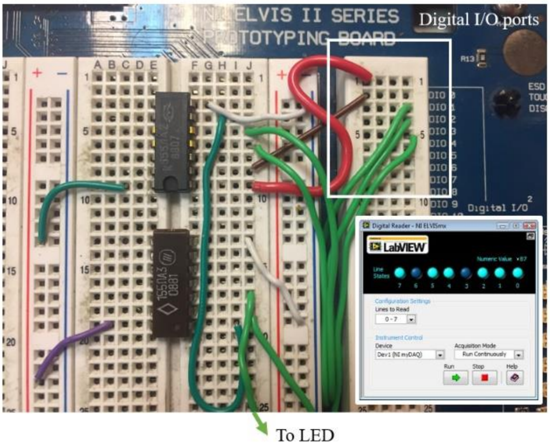


Fig. 1. Connection of the assembled homework circuit to the digital I/O ports of NI ELVIS II (marked with a rectangle) for manual testing and the Digital Writer virtual instrument (in the inset).

This procedure is repeated for all rows of the truth table. This process is very labor-intensive because testing a circuit with N inputs would require checking 2^N combination of the input variables. Besides, if the homework circuit was incorrect, the student would have to repeat this procedure all over again after eliminating all errors.

Earlier, we attempted to automate this process [6] using a portable device for homework checking, Fig. 2.

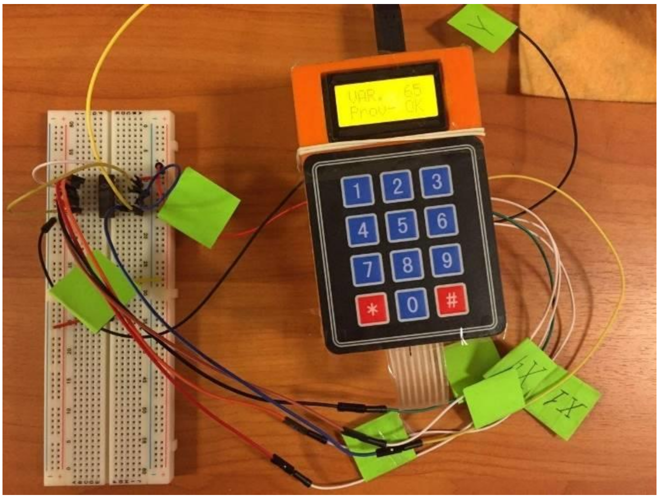


Fig. 2. Portable device for automatic checking of homework circuits.

The device is comprised of an Arduino Uno microcontroller. The device has homework logical expressions hard-coded in the microcontroller program. The expressions correspond to various options for the homework assignment. The device also has a keyboard for entering the assignment option number and an LCD for showing the test results. Although this device greatly simplified the process of testing homework circuits, it had a number of drawbacks:

- The testing could only be done for the expressions hard-coded in the Arduino program. The program would have to be re-written and re-uploaded, should the list of expressions for the homework assignment change.
- If the contact between a *correct* homework circuit and the device was bad, the device could not correctly communicate with the circuit, and it would display that the circuit is incorrect. In this case, the student could not establish that there was a connection problem because the device did not display the truth table due to the small size of the display.

There is much research devoted to developing IC testers. In [7], a reconfigurable digital IC tester implemented using the ARM integrator rapid prototyping system was proposed. In [8], a software system for testing and grading of circuits done with digital logic simulators was proposed. Wu et al. [9] presented a system for building a virtual representation of circuits for further debugging and sharing. Nolan at el [10] presented a LabView-based IC parametric tester for engineering education. Although this approach [10] is very flexible and powerful, the use of this approach is limited by the number of available workstations having a LabView license and the corresponding hardware. In [11], a digital tester comprised of an Arduino microcontroller and a MATLAB program was proposed. In [12], an Arduino-based IC tester was presented.

The device for automatic testing of homework assignment circuits must be made as flexible as possible and have as little hardware (NI ELVIS) and software (MATLAB, LabView) requirements as possible in order to create as many workstations for testing homework assignment circuits as possible. The goal of this paper is to create such a device. Besides, the device must not have the drawbacks of the earlier version [1] mentioned above.

2 Development of the device

2.1 Device architecture

From the standpoint of device architecture, Fig. 3, there are two levels:

- Level 1 contains a microcontroller. The student connects dedicated digital I/O ports of the microcontroller to the input of the circuit tested;
- Level 2 contains a special application on a desktop computer (laptop). The application uses the microcontroller to probe the circuit tested, and it also performs the main procedures of circuit testing. The application interacts with the microcontroller via a serial port.

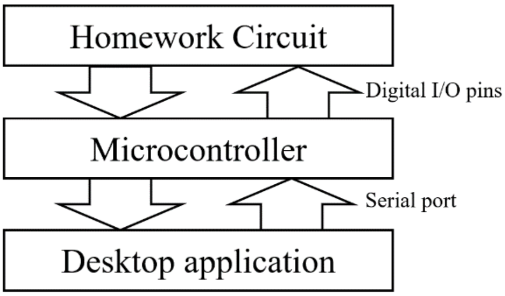


Fig. 3. Architecture of the proposed device.

Having the application at Level 2 enables a user-friendly presentation of the required and actual truth tables, choosing a logical expression from a database or entering a custom logical expression. Having the capability to test user-defined expression can be useful for testing individual ICs or sub-circuits of the circuit tested.

2.2 Choosing a Level 1 microcontroller and a programming framework for the Level 2 application

Arduino Uno (or an Arduino-compatible device) is an optimal choice for the Level 1 microcontroller because Arduino Uno is ubiquitous in the DYI community due to its simplicity and the ease of programming. Besides, it is relatively cheap (the original version costs around 30 USD and an Arduino-compatible version may be as cheap as just 5 USD). This device also has a serial port to communicate with a personal computer.

Generally, logical expressions in the homework circuit have five input variables. Therefore, the microcontroller must have at least six digital I/O pins (5 pins for setting the input variables of the homework circuit and one pin for reading the homework circuit response). Arduino Uno has 14 digital I/O pins, which fully meet this requirement.

Currently, Windows 10 is the most popular desktop operating system [13]. There-fore, if the Level 2 application is written in C# to target the Microsoft .NET framework version

3.0 (or higher), then the application can be used on many desktops and laptops, including those belonging to students. This approach can significantly increase the scale of the homework checking activities provided there are enough microcontrollers.

2.3 Algorithms of the microcontroller and the Level 2 application

Figure 4 presents a flowchart of the Level 2 application algorithm.

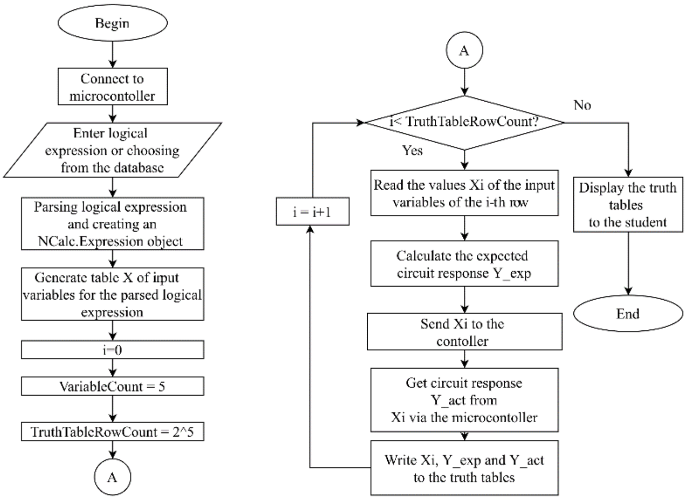
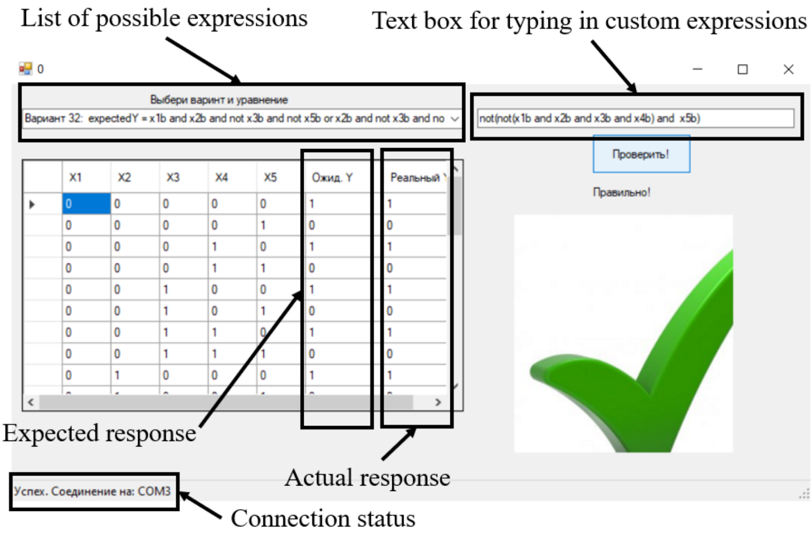

Begin
Connect to microcontoller
Enter logical expression or choosing from the database
Parsing logical expression and creating an NCalc.Expression object
Generate table X of input variables for the parsed logical expression
i=0
VariableCount = 5
TruthTableRowCount = 2^5
A
A
i< TruthTableRowCount?
No
Yes
Read the values Xi of the input variables of the i-th row
Calculate the expected circuit response Y_exp
Send Xi to the contoller
Get circuit response Y_act from Xi via the microcontoller
Write Xi, Y_exp and Y_act to the truth tables
i = i+1
Display the truth tables to the student
End

Fig. 4. Flowchart of the Level 2 application.

First, the application establishes communication with the microcontroller. If communication is established, the status of the application changes. After that, the student can either choose a logical expression from the list or enter a custom logical expression.

After that, the application parses the logical expression and makes an instance of the *NCalc.Expression* class [14]. Then, the application generates all possible combinations of the truth table for the homework circuit based on the number of input variables. For each row of the truth table, the application does the following. First, the application sends the values of the input variables to the microcontroller via the serial port. Second, the microcontroller sends them to the homework circuit and sends the circuit's response back to the application via the serial port. Third, the application calculates the expected homework circuit response according to the logical expression.

The application was written in C# using a GUI class library Windows Forms, which is a part of the Microsoft .NET Framework. Figure 5 shows the user interface of the Level 2 application and specifies various features of the application.


List of possible expressions
Text box for typing in custom expressions
Expected response
Actual response
Connection status
0
Проверить!

Выбери варинт и уравнение

Вариант 32: expectedY = x1b and x2b and not x3b and not x5b or x2b and not x3b and no

not(not(x1b and x2b and x3b and x4b) and x5b)

	X1	X2	X3	X4	X5	Ожид. Y	Реальный Y
▶	0	0	0	0	0	1	1
	0	0	0	0	1	0	0
	0	0	0	1	0	1	1
	0	0	0	1	1	0	0
	0	0	1	0	0	1	1
	0	0	1	0	1	0	0
	0	0	1	1	0	1	1
	0	0	1	1	1	0	0
	0	1	0	0	0	1	1

Правильно!

Успех. Соединение на: COM3

Fig. 5. User interface of the Level 2 application.

Figure 5 presents the flowchart of the microcontroller main loop algorithm.

At each cycle, the microcontroller gets the command identifier if the serial port is available. If the device has to be identified to establish a connection with the Level 2 application, the controller sends a unique identifier in the serial port, so that the application would distinguish the microcontroller from other devices at COM ports.

If the connection is established, the application sends the values of the input variables, the microcontroller reads them from the serial port and sets the corresponding boolean values of the digital I/O pins. After that, the microcontroller reads the home-work circuit response at a dedicated digital I/O pin and sends this response back to the Level 2 application via the serial port. Figure 6 shows the user interface of the application.

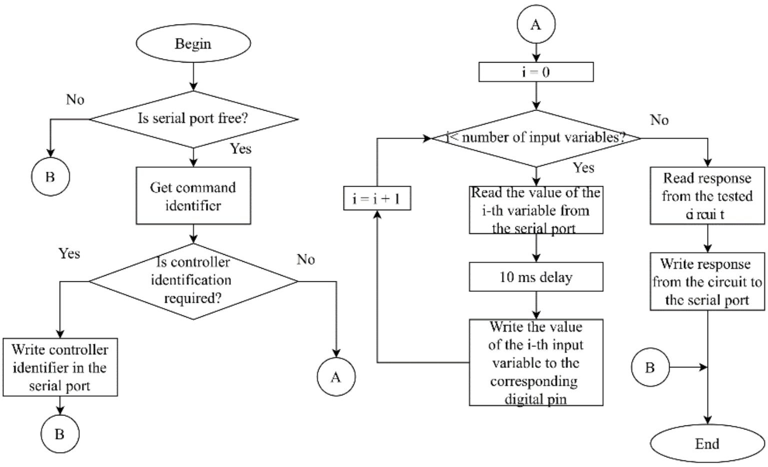

Begin
Is serial port free?
No
Yes
B
Get command identifier
Is controller identification required?
Yes
No
Write controller identifier in the serial port
B
A
A
i = 0
i< number of input variables?
No
Yes
Read the value of the i-th variable from the serial port
10 ms delay
Write the value of the i-th input variable to the corresponding digital pin
i = i + 1
Read response from the tested d rcui t
Write response from the circuit to the serial port
B
End

Fig. 6. Flowchart of the microcontroller main loop algorithm.

2.4 Device approbation

Let us show how the developed device can be used to check the homework circuit corresponding to the following logical expression:

$$Y = \overline{\overline{X_1 \wedge X_2 \wedge X_3 \wedge X_4} \wedge X_5} \qquad (2)$$

Figure 7 shows the schematic of the circuit corresponding to this logical expression. The circuit is comprised of a 4-input NAND gate (SN7420) and a 2-input NAND gate (SN7400).

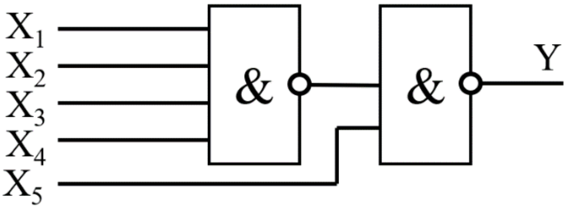


Fig. 7. Circuit schematic.

After assembly, the circuit inputs are connected to the digital pins 2…6 of the microcontroller, and the circuit output is connected to the digital pin 7. The power supply of the circuit comes directly from the microcontroller, so there is no need for a standalone battery pack. The microcontroller is connected to a laptop via a USB 2.0 cable. Figure 8 shows a photo of the arrangement.

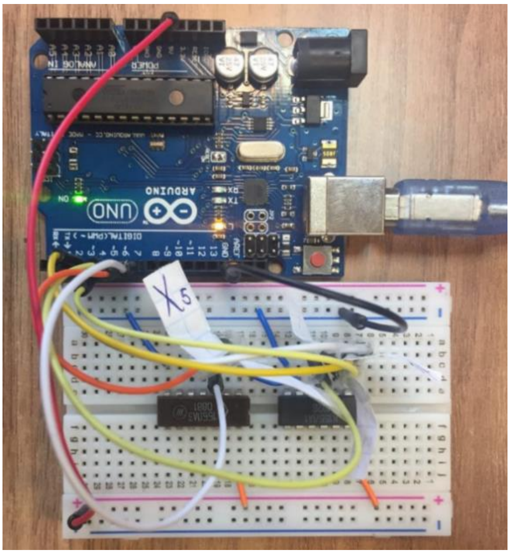


Fig. 8. Connection of the homework circuit and the microcontroller.

After that, we enter the text of the logical expression in the text box for custom expressions in the following form:

not(not(x1b and x2b and x3b and x4b) and x5b) (3)

Then we initiate the testing and inspect the truth tables, Fig. 5. If the expected and actual responses of the circuits match for all rows in the truth table, the applications display the green mark symbol, meaning that the circuit is working as intended. Otherwise, we have to check all the connections and test again.

The developed device was used to check homework circuits in the spring semester of the year 2019. Generally, the device was approved by the students due to the high speed of checking circuits and the ease of use. However, we received some comments from the students:

- If the user typed in a logical expression with syntax errors, the application raised an error because NCalc was not capable of parsing the expression;
- Students had difficulties detecting faults in the circuit based solely on the truth table, and they requested a feature whereby the device would make a recommendation where the faults in the circuit could be.

The first issue is easy to tackle; however, implementing the second feature would require additional research to be done in future studies.

3 Conclusion

In this paper, we developed the device for automatic testing of digital circuits assembled by students within the framework of completing the homework assignment in the digital electronics course. The device is comprised of the Arduino microcontroller connected to a personal computer with the special application written in C#. The application, based on the input logical expression, generates a truth table for the expression. Then it probes the homework assignment circuit via the microcontroller to generate an actual truth table. The device improved the speed and quality of the process of checking circuits in the homework assignments, but diagnosing faults based solely on the truth tables built by the device remains complicated. This issue can be solved by implementing a special algorithm for detecting possible faults in the circuit based on the truth table of the circuit assembled.

References

1. E.V. Smirnova, A.A. Dobrjkov, A.P. Karpenko, V.V. Syuzev, *Mentally structured educational technology and engineers preparation quality management*, Commun. Comput. Inf. Sci. **754**, pp. 119-132 (2017)
2. A.I. Vlasov, L.V. Juravleva, V.A. Shakhnov, *Visual environment of cognitive graphics for end-to-end engineering project-based education*, J. Appl. Eng. Sci. **17**, pp. 99–106 (2019)
3. I.G. Doronkina, E.E. Krasnovskiy, E.V. Zakharova, P.V. Ulianishchev, L.V. Ulyanishcheva, *Academic cloud services: Innovative solutions and international practice*, J. Adv. Res. Dyn. Control Syst. **11**, pp. 65–72 (2019)
4. N.M. Mezhennaya, O.V. Pugachev, *On perception of computer algebra systems and Microsoft excel by engineering students*, Probl. Educ. 21st Century. **77**, pp. 379–395 (2019)
5. N.M. Mezhennaya, O.V. Pugachev, *On the results of using interactive education methods in teaching probability theory*, Probl. Educ. 21st Century. **76**, pp. 678–692 (2018)
6. T. Fazli, S. Gavrilenkov, E. Petrenko, *A device for automatic testing of digital circuits to improve speed and quality of checking homework assignments in the automation*

equipment course, Herald of the Eurasian Science **9(3)**, p. 40 (2017) http://naukovedenie.ru/PDF/37TVN317.pdf

7. F. Pang, T. Brandon, B. Cockburn, M. Hume, *A reconfigurable digital IC tester implemented using the ARM Integrator Rapid Prototyping System*, Canadian Conference on Electrical and Computer Engineering, pp. 1931–1935, IEEE, Ontario (2004)
8. Z. Kurmas, *Improving student performance using automated testing of simulated digital logic circuits, Proceedings of the Conference on Integrating Technology into Computer Science Education*, ITiCSE. pp. 265–269, ACM, Madrid (2008)
9. T.-Y. Wu, B. Wang, J.-Y. Lee, H.-P. Shen, Y.-C. Wu, Y.-A. Chen, P.-S. Ku, M.-W. Hsu, Y. –C. Lin, M.-Y. Chen, CircuitSense: *Automatic sensing of physical circuits and generation of virtual circuits to support software tools*, Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST), pp. 311–319, ACM, Quebec City (2017)
10. L. Nolan, M.-T. Chew, S. Demidenko, M. Ooi, *Virtual instrumentation based IC parametric tester for engineering education*, Proceedings - 5th IEEE International Symposium on Electronic Design, Test and Applications, DELTA 2010. pp. 310–315, IEEE, Ho Chi Minh City (2010).
11. Arduino-based digital IC tester using MATLAB, https://www.electronicsforu.com/electronics-projects/arduino-based-digital-ic-tester-matlab, last accessed 2019/11/02.
12. Arduino-based digital IC tester, https://www.electronicsforu.com/electronics-projects/hardware-diy/arduino-based-digital-ic-tester-truth-table, last accessed 2019/11/02
13. StatCounter Global Stats report, https://gs.statcounter.com/os-market-share#monthly-201809-201908-bar, last accessed 2019/11/02.
14. NCalc library homepage on Github, https://github.com/MichaelAguilar/NCalc, last accessed 2019/11/02.