One Approach to Studying the Topic "Arithmetic Circuits: Design, Simulation and Debugging"

MICHAEL DOLINSKY
Faculty of Mathematics and Programming Technologies,
Francisk Skorina Gomel State University,
Sovetskaya Str.104, 246019, Gomel,
BELARUS

Abstract: - This article describes the technology of teaching the theme "Arithmetic circuits: design, simulation and debugging" of basic digital electronics course to first/second-year students based on the DL.GSU.BY website. The main advantages of the technology include training adapted to the student, many years of experience in practical application, and effectiveness. The following issues are consistently considered in the article: the theoretical foundations of the topic; a library of standard components; a system of step-by-step learning to design arithmetic circuits, and a technology for simulation and debugging.

Key-Words: - basic digital electronics, arithmetic circuits, design, simulation, debugging, website DL.GSU.BY, HLCCAD.

Received: July 23, 2022. Revised: August 26, 2023. Accepted: September 25, 2023. Published: November 28, 2023.

1 Introduction

Blended learning has been actively developed before COVID-19, but with the advent of this disease, it has become critical. Works, [1], [2], substantiate the need for a transition to blended learning, which combines the possibilities of traditional classroom learning and new information technologies. Approaches to the transition to blended learning includes works, [3], [4]. A variety of languages and systems used in teaching, among the proposed systems are mentioned "GeoGebra", [5], Google Classroom, [6], Moodle, [7], [8]. The real experience of blended learning with the help of computer technologies in universities is described in the following papers: programming, [9], computer thinking, [10], chemistry, [11], basic medical knowledge, [12], earth sciences, [13], anatomy, [14], WEB design, [15], neuroanatomy, [16], English language, [17], new educational technologies, [18].

The author has been using blended learning for disciplines related to teaching programming and digital electronics for many years at the Faculty of Mathematics and Programming Technologies (specialties: "Information Technology Software", "Informatics and Programming Technologies", "Applied Informatics") of Gomel State University named by Francisk Skorina.

To increase the effectiveness of training, the system for designing, simulation, and debugging digital electronics circuits HLCCAD, and the instrumental system for distance learning, both developed under the guidance of the author are used.

The author already describes approaches to study the topics "Logic and combinational circuits", and "Synthesis of combinational circuits using Carnot maps, [19].

This work presents the author's approach to the study of the topic "Arithmetic circuits: design, simulation and debugging".

2 Problem Formulation

The author has been teaching disciplines related to digital electronics at the Faculty of Mathematics of Gomel State University named by F.Skorina for many years.

All this time, learning conditions are rapidly changing: computer tools and Internet technologies are improving, the average level of preparation and motivation of students is decreasing, and at the same time, the requirements for the knowledge and skills of university graduates are increasing. This leads to the need to change the learning process, so that, on the one hand, theoretical knowledge is given in much more detail and a much simpler language than was previously accepted, on the other hand, to consolidate the acquired knowledge, not only and not so much seminars with students are used, but their solution of practical tasks.

3 Problem Solution

3.1 Theoretical Background

This work is devoted to the description of such a modified methodology for teaching the topic “Arithmetic calculations”, taking into account the trends described above. Before this topic, students have already studied the topics “Introduction to the subject”, “Synthesis of combinational circuits using truth tables”, and “Combination circuits”.

The topic “Introduction to the Subject” is particularly intended for mastering the HLCCAD system, with which you can edit, simulate, and debug functional diagrams of digital devices, as well as for becoming familiar with the basic logical operations NOT, AND, OR, XOR and corresponding basic logic gates.

The topic “Synthesis of combinational circuits using truth tables”, on the one hand, consolidates the knowledge and skills of using basic logical elements NOT, AND, OR, XOR to solve problems in the design of digital devices, on the other hand, develops the skills of analysis (and mental simulation) of circuits, composed of these basic logical elements, and on the third hand introduces the method of minimizing logical functions using Carnot maps.

The topic "Combinational Circuits" introduces students to the basic combinational circuits (decoder, encoder, multiplexer, adder) used for designing and analyzing digital devices.

3.2 The Essence of Design Tasks

When learning the fundamentals of digital electronics, students must gain good practical skills in analyzing and synthesizing a variety of digital devices. A significant help in this is the means of their simulation and debugging. For these purposes, the GSU is named after. F. Skaryna has been developing, operating, and using the HLCCAD system in the educational process for many years. To develop the skills of creating circuits of digital devices in the HLCCAD system and finding errors in them, the topic “Arithmetic circuits” was mainly introduced into the study. In general, the task that the student must complete looks like this: if some condition is met, calculate one arithmetic expression; if the condition is not met, calculate another arithmetic expression. In both expressions, as a rule, at least four variables are used, two of which have a width of two bytes, and the other two have a width of one byte. An example of such a task is shown in Figure 1:

Task No. 1

HLCCAD project name: Arithm.PRD

Input device: Arithm

Develop a device that calculates the value of an expression

```
          / a/b+cd*a              if a*d<0
RES=<
          \ b-(c+a/d)*b,          if a*d>=0
```

Information about inputs and outputs is presented in the table.

Examples:

a= 8	a=10
b= 3	b= 7
c= 5	c=-9
d=-2	d= 3

RES=23 RES=49

Name	Dimension	Type
C	8	input
D	8	input
A	16	input
B	16	input
RES	16	output

Fig. 1: Example of a task on the topic “Arithmetic calculations”

With careful attention to the dimensions of the data, the task is actually to pull out the necessary blocks of arithmetic operations on the diagram and correctly connect their outputs and inputs following the order of actions in the expressions**.**

Obviously, for a student with any level of preliminary preparation, the task is understandable and feasible - you just need to draw a diagram that performs the specified calculations. In this case, the time to complete the task will be determined by the skills of finding and eliminating errors. The process of performing such tasks, almost in its purest form, is devoted to developing technologies and developing practical skills in finding and eliminating errors in circuits.

3.3 Standard Component Library

Within the framework of the topic "Arithmetic calculations," it is necessary to use such basic digital blocks as an adder, subtractor, multiplier, or divider. In addition, a multiplexer is required to select one of the calculated results. In some tasks, you need to choose not from two expressions, but from three. This leads to the need to build a circuit that will calculate the values supplied to the address lines of the multiplexer.

Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6 show elements of the library of Standard components of the HLCCAD system used in tasks on the topic “Arithmetic calculations”: multiplexer, adder, constant.

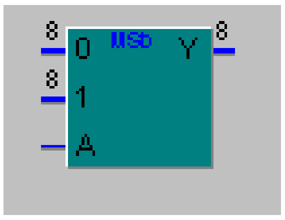


Fig. 2: Multiplexer MSb8x2

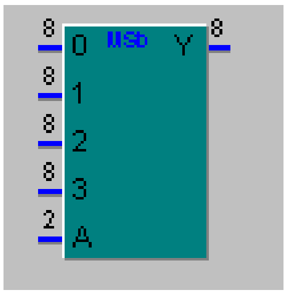


Fig. 3: Multiplexer MSb8x4

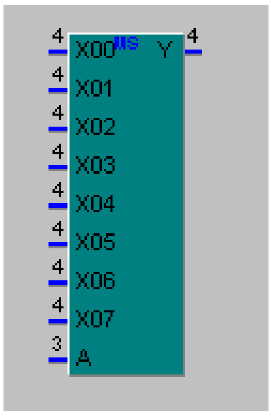


Fig. 4: Multiplexer MSb4x8

<u>Multiplexer MS8b8x2</u>- This is the so-called bus multiplexer. It has an 8-bit bus at the output two 8-bit data buses (0 and 1) and one address line A at the input. If A=0, then the values from the input bus 0 are transmitted bit by bit to the output bus Y. If A=1, then the values from input bus 1 are transmitted bit by bit to the output bus Y.

<u>Multiplexer MSb8x4</u> is also a bus multiplexer. Its output is an 8-bit Y bus, its input is four 8-bit data buses (0, 1, 2, 3) and a 2-bit address bus.

If the address input is 00, then data from bus 0 is transferred to the output.

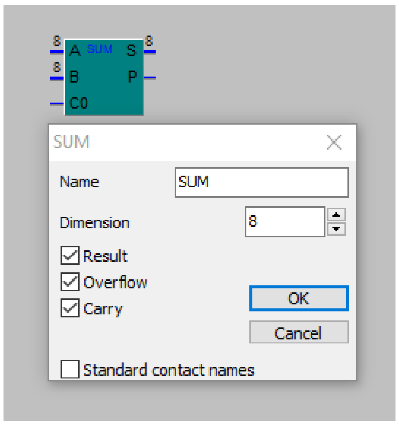
If the address input is 01 (the left bit is the most significant bit), then data from bus 1 is transmitted to the output.

If the address input is 10, then data from bus 2 is transmitted to the output.

If the address input is 11, then data from bus 3 is transmitted to the output.

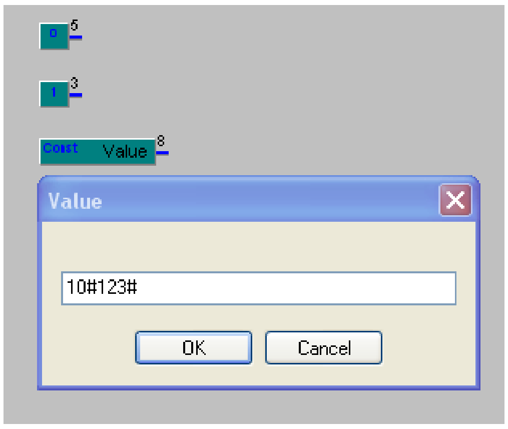
In addition, any of these multiplexers can be converted into any arbitrary multiplexer by changing the number of bits in the address line and the number of bits in the data buses. For example, Figure 4 shows the MSb4x8 multiplexer, which has 3 address lines and 8 four-bit data buses. The output is also a four-bit bus. The first number in the designation (4) indicates the width of the input and output data buses, and the second number (8) indicates how many input data buses there will be (at the same time determining the number of required address lines, as a base 2 logarithm of the number of input data buses: 3 is the logarithm of 8 based on 2).

The adder (Figure 5) is 8-bit by default. However, this bit depth (as for all devices from the Standard project) can be arbitrarily changed, in addition, you can not display (if not necessary) the transfer, overflow, and sum contacts.



Name: SUM
Dimension: 8
☑ Result
☑ Overflow
☑ Carry
☐ Standard contact names

Fig. 5: Adder



10#123#

Fig. 6: Constants

Constants (Figure 6) can be specified in several ways. Firstly, they can be formed as a union of sequences of ones and zeros, using, respectively, constants of ones and zeros of the required bit depth. In addition, there is a constant element (const), in the parameters of which you can specify the number system in which the constant is specified (before the first '#' symbol - decimal by default), and then the constant itself (between two '#' symbols). Note that specifying negative numbers in the decimal number system is not supported. This requires students to understand how negative numbers are represented in two's complement and how they are written in binary, octal, or hexadecimal.

3.4 SignedArithm Circuit Library

The Standard library components (MUL, DIV, CMP) work correctly only with positive numbers (that is, they calculate the result correctly only if the input is positive numbers). At the same time, tasks for calculating arithmetic expressions require correct work with both positive and negative numbers. To resolve this conflict, students developed a library of signed arithmetic schemes, SignedArithm. It includes the following circuits with fixed-width input and output buses: subtraction (iSUB), multiplication (iMUL), division (iDIV), comparison (iCMP), increase in bit width (sign propagation: CBW, CWD), presented in Figure 7:

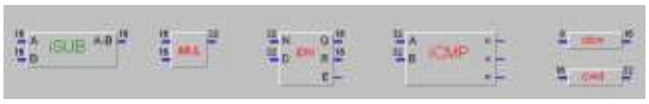


Fig. 7. Signed arithmetic circuits

When solving their problems, students simply pull out these drawings onto their diagrams. At the same time, by clicking on any of them, everyone can see exactly how the corresponding circuit is made based on components from the standard library. Figure 8 shows the circuit of a 16-bit signed subtractor.

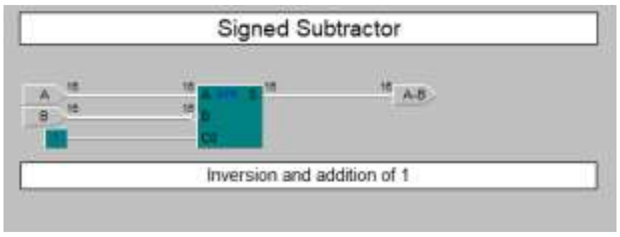


Fig. 8: 16-bit signed subtractor circuit

As is known, to represent a negative number in two’s complement code, it is enough to invert its binary representation and add 1 to the result. In Figure 8, to A subtract the number B, the additional code of the number B is added to the number A.

Figure 9 shows the circuit of a 16-bit signed multiplier. First, the sign of the result is calculated from the signs of the factors (if they have different signs, then the result is negative, and if they are the same, then the result is positive) and in parallel, for each of the factors, the module of the corresponding number is constructed. Then the resulting modules are multiplied. And finally, the output is the resulting product (if the result is positive) and its complementary code if the result should be negative.

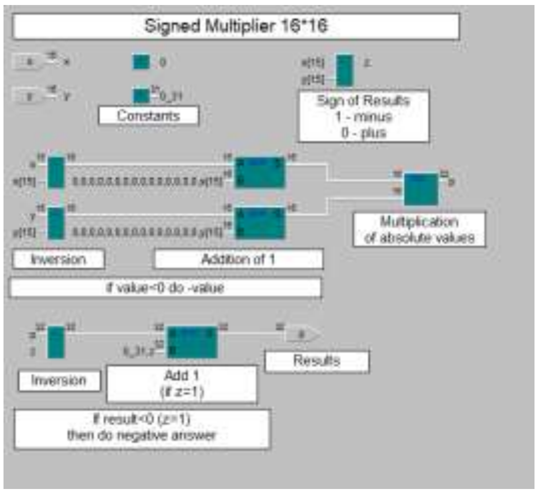


Fig. 9: 16-bit signed multiplier circuit

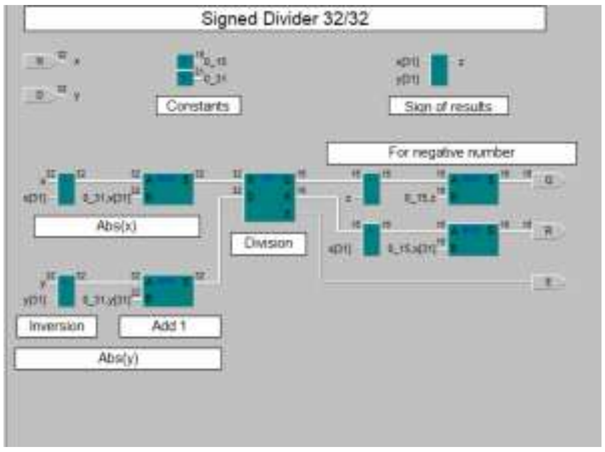


Fig. 10: 32-bit signed divider circuit

Figure 10 shows the circuit of a 32-bit signed divider. First, the result is calculated based on the signs of the dividend and divisor (if they are of different signs, then the result is negative, and if they are the same, then positive) and in parallel, the module of the corresponding number is constructed for the dividend and divisor. Then the modulus of the dividend is divided by the modulus of the divisor. Finally, the output is the resulting result (if it is positive) and its additional code if the result should be negative. When dividing (integer!), the quotient, remainder, and error flag (division by 0) are constructed, as in the standard DIV component.

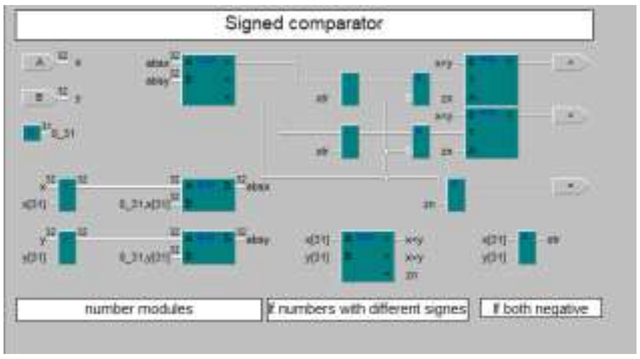


Fig. 11: Schematic of a 32-bit signed comparator

Figure 11 shows the student-developed 32-bit comparator circuit; this is a properly working design that has been used in practice for many years. A few years later, a clever student proposed an alternative option, much simpler (Figure 12):

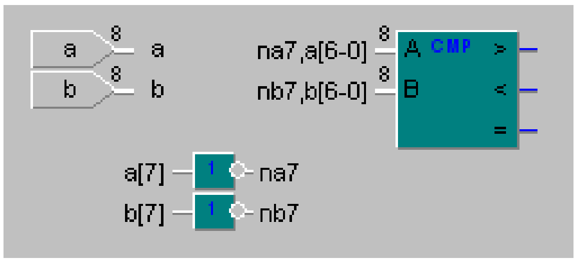


Fig. 12: The 8-bit signed comparator circuit

The most significant bits of both numbers being compared are inverted, and the resulting numbers are compared using a standard comparator that compares positive numbers. The following is an explanation of the scheme by student (its author): "*The question was how to turn an unsigned comparator into a signed one. After all, you can simply add 2^(n-1) to each of the numbers (n is the number of bits). No one will argue that this will not change the result of the comparison? It won't change, but both numbers will become positive and an unsigned comparator can be used.*

So, when we invert the most significant bit of our numbers, we get not just some other incomprehensible numbers, but positive numbers, the difference of which remains the same.

for 4-bits:
-positive
  0xxx
+1000
  1xxx - increased by 8 (2^3)
-negative
  1xxx
+1000
1|0xxx - increased by 8
^
"leaves", because we have 4 bits

We have received 2 numbers in unsigned form, to which we can apply an unsigned comparator. In short, by inverting the most significant bit, we add the same number to both numbers, making them positive. This does not change the result of the comparison."

Since the multiplication, division, and comparison circuits have fixed bit widths of the operands, we need to be able to change the bit depth of the numbers with which we operate. To increase the bit depth, the CBW (from 8 bits to 16 bits) and CWD (from 16 bits to 32 bits) devices, presented in Figure 13 and Figure 14, are used.

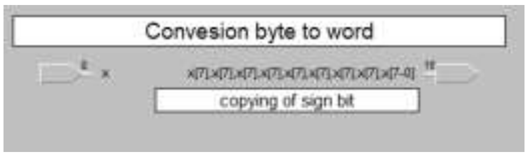


Fig. 13: Scheme for converting bytes to words

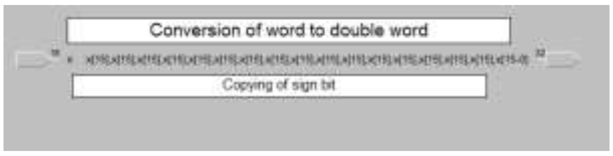


Fig. 14: Scheme for converting a word into a double word

3.5 Recommended Technology for Completing the Task

For the addition operation, use the SUM adder from the Standard project (it works correctly with both positive and negative numbers). For all other arithmetic operations (subtraction, multiplication, division, comparison), we take devices from the SignedArithm project (iSUB, iMUL, iDIV, iCMP, respectively) (Figure 15).

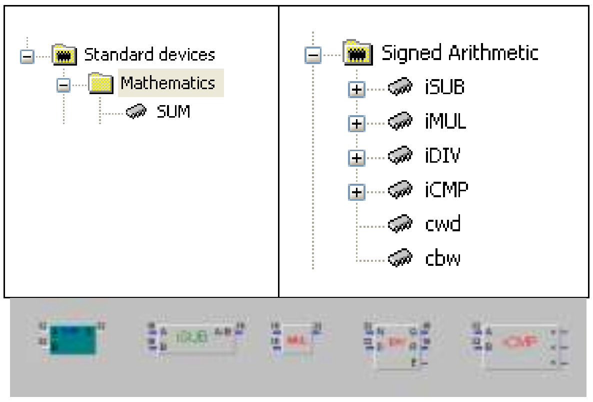


Fig. 15: Blocks of used operations: arithmetic and comparison

IT IS FORBIDDEN to change the bit depth of devices (iSUB, iMUL, iDIV, iCMP) from the SignedArithm project (unfortunately, we don't have "foolproof" there yet).

One of the problems is the different bit depths of the source data. It must be solved by leveling the source data to 16 bits using the CBW device (x8 => CBW => x16) and to 32 bits using the CWD device (x8 => CBW => x16 => CWD => x32). The bit-increasing blocks are presented in Figure 16.

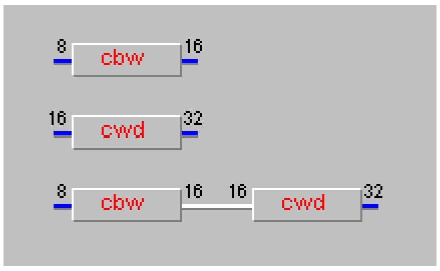


Fig. 16: Bit increasing blocks

In all tasks, it is guaranteed by tests that the answer will not exceed a signed integer of 16 bits.

If you need to reduce the bit depth (usually after multiplication), you can do this: designate the output (for example, Y, let it be 32-bit). To take the lower 16 bits of this output at the 16-bit input of the device, just write Y, [15-0] (after clicking on the input line). The method for reducing the bit depth is presented in Figure 17.

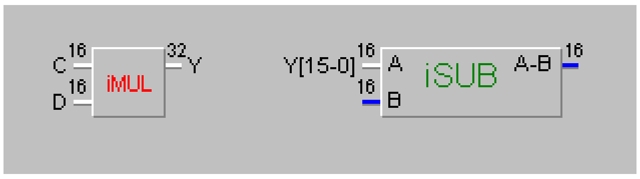


Fig. 17: Method for reducing the bit depth

To select one of several results, use the bus MULTIPLEXER (MSb8x2) from the Standard project (Figure 18):

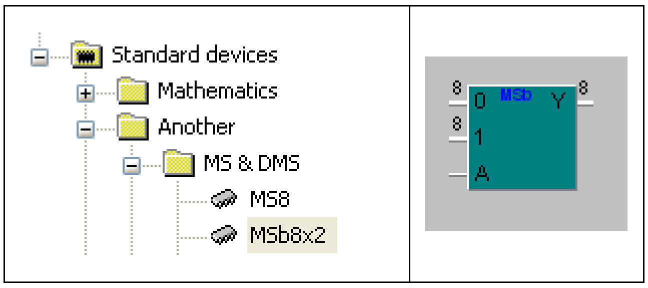


Fig. 18: Standard multiplexer MSb8x2

All tasks require a 16-bit bus multiplexer (Figure 19); it can be obtained from an 8-bit one using an external editor (by right-clicking on the case and selecting the line "External Editor").

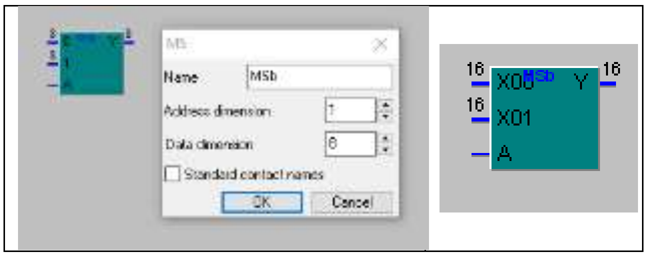


Fig. 19: 16-bit multiplexer

The result must be sent to the x01 bus res1, if the condition is met (line A is 1) res0, if the condition is NOT met (line A is 0) (Figure 20).

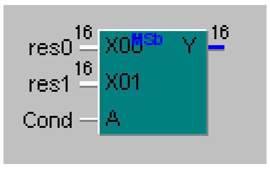


Fig. 20: The meaning of the multiplexer input contacts

The Cond condition is calculated using the iCMP comparison scheme (Figure 21):

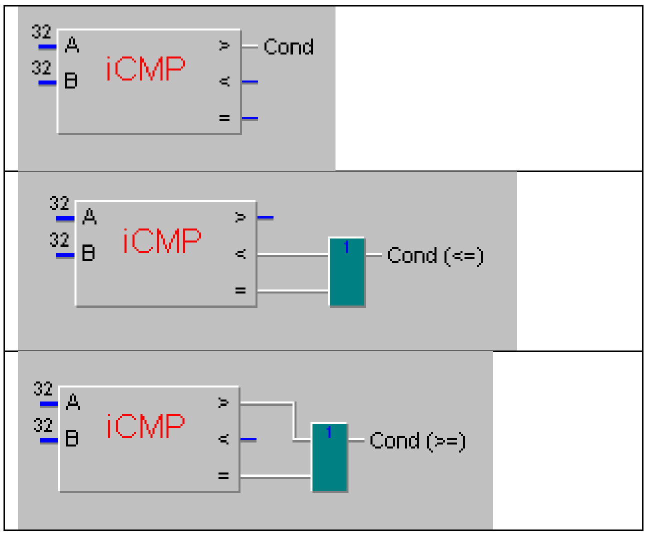


Fig. 21: Condition calculation

3.6 Circuit Debugging

Figure 22 shows a fragment of debugging a circuit for calculating an arithmetic expression. A test is already connected to the circuit, on which the circuit simulation is launched. Since an error is detected (the value at the output of the circuit is not equal to the reference value specified in the test), a debugger window automatically opens, where the circuit is presented, and the currently available value is given at each input and output pin. The current moment in time is automatically selected as the moment in time when the first error was detected.

By manually calculating the expression for the current source data, the student can accurately determine what value should be on each of the output contacts of the circuit devices. By consistently checking the output values with the results of your calculations, you can easily find the first discrepancy - here is the first design error that needs to be corrected and the simulation restarted. This will continue until correct results are obtained for all input data values.

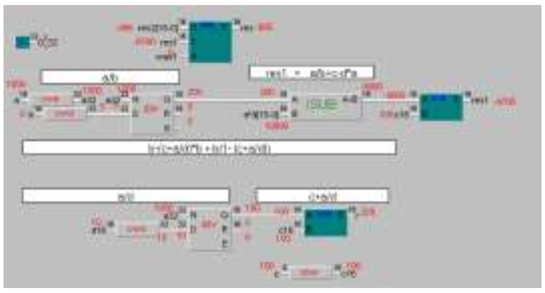


Fig. 22: Fragment of debugging the circuit for calculating an arithmetic expression

After correcting all errors, you need to send the resulting project for final automatic verification in the system DL.

3.7 Advanced Tasks

Of the 30 variants of these tasks, there are several of the most difficult ones, in the conditions of which it is indicated that when calculating arithmetic expressions, it is necessary to calculate the modulus, sign, or factorial of a number (for example, the below-presented tasks 7,21,5).

Task 7

```
         /   c/a + b/d – a* c          if a=4
Res = <
         \   (a + b – sign(c*d)/a      if a<>4
```

Task 21

```
        /   x! + y/(a-b)               if   1<=x<=7
Res = <
        \   7! + x*y/(a-b)             in other cases
```

Task 5

```
         /  | (a-b)/c|  + d*c      if d>0
Res = <
         \  | |a|  - |b|| * d / c  +  sign(a*b*c*d)    else
```

To simplify the completion of such tasks, there are special training tasks, some of which are given below in Figure 23, Figure 24, and Figure 25. In fact, in each of them, the student is offered a scheme that solves the problem. At the same time, the student is required to understand how the circuit functions to complete it by selecting the correct pin labels.

In the scheme for calculating the modulus of a number, presented in Figure 23, its complementary code is calculated for a number, and, using a multiplexer, either the number itself (if it is positive) or its complementary code (if the original number was negative) is selected.

In the calculation scheme **sign** of the number presented in Figure 24, the number is sent to the comparator with zero. As a result, we get three single lines (x>0, x=0, x<0), which are logically multiplied by the constants 1, 0, and -1 (16 ones), respectively. The disjunction of these three conjunctions determines the result.

In the calculation scheme **factorial** of number (in the range from 0 to 15) a multiplexer is used, with 16 data buses at the input, each of which is supplied with a corresponding pre-calculated constant from 0! up to 15! (Figure 25).

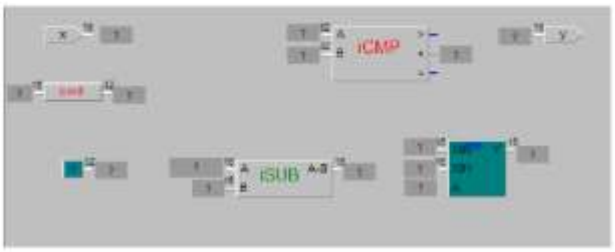


Fig. 23: Circuit for calculating the modulus of a 16-bit number

$$y = \mathrm{sign}(x) = \begin{cases} 1, & x>0 \\ 0, & x=0 \\ -1, & x<0 \end{cases}$$

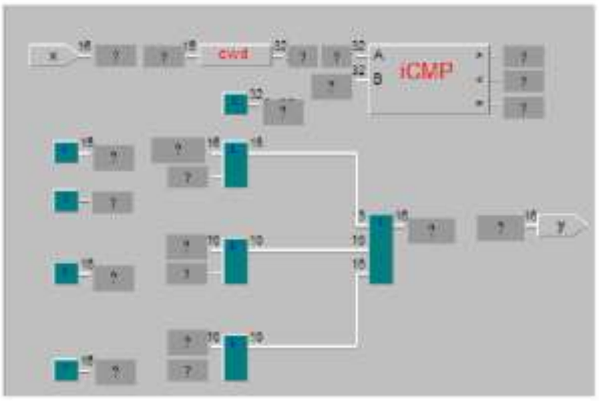


Fig. 24: Circuit for calculating the sign of a 16-bit number

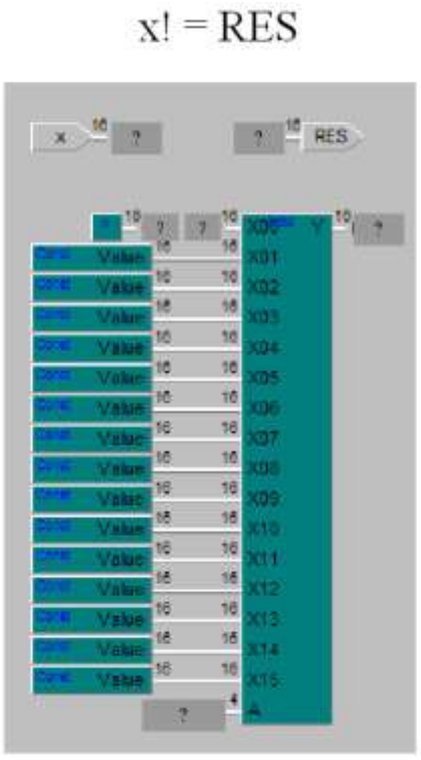


Fig. 25: Circuit for calculating the factorial of a 16-bit number (with a value from 0 to 15)

3.8 Functions of the DL.GSU.BY Distance Learning System in the Educational Process

The DL system performs the following basic functions: presentation of the task to the student, acceptance and prompt verification (within a minute) of the circuit designed by the student, and instant notification to the student of the result of checking the circuit. In case of an error, the student is given a test in which his circuit produced the wrong answer. The test describes input influences and standard responses. A table of the results of checking the work of all students is built in real-time.

System DL performs the following additional functions: provides the student with full access to the theory of the subject, including the full text of the lecture on this topic; supports a forum that has been functioning for many years, in which students ask their questions, the teacher answers and adds thematically systematized links to questions and answers on the first page of the forum; supports random selection of a test option for a student, prohibiting access to the Internet and network folders during tests.

To support students who missed classes or have difficulty mastering the subject, there are video recordings of students solving all 30 test options, which can be used during self-study.

4 Conclusion

This article describes a method developed by the author and repeatedly tested for studying the topic “Arithmetic Calculations”, aimed at working in groups of students with fundamentally different levels of motivation and preliminary training. A serious technical basis for the methodology is the developed instrumental system of distance learning (Distance Learning Belarus [20]). The introduction of this teaching methodology has ensured significant changes in the quality of education, especially for the least prepared and motivated category of students. At the same time, the most prepared students are also satisfied with this approach to learning.

References:

[1] Aznam N., Perdana R., Jumadi J, Nurcahyo H., Wiyatmo Y. The Implementation of Blended Learning and Peer Tutor Strategies in Pandemic Era: A Systematic Review *Advances in Social Science, Education and Humanities Research,* Vol. 541, 2021, pp.906-914.

[2] Kakeshita T., Ohtsuki M. Survey and Analysis of Computing Education at Japanese Universities: Non-IT Departments and Courses, *Olympiads in Informatics*, Vol. 13, 2019, pp.57-80.

[3] Jones K., Ravishankar S. *Higher Education 4.0. The Digital Transformation of Classroom Lectures to Blended Learning*, Springer Singapore, 2021.

[4] Zaugg H., Graham C., Lim C., Wang T. Current and Future Directions of Blended Learning and Teaching in Asia, chapter 16 in the book "*Blended Learning for Inclusive and Quality Higher Education in Asia*", 2021, pp.301-327.

[5] Stahl G. Redesigning Mathematical Curriculum for Blended Learning, *Education. Sciences,* Vol. 11, 2021, pp.165-177.

[6] Astarilla L., Warman D. The Effect of Google Classroom in Blended Learning on University Students’ English Ability, *Journal of English for Academic,* Vol 8, No 1, 2021, pp.12-23.

[7] Antwi-Boampong A. Blended Learning Adoption in Higher Education: Presenting the Lived Experiences of Students in a Public University from a Developing Country, *The*

Turkish Online Journal of Educational Technology, Vol. 20, Issue 2, 2021, pp.14-22.

[8] Oktaria S., Sasongko R., Kristiawan M., Development of Blended Learning Designs using Moodle to Support Academics of The Curriculum in University of Bengkulu, *Jurnal Studi Guru dan Pembelajaran (Indonesia)*, Vol. 4, No. 1, 2021, pp.118-126.

[9] Lodi M., Malchiodi D., Monga M., Morpurgo A., Spieler B. Constructionist Attempts at Supporting the Learning of Computer Programming: A Survey, *Olympiads in Informatics*, Vol. 13, 2019, pp.99-122.

[10] Weigend M., Vanicek J., Pluhar Z., Pesek I. Computational Thinking Education Through Creative Unplugged Activities, *Olympiads in Informatics*, Vol. 13, 2019, pp.171-192

[11] Chamberlain S., Elford D., Lancaster S., Silve F. Tailored Blended Learning for Foundation Year Chemistry Students, *Chimia*, Vol 75, 2021, pp.18–26.

[12] Lovey T., O'Keeffe P., Petignat I. Basic Medical Training for Refugees via Collaborative Blended Learning: Quasi-Experimental Design, *Journal of Medical Internet Research*, Vol 23, No 3, 2021 https://doi.org/10.2196/22345.

[13] Bond C., Cawood A. A role for virtual outcrop models in blended learning – improved 3D thinking and positive perceptions of learning, *Geoscience Communication*,Vol.4, Issue 2, 2021, pp.233-244.

[14] Sarkar S., Sharma S., Raheja S. Implementation of Blended Learning Approach for Improving Anatomy Lectures of Phase I MBBS Students – Learner Satisfaction Survey, *Advances in Medical Education and Practice*,Vol. 12, 2021, pp.413-420.

[15] Sulistiyarini D., Sabirin F., Ramadhani D. Effect of Project-Based Learning Through Blended Learning on Website Design Skills //*Journal of Educational Science and Technology*, Vol. 7, 2021, pp.58-66

[16] Border S., Woodward C., Kurn O., Birchall C., Laurayne, H., Anbu, D., Taylor, C. and Hall, S. Working in Creative Partnership with Students to Co-Produce Neuroanatomy e-Learning Resources in a new era of Blended Learning. *Anatomical Sciences Education,* 2021, Accepted Author Manuscript, https://doi.org/10.1002/ase.2090

[17] Jerry M., Yunu M. Blended Learning in Rural Primary ESL Classroom: Do or Don't, *International Journal of Learning, Teaching and Educational Research,* Vol 20, No 2, 2021, pp. 152-173.

[18] Musawi A., Ammar M. The Effect of Different Blending Levels of Traditional and E-Learning Delivery on Academic Achievement and Students' Attitudes towards Blended Learning at Sultan Qaboos University, *The Turkish Online Journal of Educational Technology*,Vol.20, Issue 2, 2021, pp.127-139.

[19] Dolinsky M. About One Approach to the Study of the Topic "Synthesis of Combinational Circuits Using Carnot Maps", WSEAS Transactions on Advances in Engineering Education, vol. 20, pp. 60-69, 2023, https://doi.org/10.37394/232010.2023.20.9.

[20] Distance Learning Belarus by Gomel State University after Francisk Skaryna © 1999-2023, [Online]. https://dl.gsu.by (Accessed Date: November 23, 2023).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Author contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author have no conflicts of interest to declare.