Tool HLCCAD for Blended Learning the Fundamentals of Digital Electronics

M.S. DOLINSKY

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

*Abstract***.**The article discusses the practical experience of blended teaching students the fundamentals of digital electronics using the instrumental distance learning system DL.GSU.BY (DL) and tool HLCCAD specially designed for teaching the basics of digital electronics: design, modeling and analysis of functionally complex digital systems and its integration with DL, providing automatic verification of circuits developed by students. Both software developed at the Gomel F. Skoryna State University under the guidance of the author. DL provides students with access to theory and assignments for design and analysis, as well as sending solution files for both types of problems. HLCCAD provides students with visual design and debugging of digital device functional diagrams, as well as checking the correctness of the schemes sent by students for verification by their simulation on the set of tests specified by the author of the problem. HLCCAD also provides verification of assignments for the analysis of functional diagrams of digital devices by simulating tests sent by the student on the author's solution.

Keywords: blended learning; fundamentals of digital electronics; distance learning instrumental system, high level chip computer aided design, DL.GSU.BY
Received: April 7, 2022. Revised: October 8, 2022. Accepted: November 2, 2022. Published: December 2, 2022.

1 Introduction

The papers [1,2] convincingly substantiate the need for a transition to new forms of education in universities, including, and above all, blended learning, which integrates traditional approaches with online learning and, accordingly, makes the most of both the teacher's opportunities in the classroom and means of modern information technologies in the classroom and during independent work of students.

The papers [3-9] describe examples of the successful use of blended learning in university education in a variety of specialties. The paper [10] presents the author's experience of blended teaching of first-year students the basics of programming. The paper [11] presents the author's experience of blended teaching of undergraduate students the basics of digital electronics.

2 Problem Formulation

Fundamentals of digital electronics is a new and difficult subject for undergraduate students, which most of them encounter for the first time. Additional problems are created by the significantly different learning skills and motivation of different students.

The following is the first section of the theory to be studied.

Logic elements and logical operations

Negation, conjunction, disjunction. Addition modulo 2. Basic identities. The transition from logical functions to circuits. Design and simulation environment for digital devices HLCCAD. Basic concepts: project, device, circuit, contact, library of standard devices.

Minimization of Boolean Functions

The concept of truth tables. Construction of truth tables for functional devices without memory. Minimization of logical functions by the Carnot map method. Covering of single values. Logic coverage functions. Examples of problem solving.

Combination schemes

Decoder. Conditional graphic designation. Truth table. Logic functions. The issuance of a logical signal "1" on the output line, the binary code of which is on the input lines.

Encoder. Conditional graphic designation. Truth table. Logic functions.

Getting the number of the input line on which the logical signal "1" is located.

Priority encoder. Conditional graphic designation. Truth table. Logic functions. Obtaining the number of the upper input line, on which the logical signal "1" is located.

Multiplexer. Conditional graphic designation. Truth table. Logic functions. Transfer to the output of the values of the input line, the number of which is on the address input. Bus multiplexer.

Adder. Conditional graphic designation. Truth table. Logic functions. Binary addition of multi-bit data. Input and output transfers.

Storage devices

Trigger. Conditional graphic designation. Truth table. Logic functions. The concept of a beat. Static D flip-flop. Dynamic D-flip-flop on the rising edge. Dynamic D- flip-flop on trailing edge. Time charts. Download mode. Storage mode.

Register. Conditional graphic designation. Truth table. Logic functions. Static register. Dynamic register on rising edge. Dynamic register on the trailing edge. Time charts. Download mode. Storage mode. Reset mode. Synchronous and asynchronous register control.

Counter. Conditional graphic designation. Truth table. Logic functions. Time charts. Download mode. Storage mode. Reset mode. Direct account mode. Countdown mode. Synchronous and asynchronous counter control.

Random access memory. Conditional graphic designation. Truth table. Logic functions. Two-dimensional memory matrix. Recording mode. Reading mode. Storage mode. Dependence of RAM capacity on the capacity of input and address lines. The concept of the third state (Z-state).

Read only memory. Conditional graphic designation. Truth table. Logic functions. Two-dimensional memory matrix. Load mode. Reading mode. Storage mode. Dependence of RAM capacity on the capacity of input and address lines. Use of the Z-state in read-only storage devices.

3 Problem Solution

3.1 The ideology of studying the basics of digital electronics using HLCCAD

To solve the above problem, mixed learning is carried out using the DL.GSU.BY distance learning system, which contains the necessary theory and practical tasks for the design and analysis of digital electronics circuits. Students complete tasks and check solutions using the HLCCAD (High Level Chip Computer Aided Design) system. HLCCAD provides graphical input and editing of digital electronics circuits, modeling and debugging by visualizing simulation results, as well as generating a VHDL description if necessary to create a real device, for example using an FPGA.

Thus, the consolidation of the theory and verification of its study is carried out by performing automatically verified tasks for the design and analysis of the corresponding functional circuits of digital electronics.

3.2 Examples of circuit design assignments

The study of the theory of the fundamentals of digital electronics is reinforced by solving problems for the design and analysis of digital circuits of the following types

- by logic function
- according to the truth table
- combinational circuit
- conditional arithmetic expression
- circuit with memory.

Checking the correctness of the task is carried out as follows. The schema file is sent on the client side. HLCCAD system starts on the server side with this scheme and tests prepared by the author (input influences and correct answers for these influences). An automatic verification of the answers obtained as a result of modeling the student's decision on given input actions and the answers of the author is carried out. In case of coincidence of all answers for the entire simulation time, the solution is accepted, otherwise - no, and the student is informed for which input actions the student's scheme incorrectly calculated the values at the outputs.

Examples of the problems are given below:

Task P1. Logic function design

Build a digital device that satisfies the following Boolean function:

OUT_0 = ((IN_0+IN_1) V ~IN_2) V (IN_3+IN_4).

At the input of which are one-bit numbers IN_0, IN_1, IN_2, IN_3, IN_4, and at the output OUT_0 (one-bit).

Task P2. Truth Table Design

Design a circuit that works according to the following truth table

(input variables x1 x2 x3 x4,
output variables y1 y2 y3 y4)

x1	x2	x3	x4	y1	y2	y3	y4
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Task P3. Combination Circuit Design

The input is given two numbers X (8-bit) and N (3-bit). Delete the N-th bit of the number X and output the resulting (7-bit) number Y.

Task P4. Designing a conditional arithmetic expression schema

Develop a device that evaluates the value of an expression

$$\mathbf{RES} = \begin{cases} a/b+cd*a & \text{if } a*d<0 \\ b-(c+a/d)*b & \text{if } a*d>=0 \end{cases}$$

Res,a,b - 16-bit, c,d - 8-bit

Task P5. Memory circuit design

The input is given six one-bit numbers (A,B,C,D,E,F). If the sum of these numbers is an even number, then the device memory is reset and zero is output. If the sum is an odd number and it is more than three, then the value of the input D is written to the device memory. The same value is sent to the output. If the sum is an odd number and this number is less than or equal to three, the output is the number that is currently stored in memory.

3.3 Examples of tasks for the analysis of functional diagrams of a digital device

In all tasks for the analysis of the functional diagram of a digital device, the student is presented with its drawing, random values 0 or 1 applied to the input contacts (they are displayed in the figure). The student must understand the circuit and calculate the values at the outputs for given inputs. The values at the outputs change from 0 to 1 and back by clicking on the corresponding digit. Input actions are given 10 times and the student must correctly change the output values the same number of times. Checking the correctness of the task is carried out as follows. On the client side, a test file is formed from random inputs and student responses, and this file is sent to the server. The system starts on the server side HLCCAD with the author's solution to this problem (which was presented in the task figure) and the test file received from the client. An automatic verification of the answers obtained as a result of modeling the author's solution on given input actions and the student's answers is carried out. In case of coincidence of all answers for the entire simulation time, the solution is accepted otherwise - no, and the student is informed for which input actions the student incorrectly calculated the values at the outputs.

Task A1. Analysis of a circuit from logic functions

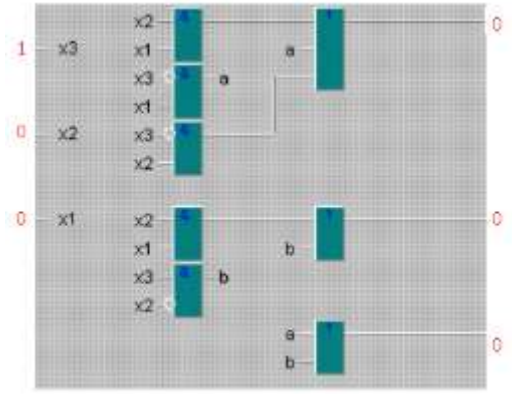


Fig.1. Analysis of a circuit from logic functions

Task A2. Combination circuit analysis

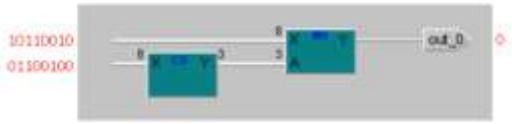


Fig. 2. Analysis of the combinational circuit

Task A3. Arithmetic circuit analysis

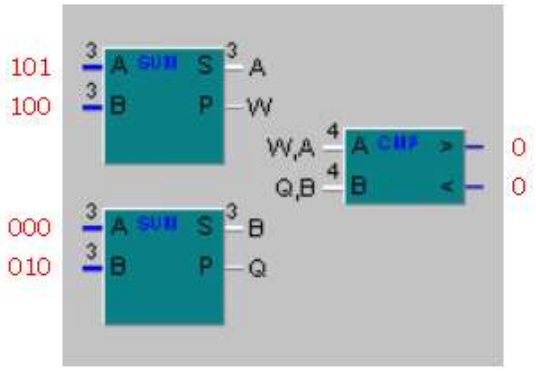


Fig. 3. Analysis of the arithmetic circuit

Task A4. Memory Circuit Analysis

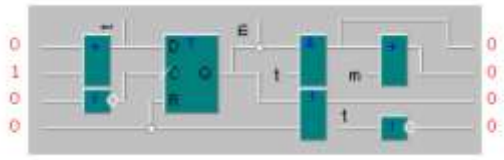


Fig. 4. Analysis of the circuit with memory

3.4 Technology for designing digital devices using HLCCAD

The HLCCAD system allows you to create devices both "bottom-up" and "top-down". To create a new project, select the menu item "File|New Project" in the main window. After specifying the file name, the new project will be added to the Project inspector window. In order to create a new device, you need to call the local menu above the project name and select the "New device" item, indicating its name in the dialog box.

After that, the developer can modify the body and circuit of the device. To open the editor, call the local menu above the device name and select the "Editor" item.

The device editor window consists of two sub-editors: the editor of the conventional graphic designation/interface (Fig. 5) and the schematic editor (Fig. 6).

The process of building a device usually starts with the interface. The device interface editor allows the developer to resize it, add, delete and modify pins or text on it.

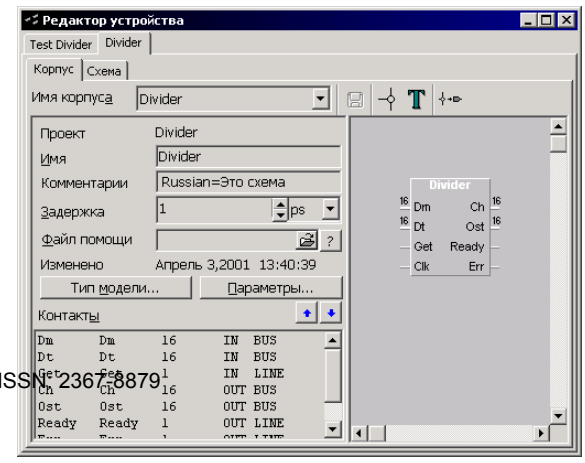


Fig.5 Editing the device interface

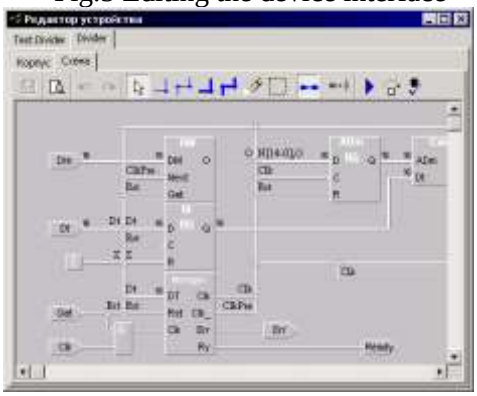


Fig. 6. Schematic editor window

The scheme is represented as a set of devices and links between them. Adding devices from other projects is done using the Drag&Drop technology. To do this, you need to select the device in the "Project inspector" window, press the left mouse button, move the mouse pointer over the diagram and release the button. In order to create a new device, you need to call the local menu and select the appropriate item.

Links are edited in 3 ways:

- drawing single lines
- polyline drawing
- using an invisible bus

When drawing lines, the developer determines the type of line (single or bus) by pressing the corresponding button on the panel. Individual lines are drawn according to the following algorithm:

- press the left mouse button on the diagram
- move the mouse pointer
- release the mouse button

Drawing broken lines goes like this:

- left click on the diagram
- move the mouse pointer.
- to draw a segment of a polyline, click the left mouse button and repeat from the previous step
- to cancel press the right mouse button

When drawing lines and tires, invisible (named) tires can be used. To enter a bus name, double-click the left mouse button above the line and enter the appropriate names in the dialog box.

Using an invisible bus allows you to reduce the number of lines in the diagram. To

enter contact lines into the bus, you must enter a name under which these lines will be available. The name entry window is called after double-clicking with the left mouse button over the contact of the body or circuit.

When developing from the bottom up, it is possible to combine ready-made devices by establishing links between them.

3.5 Built-in parameterized standard device library

Decomposition of the developed device or its block to obtain simulated components is carried out by creating a circuit using synthesized devices. These are the devices from the basic parameterized device library "Standard.prd". Table 1 lists the devices included in this library. The developer can change, add or remove contacts, change their parameters and capacity in accordance with the functionality of the device. The high-level model is automatically adjusted to the algorithm of work corresponding to this changes.

Table 1. Devices of the “Standard” library

Logics	NOT	Logic inverter
	OR	Logical OR
	XOR	Logical XOR
	AND	logical AND
Combination schemes	CD	Priority encoder
	DC	Decoder
	MS	Multiplexer
	DMS	Demultiplexer
Constants	0	Logic "0" generator
	one	Logic "1" generator
	bf	tristable buffer
Memory	T	Trigger
	RG	Register
	CT	Counter
	ROM	Read Only Memory
	RAM	RAM
Mathematical operations	CMP	comparator
	SUM	Adder
	MUL	Multiplier
	DIV	Divider

3.6. Testing and debugging the developed schemes

Test actions on the simulated device can be applied in several ways.

First, the developer can interactively change the values on the pins. To do this, you need to open the appropriate schema debugger window. Then double-click the "mouse" to call the window for changing the value. After entering a new value, just press the “Ok” button (Fig. 7).

Secondly, a batch testing mode is implemented. The developer can specify a set of test actions in the form of a specialized text file (Fig. 8). To do this, in the schematic editor window, you need to call the local menu and select the "Parameters" item. In the "Scheme Options" window, you must specify the appropriate file name. Before starting for simulation, you must set the "Use tests" flag in the simulation parameters.

Third, the developer can create an interactive debugging environment. The device will act as a generator of input actions. During simulation, the device can create an additional window in which, using the keyboard or mouse, it can change the values of the device outputs on the diagram. Similarly, the diagram may contain devices that perform the functions of testing and/or visualization.

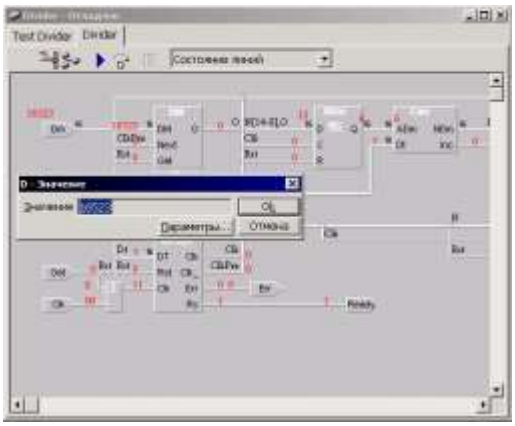


Fig.7. Example of interactive input

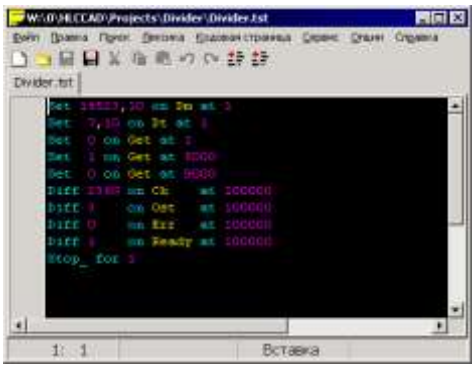


Fig.8. An example of a file of test actions

Additionally, you can specify the full test mode. As a result of the execution, after modeling the device scheme, a test file will be

automatically generated on the contacts on the trace of values, the contents of which will be used to generate a VHDL description for more complete testing.

3.7 Visualization of simulation results

Analysis of simulation results is carried out using a wide range of different visualization windows.

The structure of the simulated device is displayed in the "Model Tree" window (Fig. 9).

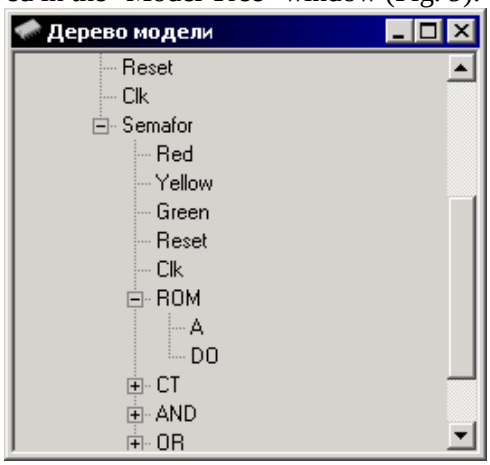
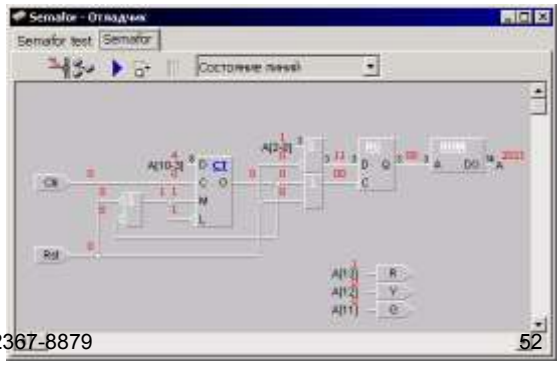


Fig.9. Model Tree

With this window, the developer can open other analysis windows for the elements of the tree. In addition, this window displays the results of simulation monitoring.

To visualize the values on the diagram, the developer can use the diagram debugger window (Fig. 10). The debugger displays the schematics of the simulated devices and the values set on the pins. Values on contacts can be displayed in an arbitrary number system from 2 to 256. For any contact, you can set an enumerated type that allows you to specify a list of replacement values for a text string. The developer has access to mechanisms for setting the parameters of values for all contacts of a certain body on the diagram, or for all contacts on the diagram. In addition, any value can be moved to an arbitrary location on the diagram. The debugger window also allows you to navigate through the device tree. The developer can open a schematic for any package in that schematic, or move up a level from the current one.



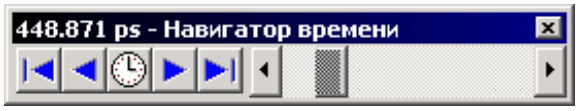


Fig.10. Schematic Debugger Window

Fig.11. Time navigator window

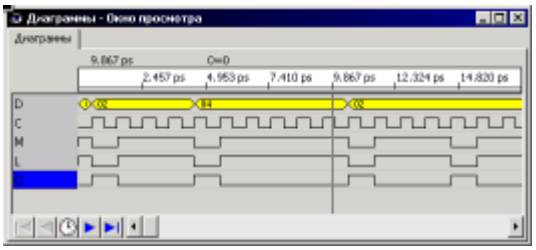
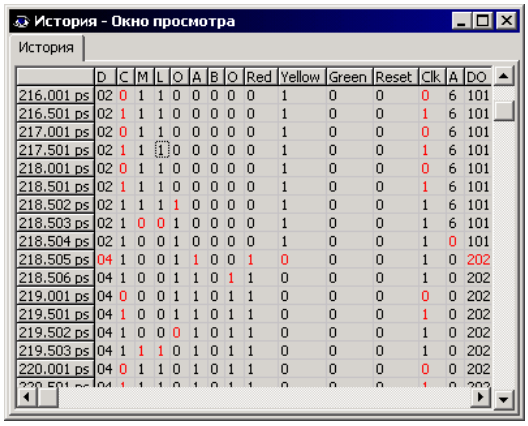


Fig.12. Timing window

Fig.13. Contact history window



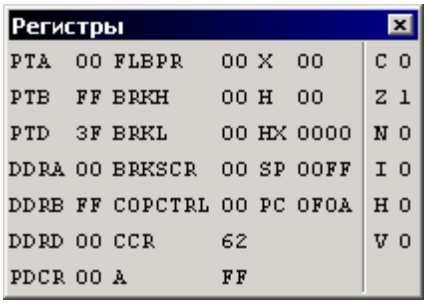


Fig.14. Registers window

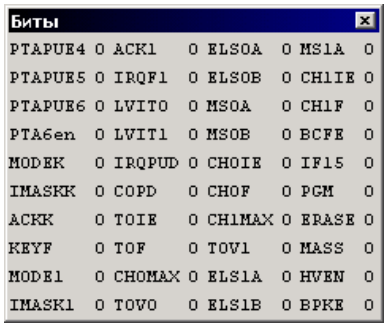


Fig.15. Bits View Window

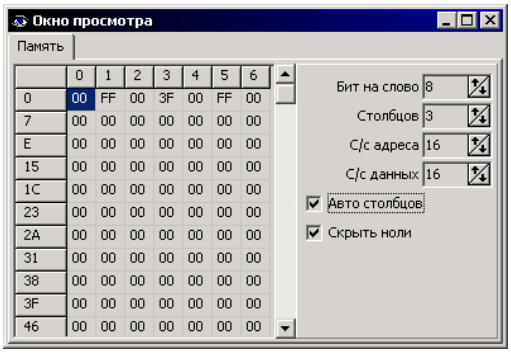


Fig.16. Memory dump viewer window (formattable)

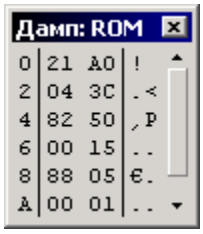


Fig.17 Memory dump viewer window (standard)

To change the analyzed model time, you can use the "Time Navigator" window (Fig. 11). The navigator allows you to change the time using the scroll bar. Numeric time input is allowed. With the help of additional buttons, you can change the time for a certain interval in any direction, as well as jump to the nearest model time, at which the value on the circuit contacts changed, opened in the circuit debugger window. In addition, there is a mode of automatic change of the analyzed time by any of the methods described above.

Two analysis windows are provided for viewing the trace of values for contacts: time diagrams and contact history. The timing diagram window (Fig. 12) is traditional for digital hardware development systems. On the left side, the contact names are displayed, and on the right side, the trace of values in the form of a diagram. You can zoom, filter, and search for a value. Navigation tools are similar to the "time navigator" window. The contact history window (Fig. 13) displays the trace of values in the form of a table. Each column of the table contains the values of the contact, at the model time specified by the row. Each row of the table is always different from the previous one, i.e. information is compressed.

To analyze the values of internal variables of models, the window for viewing registers and flags (Fig. 14), as well as the window for viewing bits (Fig. 15) is intended. The values correspond to the model time set in the "time navigator".

There are two ways to view the contents of a memory dump. The first window (Fig. 16) allows you to change the window settings arbitrarily: change the number system of the address and data, set an arbitrary size of the memory word. The second window (Fig. 17) has built-in settings.

Mechanisms for automatic arrangement of visualization windows in the form of horizontal and vertical "tiles" have been implemented.

Conclusion

This material is devoted to the system HLCCAD for debugging hardware of embedded digital systems. The development has been introduced into the educational process of the university. So blended teaching and learning are provided for students.

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

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

[4] 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

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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

[10] Dolinsky M. Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY", WSEAS Transactions on Advances in Engineering Education, vol. 19, 2022, pp. 52-57

[11] Dolinsky M. Experience of Blended Learning the Fundamentals of Digital Electronics for First/Second Year University students On Base of Distance Learning System DL.GSU.BY. International Journal of Education and Learning Systems, 2022, 7, 59-64