



# Article The Influence of the Developed Specific Multi-Paradigm Programming in Digital Logic Education

Lukas Hapl<sup>1</sup> and Hashim Habiballa<sup>2,\*</sup>

- <sup>1</sup> Department of Information and Communication Technologies in Education, University of Ostrava, Fr. Sramka 3, 70900 Ostrava, Czech Republic; lukas.hapl@roznovskastredni.cz
- <sup>2</sup> Department of Informatics and Computers, University of Ostrava, 30. Dubna 22, 70103 Ostrava, Czech Republic
- Correspondence: hashim.habiballa@osu.cz

Abstract: This article introduces the possible usage of the developed programming discourse that can be used to support training in the digital logic area. The discourse merges several programming paradigms into one solution. The intended learners are secondary school students focused on digital system programming. The main intent is to find out whether digital logic curriculum based on Digital Circuits Based Logical Programming (DCBLP) inheritance has positive impact on the students and the way they explore the digital logic itself. Students' cognitive and affective areas are in the scope of this preliminary research and questionnaires and cognitive tests will help to support the research. Experimental and control groups were used to gather relevant records. To analyse and support the interpretation of the data gathered by questionnaires, the chi-square test (two-tailed) has been used. ANOVA has been used to evaluate data for the achievement test results. The preliminary research revealed there is a possibility of using developed programming discourse DCBLP in digital logic training. Students claim overall usefulness of the discourse in the training; the strong motivation power of the programming discourse itself has not been discovered. From the test we conclude that the performance of the students trained using new programming discourse is significantly better. It is possible to use more different programming paradigms, such as imperative and declarative, in one solution to support training in the area of digital logic. Such solutions can enhance the way the students deal with the programming languages and also supports interdisciplinary relationships.

Keywords: digital logic; multi-paradigm programming; DCBLP; CS curriculum

# 1. Introduction

The schools in the Czech Republic follow the ISCED hierarchy. Secondary technical schools (sometimes called high professional schools) are typical examples of ISCED 3 (upper secondary education, ISCED 97 3A, ISCED 2011-P 354) schools in this system [1]. These schools prepare the students for certain professional/technical areas. These students are mostly aged between 15 and 18 years. From the ontogenetic perspective, it is a stage of late adolescence of a person framed by the ages 15 and 20 years. It is also a period that is suitable for professional training. It is the right time to develop abilities, knowledge and to gain important competencies [2]. The topic of the education system in Czechia itself is very complex; therefore, it is not possible to describe all of its aspects here, but it is necessary to mention that the elementary as well as secondary schools operate their curricula based on the RVP (Framework Educational Programs [3]). When selected and approved by officials, the schools have to incorporate the programs into their own school educational program(s). One of many of these programs available for secondary technical studies is a framework program named Information Technology (IT). The schools have to follow up the framework programs used in the school educational program, but they also have the possibility to extend the education in their program accordingly. If we mention the Information Technology framework program, schools often include the digital logic



Citation: Hapl, L.; Habiballa, H. The Influence of the Developed Specific Multi-Paradigm Programming in Digital Logic Education. *Mathematics* 2021, *9*, 1255. https://doi.org/ 10.3390/math9111255

Academic Editor: David Pugalee

Received: 27 April 2021 Accepted: 25 May 2021 Published: 31 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). area [4,5] in their school curriculum documents [6,7] although the topic is not directly included in the frame educational program for IT. The important role of digital logic in the computer science curriculum is mentioned by Connely [8] (Connely et al., 2004). Thus, we can consider computer science as a technical discipline with a deep mathematical basis.

It is necessary to emphasize that those schools training students based on certain electronics education framework programs (not all, it depends on the type of electronics framework educational program) have the digital logic training included automatically (e.g., Telecommunication– includes digital logic). It is obvious that digital logic is more related to electronics than to informatics. The reason is that, for more than four decades, the digital logic area has benefitted from digital integrated circuits and vice versa. From the educational perspective, this is a pleasant situation as long as the students understand the electronics or have electronics training. There are already documented experiments supporting and enhancing the education of digital logic with a tight relationship to electronics [9]. These use simulation software along with (or with the support of) HDL based languages. A deeper analysis of the significance of project-oriented learning was carried out by [10]. Unfortunately, these are not fully usable by the informatics students. They do not have the appropriate preliminary training in electronics. Still, the teachers expect the same good knowledge results in the area of digital logic and understanding of digital logic when speaking about informatics students.

# 2. Digital Logic Area Modification for IT Study Programs at Secondary Schools

Normally, the content of digital logic education in an IT curriculum follows the same scheme and requirements as if it was in the electronics based curriculum. The educational areas are as follows (e.g., [6,7]):

- Boolean logic: functions, algebra, truth tables, expressions, logic networks.
- Disjunctive and conjunctive normal forms, graphics ways of minimization (normal forms).
- Digital circuits and logic levels, standards.
- Combinational (time-independent) logic and examples.
- Sequential logic and examples.
- Application of combinational and sequential logic.

It is not so hard to educate IT oriented students in the area of Boolean logic, Boolean algebra and normal forms, but it is much harder to educate them in the rest of the proposed areas. Those are supported by the electronics and digital integrated circuits principles which are not in the scope of an IT education program as mentioned previously. There is the possibility of simply excluding these areas from the curriculum but it is more meaningful to support these digital logic areas differently than for electronics. It is also already known that the declarative logic programming utilizing PROLOG is an appropriate way of describing the functions of digital logic circuits [11]. Clocksin began the research for two main reasons: to develop logic-programming idioms, evaluating the relative utility of various techniques, and to demonstrate how logic programming can contribute effectively to design automation for digital circuits. There is also a Computer Science (CS) curriculum in Israel enhanced by the logic paradigm programming [12]). The goal of Gal-Ezer was to enhance formal logic education at high schools. Thanks to this enhancement, the students in Israel are trained in imperative and declarative programming paradigms (separately). Noticeable are also the intentions to focus on the constructivist learning method in the formal logic area [13–15]. An overall summary of the milestones related to Mathematical Logic in Computer Science is presented by Kfoury [16,17]. There are also other nonprocedural paradigm based experiments in teaching theoretical computer science [18]. Based on these facts and previous findings regarding digital logic and programming, it is possible to assume that an appropriate programming solution can possibly enhance digital logic training. An appropriate programming solution should help IT students to better understand the areas of digital logic that are normally related to electronics. The solution could also bring the possibility of substituting electronics parts by an equal software oriented portion supporting demonstration and practicing. However, the goal is not just

to include another programming paradigm into curriculum, but to use the best of the imperative paradigm (which is already mandatory and included in the IT curriculum) and declarative paradigms (it is not included as mandatory in the IT curriculum) in one solution. Surely the automata-based or already mentioned HDL programming—such as Python based FHDL [19]—can be used to support the intention, but we would like to know whether using the logic programming (as Clocksin did using PROLOG) together with imperative programming would also be beneficial. The reason is that, nowadays, programming languages offer more possibilities than 30 years ago and there certainly exists the possibility to merge some of the standalone paradigms to one programming solution. An example is the recently developed programming discourse utilizing multi-paradigm principles to support digital logic education using any current programming language that can follow the rules of this new discourse. The solution is purely based on Boolean logic without the necessity of using supportive constructions and statements that are not necessary to describe the logic. It directly transfers the principles of Boolean logic and Boolean algebra into the programming one to one as they can also be seen this way in real digital circuits made of semiconductors (the logic way, not automata-based), no matter whether we are speaking about the combinational or the sequential ones. It also emphasises the Flow-Based Programming principles [20], which belongs to the declarative paradigm [21]. The mentioned selected programming discourse is named Digital Circuits Based Logical Programming (DCBLP). Its principles are applicable for several programming languages that are nowadays widely used, such as Java, C++ and Python. The Python programming language has been chosen as the primary language in this experiment due to its rising popularity [22]. It can be easily used on desktop computers running many of the common operating systems as well as on Internet of Things (IoT) devices driven by Linux based operating systems. DCBLP discourse applied in Python programming language is called DCBLPy by the authors. The prerequisite necessary for the students to use DCBLP is to understand the basic principles of the programming language chosen as the platform to apply DCBLP (such as Python). It can be performed and implemented as suggested by the roadmap in Figure 1. An experiment described in this article focused on stage 2, presented in Figure 1.



Figure 1. Roadmap used for digital logic and IoT training.

# Digital Circuits Based Logical Programming in Details

Digital Circuits Based Logical Programming is a merge of imperative programming (structured, object oriented) and declarative programming (flow based, logic) techniques in one bundle suitable for describing digital logic [23]. This is the reason why this discourse is seen as a domain specific multi-paradigm one. However, DCBLP does not represent a bundle of programming software nor an API. It represents the philosophy and principles of digital logic circuits' description. It is how the digital logic systems can be specifically described using the currently widely used imperative programming languages. Only data-flow (inputs/outputs) and logic operators are used along with procedures or objects (and their instances). Short characteristics of DCBLP discourse can be borrowed and quoted from an article describing this discourse [24]. "The main idea of this programming paradigm builds on direct translation of hardware, physical principles and internal logic of digital circuits into software representation [25]. In the real world we are able to design logic based digital circuits, interconnection and data flow by means of electric signals Functions and properties of digital circuits can be modeled as classes—and their instances are particular digital circuits. This enables us to use principles of object oriented programming. Their interconnection is performed through inputs and outputs (in instance methods), conductors are implemented by variables, pipes or queues. Electric signals are implemented by numerical values of the inputs and outputs. In this model we use FBP principles—FBP uses black box objects with input and output data. We can use special instances of classes containing all other instances and their methods constituting the logical network. The main method of this special instance is executed in its own thread with a predefined period for repeating content, in order to simulate the real world. This leads us also into multi-threading programming, which is mentioned not only in FBP but also in articles focused on structured programming". The main characteristic and principles of DCBLP are as follows:

- It is possible to use the code to describe the function of simple digital circuits.
- Based on a multi-paradigm approach.
- It can possibly be used by technicians understanding digital logic.
- Using programming languages to directly learn the Boolean functions.
- Possibility of creating the code without using "if statements".
- Alternative way of programming without establishing a flow chart.
- Possibility to use technical, logical, digital circuits' principles in programming. Main properties of code for DCBLP include the following rules:
- Code describes the system, which can be represented on a digital level.
- Resulting logic consists of basic logical functions (not "if statement" black box).
- Truth and state tables and logic networks are essential elements with digital circuits' functions.
- Code can be based both on structured and object oriented programming.
- Code is not based on simulation particular states but it is based on internal functions logic and it includes FBP elements, that is, data flow between I/O.
- It can be directly bound to digital inputs and outputs.

The learning materials [26] provide real examples for combinational and sequential logic education (utilizing DCBLP in Python language). These learning materials were used during the experiment and can be used to directly enrich digital logic education by DCBLP.

An example of DCBLPy usage is presented in Figure 2 to support the presented principles and the background theory. It illustrates a simple digital logic comparator (identity, not magnitude) which can also be found as the digital circuit in TTL logic ( $74 \times 521$  or similar). It belongs to the combinational logic area. The logic of such a comparator can be transformed to Python code (Figure 2-bottom part of the picture) that follow the rules of DCBLP. Only Boolean logic operators are accepted and loop block is allowed. There is no need for an "if statement" at all. Note that the code is simplified and



does not contain inputs validation and handling of exceptions. The magnitude compactor can also be created.

Figure 2. N-bit digital identity comparator as logic network and DCBLP code in Python.

All of the combinational digital logic can be transformed the same way. An objectoriented programming (OOP) approach can be used as well; however, it is not mandatory for such an easy task. For sequential logic it can be more complicated to use structured programming, therefore an object-oriented approach is essential along with using the timers or multi-threading features.

### 3. Experimentation

The intention is to find out if the change of the digital logic curriculum that is based on DCBLP inheritance has a positive impact on the students and the way they explore the world of digital logic itself. Surely, we need relevant data to judge. Students' cognitive and affective areas are in the scope of this data gathering and research [27]. It is based on the constructivist approach [28] of teaching. The main questions covering these areas of interest are as follows:

- Is the discourse itself sensed as a good element for positive influence of the digital logic education?
- If so, what is the most valuable element recognized by the students in such modified education?
- Is it possible that the change in education content can lead to bigger interest among the students?
- Is it possible to use the discourse as a motivation support in the area of an IoT as well?
- What is the real impact of such change in the cognitive area?

Quantitative research methods will be used to answer these questions. We will use a questionnaire to discover what the students think about the discourse inherited in their education to cover affective aspects. Next, the quasi-standardized achievement/performance test to cover cognitive/knowledge aspects will be used [29]. To cover questions No. 1–4, the questionnaire will be used and question No. 5 is covered by the knowledge/performance test. The questionnaire consists of 11 questions divided into several categories which

will be discussed later [30]. The performance test covering the cognitive/knowledge area includes 16 questions that are also grouped into several categories based on the digital logic curriculum (with an appropriate width and depth). The details of this test will also be discussed later. Based on the main questions we can suggest the hypotheses (factual, non-statistical) that should be verified:

Hypothesis 1a (H1a). A new discourse is sensed as appropriately enhancing the digital logic education.

**Hypothesis 1b (H1b).** *A new programming discourse is sensed as helpful for demonstration of the digital systems.* 

**Hypothesis 1c (H1c).** *A new discourse is sensed as suitable for practicing in the area of digital logic education.* 

**Hypothesis 1d (H1d).** A new discourse brings the motivation possibilities for the digital logic education.

Hypothesis 1e (H1e). A new discourse brings the motivation possibilities for an IoT usage.

**Hypothesis 2 (H2).** *A new discourse has the measurable impact in the cognitive area and the students' knowledge in the field of digital logic.* 

H1a–H1e cover the main questions 1–4 for an affective area presented earlier. H2 covers the cognitive part of the research. Therefore null hypothesis [29] for H2 can be formulated as follows: H20: There is no difference between students' overall performance in the area of digital logic with and without the support of a new discourse. Additional hypotheses that should also be verified can be formulated as follows:

**Hypothesis 2a0 (H2a0).** It has no measurable impact in the block of Boolean logic functions.

**Hypothesis 2b0 (H2b0).** It has no measurable impact in the block of combinational logic.

**Hypothesis 2c0 (H2c0).** It has no measurable impact in in the block of sequential logic.

**Hypothesis 2d0 (H2d0).** *It has no measurable impact in the programming language usage across the logic blocks.* 

# 3.1. Affective Area Research Details

There has been a decision made to determine whether DCBLPy (a.k.a. DCBLP discourse application)—as an important control variable—has a positive/negative impact on the students during the training, before the cognitive performance of the students is measured. The questionnaire was prepared to fulfil this goal. There are 11 questions in the questionnaire used in this research (of a total of 20) divided into several categories corresponding to the proposed hypothesis: "Appropriate enhancement" means how the students sense the way DCBLPy is used during the education and whether it helps to enhance the education content or whether it brings additional unwanted/inappropriate load. "Demonstration"—should uncover whether transforming logic areas and algebra formulas into programming code of DCBLPy and demonstrating DCBLPy instead can be seen from the perspective of the students as helpful. "Practicing" means utilization and use of DCBLPy by the students themselves and their expectation regarding the necessity for practicing at school during the training, which is currently not possible because of the allowed educational form. "Motivation for Digital Logic" expresses whether DCBLPy motivates the students to even think by themselves about the Boolean logic functions and Boolean logic based tasks—and if so, in which part of the curriculum. "IoT programming" represents the students' opinion about future IoT programming using DCBLPy. Data

gathered from the "Enhancement", "Demonstration" and partially from other areas can be additionally observed from the combinational and sequential logic perspective. Because the research in this part covers the students' affect area, it can be later enhanced/reduced to more/less categories. Additional hypotheses can also be formulated during future research. The main categories covered by the questionnaire in relation to hypotheses H1x for the affective area (described in the text above) are visualized in Figure 3.



Figure 3. Relation between areas of interest and hypotheses.

The questions in the questionnaire are divided into the above specified categories as described in the Table 1.

Table 1.	Questionnaire	categories.
----------	---------------	-------------

Areas of Interest	Questions Included in the Area
H1a: Appropriate Enhancement	<ul><li>(A) The scope of DCBLP integration into digital logic is,</li><li>(B) Description of combinational logic by DCBLP was,</li><li>(C) Description of sequential logic by DCBLP was</li></ul>
H1b: Demonstration Usefulness	(A) Demonstration of digital logic with DCBLP helped, (B) What was the most helpful during the training
H1c: Self Practicing	<ul><li>(A) Do you like to include practicing DCBLP in ICT labs,</li><li>(B) During the training I missed the most</li></ul>
H1d: Motivation for Digital Logic (learning)	<ul><li>(A) Including DCBLP tasks in digital logic was for me,</li><li>(B) Programming included in digital logic</li></ul>
H1e: IoT Programming	(A) Motivation for the future programming on IoT
Not specified	(A) Using the programming language to solve the tasks is

# 3.2. Cognitive Area Research Details

First of all, the education environment was observed and then the variables with the influence capability for the digital logic training were chosen. The control and dependent variables were determined accordingly. The control variables have the power to adjust the training process by selecting an appropriate target audience and the training attributes. These are: age of the students (15–18), the study program (based on IT framework), previous passed training in selected programming language (e.g., Python), the teaching method (based on transmission and demonstration, practicing at school during the lessons

was limited), educator (more than one teacher) and training content (DCBLPy included). Dependent variables' values bring us the results of the students' overall performance in the cognitive test and the particular results for Boolean logic and combinational and sequential logic. Variable names, their types and values/descriptions are summarized in Table 2.

Variable Name	Variable Type	Value/Description
Overall Performance in digital logic test	Dependent	Total score of the test results including Boolean logic, combinational and sequential tasks
Performance in Boolean logic area	Dependent	Sum score for Boolean logic tasks
Performance in Combinational area	Dependent	Sum score for combinational tasks
Performance in Sequential test	Dependent	Sum score for sequential tasks
Age of the students	Control	15–18 (upper secondary)
Educational Framework	Control	Information technology
Previous programming language training	Control	Python
Teaching methods	Control	Combined instructive and demonstrative
Training content	Control	With/without DCBLP usage
Educator	Control	More than one
Digital logic knowledge	Concluded	Conclusions made (deduced) from the entire performance tests

**Table 2.** Identified Variables for Cognitive Area.

Input (control) variables impact the function of the training process and an output of the function is collected using the values of standalone digital logic categories. The corresponding relations are presented in Figure 4.



Figure 4. Variables influencing the training and relation to hypotheses.

An achievement test explores the cognitive aspects of DCBLP(y) inheritance into the training along with the other control variables set. The test contains 16 tasks with 1 point per task (16 points in total). The test was divided into several categories that should provide data for hypothesis H2 verification (transformed to statistical hypothesis H0 and alternatives). Categories in the test directly copy the need to answer the null and alternative hypotheses, respectively. Boolean logic and functions (7 questions), combinational (2 questions) and sequential logic (2 questions) knowledge could be tested. Additionally, programming skills covering all previous categories based on DCBLP principles were measured (5 questions: 1 covering Boolean logic, 3 combinational, 1 sequential). All of the areas contain open-ended questions; there is no close-ended question. At least one question of each category was focused on the lower levels of the Bloom's taxonomy of cognitive aims (remember or understanding) and at least one question was focused on application. This test was also used during the proof of concept (pilot) stage and adjusted accordingly afterwards. Discrepant criteria were calculated to eliminate the easiest questions. Based on the initial results (in pilot), it was possible to express that the Upper-Lower Index was between 0–0.5 (neutral to slightly difficult) and the Kudera-Richardson's reliability coefficient [29,31] was 0.78 (acceptable reliability). The validity of the test was assured and acquired by the peer review.

### 3.3. Participants: Control and Experimental Groups

The experiment took place during the school year 2018/2019. The participating school was the Secondary professional school of informatics, electronics and crafts located in Moravia, Czechia. One of their school framework programs-Modern Information Technologies—is based on the educational framework program, Information Technology. The classes using this program were selected for this experiment. All of the students had to pass the standardized entrance examination (mandatory for all of the secondary schools of this type in this country) to be able to study this school program. The participants went through a course on the Python programming language during the first year of their studies. All of the prerequisites defined by the allowed values of the control variables were met. Two classes participated in the research. One class served as the experimental group meanwhile the other one served as the control group. The experimental group contained 28 students (n = 28, average age = 16.32, standard deviation = 0.61) and the control group contained 25 students (n = 25, average age = 16.28, standard deviation = 0.54). From the perspective of measuring the change impact, this sample selection corresponds to the Lindquist's plan of type 1 with low reliability [32]; however, with the usage of the control variables (age, prior programming course, the same school educational framework), selection of the students using the entrance examination and using an appropriate statistical methods (later in the text) it can be categorized as a plan of type 5 with higher reliability [29].

# 3.4. Usage of the Questionnaire and Achievement Test

The questionnaires investigating the students' opinions were filled out after 9 months of the training (almost at the end of the course–1 month left) by the participants of the experimental group only, as they contained the questions related to the new educational content, which was unknown to the students of the control group. Of the 28 students contacted, 24 (86%) attended the survey. The knowledge in the digital logic area was measured by the achievement test after 10 months from the beginning of the training (one month left to end of the course) in both groups. The test was used as the last countable exam of the course in both groups. Even if the control group did not participate in the education of the modified training content, they had the same previous programming language training as the students in the experimental group. The part of the test directly dealing with the DCBLPy code (only one question) was not measured in the control group. The programming questions/tasks dealing with the opened answers were measured in both groups as the control group could use standard programming language Python syntax to express an acceptable solution. Time taken to fill the test was 45 min (one education unit). All of the students (100%) in both groups participated. No student missed the test.

# 4. Education Process and Training Content Description

The digital logic area training is realized during the second year of study (total length of the study program is 4 years). Standard training in the digital logic area is spread over 10 months, with a total of 64 lessons, 45 min per lesson. Both groups in the experiment were educated with respect to the "Teaching methods" control variable allowed value, that is: combined transmissive, demonstrative (illustrative) methods. Although the content

of instruction in both groups is the same, inheritance of DCBLPy slightly changes the possibility of the methods' usage. Without DCBLPy the content is more instructive and the practicing is focused on manual calculations, meanwhile the digital logic content enriched by DCBLPy allows more logic solutions' demonstration using the programming code and practicing is also more focused on programming. Still, both groups did not have access to computer labs during the lessons because of the theoretical nature of the training itself. Most of the time transmissive education with demonstrative parts prevailed. If the computers were used by students then solely and voluntarily out of the lessons realized at school (what cannot be denied) to assure comparable conditions. The comparison of the content in experimental and control groups is presented in Table 3.

Theme Topic	Experimental Group	Control Group
Boolean logic/functions	Logic functions, symbols Expressions; Truth tables; Logic networks; <i>Python code for</i> <i>logic functions</i>	Logic functions, symbols; Expressions; Truth tables; Logic networks
Boolean algebra rules	Laws of Boolean algebra; de Morgan rules; Normal forms; Karnaugh maps	Laws of Boolean algebra; de Morgan rules; Normal forms; Karnaugh maps
Digital logic circuits, standards and logic levels	Gates; Logic levels (TTL/CMOS); Representative circuits; Usage of buttons and LED	Gates; Logic levels (TTL/CMOS); Representative circuits; Usage of buttons and LED
Combinational logic	Representative logic solutions; Truth tables, functions, networks; <i>Structured programming</i> <i>refresh</i> ; DCBLP rules; DCBLP for combinational logic	Representative logic solutions; Truth tables, functions, networks
Sequential logic	Representative logic solutions; Truth tables, functions, networks; OOP principles refresh; DCBLP for sequential logic	Representative logic solutions, Truth tables, functions, networks
Applications	Counters; Registers; DCBLP supporting projects	Counters; Registers

Table 3. Training content in experimental and control groups.

### Training Content Enhancements in Detail

Each of the areas enhanced by DCBLP and mentioned in Table 3 can be observed in more details to understand the impact on the curriculum. This part explains how the training content in the area of logic functions, combinational logic and sequential logic can be enhanced by programming based on DCBLP. Flow-charts for each of mentioned areas impacted by DCBLP are available in Figure 5. Please note the areas that are not impacted (as presented also in the Table 3, e.g., rules, logic levels) are not discussed in detail as the training content is not modified compared to standard training in the control group. The goal is to use something the students already know—programming language, to describe something they learn and try to understand—the logic. It is expected to directly practice the logic, fixing the knowledge and observe the final result. Moreover, the students learn how to enhance the programming code, use inheritance and build more complex solutions based on previously assembled code. They will be thinking about the solution within the Boolean logic and permitted operators' boundaries.



**Figure 5.** Flow-charts for the training areas enhanced by DCBLP. \* means sequence how the particular areas continue.

A complete textbook [24] was not yet available to support the students' training during the time of this research. However, standalone tasks and methodology were available and were used to support the training. These tasks (as well as the textbook) utilize Python as the programming language chosen to implement DCBLP (previously described as DCBLPy). The textbook is currently available in Czech localization and will be used in the next stage of the research. However, the teacher can create their own DCBLPy materials following the DCBLP principles and standard logic solutions. Standalone tasks as well as the textbook follow the same scheme (Figure 5) and following explanation (Figures 6–8).

1. Training area of Boolean logic functions

The goal: teach the students the logic functions and prepare them to use the programming language.

- a. Describe the Boolean logic, logic functions, schematics, algebraic formulas.
- b. Show the students programming language expressions to describe logic functions.
- c. Show the students how the algebraic expression will be transformed to the chosen language.
- d. Demonstrate each logic function in chosen programming language using different values on inputs and observe the output results.

2. Training area of combinational logic functions. The goal: show and teach the students the standardized combinational logic functions (binary adder, parity generator, comparators, decoders, etc.).

- a. Refresh the principles of the structured programming in selected programming language.
- b. Present and highlight the principles of DCBLP (e.g., "no if statements").
- c. Teach the students combinational logic standardized solutions (one by one).
- d. Combinational logic of the current interest will be transformed to the selected language using DCBLP principles (for example Python) and the function will be demonstrated to the students by the teacher. Structured programming will be used to support this part. Optionally the students can rewrite the logic to the selected language by themselves utilizing the idea of inquiry based tasks.
- e. The results will be verified, summarized and discussed with the students.

#### Truth table Expression Schematics $\mathbf{Y} = \mathbf{A} \wedge \mathbf{B}$ $\mathbf{Y} = \mathbf{A} \cdot \mathbf{B}$ в A Y & А 0 0 0 Υ в Python code 0 1 0 1 0 $\mathbf{Y} = \mathbf{A}$ and $\mathbf{B}$ 0 1 1 1

AND function

Figure 6. Description of one of the logic functions (fAND) with Python code included.

Return to point b. of this part and continue the same way of describing another standardized combinational logic solution when necessary. An example of one of the combinational logic solutions is shown in Figure 7.



**Figure 7.** Illustration of the steps to be taken during the combinational logic training in experimental group.

There can be numerous tasks created to cover combinational logic area (depends on the time available for the combinational part in the curriculum and an amount of the combination logic functions the students should discover and learn).

3. Training area of sequential logic The goal: teach the students standardized bi-stable sequential solutions and use the expressions and programming language in the sequential logic area.

- a. Refresh the OOP principles of the selected programming language, as the OOP will be used to support this part.
- b. Use the OOP template presented in learning materials [24] or create your own following DCBLP principles.
- c. Teach the students selected bi-stable sequential circuits; start with Reset-Set circuit.

d.

Let the students create their own classes, methods or even modules that can be used or imported later. Return to the first point c. of this part and continue the same way of describing Delay, Toggle and Counter sequential logic solutions one by one. Each logic will be built on the one previously presented. This means that the Counter is based on Toggle, which inherits from Delay, which inherits from Reset-Set. An example of the task that can be used to cover part of the combinational logic is presented in Figure 8. Similar steps will be taken for each sequential logic solutions (RS, D, JK, T, counters).

logic to the selected language by themselves utilizing the idea of inquiry based tasks.



**Figure 8.** Illustration of the steps to be taken during the sequential logic training in experimental group.

Moreover, the students can create complex solutions based on combinational and sequential circuits (as expressed in Table 3). They will use previously prepared code (functions, classes, methods) describing combinational and sequential circuits they already passed through. Complex tasks can cover solutions using counters and decoders to create PWM controlling LED or projects such as digital clock.

# 5. Ethical Aspects

The permission to work with the data of all of the participants in both of the groups was acquired before the experiment began. The procedure followed the local rules for the General Data Protection Regulation (GDPR). The GDPR officer and the headmaster of the institution were informed. Informed consent was acquired from the participants. The goals, changes in the training content and basic information gathered about the students (age, class), place of the store and time availability were included in the consent form. All participants had the right to refuse to store their data related to the research in any time. All of the personal or personalized records are considered private. Only anonymous data are available under the experimental IDs for each student participating in the research. The relationship between these identifications and the exact students were carried out by the standalone conversion table, which is stored only electronically on an allowed internal AES ciphered storage of the institution where the research was made. Data analysis was stored the same way. Printed and filled questionnaires and the achievement are available from the researchers. No personal information other than the age, class, score gained and opinions about the modified content was gathered. Based on the internal institution regulation, the

stored data will be automatically discarded and destroyed after 5 years from the end of the research. There will be no copy available after this period.

# 6. Statistical Methods Usage

To support correct interpretation of the data gathered by questionnaires (nominal measurements) the goodness of fit chi-square test (Pearson's) was used to prove that the differences between the answers within the individual questions are significant. The expected observation counts are values of the same size (the probability is distributed equally). Thus we expect the observed frequencies statistically equal to expected frequencies to accept the null hypotheses for each test. The results of the survey are presented by the graphs (pie-charts) in the section "Questionnaire Data Presentation". To provide the proof that the samples of the achievement test results in both groups were selected from the same complete sample, the Kolmogorov-Smirnov's test was used. Even if it is more powerful for ordinal samples, we can carefully use it for our discrete values as well with the reliability limitation that it brings (n1 = 25, n2 = 28, for the nearest tabulated critical value  $C_{(20,21)} = 173$ , M = 700, D = 0.143, C = 100.1, where the values  $C_{(n1tab,n2tab)}$  are tabulated values for n1 and n2 groups). From the results can be judged that both groups were acquired from the same complete sample.

The next step utilized Fisher–Snedecor's F-test to discover whether there is no statistically significant difference between the data variance in both groups ( $F_{(24,27)} = 1.96$  or the nearest tabulated  $F_{(20,20)} = 2.12$ , counted F = 1.14). It was calculated that F is lower than F critical. It can be expressed that there is no significant difference of data variance inside of both groups; therefore analysis of variance can be performed as the next step. Fisher's analysis of variance (ANOVA) was used to evaluate data for the achievement test results. This test will help to judge whether there is a significant difference between the results in the achievement tests of experimental and control group, especially:

- between the results of the whole test (all tasks) between the groups;
- between the results of the selected tasks per topic (e.g., combinational, sequential logic).

This method was preferred over the Mann-Whitney's U-test because of the possibility of evaluating groups with a significantly different number of the participants and at the same time with more than 20 participants [33]. Based on this test the null hypothesis and alternative hypotheses respectively were verified. All of the mentioned tests were used under the circumstance of 5% probability error (significance level  $\alpha = 0.05$ ). Visualization of data is supported by column graphs showing frequency of correct answers per tasks and quartile box-plots to show differences between the groups. Results and visualization are available in the section "Achievement Test Results Data Presentation". Statistical tools "Statistica.cz", "SPSS by IBM" and "LibreOffice Calc" have been used to support the calculations.

### 6.1. Questionnaire Data Presentation

Every hypothesis for the affective area is verified based on data acquired from the questioner using Pearson's goodness of fit chi-squared test and the following formula:

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i},\tag{1}$$

where *O* is observed count, *E* is expected count (asserted by null hypothesis). The significance level is 0.05 as already stated. Counted  $\chi^2$  will be compared to the critical value using an appropriate degrees of freedom (depends on the response categories). If the counted  $\chi^2$  is greater than the critical value we will reject the null hypothesis and accept the alternative hypothesis, otherwise we will not refuse the null hypotheses. The critical values will be presented in the form of:  $\chi^2_{significancelevel}$  (*degreesoffreedom*). For example:  $\chi^2_{0.05}$  for significance level 0.05 and 1 level of freedom for two answer categories in the question.

The graphs and statistical results are available to support the decision regarding each of the hypotheses. We start with the factual hypothesis H1a area (zero hypothesis H0 and alternative HA) and continue with H1b (zero, alternative), and so forth.

For the Area of Interest H1a-Appropriate Enhancement

We need to judge if the discourse is appropriately enhancing the digital logic education. There are three questions in the questionnaire related to this area of interest (Table 1) that must be evaluated one by one before we judge.

(A) The scope of DCBLP integration into digital logic.

The statistical null hypothesis and alternative hypothesis are formulated:

# **Hypothesis 0 (H0).** There is no significant difference in responses sensing DCBLP.

**Hypothesis A (HA).** *There is a significant difference in responses sensing DCBLP.* 

The questionnaire responses for answering the hypothesis are gathered in the Table 4.

Answers	0	Ε	O - E	$(O - E)^2$	(O – E) <sup>2</sup> /E
adequate	21	12	9	81	6.75
too much	3	12	-9	81	6.75
					Σ13.50

 Table 4. Responses for the question Sensing DCBLP integration.

From the Table 4 (and leftmost graph in Figure 9 is obvious that there is a significant difference in sensing DCBLP ( $\chi^2 = 13.5, \chi^2_{0.05}(1) = 3.84$ ). Thus  $H_0$  is rejected and alternative hypothesis  $H_A$  for the area of interest (H1a) can be accepted. It is possible to express that the positive attitude is observed.

Next, the differences between the learning areas of combinational and sequential logic are investigated. The results are visualized by the middle and right graphs in Figure 9. It is now necessary to answer the question: "where is the significant positive impact (meaning the area of the training) when positive (question H1a-A) attitude has been expressed by the students?"

(B) Description of combinational logic by DCBLP

The statistical null hypothesis and alternative hypothesis are formulated:

**Hypothesis 0 (H0).** *There is no significant difference in responses regarding usefulness of DCBLP in relation to combinational logic programming.* 

**Hypothesis A (HA).** *There is a significant difference in responses regarding usefulness of DCBLP in relation to combinational logic programming.* 

Calculation of the results are based on the same approach as for the question H1a-A. The frequencies of responses are: quite useful = 16, very useful = 4, rather not useful = 4. For the comparison calculated  $\chi^2 = 10$ ,  $\chi^2_{0.05}(2) = 5.991$  it is possible to conclude there is a significant difference between the individual answers thus rejecting  $H_0$  and accepting  $H_A$ . Since there is a difference, the comparison can continue. For the combinational logic can be expressed that the students evaluate it as "quite useful". It is significantly more than expressing it as a "very useful" ( $\chi^2 = 7.2$ ,  $\chi^2_{0.05}(1) = 3.84$ ). There is no student that evaluates it as a "totally bad".

(C) Description of sequential logic by DCBLP

The statistical null hypothesis and alternative hypothesis are formulated:

**Hypothesis 0 (H0).** There is no significant difference in responses regarding usefulness of DCBLP in relation to sequential logic programming.

**Hypothesis A (HA).** *There is a significant difference in responses regarding usefulness of DCBLP in relation to sequential logic programming.* 

The frequencies of responses are: very useful = 1, quite useful = 16, rather not useful = 1, totally bad = 1. For the sequential logic the same amount of the students (the same students) as described in H1a-B expressed that DCBLP is "quite useful". However, there is significantly less amount of the students that marked DCBLP as a "very useful" compared to combinational logic area ( $\chi^2 = 13.24$ ,  $\chi^2_{0.05}(1) = 3.84$ ). There are also more students that stated DCBLP for sequential logic area as "rather not useful" compared to combinational logic. Thought the difference between these two statements (quite useful vs. rather not useful) is still statistically significant ( $\chi^2 = 4.55$ ,  $\chi^2_{0.05}(1) = 3.84$ ). For the case of simplification (aggregating positive attitude, aggregating negative attitude) the difference is still significant ( $\chi^2 = 4.20$ ,  $\chi^2_{0.05}(1) = 3.84$ ). We also compared the overall responses with the result  $\chi^2 = 25.0$ ,  $\chi^2_{0.05}(3) = 7.815$ . Alternative hypothesis  $H_A$  (with countable 5% error probability) can be accepted. Additionally, it is possible to express that in both of the areas (combinational and sequential logic) the discourse is statistically sensed by the students as "quite useful".



Figure 9. Questions related to usefulness of the programming discourse.

Based on the counts of the responses for the questions A, B and the comparison of expected observation the whole area H1a can be expressed that a new discourse is sensed by the experimental group as appropriately enhancing the digital logic education.

For the Area of Interest H1b—Demonstration Usefulness

We need to judge whether the discourse is positively sensed during the demonstration of logic. There are two questions in the questionnaire related to this area of interest (Table 1) that must be evaluated one by one before we judge. The null hypothesis can be verified using two questions in the survey (Figure 10). The first one is direct: "In which (if any) area the demonstration of digital logic supported by DCBLP helped with the study?" The second is indirect: "What was the most useful aspect noticed during the training?"

(A) Demonstration of the digital logic supported by DCBLP

The statistical null hypothesis and alternative hypothesis are formulated:

**Hypothesis 0 (H0).** There is no significant difference in responses regarding DCBLP demonstration usefulness.

**Hypothesis A (HA).** *There is a significant difference in responses regarding DCBLP demonstration usefulness.* 

The frequencies of responses are: yes—for combinational logic = 4, yes—for sequential logic = 4, yes—both types = 16, did not help = 0. There is no need to utilize chi-square test to interpret the data gained for the first question as the difference between the amounts

of answers is obvious (Figure 10-left), however the calculation is as follows:  $\chi^2 = 33.0$ ,  $\chi^2_{0.05}(3) = 7.815$ . Accepting the HA: there is a significant difference in responses regarding DCBLP demonstration usefulness. Based on the second question (Figure 10-right) an additional hypothesis (B) can be expressed.

(B) What was the most helpful during the training

**Hypothesis 0 (H0).** *There is no significant difference between demonstration based on DCBLP and other helpful aspects.* 

**Hypothesis A (HA).** *There is a significant difference between demonstration based on DCBLP and other helpful aspects.* 

The students could select multiple answers, therefore the frequencies of responses are higher. The frequencies of responses are: nothing = 1, demonstration based on DCBLP = 10, own notes = 6, whiteboard writings = 19. Also for the question the chi-square has been calculated ( $\chi^2 = 19.33$ ,  $\chi^2_{0.05}(3) = 7.815$ ). Although the answers are significantly different, after additional analysis comparing individual answers one to each other it cannot be confirmed that DCBLP acts as the most significant helpful aspect in the training. Null hypothesis ( $H_0$ ) cannot be rejected in this case.



Figure 10. Questions related to sensing the demonstration usability.

For the H1b area it is possible to conclude that the declared usefulness is mostly sensed under both areas of learning content, and in overall significantly more than declared unhelpfulness, however there is no significant difference between declared aspects that can help the most. Demonstration is not sensed as the most important aspect during the training.

For the Area of Interest H1c—Self Practicing

We need to judge if the discourse is positively sensed during the practicing or for practicing the logic. There are two questions in the questionnaire related to this area of interest (Table 1) that must be evaluated one by one before we judge.

(A) Do you like to include practicing DCBLP in ICT labs?

The statistical null hypothesis and alternative hypothesis are formulated:

**Hypothesis 0 (H0).** There is no declared (sensed) need for practicing DCBLP during the training.

**Hypothesis A (HA).** There is declared (sensed) need for practicing DCBLP during the training.

The responses are as follows: yes—I don't have the possibility to train at home = 18, yes—need more practicing and feedback = 4, no—I doubt I can ever understand it = 1, no—it's useless = 1. There is an obvious difference (Figure 11-left) between the answers distribution ( $\chi^2 = 33.0, \chi^2_{0.05}(3) = 7.815$ ). Based on data we can reject  $H_0$  and accept  $H_A$ : There exists students' sensed need for practicing logic when using DCBLP.

(B) During the training I missed the most ...

The statistical null hypothesis and alternative hypothesis are formulated:

**Hypothesis 0 (H0).** *There are no declared differences between practicing needs.* 

Hypothesis A (HA). There are declared differences between practicing needs.

The responses for the answer are (multiple choices were allowed): learning materials = 18, more programming at school = 10, more tasks for home preparation = 7, more challenging programming tasks = 4, easier programming tasks = 9. Answers for missing aspects in training ( $\chi^2 = 11.38$ ,  $\chi^2_{0.05}(4) = 9.488$ ) show that there is a difference between preferences, however there is no significant difference between the needing of programming at school ( $\chi^2 = 0.53$ ,  $\chi^2_{0.05}(1) = 3.84$ ) and also no significant difference between missing more challenging programming tasks and easier programming tasks ( $\chi^2 = 1.92$ ,  $\chi^2_{0.05}(1) = 3.84$ ).

There is also no significant difference between the necessity of the learning materials and more tasks for programming (neither separated nor aggregated programming at school/home). However there is a strong needing for including practical programming into the training (aggregated for Figure 11-left  $\chi^2 = 16.66$ ,  $\chi^2_{0.05}(1) = 3.84$ ), and has been revealed, there is also a strong need for learning materials that were not available during the period the research was in progress (as already mentioned in the section Training Content Enhancements in Details). Therefore null hypothesis is rejected and  $H_A$  is accepted.



Figure 11. Questions related to practicing and missing training elements.

For the H1c area it is possible to conclude that, based on the declared answers, there are significant differences in the answers in both subcategories. It has been investigated that there is a strong need to include practicing into the training and a strong need for the learning materials that were not available during the time of preliminary training.

For the Area of Interest H1d—Motivation for Digital Logic (Learning)

We need to judge if the discourse is sensed as motivating for the subject learning. There are two questions in the questionnaire related to this area of interest (Table 1) that must be evaluated one by one before we judge.

(A) Including (inheriting) DCBLP into digital logic was for me ...

The statistical null hypothesis and alternative hypothesis are formulated:

**Hypothesis 0 (H0).** There is no difference between answers regarding declared motivation.

**Hypothesis A (HA).** *There is a difference between answers regarding declared motivation.* 

The count responses for the answer are: motivating for digital logic = 9, neutral = 10, demotivating = 3. There is a difference between answers when asking about motivation, but there is no significant difference between individual answers compared in overall

 $(\chi^2 = 3.90, \chi^2_{0.05}(2) = 5.99)$ . There is no statistically measurable additional motivation aspect expressed by the students when compared the amount of answers in option "Motivating" to answers in option "Demotivating" ( $\chi^2 = 3.0, \chi^2_{0.05}(1) = 3.84$ ). Therefore  $H_0$  cannot be rejected.

(B) Programming included in digital logic

The statistical null hypothesis and alternative hypothesis are formulated:

Hypothesis 0 (H0). There is no difference between answers regarding declared popularity.

Hypothesis A (HA). There is a difference between answers regarding declared popularity.

The count responses for the answer are: dislike = 3, like = 16, neutral = 5. Although there is no strong motivation aspect measured, students do like the way of enhancing digital logic by proposed programming (Figure 12-right). There is a significant difference ( $\chi^2 = 12.25, \chi^2_{0.05}(2) = 5.99$ ) between the frequency of answers dislike/like/neutral. The popularity is the most frequent answer. The null hypothesis is rejected and the alternative hypothesis  $H_A$  is accepted.



Figure 12. Questions related to motivation for learning digital logic.

For the area H1d is possible to express that there is no significant difference between declaration of motivation or demotivation. We cannot say that most of the students are motivated thanks to the discourse. However, it is possible to express that the students sensed the discourse as suitable and they like the inheritance into the training.

For the Area of Interest H1e—Iot Programming

We need to judge whether the discourse is sensed as motivating for the future programming on IoT devices. There is one question in the questionnaire related to this area of interest (Table 1) that must be evaluated one by one before we judge.

(A) Motivation for the future programming on IoT

The statistical null hypothesis and alternative hypothesis are formulated:

**Hypothesis 0 (H0).** *There is no difference between answers regarding declared motivation for future programming on IOT.* 

**Hypothesis A (HA).** *There is a difference between answers regarding declared motivation for future programming on IOT.* 

The count responses for the answer are: very motivated = 4, slightly motivated = 14, slightly demotivated = 4, demotivated = 2. There is a significant difference ( $\chi^2 = 14.67$ ,  $\chi^2_{0.05}(3) = 7.81$ ) in distribution of the answers. When comparing only options "slightly motivated" and "very motivated", the option "slightly motivated" must be selected as statistically significant ( $\chi^2 = 5.56$ ,  $\chi^2_{0.05}(1) = 3.84$ ). The null hypothesis must be rejected and HA is accepted.

Motivation for the future programming on IoT





The students claim they are slightly motivated for future programming on IoT devices after completing the training although it cannot be assured that the only reason is a new discourse (Figure 13).

There is an additional finding beside the verification of the proposed hypothesis. There ware more questions regarding the programming used in the educational process. It was not just about an impact of DCBLP on digital logic. It was about the students' perception of the programming language support. It is clear from the graph below (Figure 14) that there is no significant difference between the amounts of answers gathered for each option.



Figure 14. Question related to using programming language during the training.

It partially corresponds to previous students' claims regarding DCBLP included in digital logic. There are slight differences although they are not significant. The interesting matter can be revealed when joined with another question about the study program preferences. Of the respondents, 20% claimed they would prefer electronics study content instead of a focus on programming. Of these students, 80% also claimed that programming was not supportive for solving any tasks during the whole study program. Creating a contingency table and putting data in the test for these two areas (if there is a relation between preferences of study program and willingness to using the programming language) brought the result that there is a relation ( $\chi^2 = 13.71$ ,  $\chi^2_{0.05}(2) = 5.99$ ) between the study program preferences and willingness to use the programming language. It means that the students preferring electronics instead of programming also perceive the new discourse as not supportive. From this small sample cannot be concluded it is valid for whole population, however the students' preferences regarding the study branch that they prefer should be taken into account.

### 6.2. Achievement Test Results Data Presentation

The achievement test provides data for the H2 related hypothesis. Part of the analysis is based on exploratory data analysis supported by box-plots and the decisions of significant differences are based on ANOVA. We compare the answers in the achievement tests in both groups (experimental/control). Training areas are discussed individually in details, supported by statistics and appropriate graphs:

First the overall performance comparison in the achievement test:

**Hypothesis 0 (H0).** *There is no difference between students overall performance in the area of digital logic with and without a support of a new discourse.* 

**Hypothesis A (HA).** There is a difference between students overall performance in the area of digital logic with and without a support of a new discourse.

The calculation is based on summarized points (for all of the tasks) gained by each student (in the achievement test.). Based on that, average points for both groups have been calculated (Figure 15-left). Standard deviance is also included in the graph. For better comparison of both groups the quartile deviance has been calculated as well (Figure 15-right). The experimental group  $\tilde{x} = 10$ ,  $Q_1 = 7.25$ ,  $Q_3 = 12.9$  (where  $\tilde{x}$  is median,  $Q_1$  the first quartile,  $Q_3$  is the third quartile), the lowest gained points equal 4.5 and the highest points gained equal 16. For the control group  $\tilde{x} = 3$ ,  $Q_1 = 2$ ,  $Q_3 = 5$ , the lowest points gained equal 0, the highest points gained equal 14. The maximum points that could be gained by each student (for the whole test) is 16.



Figure 15. Comparison of average points gained per group (left) and quartile graphs for points gained (right).

Alpha	0.05				
Groups	Count	Sum	Mean	Variance	
Experimental	28	278.500	9.946	12.710	
Control	25	90.500	3.620	9.006	
Source of Variation	SS	df	MS	F	F critical
Between Groups	528.615	1.000	528.615	48.201	4.030
Within Groups	559.310	51.000	10.967		
Total	108.925	52.000			

Table 5. ANOVA (single factor) results for overall achievement tests group comparison.

The points gained by each student are also used to calculate analysis of variance ANOVA (Table 5) F = 48.2,  $F_{0.05}(1,51) = 4.03$ , from which it is possible to accept alternative hypothesis  $H_A$  for the area of overall performance.

Additional investigation regarding hypothesis must be performed:

Hypothesis 2a0 (H2a0). The discourse has no measurable impact in the block of Boolean logic functions.

Hypothesis 2aA (H2aA). The discourse has a measurable impact in the block of Boolean logic functions.

Because of the different count of participants, the data cannot be directly compared by simple calculation and visualization of the amount (frequency) of correct answers. Therefore, this comparison (Figure 16-left) is only illustrative. Better results comparison brings box-plots (quartile based graphs: Figure 16-right). The experimental group  $\tilde{x} = 4.75$ ,  $Q_1 = 4.38$ ,  $Q_3 = 6.13$ , the lowest gained points is 1.5 and the highest points gained is 7. For the control group  $\tilde{x} = 1.5$ ,  $Q_1 = 1$ ,  $Q_3 = 3$ , the lowest points gained equal 0, the highest points gained = 7. Maximum possible points that could be gained by each students for this part = 7.



Figure 16. Correct answers in total per task (l) and quartile graphs for points gained (r).

Alpha	0.05				
Groups	Count	Sum	Mean	Variance	
Experimental	28	141.5	5.05	1.78	
Control	25	53	2.12	2.88	
Source of Variation	SS	df	MS	F	F critical
Between Groups	113.66	1	113.66	49.41	4.03
Within Groups	117.31	51	2.3		
Total	230.97	52			

Table 6. ANOVA (single factor) results for Boolean logic block achievement tests comparison.

Analysis of variance brings the following results (Table 6): F = 49.41,  $F_{0.05}(1,51) = 4.03$ , rejecting H2a0 and accepting H2aA. There is a significant difference between the groups in the selected questions with the content of Boolean logic.

As next the null and alternative hypotheses are formulated for the combinational logic area:

Hypothesis 2b0 (H2b0). The discourse has no measurable impact in the block of combinational logic.

Hypothesis 2bA (H2bA). The discourse has a measurable impact in the block of combinational logic.

The experimental group  $\tilde{x} = 1$ ,  $Q_1 = 1$ ,  $Q_3 = 2$ , the lowest gained points = 0 and the highest points gained = 2. For the control group  $\tilde{x} = 0$ ,  $Q_1 = 0$ ,  $Q_3 = 1$ , the lowest points gained = 0, the highest points gained = 1. Maximum possible points that could be gained by each student for this part = 2 (Figure 17).



Figure 17. Correct answers in total per task (l) and quartile graphs for points gained (r).

Alpha	0.05				
Groups	Count	Sum	Mean	Variance	
Experimental	28	34	1.21	0.45	
Control	25	12	0.48	0.43	
Source of Variation	SS	df	MS	F	F critical
Between Groups	7.12	1	7.12	16.17	4.03
Within Groups	22.45	51	0.44		
Total	29.58	52			

Analysis of variance brings the following results (Table 7): F = 16.17,  $F_{0.05}(1,51) = 4.03$ , rejecting null hypothesis, accepting H2bA.

Next, the null hypotheses are formulated for the sequential logic area:

Hypothesis 2c0 (H2c0). The discourse has no measurable impact in the block of sequential logic.

**Hypothesis 2cA (H2cA).** The discourse has a measurable impact in the block of sequential logic.

The experimental group  $\tilde{x} = 0$ ,  $Q_1 = 0$ ,  $Q_3 = 1.5$ , the lowest gained points is 0 and the highest points gained is 2. For the control group  $\tilde{x} = 0$ ,  $Q_1 = 0$ ,  $Q_3 = 0.5$ , the lowest points gained equal 0, the highest points gained = 1. Maximum possible points that could be gained by each students for this part = 2 (Figure 18).



Figure 18. Correct answers in total per task (l) and quartile graphs for points gained (r).

Alpha	0.05				
Groups	Count	Sum	Mean	Variance	
Experimental	28	25.5	0.91	0.63	
Control	25	7.5	0.3	0.19	
Source of Variation	SS	df	MS	F	F critical
Between Groups	4.93	1	4.93	11.67	4.03
Within Groups	21.53	51	0.42		
Total	26.45	52			

Table 8. ANOVA (single factor) results for the Sequential block achievement tests comparison.

Analysis of variance brings the following results (Table 8): F = 11.67,  $F_{0.05}(1,51) = 4.03$ , rejecting null hypothesis and accepting H2cA.

Next the null and alternative hypotheses are formulated for the programming area:

**Hypothesis 2d0 (H2d0).** *It has no measurable impact in the programming language usage across the logic blocks.* 

**Hypothesis 2dA (H2dA).** It has the measurable impact in the programming language usage across the logic blocks.

All tasks/questions require the usage of a programming language. There was no necessity to use DCBLP in the answers for the question as the control group students have had no training in this new programming discourse. Although both groups went through previous training in the programming language (see Figure 1 stage 1). Every piece of code using any programming language and leading to a task's solution was acceptable. The graph on the left side of Figure 19 shows the frequency (sum amount) of acceptable solutions per task, on the right are the quartile graphs for groups' comparison. The experimental group  $\tilde{x} = 2.5$ ,  $Q_1 = 1.5$ ,  $Q_3 = 4.5$ , the lowest gained points = 0 and the highest points gained = 5. For the control group  $\tilde{x} = 0$ ,  $Q_1 = 0$ ,  $Q_3 = 1$ , the lowest points gained equal 0, the highest points gained = 4. Maximum possible points that could be gained by each students for this part = 5. Graphs for this part are available in Figure 19.



Figure 19. Correct answers in total per task (l) and quartile graphs for points gained (r).

Alpha	0.05				
Groups	Count	Sum	Mean	Variance	
Experimental	28	77.5	2.77	2.64	
Control	25	18	0.72	1.88	
Source of Variation	SS	df	MS	F	F critical
Between Groups	55 39	1	55 30	24.20	4.02
1	00.07	1	55.59	24.29	4.05
Within Groups	116.28	51	2.28	24.29	4.05

Table 9. ANOVA (single factor) for Programming block achievement tests comparison.

Analysis of variance brings the following results (Table 9): F = 24.29,  $F_{0.05}(1,51) = 4.03$ , rejecting null hypothesis, accepting H2dA.

It is also possible to express that it has no impact on the previous hypotheses' acceptance in certain areas.

- The task no. 4 into Boolean logic tasks and functions: F = 49.48,  $F_{0.05}(1,51) = 4.03$ , still accepting H2aA;
- The tasks no. 7, 8c, 9 into combinational logic tasks: F = 23.49,  $F_{0.05}(1,51) = 4.03$ , still accepting H2bA;
- The task no. 12 into sequential tasks: F = 14.80,  $F_{0.05(1,51)} = 4.03$ , still accepting H2cA. In all of the measured areas the achievements of the students in the experimental group were significantly better.

# 7. Future Work and Possible Improvements

There are several areas of possible training improvements:

- Possibility to use and offer to the students the developed learning materials that were not available during the training and preliminary research. Learning materials can be included as another control variable that can impact the learning process.
- Using constructive learning methods along with the instructive learning methods and using the computers during the training to directly practice logic networks using the programming language. This will affect the allowed value in the "Teaching methods" control variable.
- Possibility to utilize inquiry based education.

The next research plan is as follows:

- Repeat the experiment with different experimental and control groups.
- Realize the experiment at different/another school(s).
- Focus on IoT education impact (Figure 1—stage 3).
- Find the implications. The reason why and in what circumstances DCBLP does and does not work.

# 8. Conclusions

The preliminary research revealed that there is the possibility of using developed programming discourse DCBLP in digital logic training at secondary schools focused on IT framework education. Thanks to the survey focused on gathering students' opinions we can expect that the students will probably accept enhancements of the digital logic curriculum by the proposed programming discourse. The multi-paradigm programming in this specific domain is possible. From the data available it is possible to conclude that the programming discourse used during the instructive education supported by the demonstrative aspects is sensed by the students to be appropriate, quite useful, and positively affects the combinational area the most. However, the discourse itself in such training is not sensed to be the most important aspect of the education. The students also ask for more practice during the lessons when the discourse is in use and along with it they ask for learning materials. The students who do not prefer programming and prefer electronics instead will probably claim the programming discourse is not supportive. However, the students claim that the overall usefulness of the discourse in the training, the strong motivation power of the programming discourse itself has not been discovered based on their answers. From the achievement test we conclude that the overall performance of the students trained using the new programming discourse is significantly better. They performed better compared to the students in control group in every area of interest. The least significant difference was measured in the area of sequential logic. The experiment should continue to verify data from this preliminary experiment and to reveal new information that can be still hidden due to the low number of participants. It would also be interesting to compare the results from the different classes at different schools who would be curious to participate in this research. It is still too soon to answer all of the questions asked at the beginning of the research but the direction towards answering them is set.

**Author Contributions:** Conceptualization, L.H. and H.H.; methodology, L.H.; software, L.H.; validation, H.H. and L.H.; formal analysis, H.H. and L.H.; writing—original draft preparation, L.H.; writing—review and editing, H.H. and L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by University of Ostrava specific research budget.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

# References

- NUV. Nová Klasifikace ISCED 2011 (New Classification ISCED 2011). 2015. Available online: www.nuv.cz/isced (accessed on 31 May 2020).
- 2. Galotti, K.M. Cognitive Development: Infancy through Adolescence, 2nd ed.; SAGE: London, UK, 2016; ISBN 978-1483379173.
- 3. Rámcový Vzdělávací Program (Framework Educational Program) pro Obor Vzdělání Informační Technologie; MŠMT: Praha, Czech Republic, 2008; 85p.
- 4. Roth, C.H. Fundamentals of Logic Design, 5th ed.; Thomson/Brooks/Cole: Belmont, CA, USA, 2004; ISBN 978-0534378042.
- 5. Bindal, A. Fundamentals of Computer Architecture and Design; Springer International Publishing: Cham, Switzerland, 2017.
- Školní Vzdělávací Program (School Educational Program): Informační Technologie. Obchodní Akademie, Vyšší Odborná Škola a Jazyková Škola s Právce Státní Jazykové Zkoušky Uherské Hradiště. 2018. Available online: https://www.oauh.cz/ (accessed on 31 May 2020).
- Školní Vzdělávací Program (School Educational Program): Moderní Informační Technologie. Střední Škola Informatiky, Elektrotechniky a řemesel Rožnov pod Radhoštěm. 2016. Available online: <a href="https://www.roznovskastredni.cz/obor-18-20-m-01-moderni-informacni-technologie">https://www.roznovskastredni.cz/obor-18-20-m-01-moderni-informacni-technologie</a> (accessed on 31 May 2020).
- 8. Connelly, R.; Gousie, M.; Hadimioglu, H.; Ivanov, L.; Hoffman, M. The role of digital logic in the computer science curriculum. *J. Comput. Sci. Coll. JCSC* 2004, *19*, 5–8.
- 9. Noga, K.M.; Radwanski, M. Our experiences in teaching of digital logic. In *Innovations in E-learning, Instruction Technology, Assessment, and Engineering Education;* Iskander, M., Ed.; Springer: Dordrecht, The Netherlands, 2007; ISBN 978-1-4020-6262-9.
- 10. Clark, M.A.C.; Boyle, R.D. A Personal Theory of Teaching Computing Through Final Year Projects. *Comput. Sci. Educ.* **1999**, *9*, 200–214. [CrossRef]
- 11. Clocksin, W.F. Logic programming and digital circuit analysis. J. Log. Program. 1987, 4, 59–82. [CrossRef]
- 12. Gal-Ezer, J.; Harel, D. Curriculum and Course Syllabi for High-School Computer Science Program. *Comput. Sci. Educ.* **1999**, *9*, 114–147. [CrossRef]
- 13. Habiballa, H.; Kmet, T. Theoretical branches in teaching computer science. *Int. J. Math. Educ. Sci. Technol.* **2004**, *35*, 829–841. [CrossRef]

- Habiballa, H.; Jendryscik, R. Constructivistic Mathematical Logic Education. In AIP Conference Proceedings, Proceedings of the International Conference of Computational Methods in Sciences and Engineering 2017, Rhodes, Greece, 15–17 June 2017; AIP Publishing Inc.: Melville, NY, USA, 2018; Volume 2040, pp. 0300061–0300064, ISBN 978-073541766-3.
- 15. Habiballa, H.; Kmet, T. Mathematical Logic and Deduction in Computer Science Education. *Inform. Educ.* **2008**, *7*, 75–90. [CrossRef]
- 16. Kfoury, A. Mathematical Logic in Computer Science. *arXiv* **2018**, arXiv:1802.03292.
- 17. Kfoury, A. Personal Reflections on the Role of Mathematical Logic in Computer Science. *Fundam. Inform.* **2019**, 170, 207–221. [CrossRef]
- Devedzic, V.; Debenham, J. An Intelligent Tutoring System for Teaching Formal Languages. In International Conference on Intelligent Tutoring Systems; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; p. 514.
- 19. M-Labs. Migen: FHDL. 2020. Available online: http://m-labs.hk/gateware/migen (accessed on 4 March 2020).
- 20. Morrison, J.P. Flow-Based Programming: Links to External References (as of September 2015). 2015. Available online: http://www.jpaulmorrison.com/fbp/links\_external.html (accessed on 31 May 2020).
- Gabbrielli, M.; Martini, S. Programming Languages: Principles and Paradigms; Springer: Berlin/Heidelberg, Germany, 2010; pp. 369–378, ISBN 978-1-84882-913-8.
- Robinson, D. The Incredible Growth of Python. Stackoverflow: Blog. 2017. Available online: https://stackoverflow.blog/2017/0 9/06/incredible-growth-python/ (accessed on 25 September 2018).
- 23. Morrison, J.P. Flow-Based Programming: A New Approach to Application Development; Van Nostrand Reinhold: New York, NY, USA, 1994; ISBN 0-442-01771-5.
- Hapl, L.; Habiballa, H. Introduction to Digital Circuits Based Logical Programming (DCBLP), the multi-paradigm way of logical based programming. In AIP Conference Proceedings, Proceedings of the International Conference of Computational Methods in Sciences and Engineering 2019, Rhodes, Greece, 1–5 May 2019; AIP Publishing Inc.: Melville, NY, USA, 2019.
- Hapl, L. Possibility of Raspberry minicomputer in teaching. In *Perspektivy Elektroniky*; SŠIEŘ Rožnov pod Radhoštěm: Rožnov pod Radhoštěm, Czech Republic, 2018; pp. 1–10, ISBN 978-80-270-5415-1. (In Czech)
- 26. Hapl, L. Digital Systems and DCBLPy for Informatics Students (CZ); University of Ostrava: Ostrava, Czech Republic, 2019; ISBN 978-80-7599-137-9.
- 27. Atkins, V.; Wallace, S. Qualitative Research in Education; SAGE: London, UK, 2012.
- 28. Ben-Ari, M. Constructivism in computer science education. J. Comput. Math. Sci. Teach. 2001, 20, 45–73.
- 29. Mcmillan, J.H. Fundamentals of Educational Research; Pearson Education: London, UK, 2015; ISBN 9780133580723.
- 30. Gaspar, H.; Morgado, L.; Mamede, H.; Oliveira, T.; Manjón, B.; Gütl, C. Research priorities in immersive learning technology: The perspectives of the iLRN community. *Virtual Real.* **2020**, *24*, 319–341. [CrossRef]
- 31. Freedman, D.A. Statistical Models: Theory and Practice; Cambridge University Press: Cambridge, UK, 2005; ISBN 978-0-521-67105-7.
- 32. Lindquist, E.F. Statistická Analýza v Pedagogickém Výzkumu; SPN: Praha, Czech Republic, 1967. (In Czech)
- Keselman, H.J.; Huberty, C.J.; Lix, L.M.; Olejnik, S.; Cribbie, R.A.; Donahue, B.; Kowalchuk, R.K.; Lowman, L.L.; Petoskey, M.D.; Keselman, J.C.; et al. Statistical Practices of Educational Researchers: An Analysis of their ANOVA, MANOVA, and ANCOVA Analyses. *Rev. Educ. Res.* 1998, 68, 350–386. [CrossRef]