ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 10, No. 1, 2025

RESEARCH AND DESIGN OF A MATRIX MULTIPLIER ON FPGA

Ihor Kogut[1], Volodymyr Hryha[1], Bohdan Dzundza[1], Liudmyla Hryha[1], Iryna Hatala[2]

[1] Vasyl Stefanyk Precarpathian National University, 57, Shevchenko str., Ivano-Frankivsk, 76018, Ukraine,
[2] Wayne State University, 42, Warren Ave, Detroit, 48202, USA

Authors' e-mails*: igor.kohut@pnu.edu.ua, volodymyr.gryga@pnu.edu.ua, bohdan.dzundza@pnu.edu.ua, liudmyla.hryha@pnu.edu.ua, iryna.hatala@gmail.com*

https://doi.org/10.23939/acps2025.01.010

Submitted on 10.03.2025

***Abstract*: This paper presents a comprehensive investigation and hardware implementation of a multi-bit Brawn matrix multiplier architecture. The research focuses on analyzing the system characteristics of binary multipliers realized with both conventional and optimized full and half adders. Particular attention has been given to the applicability of such multipliers within arithmetic logic units (ALUs) for vector and scalar processing architectures. Analytical models have been formulated to quantify hardware resource utilization and computational latency across various logic base configurations. The proposed multiplier has been described using the VHDL hardware description language and validated through functional simulation. The designs have been synthesized and implemented on Xilinx FPGA platforms. It has been established that the use of improved full and partial binary adders as part of the Brown matrix multiplier reduces the hardware complexity by a factor of 1.7 and increases the performance by a factor of 2.9 compared to the known classical structures of binary adders. The use of multibit Brown matrix multipliers with an improved element base allows to significantly speed up the execution time of the multiplication operation of special-purpose processors and vector and scalar supercomputers.**

***Index terms*: matrix multiplier, half and full adder, ALU, FPGA, synthesis, topology.**

I. INTRODUCTION

Binary multiplication is a fundamental operation widely employed in cryptanalysis tasks, digital image and signal processing, RGB pixel transformations, and spectral analysis. This operation is typically implemented using either software-based or hardware-based techniques [1–3].

Multi-bit binary multipliers are extensively utilized in coprocessors, processor control units, arithmetic logic units (ALUs), specialized signal and image processing processors, as well as in both vector and scalar processing architectures [1–5].

Functionally and structurally, multiplication can be performed using software methods that rely on sequential algorithms or other approaches based on analytical expressions [3, 4, 6]. Alternatively, hardware-based implementations offer another approach, including parallel, sequential, and hybrid parallel-sequential architectures [3–6].

One of the primary indicators of efficiency in multibit matrix and tree-structured multipliers is their ability to maintain high computational throughput, which is largely determined by the minimization of signal propagation delays during the formation of partial sums and the transmission of carries within the underlying single-bit binary adder cells.

Additional essential performance metrics include minimal structural complexity in reduced input/output overhead and low hardware complexity.

A wide array of architectural implementations exists for single-bit half binary adders and full binary adders, many of which are integral building blocks in the design of matrix and multilayer binary multipliers.

Accordingly, the development of multi-bit matrix multipliers utilizing optimized single-bit adder cells, along with a comprehensive evaluation of their system-level performance characteristics, represents a timely and practically important direction of research.

II. LITERATURE REVIEW AND PROBLEM STATEMENT

Arithmetic operations encompass addition, subtraction, multiplication, division, comparison, and square root extraction [1, 3, 5, 6].

Among these, multiplication is the second most frequently performed operation in computational systems, following addition. Several methods exist for accelerating multiplication, such as altering the encoding of the operands to reduce the number of partial products (Booth's algorithm) [7], utilizing more efficient summation techniques for partial products that minimize the time spent on carry propagation, and implementing parallel computation of all partial products [2, 3]. These approaches are typically realized using combinational logic circuits.

Parallel computation of partial products is a fundamental aspect of all multiplication schemes. The primary distinction between different architectures lies in the method of summing the partial products. Based on this, multipliers can be categorized into matrix-based and multi-layer tree-structured designs. The difference between these types is expressed in the number of single-bit adders utilized, their configuration, and the technique employed for carry propagation during summation.

In matrix multipliers, summation is performed by a matrix of adders, which consists of sequential rows of single-bit adders with carry propagation. As the data progresses downward through the array of adders, each row of the carry-save adders adds the next partial product to the accumulated sum of partial products. Matrix multipliers achieve high performance and exhibit a significant degree of regularity, which is particularly advantageous when implementing such multipliers as integrated circuits. On the other hand, such circuits occupy a large area on the chip, and with the increase in the bit depth of the multipliers, this area increases in proportion to the square of the number of bits. Another disadvantage of matrix multiplier is its low hardware utilization. As the sum of partial products propagates down the matrix, each row is used only once when it intersects with the active computation phase. This characteristic can be exploited to enhance computational efficiency by pipeline processing, where, as each row is freed up, it can be reused to multiply the next pair of operands.

Some of the most well-known matrix multipliers include the Brown multiplier, the carry-propagating multiplier [5, 9], as well as multipliers based on the Boole-Vulli [3, 4] and Pezaris [3] algorithms, designed for multiplication of binary numbers in two's complement representation.

To reduce the delay typical of matrix multipliers, designs based on tree-like structures can be utilized. While multi-layer multipliers exhibit faster performance than matrix-based structures, their implementation requires additional connections to merge bits of the same weight. This can result in a larger chip area than in matrix-based adder configurations.

In multi-layer multipliers, summation is achieved through a tree-like structure, primarily focusing on the compression of partial products with the same weight by employing single-bit full adders (3–2) and half adders (2–2). Known single-bit adders with input-output ratios such as (5–3), (7–3), (15–4), and others are referred to as counters. This is because the output code of these adders, similar to that of a binary counter, corresponds to the number of units at the inputs [2–5].

Among the most widely used multi-layer multipliers are tree-based schemes implemented using the Wallace [9, 10] and Dadda [11, 12] algorithms.

III. SCOPE OF WORK AND OBJECTIVES

The development of high-performance specialized computing systems is an urgent task in the current conditions of dynamic growth in demand for high-speed multi-bit components of ALUs vector and scalar processors used in the field of artificial intelligence (AI). At the same time, an important task is to investigate the system characteristics of high-speed multi-bit matrix multipliers based on improved structures of half adders and full adders [1–6].

The main goal of the work is to study and design multi-bit high-speed matrix multipliers on FPGAs for their further use in signal and image processing systems, cyber-physical systems, and tasks of processing RGB pixels of color images [4–6].

This article uses the materials and results obtained by the authors during the research work "Multifunctional sensor microsystem for non-invasive continuous monitoring and analysis of human biosignals" state registration number 0124U000384 dated 01.01.2024, which is carried out at the Department of Computer Engineering and Electronics, of the Vasyl Stefanyk Precarpathian National University in 2024-2026.

IV. DESIGN OF THE ARCHITECTURE OF THE BRAWN'S MATRIX MULTIPLIER

Fig. 1 shows the structure of a 4×4 Braun matrix multiplier. This architecture incorporates single-bit binary half adders (HA) and full adders (FA) [2–6].

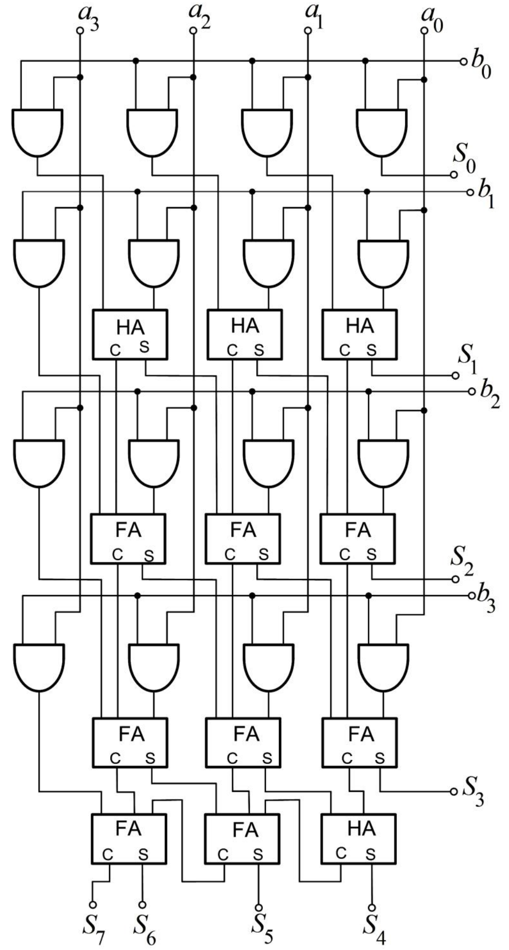


Fig. 1. Structure of the Brown 4×4 matrix multiplier

The overall hardware complexity of the Braun matrix multiplier can be estimated using the following expression:

$$A_{MM} = (n^2 \times A_{LE}) + (n \times A_{HA}) + (n^2 - 2n) \times A_{FA}, \quad (1)$$

where n – bit-length of the input data; A_{LE} – hardware complexity of logical element "AND"; A_{HA} – hardware complexity of a HA; A_{FA} – hardware complexity of a FA.

Let's calculate the hardware complexity of a matrix multiplier based on the well-known classical full adders and half adders [3], the structures of which are shown in Fig. 2.

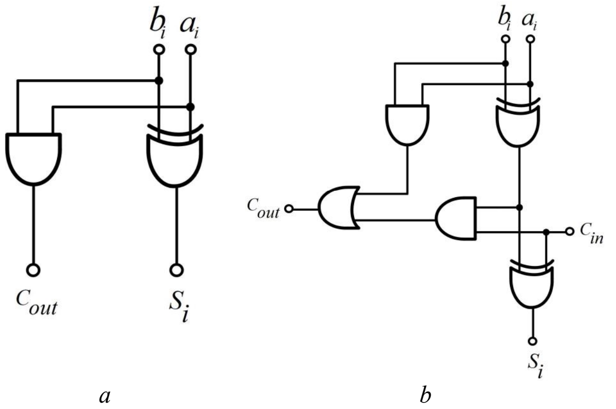


Fig. 2. Structure of a classical half adder (a) and full adder (b)

The hardware complexity of a binary half adder is represented by 5 logic gates, with a time complexity of 3 clock cycles for the sum signal generation and 1 microcycle for the carry signal generation.

The hardware complexity of a full binary adder is composed of 11 logic gates, with a time complexity of 6 microcycles for sum signal generation and 5 microcycles for carry signal generation.

For example, with $n = 4$ using classical full and half adders, the hardware complexity of the Brown matrix multiplier is calculated as follows:

$A_{MM1} = (4^2 \times 1) + (4 \times 5) + (4^2 - 8) \times 11 = 124\,(gate)$.

In [13], an improved structure of a half binary adder was proposed, which is shown in Fig. 3.

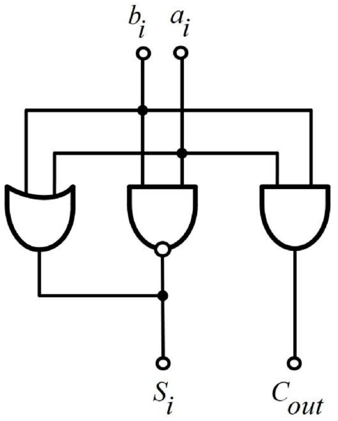


Fig. 3. Structure of an improved half adder

The hardware complexity of the improved binary half adder is 3 gates, and its time complexity is 1 microcycle when generating the sum and carry-out signals.

An improved structure of a full binary adder is proposed in [14], which is shown in Fig. 4. The hardware complexity of the improved full binary adder is comprised of 6 logic gates, while its time complexity is 2 microcycles for the generation of the sum and carry-out signals.

For example, with $n = 4$ the use of the improved HA and FA, the hardware complexity of a matrix multiplier is as follows:

$$A_{MM2} = (4^2 \times 1) + (4 \times 3) + (4^2 - 8) \times 6 = 76\,(gate).$$

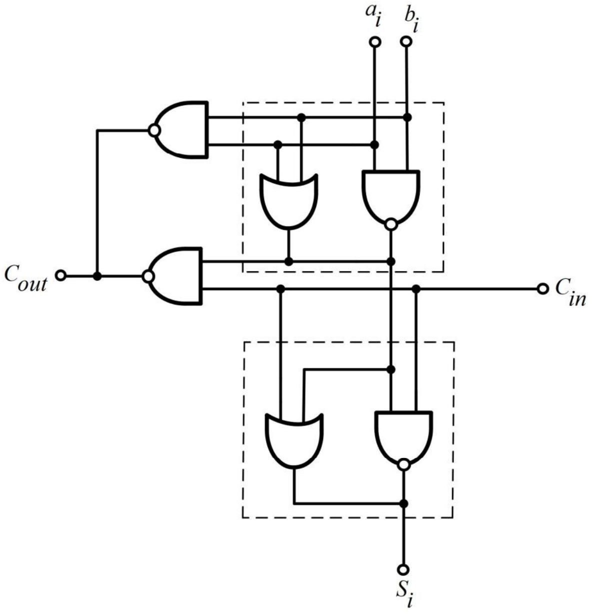


Fig. 4. Structure of an improved full adder

That is, the hardware complexity of the Brown matrix multiplier will be 1.7 times less with the use of improved HA and FA.

Table 1 shows the obtained values of the hardware complexity of multi-bit Brown's matrix multipliers based on classical and improved HA and FA

Table 1

The hardware complexity of Brown's matrix multipliers

Brown's matrix multiplier bit-length	Number of gates when using classic HA and FA (A_{MM1})	Number of gates when using advanced HA and FA (A_{MM2})
8	624	372
16	2784	1648
32	11712	6880
64	48000	28096
128	194304	113536
256	781824	456448
512	3136512	1830400

Based on the values of hardware complexity obtained in Table 2, we see that the hardware complexity of a multibit matrix multiplier (n = 512) is 393984 gates when using improved NS and PS.

The time complexity of the Brown matrix multiplier is calculated according to the following formula:

$$t_{MM} = t_{LE} + (2 \times t_{HA}) + (2n - 4) \times t_{FA}, \qquad (2)$$

where n – bit-length of the input data; t_{LE} – time complexity of logical element "AND"; t_{HA} – time complexity of a HA; t_{FA} – time complexity of a FA.

For example, with $n = 4$ the use of the improved HA and FA, the time complexity of a matrix multiplier is as follows: $t_{MM1} = 1 + (2 \times 3) + (8 - 4) \times 6 = 31\,(microcycle)$.

When using improved HA and FA, the time complexity of the Brown matrix multiplier will be: $t_{MM2} = 1 + (2\times1) + (8-4)\times2 = 11(microcycle)$.

That is, the time complexity of the 4×4 matrix multiplier will be 2.9 times less with the use of improved HA and FA.

Table 2 shows the obtained values of the time complexity of multibit Brown's matrix multipliers based on classical and improved HA and FA.

Table 2

The time complexity of Brown's matrix multipliers

Brown's matrix multiplier bit-length	Time complexity when using classic HA and FA (t_{MM1})	Time complexity when using advanced HA and FA (t_{MM2})
8	79	27
16	175	49
32	367	123
64	751	249
128	1519	507
256	3115	1039
512	6127	2043

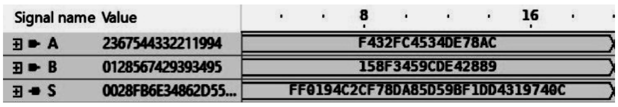
Based on the values of the time complexity obtained in Table 2, we see that the time complexity of the multi-bit matrix multiplier (n = 512) is 2043 microcycle when using improved HA and FA.

V. MODELLING AND SYNTHESIS OF BROWN'S MATRIX MULTIPLIER ON FPGA

The development of multi-bit Brown multipliers was carried out using the Verilog hardware description language within the Active HDL SE integrated environment [15].

Fig. 12 shows a diagram of the functional simulation of the 64×64 Brown matrix multiplier.

This diagram shows the transmission of 128-bit integer binary values to the inputs of *A* and *B*. The output *S* is the 128-bit result of the product of the input data.



```
Signal name Value                          8               16
A           2367544332211994   F432FC4534DE78AC
B           0128567429393495   158F3459CDE42889
S           0028FB6E34862D55...  FF0194C2CF78DA85D59BF1DD4319740C
```

Fig. 5. Functional diagram of the 64-bit Brown matrix multiplier

The synthesis of the 64-bit Brown matrix multiplier was performed on the Artix-7 FPGA family, specifically the XC7A100TCSG324-1 chip from Xilinx.

In an enlarged view, Fig. 5 represents a section of the FPGA crystal on which the structure of the 64-bit Brown matrix multiplier was implemented in the Vivado Design Suite CAD software.

As a result of the synthesis and implementation of a 64-bit Brown matrix multiplier based on improved HA and FA on the selected FPGA chip, the following hardware costs were obtained. In total, the implementation of such an MPB requires 376 LUTs out of the available 63400 on the FPGA chip.

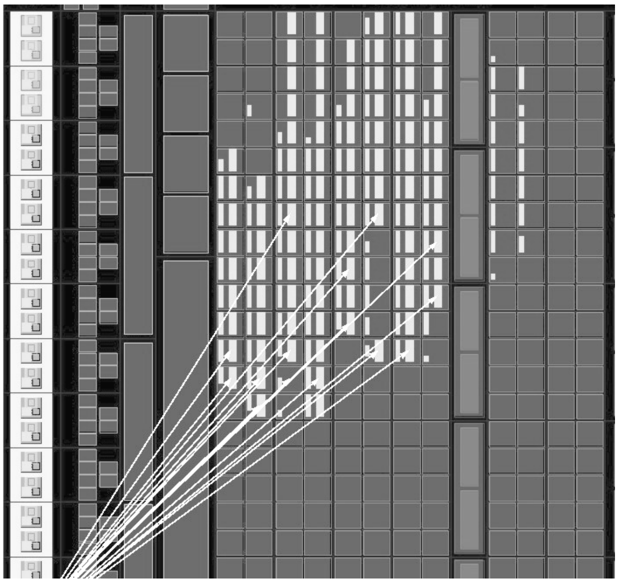


Fig. 6. Implementation of a 64-bit Brown matrix multiplier on a FPGA chip in Vivado CAD

Table 5 shows the results of synthesizing a 64-bit Brown matrix multiplier on an Artix-7 FPGA using classical and advanced components of single-bit HA and FA.

Table 3

Results of synthesis of a 64-bit Brown matrix multiplier on FPGA

System characteristics	Brown's matrix multiplier is based on classical HA and FA	Brown's matrix multiplier is based on advanced HA and FA
Hardware cost (Number of LUTs)	674	376
Performance (Frequency, MHz)	132	365

From the results shown in Table 3, we can see that the hardware costs for implementing a 64-bit Brown matrix multiplier based on advanced HA and FA are about 1.8 times less than the cost of implementing such a multiplier on classical HA and FA, and the speed is about 2.8 times faster.

VI. CONCLUSION

In this paper, a matrix multiplier of Brown's binary numbers was developed, which is a component of arithmetic and logic devices of vector and scalar processors. A review of hardware methods of binary number multiplication was carried out and it was determined that matrix and tree multipliers have high speed of implementation of this operation. The system characteristics of the implementation of such multipliers on classical and advanced components of half and full binary adders were analyzed. The design and modeling of a 64-bit Brown matrix multiplier using the VHDL hardware description language were considered. When synthesizing such a multiplier on FPGAs in Vivado CAD, a 1.8-fold reduction in hardware complexity and a 2.8-fold increase in performance were obtained compared to the use of the classical element base of HA and FA, which was confirmed by theoretical calculations.

References

[1] P. Kumar, N. S. Bhandari, L. Bhargav, R. Rathi and S. C. Yadav (2017). Design of low power and area efficient half adder using pass transistor and comparison of various performance parameters, *International Conference on Computing, Communication and Automation (ICCCA)*, pp. 1477–1482. DOI: https://doi.org/10.1109/CCAA.2017.8230033.

[2] Kogut I., Hryha V., Dzundza B., Hryha L., Hatala I. (2024). Research and Design of Multibit Binary Adders on FPGA, *Advances in Cyber-Physical Cystems*, Lviv. Vol. 9(2), 108–114. DOI: https://doi.org/10.23939/acps2024.02.108.

[3] A. Anand Kumar (2016). Fundamentals of Digital Circuits. *PHI Learning Private Limited Delhi-110092*, 4th edition, 1070 p.

[4] Y. Nyckolaychuk, V. Hryha, N. Vozna, A. Voronych, A. Segin, P. Humennyi (2022). High-performance coprocessors for arithmetic and logic operations of multi-bit cores for vector and scalar supercomputers. *Advanced Computer Information Technologies. 12th International Conference. ACIT 2022*. Spišská Kapitula, Slovakia, 410–414. DOI: https://doi.org/10.1109/ACIT54803.2022.9913198

[5] Y. Nykolaychuk, V. Hryha, N. Vozna, I. Pituhk, L. Hryha (2023). High-Performance Components of Hardware Multi-Bit Specific Processors for the Addition and Multiplication of Binary Numbers. *Computer systems and information technologies*, (2), 25–32. DOI: https://doi.org/10.31891/csit-2023-2-3.

[6] Nykolaychuk, Y., Vozna, N., Davletova, A., Pitukh, I., Zastavnyy, O., & Hryha, V. (2021, September). Microelectronic Structures of Arithmetic Logic Unit Components. In *2021 11th International Conference on Advanced Computer Information Technologies (ACIT)* (pp. 682–685). IEEE. DOI: https://doi.org/10.1109/ACIT52158.2021.9548512.

[7] Liang, C., Su, L., Wu, J., & Xiong, J. (2016, October). An innovative Booth algorithm. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (pp. 1711–1715). IEEE. DOI: https://doi.org/10.1109/IMCEC.2016.7867510.

[8] Sharma, A., Mittal, A., Singh, A., & Kapoor, R. (2024, July). Design and Implementation of Braun Multiplier using Verilog. In *2024 Asia Pacific Conference on Innovation in Technology (APCIT)* (pp. 1–6). IEEE. DOI: https://doi.org/10.1109/APCIT62007.2024.10673483.

[9] Asif, S., & Kong, Y. (2015, December). Analysis of different architectures of counter based Wallace multipliers. In *2015 Tenth International Conference on Computer Engineering & Systems (ICCES)* (pp. 139–144). IEEE. DOI: https://doi.org/10.1109/ICCES.2015.7393034.

[10] devi Ykuntam, Y., Pavani, K., & Saladi, K. (2020, July). Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs. In *2020 11th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1–6). IEEE. DOI: https://doi.org/10.1109/ICCCNT49239.2020.9225404.

[11] Poornima, H. S., Nagaraju, C., & Yadav, S. S. (2022, December). Synthesis and Simulation of A Low-Power, High-Efficiency and Effective Dadda Multiplier. In *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)* (pp. 1–6). IEEE. DOI: https://doi.org/10.1109/ICERECT56837.2022.10059592.

[12] Gangwar, P., Gupta, R., & Kaur, G. (2021, September). A Design Technique for Delay and Power Efficient Dadda-Multiplier. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 66–71). IEEE. DOI: https://doi.org/10.1109/ICIRCA51532.2021.9544746

[13] Single-bit half-adder (2017) Patent of Ukraine. No. 115861. https://sis.nipo.gov.ua/uk/search/detail/805118/

[14] Single-bit full-adder (2022) Patent of Ukraine. No. 150330. https://sis.nipo.gov.ua/uk/search/detail/1676025/

[15] Amano, H. (Ed.). (2018). *Principles and structures of FPGAs*. Springer. 231 p.

Igor Kohut, D. Sc, was born in Zelene village of Husiatyn district, Ternopil region, in 1953. He graduated from Lviv Polytechnic Institute in 1967. Ph.D. Solid State Electronics at Lviv Polytechnic Institute (1993), and his Doctor of Technical Sciences Degree in Electrical and Computer Engineering at Lviv Polytechnic University in 2010. He is the author of more than 250 scientific papers, including 50 copyright certificates for inventions and patents of Ukraine, 1 monograph and textbooks. Research interests: research and development of advanced element base for computing, specialized computer systems for control systems, sensor electronics, microsystems-on-a-chip based on advanced microelectronic technologies of large and ultra-large integrated circuits with silicon-on-insulator structures, including their three-dimensional architectures

Volodymyr Hryha, PhD, was born in Nadvirna, Ivano-Frankivsk region in 1982. Engineering of Computer Systems. Lviv Polytechnic University, Ukraine (2004). He received Ph. D. in Technical (Computer) Sciences in 2015 at Ternopil National Economic University. He holds the title of Associate Professor of Department of Computer Engineering and Electronics of Faculty of Physics and Technology of Vasyl Stefanyk Precarpathian National University. He is the author of more than 150 papers (of which 25 patents of Ukraine). His main Areas of Research Interest are the design of specialized computer systems, FPGA, spatial-temporal graphs, Hardware Design Language, Programmable systems on the Chip, Internet of Things.

Bogdan Dzundza, D. Sc., was born in Bodnariv village, Kalush district, Ivano-Frankivsk region in 1982. 2004–2007 – postgraduate studies at Vasyl Stefanyk Precarpathian National University, specialty Solid State Chemistry. In 2008 he successfully defended his thesis and received a PhD. He is co-author of ten textbooks, two monographs, more than 130 publications in professional scientific journals, and has participated in many international scientific conferences. Research interests: studying the physical properties of semiconductor thin films and nanostructures (pure and doped materials, solid solutions, and multicomponent semiconductor compounds), thermoelectric and photovoltaic semiconductor materials development of automated laboratory research systems computer science, microcontroller systems and the Internet of Things.

Liudmyla Hryha, was born in Shepetivka, Khmelnytskyi region, in 1986. Engineer in the field of electronics and telecommunications. Lviv Polytechnic University, Ukraine (2008). Her main Areas of Research Interest are the design of specialized computer systems, databases, components computer system, sensor electronics, Programmable systems on the Chip, Internet of Things.

Iryna Hatala was born in Ivano-Frankivsk. She is a PhD student in the Physics and Astronomy Department at Wayne State University. Over the past three years, Iryna Hatala has worked as a graduate teaching assistant for five different classes within the Physics and Astronomy Department, including three of the introductory physics courses for undergraduate students. Research interests: physics and astronomy, condensed matter physics, studying the physical properties of semiconductor thin films and nanostructures (pure and doped materials, solid solutions, and multicomponent semiconductor compounds).