ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 10, No. 2, 2025

SYNTHESIS OF MULTI-BIT PYRAMIDAL ADDERS ON FPGA

Volodymyr Hryha[1], Nataliia Vozna[2], Lesya Mychuda[3], Lidliia Shtaier[4], Iryna Hatala[5]

[1]*Vasyl Stefanyk Carpathian National University, 57, Shevchenko Str., Ivano-Frankivsk, 76018, Ukraine,*
[2]*West Ukrainian National University, 11, Lvivska Str., Ternopil, 46009, Ukraine,*
[3]*Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine,*
[4]*Ivano-Frankivsk National Technical University of Oil and Gas, 15, Karpatska Str., Ivano-Frankivsk, 76019, Ukraine,*
[5] *Wayne State University, 42, Warren Ave, Detroit, 48202, USA.*
Authors' e-mails: *volodymyr.hryha@cnu.edu.ua, nvozna@ukr.net, lesia.z.mychuda@lpnu.ua, lidiia.shtaier@nung.edu.ua, iryna.hatala@gmail.com*

https://doi.org/10.23939/acps2025.02.135

Submitted on 11.09.2025

***Abstract*: The paper analyses the system characteristics and functional capabilities of multi-bit pyramidal adders that can be used in the structures of discrete perceptron of modern neural networks for summing weight coefficients and input signals. The methodology for designing multi-bit adders using flow and spatial-temporal graphs is described. Algorithmic and recursive pyramidal adders have been developed and their main system characteristics have been determined. Software models of multi-bit algorithmic and recursive pyramid adders have been developed using the VHDL hardware description language. Functional modelling and synthesis of developed structures of multi-bit pyramidal adders on FPGA were performed. It has been established that when using multi-bit numbers (n=1024), the hardware complexity of recursive pyramidal adders is reduced by 64 times.**

***Index terms*: synthesis, perceptron, neural network, half-adder, discrete signals, functional modelling, FPGA.**

I. INTRODUCTION

The binary addition operation is most often used in computational algorithm structures to perform arithmetic and special operations [1]. In addition, at the current stage of development of computer systems, their integration with artificial neural networks is observed [2]. Artificial neural networks (ANNs) are built on the basis of perceptron's, which contain adders for calculating weighted sums of input signals [3]. Therefore, the intensive development of ANNs is driven by the need to find possible ways to reduce their hardware complexity, in particular the addition operation.

The use of specialized operational devices in neural network component structures will minimize hardware costs and ensure high-speed execution of the relevant algorithm and its functions [4].

Multi-bit pyramid adders are widely used components of computing devices, arithmetic logic units (ALUs), matrix and multi-layer multipliers, coprocessors, and artificial neural networks, thanks to the parallel execution of incomplete binary addition operations [5,6].

An important criterion for their effectiveness is ensuring minimal hardware complexity and maximum speed with minimal delay of sum bit formation signals and carry-through transfers [7]. A key feature of the use of pyramidal adders in computer systems and artificial neural networks is the uniformity of binary incomplete addition operations in their structure.

Therefore, constructing structures for such adders that are optimal in terms of hardware complexity is important and can be achieved using spatial-temporal graphs [8, 9].

Thus, the relevant scientific and applied task is to develop hardware structures of multi-bit pyramidal adders based on spatial-temporal graph (STG) and to study their system characteristics.

II. LITERATURE REVIEW AND PROBLEM STATEMENT

The widespread use of specialized computer systems in computing plays an important role in solving a specific algorithm or class of algorithms, as it achieves high system performance and reliability at low equipment costs for its implementation and increases the possibility of implementing this system on a FPGA crystal [10]. High data processing performance in specialized devices is achieved by approximating the structure of operating devices to the structure of the flow graph of the executed algorithm using spatial parallelism.

The works [11-14] show that algorithmic operational devices (AODs) are best suited to the flow graph structure of the executed algorithm. Their design is based on information about the graph structure of the executed algorithm, taking into account equipment cost constraints and performance requirements. When using flow graphs of an algorithm, it is possible to construct a single-cycle stream-type AOD and a multi-cycle pipeline-type AOD. Although stream graphs of executable algorithms almost exactly correspond to the structure of AODs, in many cases, from the point of view of hardware resources (the

size of the integrated circuit crystal) or non-critical time parameters, it is advisable to implement the calculation of several operations using a single computing element. In this case, the design process becomes ambiguous, since the correspondence between the structure of the algorithm graph and the structure of the AOD disappears.

The works [8] demonstrate the application of spatial-temporal graphs, the main purpose of which is to find the optimal balance between equipment costs and AOD productivity. In this case, data processing performance will be achieved by approximating the structure of the operating device to the structure of the spatial-temporal graph of the executed algorithm using spatial and temporal parallelism. When using spatial-temporal graphs of the algorithm, it is possible to construct various types of multi-tact AODs depending on the degree of compression of the stream graphs of the executed algorithms.

The paper [15] develops a vertical-parallel method for sorting one-dimensional arrays of numbers. A hardware implementation structure using flow graphs has been developed. Synthesis on FPGA has been performed.

In the work [16], a vertically parallel method and structures for calculating maximum and minimum values in one-dimensional and two-dimensional arrays were developed. A hardware implementation structure using flow graphs was developed and its system characteristics were investigated.

Of particular interest are recursive spatial-temporal graphs [8,9], which are constructed by combining the vertices of the algorithm's flow graph by height and width. The result is a single vertex that sequentially performs all identical or similar operations of the algorithm in time. The use of this type of spatial-temporal graph significantly reduces the hardware complexity of the algorithm structure.

III. SCOPE OF WORK AND OBJECTIVES

The development of components for computer systems and artificial neural networks is a pressing task in the field of information technology, driven by the emergence of a new class of complete and incomplete binary adders with minimal hardware complexity and maximum speed of sum formation and carry-through [5,7,10]. At the same time, an important task is to develop structures of multi-bit adders' devices based on spatial-temporal graphs with minimal hardware complexity.

The main objective of this work is to design algorithmic and recursive structures of multi-digit pyramidal adders using stream and spatial-temporal graphs and to study their system characteristics [8].

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 Carpathian National University in 2024-2026.

IV. RESEARCH AND DESIGN OF VARIOS TYPES OF PYRAMIDAL ADDERS

Fig. 1 shows the structure of an 8-bit pyramidal algorithmic adder (AD). This pyramidal adder is based on half adders [5].

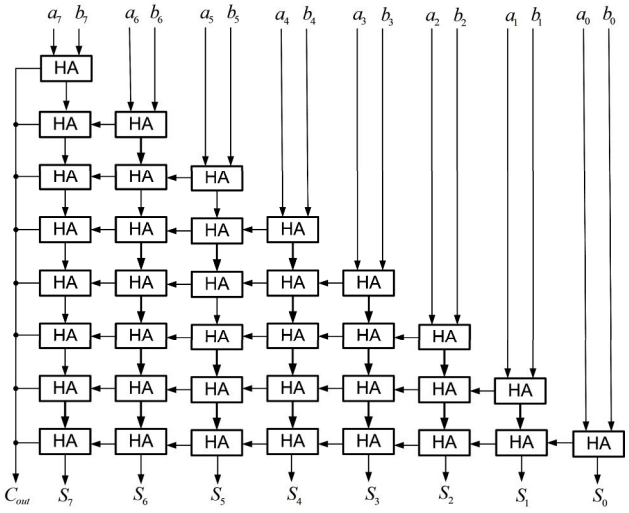

Fig. 1. Structure of a 4-bit pyramid adder built on the basis of a half-adder

The hardware complexity of this type of pyramid binary adders is calculated according to the following formula:

$$A_{AD} = \frac{m^2 + m}{2} \times A_{HA}\,, \qquad (1)$$

where, A_{HA}- hardware complexity of a half-adder, m – bit-length of the input data.

Another improved variant of the binary half-adder structure based on «AND», «OR», «NAND» logic gates has been developed in a related research paper [5,17], minimaxing hardware and time complexity characteristics (Fig. 2).

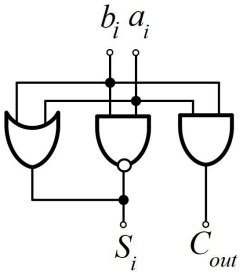

Fig. 2. Structure of half adder based on «AND», «OR», «NAND» logic gates

The hardware complexity of a previously developed binary half-adder will be A_{HA}=3 (gates), while the sum time complexity and output transfer is minimal and amounts to 1 microcycle.

When using improved half-adders

$$(A_{HA}=3)\ A_{AD} = \frac{8^2 + 8}{2} \times 3 = 108\,(gates)\,.$$

The time complexity of this type of pyramid-like multi-bit binary adder is calculated according to the following formula:

$$t_{AD} = m \times t_{HA}, \qquad (2)$$

where, t_{HA}- time complexity of a half-adder, m – bit-length of the input data.

When using improved half-adders (t_{HA}=1) $t_{AD} = 8 \times 1 = 8\,(microcycles)$.

The mathematical basis for representing the algorithmic structure of a pyramidal adder is a flow graph of the algorithm. Identifying parallelism and even controlling it, thereby ensuring the possibility of finding compromise spatial-temporal relationships, which is essential when choosing the structure of a computing device, is made possible by representing the stream graph of the algorithm in a tiered-parallel form [8,9]. In a tiered-parallel form of a stream graph, all vertices of one tier depend on the results of the previous tier and do not depend on the vertices of subsequent tiers.

Fig. 3 shows the tiered-parallel form of the flow graph of the binary number addition algorithm for a pyramidal 8-bit adder.

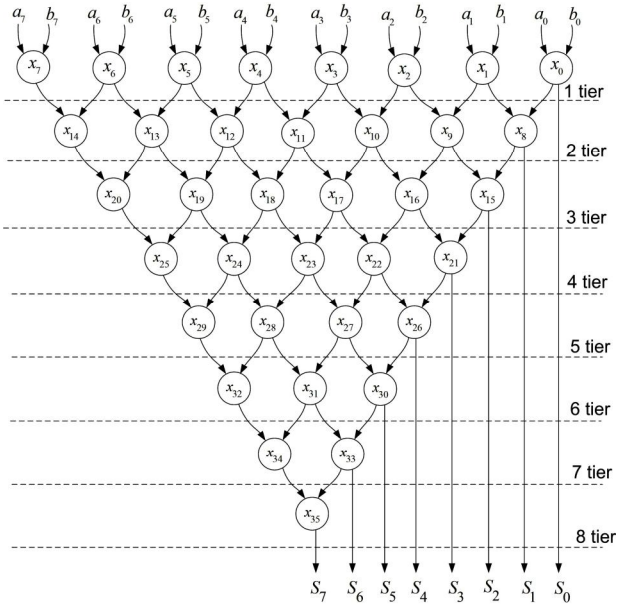

Fig. 3. Structure tiered-parallel form of the flow graph of the pyramid adder algorithm

Fig. 3 shows the vertices of the graph (x_0,…,x_{35}) that act as functional operators of the summation algorithm, namely, they sum the input data a_i and b_i.

The tiered-parallel form determines the degree of parallelism of the graph (the maximum number of vertices on one layer, h=8), as well as the minimum possible computation time of this algorithm (the number of layers, k=8).

Spatial-temporal graphs are used to construct recursive devices [8]. To obtain a recursive spatial-temporal graph, it is necessary to convert the flow graph of the algorithm into a recursive STG, taking into account that similar summation operations can be combined into one in terms of the height and width of the graph.

Fig. 4 shows a recursive spatial-temporal graph of binary number addition for an 8-bit pyramidal adder.

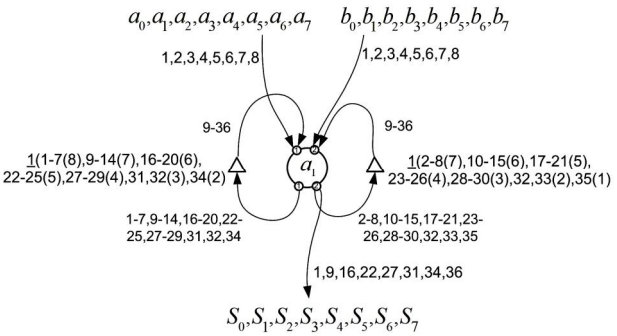

Fig. 4. Recursive spatial-temporal graph of the pyramid adder algorithm

This STG contains one vertex, which sequentially performs all 36 identical operations of incomplete binary addition of an 8-bit pyramidal adder.

The input 8-bit data (a_0,…,a_7) and (b_0,…,b_7) are fed to the input ports (1,2) of the graph vertex (a1). The intermediate results at the outputs of the output ports (1,2) are fed to the delay elements, which coordinate the simultaneous execution of the corresponding intermediate results by the vertex of the recursive STG. The output results (S_0,…,S_7) are formed at different points in time at the graph output.

Fig. 5 shows the structure of a recursive device (RD) for adding binary data on a pyramidal 8-bit adder.

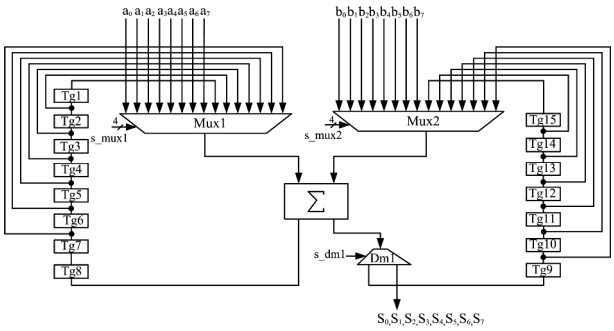

Fig. 5. Recursive device for adding binary data

The recursive binary data summing device contains two 15-input multiplexers (Mux1, Mux2), one incomplete binary adder (Fig. 2), a demultiplexer with 2 outputs, and 15 flip-flops. Multiplexers are used to feed input data and intermediate results to the inputs of the HA. The demultiplexer sends the intermediate results to the inputs of the flip-flops for delay, and the final summation results are formed at one of its outputs.

The hardware complexity of a recursive addition device is calculated according to the following formula:

$$A_{RD} = 2A_{Mux} + A_{HA} + A_{Dmux} + A_{Trig}, \qquad (3)$$

where, A_{Mux}- hardware complexity of a multiplexer, A_{HA}- hardware complexity of a half-adder, A_{Dmux} - hardware

complexity of a demultiplexer, A_{Drig} - hardware complexity of a triggers.

The hardware complexity of the recursive addition device (Fig. 5) will be: $A_{RD} = (2 \times 75) + 3 + 4 + (2 \times 15) = 187\,(gates)$.

The time complexity of a recursive addition device is calculated according to the following formula:

$$t_{RD} = (t_{Mux} + t_{HA} + t_{Dmux} + t_{Trig}) \times (m^2 + m)/2\,, \qquad (4)$$

where, t_{Mux} - time complexity of a multiplexer, t_{HA} - time complexity of a half-adder, t_{Dmux} - time complexity of a demultiplexer, t_{Trig} - time complexity of a triggers, m – bit-length of the input data.

The time complexity of the recursive addition device (Fig. 5) will be: $t_{RD} = (5 + 1 + 2 + 2) \times 36 = 360\,(microcycles)$.

Table 1 shows the results of hardware complexity for algorithmic and recursive binary data addition devices.

Table 1

Results of calculating hardware complexity for algorithmic and recursive devices

Adder bit-length	Hadrware complexity AD (A_{AD})	Hadrware complexity RD (A_{RD})
8	108	187
16	408	379
32	1584	763
64	6240	1531
128	24768	3067
256	98688	6139
512	393984	9331
1024	1574400	24571

As it can be seen from Table 1, the hardware complexity of a recursive addition device starting from 64-bit precision decreases by 4, 8, 16, 32 and 64 times compared to an algorithmic addition device.

Table 2 shows the results of time complexity for algorithmic and recursive binary data addition devices.

Table 2

Results of calculating time complexity for algorithmic and recursive devices

Adder bit-length	Time complexity AD (t_{AD})	Time complexity RD (t_{RD})
8	8	360
16	16	1360
32	32	5280
64	64	20800
128	128	82560
256	256	328960
512	512	1313280
1024	1024	5248000

As it can be seen from Table 2, the time complexity of a recursive summing device starting from 64-bit precision increases by 2-3 orders of magnitude compared to an algorithmic summing device.

In order to find the optimal balance between equipment costs and the speed of pyramid-type summing devices, it will be necessary in the future to investigate combined and parallel-serial types of such summing devices based on STG.

V. MODELLING AND SYNTHESIS OF DEVELOPED ADDERS ON FPGA

The development of an algorithmic and recursive addition device was carried out using the VHDL hardware description language in the Active HDL SE integrated environment.

Fig. 6 shows a functional simulation diagram of an 8-bit recursive pyramid-type addition device.

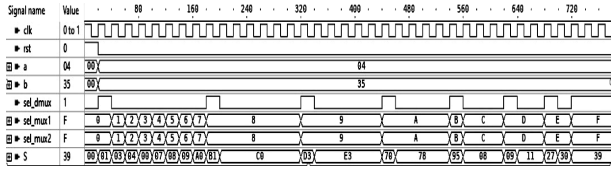

Fig. 6. Functional diagram of an 8-bit recursive pyramid-type summing device simulation

The diagram shows the input of 8-bit binary values to inputs (a and b). Signals (sel_mux1, sel_mux2, sel_dmux) are used to control two multiplexers and one demultiplexer. The output (S) generates an 8-bit sum result at 36 micro cycles.

The synthesis of algorithmic and recursive devices was performed on the Artix-7 family FPGA crystal XC7a100Tcsg324-1 from Xilinx [18,19].

Fig. 7 shows an enlarged view of the section of the FPGA crystal on which the structure of an 8-bit recursive pyramid-type summing device was implemented in the Vivado Design Suite CAD system.

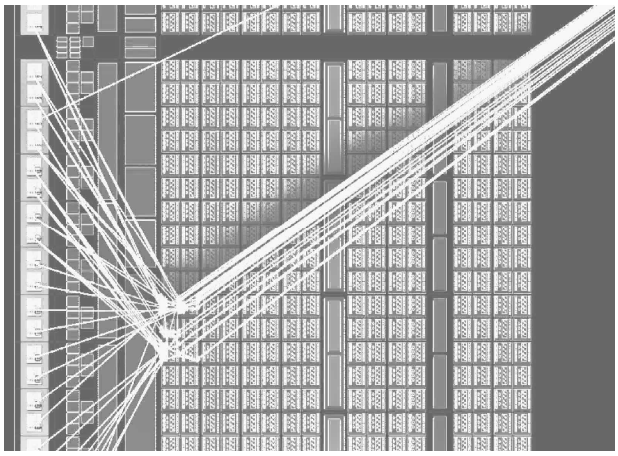

Fig. 7. Implementation of an 8-bit recursive pyramid-type adder on a FPGA crystal

Table 3 presents the results of synthesizing 64-bit algorithmic and recursive pyramid-type addition devices on Artix-7 family FPGAs.

Table 3

Results of the synthesis of algorithmic and recursive adders on FPGA

Adder bit-length	Algorithmic adder (LUT usage)	Recursive adder (LUT usage)
64	1250	317

From the results shown, we can see that the implementation of a 64-bit algorithmic addition device requires approximately four times more hardware than the implementation of a recursive addition device.

VI. CONCLUSION

The paper describes the main areas of application of multi-bit binary pyramid-type adders, which are components of modern neural networks, graphics accelerators, and arithmetic-logic units of superscalar and vector processors. The internal structure of digital logic elements and the main system characteristics of improved incomplete binary adders, which are components of pyramidal adders, are presented. Using the methodology of spatial-temporal graphs, a recursive summing device was constructed, which had significantly lower hardware complexity. Design and modelling of a 64-bit recursive pyramid-type summing device using the VHDL hardware description language in the Active-HDL package was completed. When synthesizing a recursive pyramid-type summing device on a PLDS in the Vivado CAD system, a 4-fold reduction in hardware complexity was achieved compared to an algorithmic device, which is confirmed by theoretical calculations.

VII. CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

VIII. DECLARATION ON GENERATIVE AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Allemang, R., Avitabile, R. (2022). Applied Digital Signal Processing. Handbook of Experimental Structural Dynamics. *Springer New York,* NY, 1426. DOI: https://doi.org/10.1007/978-1-4614-4547-0.

[2] A. Anand Kumar. (2016). Fundamentals of Digital Circuits. *PHI Learning Pvt. Ltd* , 4 edition, 1070.

[3] Ramanaiah, K., & Sridhar. S., (2015). Hardware Implementation of Artificial Neural Networks. *i-managers Journal on Embedded Systems, 3(4),* 31-34. DOI: https://doi.org/10.26634/jes.3.4.3514.

[4] Jagtap, A.D., Kawaguchi, K., & Karniadakis (2019). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *J. Comput. Phys., 404*, 109-136. DOI: https://doi.org/10.1016/j.jcp.2019.109136.

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

[6] Kogut I., Hryha V., Dzundza B., Hryha L., Hatala I. (2025) Research and Design of Matrix Multiplier on FPGA. *Advances in Cyber-Physical Systems*, 10(1), 10-15. DOI: https://doi.org/10.23939/acps2025.01.010.

[7] Nykolaychuk, Y., Vozna, N., Davletova, A., Pitukh, I., Zastavnyy, O., and Hryha, V. (2021) Microelectronic Structures of Arithmetic Logic Unit Components. *11th International Conference on Advanced Computer Information Technologies (ACIT'2021)*, Deggendorf, Germany, 682-685. DOI: 10.1109/ACIT52158.2021.9548512.

[8] Hryha, V., Dzundza, B., Melnychuk, S., Manuliak, I., Terletsky, A., & Deichakivskyi, M. (2023). Design of various operating devices for sorting binary data. *Eastern-European Journal of Enterprise Technologies*, *124*(4), 6-18. DOI: https://doi.org/10.15587/1729-4061.2023.285997.

[9] Gryga, V., Nykolaichuk, Y., Voronych, A., Pitukh, I., & Volynskyi, O. (2019, February). Spatial-temporal Transformation of Sorting Algorithm with" Perfect Interleaving". In *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)* (pp. 1-5). IEEE. DOI: https://doi.org/10.1109/CADSM.2019.8779341

[10] Heath Jr, R. W. (2017). *Introduction to wireless digital communication: a signal processing perspective*. Prentice Hall, 464.

[11] Kachhadiya, R. J., Agrawal, Y., & Parekh, R. (2021, August). Implementation of ALU using RTL to GDSII flow and on NEXYS 4 DDR FPGA board. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 481-488). IEEE. DOI: 10.1109/ICESC51422.2021.9532899.

[12] Korolija, N., Milutinović, V., & Furht, B. (2025) Algorithm profiling for architectures with dataflow accelerators. *J Big Data* , 12, 31. DOI: https://doi.org/10.1186/s40537-025-01089-7.

[13] Lee, K., Lee, Y., Raina, A. (2021) Software synthesis from dataflow schedule graphs. *SN Appl. Sci.* 3, 142. DOI: https://doi.org/10.1007/s42452-020-04135-6.

[14] Alotaibi, A., Almutairi, A., Kurdi, H. (2020). OneByOne (OBO): A Fast Sorting Algorithm. *Procedia Computer Science, 175,* 270–277. DOI: https://doi.org/10.1016/j.procs.2020.07.040.

[15] Tsmots, I., Skorokhoda, O., Antoniv, V., & Rabyk, V. (2018, September). Vertically-parallel method and VLSI-structure for sorting of one-dimensional arrays. In *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)* (Vol. 1, pp. 112-116). IEEE. DOI: 10.1109/STC-CSIT.2018.8526598.

[16] Tsmots, I., Rabyk, V., Skorokhoda, O., & Antoniv, V. (2017, February). FPGA implementation of vertically parallel minimum and maximum values determination in array of numbers. In *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)* (pp. 234-236). IEEE. DOI: 10.1109/CADSM.2017.7916123.

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

[18] Hideharu Amano. (2018). Principles and Structures of FPGAs. *Published in Springer Singapore*, 231. DOI: https://doi.org/10.1007/978-981-13-0824-6.

[19] Kobayashi, R., Miura, K., Fujita, N., Boku, T., Amagasa, T. (2022). An Open-source FPGA Library for Data Sorting. *Journal of Information Processing, 30,* 766–777. DOI: https://doi.org/10.2197/ipsjjip.30.76.

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.

Nataliia Vozna was born in Ternopil, Ternopil region in 1975. She graduated from Ternopil Academy of National Economy in 1998. She received Ph.D. in Technical (Computer) Sciences in 2009 at Ternopil National Economic University and DSc in Technical (Computer) Sciences in 2020 at Lviv Polytechnic National University.

From 2005 to 2011, she was an Assistant, then Associated Professor at the Department of Specialized Computer Systems of Ternopil National Economic University. Starting from 2022 – professor of the Department of Specialized Computer Systems of West Ukrainian National University. She is an author of 6 monographies (1 single), more than 150 papers (of which 30 patents of Ukraine). Her research interest includes computer science, structural transformations of problem-oriented data in distributed computer systems, processes of structuring and evaluation of the structural complexity of complex computerized system elements.

Lesya Mychuda She was born in Lviv, Lviv region in 1974. Graduated from Lviv Polytechnic National University in 1996. (Computerized Control and Automation Systems, Systems Engineer). From 2000 to 2004, she was an Assistant, then Associated Professor at the Department of Automation of Thermal and Chemical Processes at Lviv Polytechnic National University. From 2004 to 2018, she was an Assistant, then Associated Professor at the Department of Automation and Computer-Integrated Technologies at Lviv Polytechnic National University. Since 2019, she has been an associate professor at the Department of Information Technology Security at Lviv Polytechnic National University.

Research interests: signal processing, analogue and analogue-to-digital functional converters based on switched capacitors, telecommunications technologies, electrical measurements.

Lidiia Shtaier was born in Nadvirna, Ivano-Frankivsk region in 1982. She is an Associate professor, associate professor of the Department of Information and Telecommunication Technologies and Systems at Ivano-Frankivsk National Technical University of Oil and Gas. In 2004, she graduated with honors from Ivano-Frankivsk National Technical University of Oil and Gas with a degree in Control and Automation Systems and received a Master's degree. In 2010, she defended her candidate's thesis in the specialty 05.11.13 "Instruments and Methods for Control and Determination of Substance Composition" at Ivano-Frankivsk National Technical University of Oil and Gas and received the degree of candidate of technical sciences. In 2013 at Ivano-Frankivsk National Technical University of Oil and Gas, she received the academic title of associate professor of the Department of Computer Technologies in Control and Automation Systems. Her research interests include: mathematical modeling and analysis of technical and natural processes; information technologies in engineering; signal processing; development of intelligent systems; design and development technologies of software systems

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).