Development and Research of Conveyor Structures of Binary Number Sorting Algorithms

Volodymyr Gryga[1], Yaroslav Nykolaichuk[2], Nataliia Vozna[2], Artur Voronych[3], Boris Krulikovskyi[4]

1. Department of Computer Engineering and Electronics, Vasyl Stefanyk Precarpathian National University, UKRAINE, Ivano-Frankivsk, 57 Shevchenko str., email: v.dr_2000@ukr.net

2. Department of Specialized Computer Systems, Ternopil National Economic University, UKRAINE, Ternopil, 8 Chekhova str., email: ya.nykolaichuk@tneu.edu.ua, n.vozna@ukr.net

3. Department of Computer Systems and Networks, Ivano-Frankivsk National Technical University of Oil and Gas, UKRAINE, Ivano-Frankivsk, 15 Karpatska str., email: a.voronych@it.nung.edu.ua

4. Department of Computer Engineering, National University of Water Management and Nature Resources Use, UKRAINE, Rivne, 11 Soborna str., email: kboris@ukr.net

Abstract: **The characteristics of the structures complexity of conveyor sorting devices, constructed on the basis of parallel algorithms for sorting binary arrays are analyzed. The improved structure of the conveyor sorting device, which compared to the known one had 1.9 times less hardware complexity and 1.5 times higher performance, was developed and researched. Modeling of VHDL models of conveyor sorting devices was carried out, their synthesis and implementation was made in the Xilinx FPGA by automated designing Vivado system applications. The received practical results of the complexity characteristic of the developed conveyor devices coincide with theoretical calculations. There are also determined the relevant areas of application of the developed devices.**

Keywords: **special device for sorting arrays of data, conveyor sorting device, basic sorting elements, comparison scheme.**

I. Introduction

Sorting is one of the common problems of data processing and it is generally understood as a problem of placing elements of disordered set of data sets values in order of monotonic increase or decrease [1]. The sorting operation takes on average 25% of machine time [2] and it is most often used in the tasks of digital processing of signals and images, similar to convolution operations, fast Fourier transform, and others [3]. We know many methods of sequential and parallel sorting of binary numbers [1-9]. There is a large number of algorithms to organize sorting in modern digital signal processors, and each of these algorithms has its advantages and disadvantages. Performing a software sorting operation is time-consuming and generally used to perform consistent data sorting methods. Particularly effective is the parallel performance of algorithm sorting operations using hardware methods that significantly accelerate the execution time of the algorithm. Such methods of parallel sorting algorithms, in which the sequence of executed operations only depends on the number of input data, are the following: the Batcher's method [2], the modified "bubbles" method [1], the "pairwise-odd" permutation method, the fusion method and other [3].

If we use the hardware method, then there is ensured implementation of the binary number sorting operation in real time and the possibility of applying new circuit design solutions with the use of a modern element base with an orientation towards implementation in the FPGA form.

The implementation of highly efficient conveyor devices for sorting binary arrays requires extensive use of a modern element base, the improvement of existing and the development of new methods, algorithms and device structures.

In this regard, an important t ask is to develop efficient structures for conveyor sorting devices (CSDs), which will improve the time characteristics of multithreaded data processing. The implementation of such devices in the languages of the hardware description and their synthesis in FPGA will enable the designer to choose the optimal hardware cost and time-efficient characteristics of the structure of the CSD.

II. Conveyor Structures Analysis of Parallel Algorithms for Binary Numbers Sorting

The conveyor processing principle involves the alignment in time operators of the algorithms sorting on different data [5-8].

Conveyor structures of sorting devices of binary arrays is developed and analyzed below. They are based on the known parallel sorting algorithms presented in a graphical form, and their hardware and time complexity are estimated. (Their system characteristics)

The hardware implementation of the known parallel sorting algorithms in the conveyor structures involves the full reflection of their flow graphs algorithm (tiered-parallel algorithm forms) [4-8,10,11,13] into the structure of the operating device. The vertices of the graph (functional operators) will correspond to the hardware block (operation) and the arcs will correspond to lines for the transmission of input data of intermediate and final results. The conveyor registers are placed on the lines that connect the operating

units of one tier with the operating units of the previous tier.

Tier-parallel algorithm forms determines the degree of parallelism of the graph (the maximum number of vertices in a single tier or the width of the graph), as well as the minimum-possible time of calculation of the given algorithm (number of tiers or height of the graph) [6-8].

It is shown the structure of the CSD in Fig. 1, which is based on the numbers sorting by modified “bubbles” sorting method [1,13].

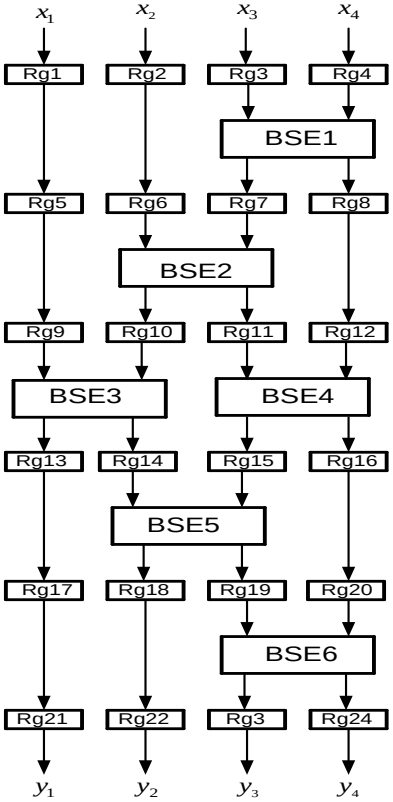

Fig.1. The structure of Conveyor sorting device for 4 values by the modified “bubbles” sorting method.

The basis for the above-presented structure of conveyor sorting device is the basic sorting elements (BSE) and conveyor register (Rg).

The structure of this conveyor device consists of the same basic sorting operations (max/min). CSD structure (Fig. 1) contains $N(N\text{-}1)/2$ basic sorting operations and $N(2N\text{-}3)$ conveyor registers for input N values.

The time complexity of this conveyor device structure is determined by the critical distribution of the signal through $(2N\text{-}3)$ basic operations of sorting and $(2N\text{-}3)+1$ conveyor registers.

It is shown the structure of the CSD in Fig. 2, which is constructed on the basis of sorting algorithm by "pairwise-odd" permutation method [2,13].

CSD structure (Fig. 2) contains $N(N\text{-}1)/2$ basic sorting operations and $(N \times N)$ conveyor registers for N input values.

The time complexity of this conveyor device structure is determined by the critical distribution of the signal through N basic operations of sorting and $(N+1)$ conveyor registers.

It is shown the CSD structure in Fig. 3, which is constructed on the basis of the sorting algorithm graph by Batcher’s method [3,13].

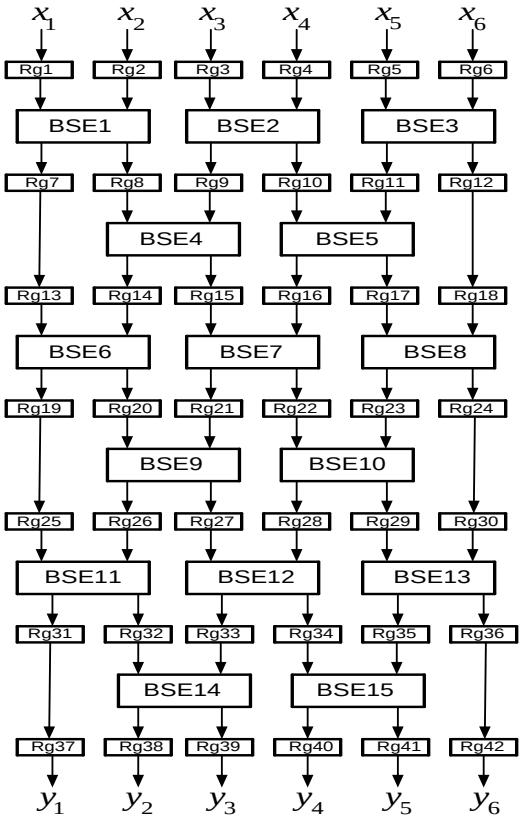

Fig.2. The structure of Conveyor sorting device for 6 values by "pairwise-odd" permutation method.

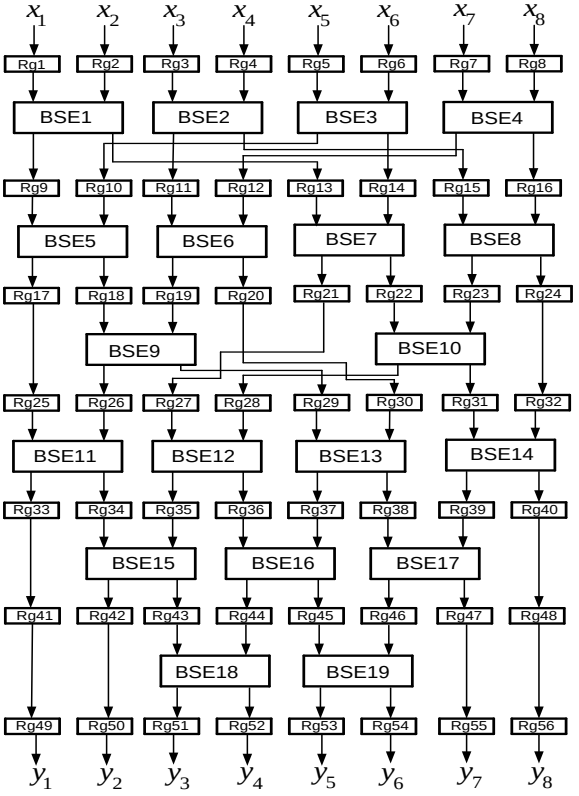

Fig.3. The structure of Conveyor sorting device for 8 values by Batcher`s method.

CSD structure (Fig. 3) contains $0,48N\ln^2 N$ basic sorting

operations and $\frac{1}{2}(\log_2 N)([\log_2 N+1])\times N$ conveyor registers for *N* input values.

The time complexity of this conveyor device structure is determined by the critical distribution of the signal through $\frac{1}{2}(\log_2 N)(\log_2 N+1)$ basic operations of sorting and (*N*-1) conveyor registers.

The conveyor structures of the sorting devices (Fig. 1-3) consist of the same type of basic sorting elements that compare two numbers and more of them are sent to the first output and less to the second output. The internal structure of sorting basic element is presented in Fig. 4.

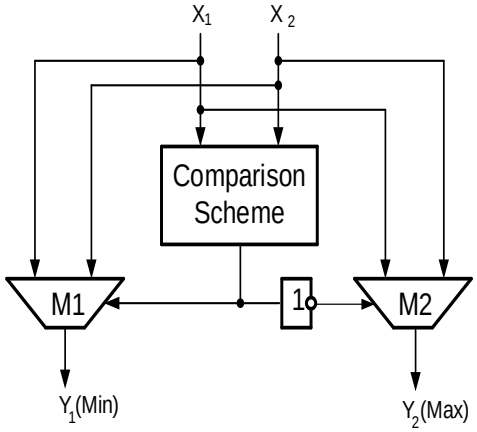

Fig.4. Internal structure of the basic sorting element

When comparing two numbers (X_1, X_2) in the comparison scheme "for more" ($X_1 > X_2$), the output of the scheme is formed a comparison sign, the direct value of which controls the multiplexer M1, to issue a smaller number to the output (Y_1) and the inverse value controls the multiplexer M2 to issue a larger number at the output (Y_2) [5,13].

The internal structure of the comparison scheme can be constructed on the basis of logical elements [5] or accelerated carry adders [13].

The calculation of the hardware and time complexity of different comparison schemes and 2-input multiplexers is given in paper [6].

With the account to these data, the hardware complexity of the CSD by the modified "bubbles" method for N = 8 and n= 4 equals:

$$A_{CCSD} = N\times(N\text{-}1)/2\times(\mathrm{A_{CS}}+2\times\mathrm{A_{MP}})+(\mathrm{n}\times\mathrm{A_{REG}}\times \times L_{REG}) = 28\times(24+56)+(4\times4\times112) = 4032\,(gates),$$

where A_{CCSD} - hardware complexity of classical CSD,

$\mathrm{A_{CS}}$ - hardware complexity of comparison scheme,

$\mathrm{A_{MP}}$ - hardware complexity of multiplexer,

$\mathrm{A_{REG}}$ - hardware complexity of register,

L_{REG} -quantity of conveyor register.

The time complexity of this conveyor device equals:

$$\tau_{CCSD} = (\tau_{CS}+\tau_{MP})+(14\times\tau_{REG}) = = (6+3)+(14\times2) = 37\nu\,(micro-cycles),$$

where τ_{CCSD} - time complexity of classical CSD,

τ_{CS} - time complexity of comparison scheme,

τ_{MP} - time complexity of multiplexer,

τ_{REG} - time complexity of register.

III. Development of an Improved Structure of the Conveyor Sorting Device

The bubble CSD (Fig. 1) can be improved by performing independent base-sorting operations separately for the first and second half of the input data on the two conveyor structures.

Values X_i are already ordered by recession and need to be further applied to $((N/2)^2 - N/2)/2 + N/2$ basic sorting elements and 4*N* conveyor registers.

It is shown an improved CSD for 4-bit binary numbers (n = 4) for 8 input values (N =8) in Fig. 5.

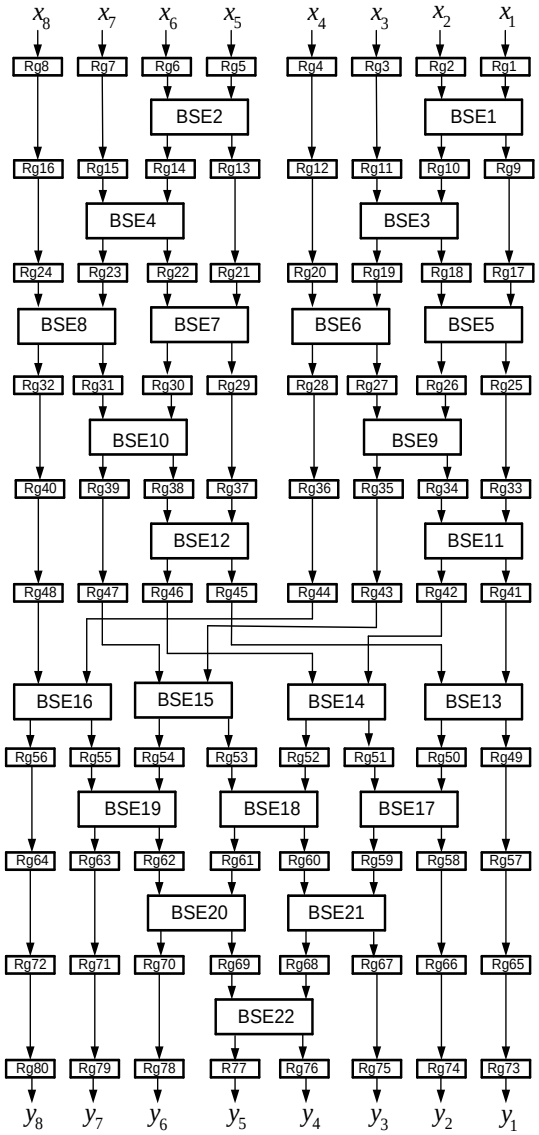

Fig.5. An improved structure of the CSD

The number of basic sorting elements for this CSD for the binary numbers is equal to $3((N/2)^2 - N/2))/2 + N/2$, and

the number of conveyor registers is $6N+4N$ for N input values.

The hardware complexity of an improved CSD for N = 8 and n = 4 is equal to:

$$A_{ICSD} = 22\times(A_{CS}+2\times A_{MP})+(n\times A_{REG}\times L_{REG}) =$$
$$=22\times(16+24)+(4\times4\times80)=2160\,(gates),$$

where A_{ICSD} - hardware complexity of improved CSD,

A_{CS} - hardware complexity of improved comparison scheme [13],

A_{MP} - hardware complexity of improved multiplexer [13],

A_{REG} - hardware complexity of register,

L_{REG} -quantity of conveyor register.

The time complexity of such CSD is equal to:

$$\tau_{ICSD} = (\tau_{CS}+\tau_{MP})+(10\times\tau_{REG}) =$$
$$=(3+2)+(10\times2)=25\nu\,(micro-cycles),$$

where τ_{CCSD} - time complexity of classical CSD;

τ_{CS} - time complexity of comparison scheme;

τ_{MP} - time complexity of multiplexer;

τ_{REG} - time complexity of register.

So, in comparison with the classic device, we get a reduction in the hardware complexity in $K_A = 4032/2160=1{,}9$ times and increase the speed in $K_\tau = 37/25=1{,}5$ time.

IV. Results of the Conveyor Sorting Devices Research

It is shown a dependence graph of logical elements (valves) number on the input data number for the conveyor structures of the known (classical) and improved sorting devices by the modified "bubbles" method.

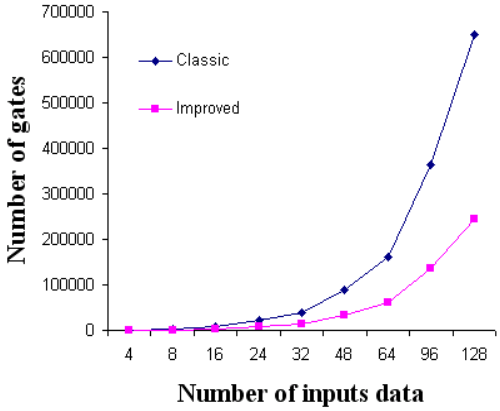

Fig.6. Graph of the dependence of the number of valves on the input data number for the CSDs

The graph shows that the number of logical elements for an improved CSD is 1.9 times less than for a classic CSD.

It is shown a graph of the dependence of the time complexity in Fig. 7 expressed in the microtacts on the input data value for the structures of the known (classical) CSD and the improved CSD with the modified "bubbles" method.

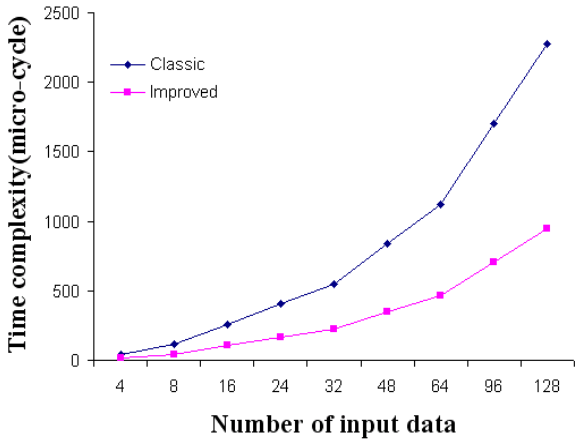

Fig.7. The dependence graph of the time complexity on the number of the input data for the CSDs

It is shown in the graph that the time complexity of an improved CSD requires about 1.5 times less microtacts than the classical CSD.

V. Results of CSD's Simulation and Synthesis on FPGA

Structures of classical and improved CSD for 8 input –one-byte numbers are described in VHDL (Virtual Hardware Description Language). The simulation of the developed CSDs on the functional level was carried out, their RTL circuits were obtained and the Xilinx FPGA synthesis was performed.

It is shown in Fig. 8 a functional diagram of the improved CSD operation for 8 input one-byte numbers.

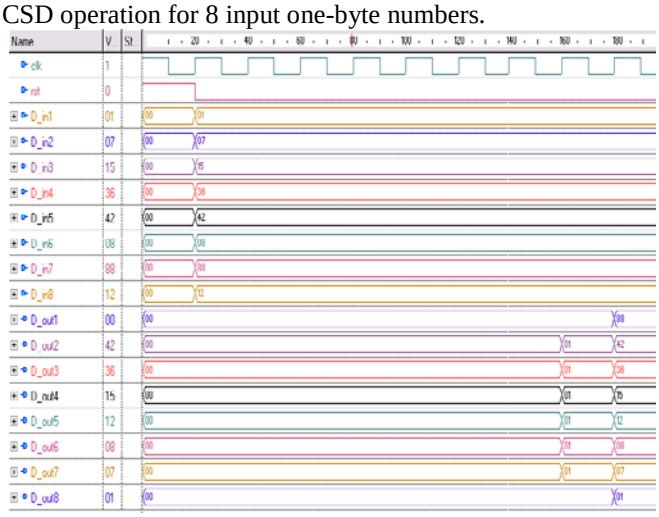

Fig.8. Functional simulation diagram for CSD of 8 one-byte numbers

The diagram shows that an array of unsorted 8-bit values is given at the inputs of the CSD (D_in1, ..., D_in8) on the 1st cycle. Then we get a sorted descending order at the outputs (D_out1, ..., D_out8) at the 9th cycle.

It is shown in Fig. 9 the topology view of improved CSD with the VHDL implementation model on the xc7a100tcsg324-1 crystal (Artix-7 family) of the Xilinx firmware by Vivado CAD.

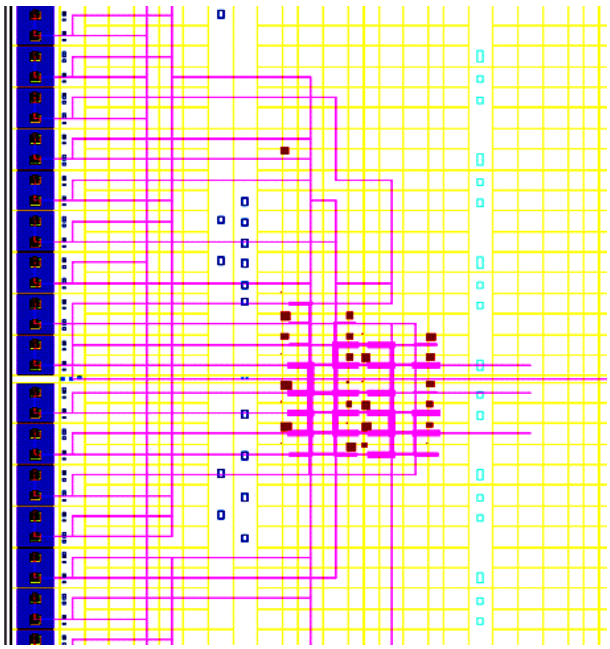

Fig.9. View of the crystal topology of CSD during implementation at the FPGA

Vivado CAD tools placed the implemented VHDL project of the improved CSD almost in the center of the crystal.

Table 1 presents the results of the synthesis of implemented CSDs for sorting 8 one-byte numbers on the FPGA of the Xilinx firmware.

TABLE 1. RESULTS OF THE CSDS SYNTHESIS ON FPGA

	FPGA	Classic CSD		Improved CSD	
		Blocks quantity FPGA (CLB)	Clock frequency (MHz)	Blocks quantity FPGA (CLB)	Clock frequency (MHz)
1	Artix 7	950	109,9	475	174,7

As can be seen from Table 1, experimental results coincide with analytical calculations.

VI. CONCLUSION

During the research of CSDs it is determined that they are widely used in multithreaded data processing, and are often used in digital processing of signals, images and sorting networks for the rapid transfer of large data arrays.

The improved structure of the CSD of binary arrays by the modified "bubble" method was developed, the hardware and time complexity calculation was performed.

As a result of the comparison with the classical CSD for binary number array by the modified bubble method it is received decreasing the cost of the equipment in 1.9 times and increasing of speed in 1.5 times, which is confirmed by the results of the practical implementation of the Xilinx FPGA.

REFERENCES

[1] D. Knuth, The art of computer programming, *V.3: Transl. from eng.* M.: *"Peace"*, pp. 841, 1978.

[2] Thomas Braunl,. Parallel Programming: *Transl. from german* ,K.: *Vyscha Shkola*., pp. 358, 1997.

[3] S.Y. Kung, VLSI array processors. *Trans. from English* - M.: *"Peace"*, pp. 672, 1991.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. Introduction to algorithms, *3rd edition, Massachusetts Institute of Technology*, pp.1313, 2009.

[5] A. Melnyk, Memory with ordered access: monograph, Lviv: *Press Lviv Polytechnic*, pp. 296, 2014.

[6] A. Melnyk, Computer Architecture, Lutsk:*Regional publisher*, pp. 470, 2008.

[7] V. Voevodin, Parallel Computing. Textbook for high schools, *BHV:Peter*, pp. 608, 2002.

[8] Andrew S. Tanenbaum Computer Architecture Austin. 6th ed., St. Petersburg: *"Peter"*, pp. 816, 2013.

[9] O. Kehret, A. Walz and A. Sikora, "Integration of hardware security modules into a deeply embedded TLS stack" *International Journal of Computing, vol. 15, issue 1*, pp. 24-32, 2016.

[10] V. Zadiraka, Ya. Nykolaichuk. Methods of effective protection of information flows, Ternopil: *Terno-graf*, pp.308, 2014.

[11] S. Melnychuk, A. Voronych, L. Nykolaichuk and B. Krulikovskyi "The Structure and Components of Embedded Special Processors for Determination of Entropy Signals and Random Massages" *Perspective technologies and methods in MEMS design. Proceedings of XIIIth International Conference. MEMSTECH 2017*, Lviv-Svalyava, Ukraine, pp.81-84, February, 2017.

[12] B. Krulikovskyi, A. Davletova, V. Gryga and Y. Nykolaichuk "Synthesis of Components of High Performance Special Proccessors of Execution of Arithmetic and Logical Operations Dsts Proccessing in Theoretical and Numerical Basis Rademacher", *Proceedings of XIVth International Conference. CADSM'2017*, Lviv-Poljana, Ukraine, pp. 214-217, February,2017.

[13] V. Gryga, Y. Nykolaichuk, N. Vozna and B. Krulikovskyi "Synthesis of a microelectronic structure of a specialized processor for sorting an array of binary numbers" *Perspective technologies and methods in MEMS design. Proceedings of XIIIth International Conference. MEMSTECH 2017*, Lviv-Svalyava, Ukraine, pp.170-173, April, 2017.

[14] V. Gryga, I. Kolosov and O. Danyluk "The development of a fast iterative algorithm structure of cosine transform" The Experience of Designing and Application of CAD System in Microelectronics. *Proceedings of XIIIth International Conference. TCSET'2016*, Lviv-Poljana, Ukraine, pp.506-509, February,2016.

[15] R. Dunets and V. Gryga "Spatio-temporal synthesis of transformation matrix of reverse fast cosine transformation", *Proceedings of XIIIth International Conference. CADSM'2015*, Lviv-Poljana, Ukraine, pp.45-49, February, 2015.