ISSN 1063-7397, Russian Microelectronics, 2023, Vol. 52, No. 7, pp. 641–647. © Pleiades Publishing, Ltd., 2023.
Russian Text © The Author(s), 2022, published in Izvestiya Vysshikh Uchebnykh Zavedenii, Elektronika, 2022, Vol. 27, No. 3, pp. 356–366.

Hardware Implementation of the Digital Signal Processing Algorithms in Recurrent Signal Processor on FPGA

Yu. A. Stepchenkov[a], *, N. V. Morozov[a], Yu. G. Diachenko[a], D. V. Khilko[a], D. Yu. Stepchenkov[a], and Yu. I. Shikunov[a]

[a] *Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia*
**e-mail: YStepchenkov@ipiran.ru*

Received February 4, 2022; revised February 11, 2022; accepted May 4, 2022

Abstract—Dataflow architecture is an alternative to traditional von Neumann computing architecture. However, known variants of dataflow architecture have a range of serious problems with no effective solutions up to the present day. This paper represents Hybrid Recurrent Signal Processor's (HRSP) hardware verification results. It describes HRSP's register transfer level model implementing its architectural specification and hardware prototype on the HAN Pilot Platform development board with Intel Arria10 field-programmable gate array. HRSP consists of a von Neumann master processor on a control layer and a recurrent dataflow unit on an operational layer. Dataflow unit includes four computing cores. HRSP's hardware model combines either software or hardware implementation of the control processor and the hardware model of the operational layer. Testing the HRSP's hardware prototype on the development board using an isolated word recognizer (IWR) as a typical data processing application has proven that the hardware model is bit-exact with both HRSP's imitation model and the original IWR C++ model. The HRSP's hardware prototype's achieved performance ensures IWR's operation in real-time mode on the development board. It is slightly better than the performance of the TMSC55x (Texas Instruments) digital signal processor. Verification of the HRSP's hardware implementation on synthetic tests showed that its average performance is 5% higher than the performance of the DSP TMSC55x digital signal processor. The results of the proposed optimization of hardware support for Fast Fourier Transform (FFT) in HRSP prove that such an optimization speeds up the FFT calculation, significantly reduces the capsule size, reduces the required hardware resources and simplifies FFT scaling.

Keywords: recurrent signal processor, hybrid dataflow architecture, hardware model, FPGA, fast Fourier transform, FFT

DOI: 10.1134/S106373972307017X

INTRODUCTION

Dataflow computing architectures (DFCAs) [1, 2] are an alternative to the traditional computing von Neumann architecture. However, the following problems have not yet been solved in the existing versions of DFCA: the implementation of recursions, loops (cycles) and iterations; work with constants, etc. An effective DFCA implementation is based on the use of large associative memory, which is advisable only for mass parallelism systems [1].

Paper [3] states that the known dataflow processors require 2–3 times more instructions to execute the program compared to the traditional architecture processors. Paper [2] notes that the orientation of developers towards the prevailing design methodology (the "collection" paradigm) ultimately led to the loss of the competitive advantages of DFCA.

Attempts to use the dataflow paradigm in the field of digital signal processing also have a long history [4]. The principles of DFCA and the requirements of digital signal processing algorithms work well together in applications that are characterized by a high degree of internal parallelism. The main limiting reason for the wide practical application of such integration is the cost factor, which makes it inadvisable to directly use solutions from the field of massively parallel DFCA systems [4, 5].

The work [6] considers the implementation of the principles of recurrence and self-sufficiency of data by combining the actual data and necessary information in one operand: the performed operation code, and the obtained result destination. The set of such operands forms a capsule designed to execute a certain algorithm similar to the program code for the von Neumann architecture. It is a compressed flow graph of the computational process. When a capsule is executed in DFCA, the operands are subjected to a recurrent involution, which is determined both by the original operand forms and operation logic of recurrent converters as DFCA hardware's part.

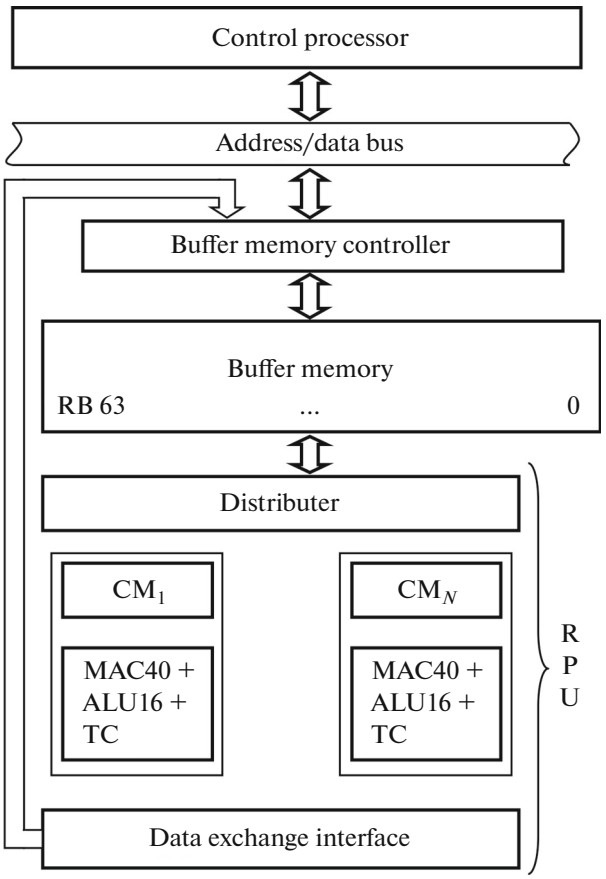

Fig. 1. HRSP hardware model structure (CM—coincidence memory; RB—ready bits; TC—tag converter; RPU—recurrent processing unit).

The hybrid recurrent signal processor (HRSP) [5, 7] developed at the Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences (Moscow) is one of the DFCA family representatives. It does not require a large amount of associative memory. For the development and debugging of capsules that implement specific algorithms for digital signal processing in HRSP, its simulation model in C# [8] and the hardware model in VHDL [9] are used. The first facilitates and speeds up the process of developing and debugging capsules, and the second ensures the HRSP implementation's equivalency on the development board with FPGA and allows one to evaluate the digital signal processing algorithms implementation effectiveness on the HRSP in real time. The experimental operation of the HRSP on a subset of typical digital signal processing algorithms has demonstrated the high efficiency of using the HRSP for solving digital signal processing problems, which provides greater performance compared to advanced digital processors [9]. At the same time, it identified the need to expand hardware support for some typical digital signal processing algorithms, in particular the fast Fourier transform (FFT) and a number of digital filters.

This paper describes the features of the HRSP architecture implemented on the HAN Pilot Platform development board with the Intel Arria10 SoC FPGA [10], considers its modernization in order to expand hardware support for typical digital signal processing algorithms, discusses the technical characteristics of the FPGA-based HRSP and test results.

HRSP HARDWARE MODEL

The hardware model of the HRSP includes a control processor and a recurrent signal processor, represented by a buffer memory controller, a buffer memory, a distributor, a data exchange interface, and N parallel computing sections (Fig. 1). Each computing section includes a coincidence memory and a calculator, which includes a multiplier with a 40-bit accumulator (MAC40) and a hardware shift device to the right and left by 1–15 bits, a 16-bit arithmetic logic unit (ALU16) and tag converter. Together with the distributor and the data exchange interface, the computing sections constitute a recurrent processing unit. The buffer memory controller contains an arbiter of accesses to the buffer memory from the control processor (read and write) and the recurrent processing unit (write).

Any type of processor can execute the role of the control processor in the HRSP architecture. The main requirement of its characteristics is enough performance to ensure the required speed and nature of processing the intermediate results of the recurrent processing unit and writing data to the recurrent signal processor. In the current implementation, the NIOS-II software processor integrated into the project at the stage of logical synthesis in the Quartus Prime SE 18 CAD system [11] or the ARM Cortex-A9 processor implemented in FPGA hardware performs the control processor function.

The buffer memory consists of four memory banks with a total capacity of 64 bits. A separate register stores the readiness bits of the capsule operands, allowing the reading of operands from the buffer memory, the banks of which are dual-port RAM. The first port is used to write data to the buffer memory from the control processor and the recurrent processor, as well as to read data from the buffer memory to the control processor. It is driven by the address/data bus system frequency. The second port is used to read operands from the buffer memory to the recurrent processor and is controlled by the latter's internal frequency.

The recurrent signal processor's functional modules form a ring pipeline. The communication interface sends the intermediate and final capsule processing results to the buffer memory for recursive use.

Table 1. Hardware resources for the HRSP implementation with NIOS-II

Functional units of HRSP	FPGA resources			
	ALM*	registers	memory, bits	DSP
Control level	3641	4048	33788672	4
Buffer memory	15262	10223	141900	0
Distributer	16666	4397	0	0
Computing sections	36826	26788	5248	4
Data exchange interface	2042	1242	0	0
Summary	74440	46698	33935820	8

*ALM—adaptive logic module

HRSP IMPLEMENTATION ON FPGA

The HRSP implementation on FPGA is the only possible (fast and affordable) option for hardware testing of the developed architecture. In principle, it cannot have a higher performance compared to other implementation types (both custom and semi-custom ICs on a gate array chip). However, it is convenient for debugging and allows one to evaluate the HRSP architecture's performance in the recurrent processing unit's clocks. Table 1 presents the hardware resources required for HRSP implementation on the Intel Arria10 SoC 10AS066K3F40E2SG FPGA in the automatic synthesis mode using the Quartus Prime SE 18.0 CAD system [6] with NIOS-II software processor as the control processor. Hardware costs are defined in terms of adaptive logic modules (ALM), registers, memory bits, and digital signal processor (DSP) blocks.

Testing the hardware model. The HRSP testing goals on the development board are as follows:

— Hardware model verification for compliance with the simulation model by comparing the obtained results;

— HRSP performance comparison against the modern DSP performance on standard data processing algorithms.

To verify the hardware model, a typical application of a DSP was chosen—an isolated word recognizer (IWR). The run of capsules generated for all algorithms of the IWR according to the initial data, extracted from the test pronunciations of 100 English words, in the simulation and hardware models showed a complete coincidence of all intermediate results. The total recognition accuracy (95%), shown by the prototype, coincided with the recognition accuracy provided by the original C++ IWR version on a personal computer.

Table 2 shows an estimate of the HRSP performance compared to the DSP TMSC55x from Texas Instruments when executing synthetic test algorithms [12, 13], which are typical for a wide class of digital signal processing problems and similar to tests from the set [14]. At the same time, the DSP TMSC55x's performance is taken as a reference. Data analysis shows that the HRSP performs filtering algorithms (with the exception of the adaptive filter) and some others better than DSP TMSC55x. For example, HRSP calculates filters with coefficient value $K_c = 6$ and filter section value $K_s = 3$ 1.14–1.50 times faster than DSP TMSC55x. However, for some tasks, the HRSP works 1.6–3.2 times slower than the DSP TMSC55x. This is due to the presence of hardware support for the corresponding algorithms in the DSP TMSC55x, which significantly speeds up their execution. This shortcoming can be eliminated by expanding the HRSP's functionality, in particular, by implementing wider hardware support for superscalar operations in the computing sections; increasing the number of working registers in the calculator; introducing mechanisms for fast loading of data into the registers of computer blocks, bypassing some stages of pipeline processing.

Expansion of hardware support for typical digital signal processing algorithms. The current version implements a 256-point FFT algorithm (Radix-2) in a fixed-point format with results written in place of the input data and has the following features:

— Four 16-bit input data are packed into one operand, the imaginary and real parts are stored in different sections of the capsule in a bit-reversed order;

— Turning coefficients are fully stored in the sectional constant memory in each computing section;

— A special instruction Butt, introduced into the instruction set, provides a four-cycle calculation of the Radix-2 "butterfly."

Figure 2 shows the execution of the Butt instruction in the steady state. The efficiency of such an FFT calculation using one HRSP's section is comparable to the DSP TMSC55x. But it can be improved through hardware optimization.

The main problems of hardware support for the FFT in the HRSP are the following:

(1) Excess overhead for storing turning coefficients;

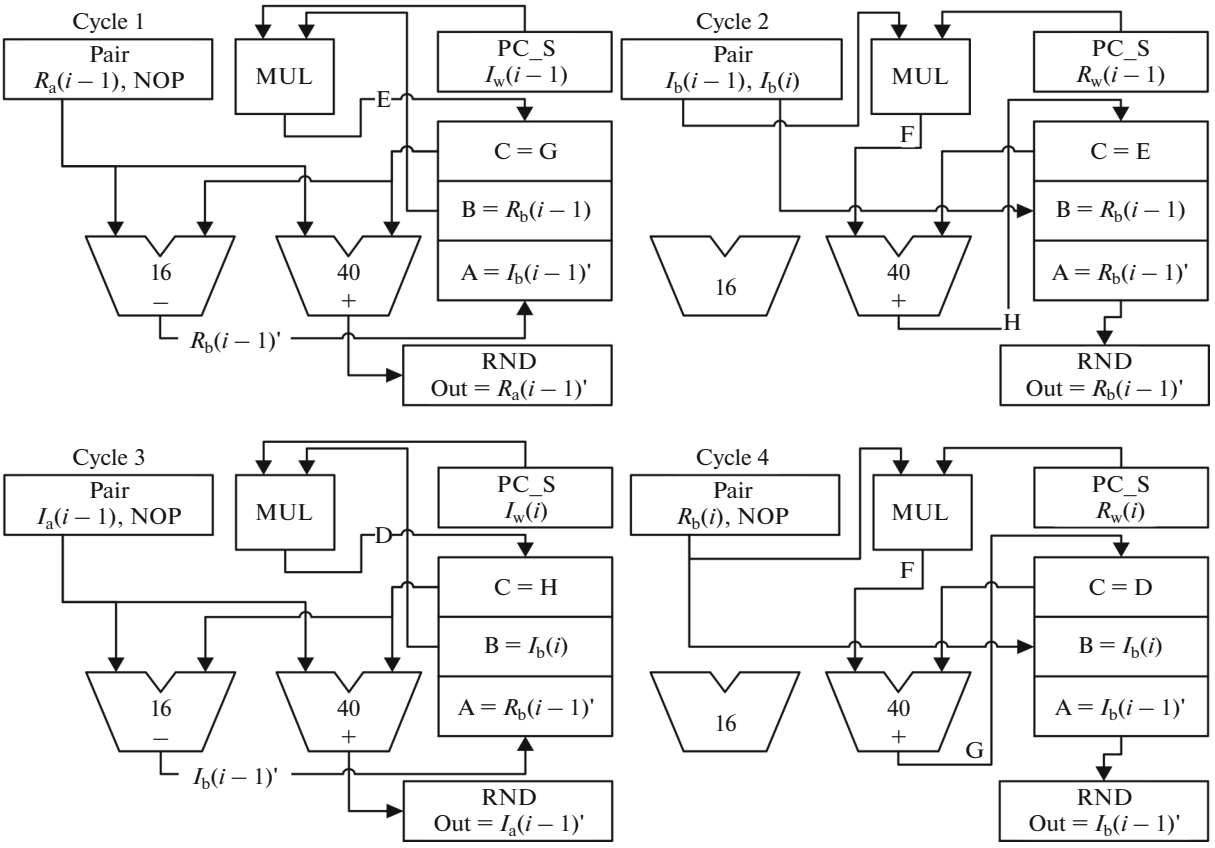

Fig. 2. Current Butt instruction algorithmic diagram.

(2) Storing of samples in a capsule in packed operands;

(3) Non-cyclic mode for reading the real and imaginary sample's parts;

(4) Loop 2 of the Butt instruction requires a special mode of unpacking and distribution of packed data;

(5) Changing the FFT order requires the capsule to be rewritten.

Table 2. Performance features of HRSP and DSP TMSC55x on synthetic tests

Algorithm		DSP TMSC55x	HRSP
Finite impulse response filter	for real data	1	$1 + (1/(1 + K_c))$
	for single samples	1	1
	for complex data	1	$1 + (3/(1 + K_c))$
Least root mean square adaptive filter		1	$1 - (K_c - 1)/(4 + 3K_c)$
Bisquare filter with infinite impulse response		1	$1 + (3 + 10K_s)/(4 + 21K_s)$
Sum of bitwise products of two vectors		1	1
Bitwise sum of two vectors		1	1.50
Finding the maximum in a vector		1	0.60
Viterbi decoder		1	0.31
256-point FFT		1	1.25

K_c is the number of filter coefficients; K_s is the number of filter sections.

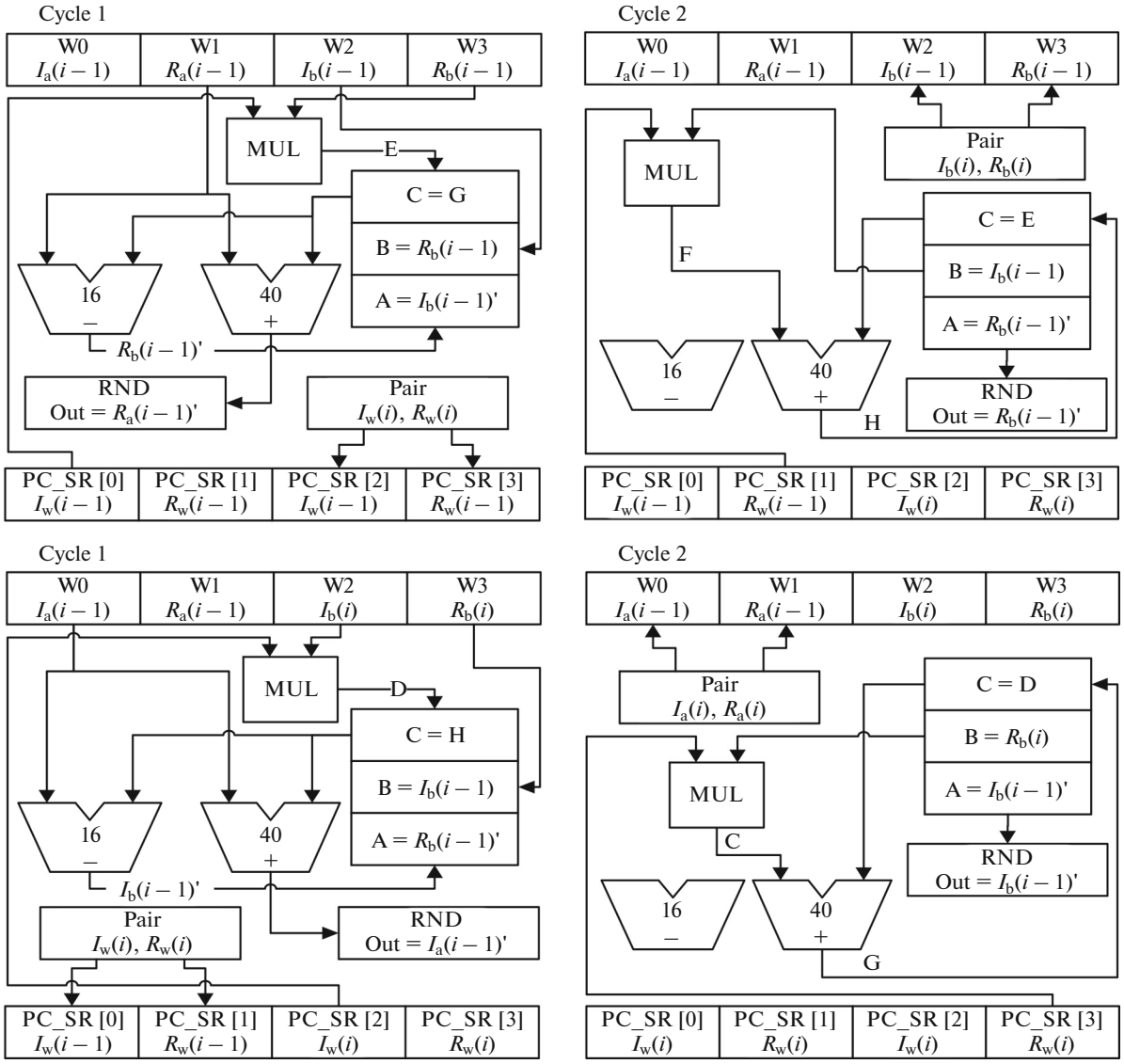

Fig. 3. Advanced Butt instruction algorithmic diagram

An analysis of the FFT hardware support's current version indicates a large degree of hardware redundancy. There are two main directions of architecture optimization: Butt instruction optimization and buffer memory modification.

Figure 3 shows the new Butt instruction's four-cycle implementation that solves problems 3 and 4 mentioned above by adding two registers W2 and W3 to the section calculator, as well as register memory for four 16-bit constants. The buffer memory includes autonomous FFT memory banks for storing real and imaginary values of samples and FFT coefficients, and a control unit for this memory, which provides address generation and read-write operation of its contents. The features of the new algorithm are as follows:

— Banks of real and imaginary values are addressed by one address;

— The addresses of the very first "butterflies" do not depend on the FFT size, so they can be set during the hardware initialization;

— The addresses of the previous "butterfly's" components are stored for subsequent writing the "butterfly" calculation results to them;

— The read values of samples and coefficients are combined to the packed operands and transferred to the recurrent signal processor in the general way.

Table 3 shows the results of comparing old and new versions of the 256-point FFT implementation. The new version is invariant to the FFT dimension, requires less memory and reduces the capsule length by almost 6 times.

Thus, the implemented HRSP architecture modification requires significantly less hardware resources

Table 3. Comparison of two FFT implementation cases on HRSP

Parameter	Old version	New version	
Supported FFT, point number	256	8 through 256	8 through 1024
Coefficient memory, number of 16-bit words	256 × 4 = 1024	256	1024
Samplememory, number of 16-bit words	528	512	
Capsule size, operand number	155	23	
Benchmark, cycle number	1024	1024	
Overhead, cycle number	44	16	
Summary, cycle number	1068	1040	

with the same size of supported FFT, is more productive and scalable when executing the FFT algorithm.

CONCLUSIONS

The hardware VHDL model of the HRSP ensured the correctness of its solutions on a prototype hardware based on the HAN Pilot Platform development board, which proved the effectiveness of its architecture for tasks that allow parallel calculations. In synthetic tests, the HRSP showed an average performance of 5% higher than the DSP TMSC55x. Testing the hardware implementation of the HRSP on the isolated word recognizer as a typical DSP application has proved its bit-exactness with the HRSP simulation model and the original C++ recognizer model.

The results obtained prove the success in optimizing the FFT hardware support in the HRSP. The new version of the tools is both more flexible and faster, and reduces the capsule's size.

ACKNOWLEDGMENTS

The materials of the article were reported at the X All-Russian scientific and technical conference with international participation "Problems in the development of advanced micro- and nanoelectronic systems" (MES-2021) (March 1–November 1, 2021, Moscow, Zelenograd).

FUNDING

The work was supported by the state task no. 0063-2019-0010.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Burtsev, V.S., *Parallelizm vychislitel'nykh protsessov i razvitie arkhitektury superEVM. Sb. statei* (Parallelism of Computing Processes and Development of the Supercomputer's Architecture), Moscow: Torus Press, 2006.
2. Klimov, A.V., Levchenko, N.N., Okunev, A.S., and Stempkovsky, A.L., The application and implementation issues of dataflow computing system, *Probl. Razrab. Perspektivnykh Mikro- Nanoelektronnykh Sistem*, 2016, no. 2, pp. 100–106.
3. Dikarev, N.I., Shabanov, B.M., and Shmelev, A.S., The use of fine-grained parallelism in dataflow processor, *Probl. Razrab. Perspektivnykh Mikro- Nanoelektronnykh Sistem*, 2016, no. 2, pp. 144–150.
4. Sundararajan, S., Minimizing communication and synchronization overhead in multiprocessors for digital signal processing, *PhD Dissertation*, Berkeley: Calif.: Univ. of California, 1995.
5. Stepchenkov, Yu.A., D'yachenko, Yu.G., Khil'ko, D.V., and Petrukhin, V.S., Recurrent data-flow architecture: Features and realization problems, *Probl. Razrab. Perspektivnykh Mikro- Nanoelektronnykh Sistem*, 2016, no. 2, pp. 120–127.
6. Khil'ko, D.V., Stepchenkov, Yu.A., Shikunov, D.I., and Shikunov, Yu.I., Recurrent data-flow architecture: Technical aspects of implementation and modeling results, *Probl. Razrab. Perspektivnykh Mikro- Nanoelektronnykh Sistem*, 2016, no. 2, pp. 128–135.
7. Shikunov, Yu., Stepchenkov, Yu., and Khilko, D., Recurrent mechanism developments in the data-flow computer architecture, *2018 IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, Moscow, 2018, IEEE, 2018, pp. 1413–1418. https://doi.org/10.1109/EIConRus.2018.8317362
8. Khil'ko, D.V., Stepchenkov, Yu.A., Shikunov, Yu.I., and Orlov, G.A., Development of capsule programming tools for recurrent data-flow architecture, *Probl. Razrab. Perspektivnykh Mikro- Nanoelektronnykh Sistem*, 2018, no. 3, pp. 2–9. https://doi.org/10.31114/2078-7707-2018-3-2-9
9. Stepchenkov, Yu.A., Morozov, N.V., Diachenko, Yu.G., Khilko, D.V., Stepchenkov, D.Yu., and Shikunov, Yu.I., Hardware verification of the recurrent signal processor on FPGA, *Probl. Razrab. Perspektivnykh Mikro- Nanoelektronnykh Sistem*, 2021, no. 2, pp. 77–82. https://doi.org/10.31114/2078-7707-2021-2-77-82

10. HAN pilot platform, specifications, terasIC. https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=216&No=1133&PartNo=2. Cited March 28, 2022.

11. Intel Quartus Prime software, Intel. https://www.intel.ru/content/www/ru/ru/software/programmable/quartus-prime/download.html. Cited March 28, 2022.

12. Stepchenkov, Yu.A., Khilko, D.V., Shikunov, Yu.I., and Orlov, G.A., DSP filter kernels preliminary benchmarking for recurrent data-flow architecture, *2021 IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, St. Petersburg, 2021, IEEE, 2021, pp. 2040–2044. https://doi.org/10.1109/ElConRus51938.2021.9396594

13. TMS320C55x DSP CPU Reference Guide. Literature Number: SPRU371F. https://www.ti.com/lit/ug/spru371f/spru371f.pdf?ts=1654850280246&ref_url=https%253A%252F%252Fwww.google.ru%252F.

14. Berkeley Design Technology, Inc. The BDTImark2000™: A measure of DSP execution speed, Muhammad Shaaban's home page. http://meseec.ce.rit.edu/eecc722-fall2001/papers/dsp/4/bdtimark2000.pdf. Cited March 28, 2022.