Routing Algorithms in Optimal Degree Four Circulant Networks Based on Relative Addressing: Comparative Analysis for Networks-on-Chip

Emilia A. Monakhova, Oleg G. Monakhov, and Aleksandr Y. Romanov, *Member, IEEE*

***Abstract*—The solution of the problem of organizing optimal communications in circulant networks of degree four is considered. For a family of optimal circulant networks with the minimum diameter and average distance for any number of nodes in a graph, we propose an optimal pair routing algorithm of constant complexity based on using the relative addressing of nodes in a network. The new algorithm is an analytical extension to any number of nodes in the network of the routing method proposed for dense Gaussian networks, and it does not use division operations, which are very expensive to implement in fixed-point format. This extension is based on the proposed scheme of transformations on the plane of geometrical patterns of optimal circulant networks. The developed routing algorithm is the basis for generating the series of routing algorithms for different subfamilies of the optimal two-dimensional circulants. The general routing algorithm and its modification for a separate subclass of circulants are implemented in the HDL NoC model with circulant topology. All the algorithm parameters important for networks-on-chip, including the consumption of memory, logical resources, and execution time are comprehensively investigated. The results of a comparative analysis of the new algorithms with other routing algorithms, previously implemented in the networks-on-chip, are presented.**

***Index Terms*—Graph Theory, Network topology, On-chip interconnection networks, Routing protocols.**

I. Introduction

In this paper, we investigate the solution to the problem of organizing efficient routing algorithms in networks-on-chip (NoCs) [1], [2], [3], [4], [5] with a topology of the class of undirected degree four circulant networks. Circulant networks (graphs) (see the surveys [6], [7], [8], [9]) are a class of regular graphs well-known both in applied practical solutions and in theoretical research (see, for instance, [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]).

Circulant graphs are studied as topologies in multiprocessor systems [6], [7], [8] and large hierarchical networks [10], and also in networks-on-chip [23], [24], [25], [26]. Compared to popular NoC topologies, such as 2D mesh and 2D torus [4], [5], they have a smaller diameter and lower average distance for the same number of nodes and links in the network [10], [23], [25]. Therefore, they are very promising for applications in networks-on-chip. The problem of the organization of an effective communication subsystem (and, primarily, optimal communications in NoCs with a circulant topology) is very acute. There are many routing algorithms developed for circulant networks, but one must take into account the special requirements for organizing communications in NoCs. They are restrictions on the number of connections between routers, on the total number of elements of all routers, and on the chip resources occupied by all routers.

Let us give the basic definitions further used in the paper. An undirected circulant graph $C(N; s_1, \ldots, s_k)$ has a set of vertices $V = Z_N = \{0,\ 1,\ \ldots,\ N-1\}$ and a set of edges $E = \{(v,\ v \pm s_i (mod\ N)) | v \in V,\ i = \overline{1,\ k}\}$, where N is the number of nodes; $S = \{s_1,\ s_2,\ \ldots,\ s_k\}$ is the set of generators, with $1 \leq s_1 < \ldots < s_k < N$; k is the dimension. The degree of a vertex of a circulant is $\delta = 2k$, if $s_k \neq N/2$ or $2k+1$ otherwise. In this paper, we consider a family of optimal degree four circulant graphs (two-dimensional circulants), which have simultaneous a minimum diameter and an average distance, and maximum connectivity. The diameter of a graph is $D = max_{u,\ v \in V} l(u,v)$, where $l(u,v)$ is the length of the shortest path between the vertices u and v. Diameter and average distance evaluate structural delays in a network, as well as its connectivity and survivability [10], [24], [27]. The minimum diameter and average distance are important parameters for topologies of multiprocessor systems with an equal number of nodes and communication lines.

The aim of this paper is to develop an effective routing algorithm for the implementation in a NoC with two-dimensional optimal circulant topology [6], [24]. The mathematical foundations of the proposed algorithm were reported at the conference in [28]. Here this algorithm was finalized, formalized, and implemented in the HDL NoC model. The comparison of the results of operation of the developed algorithm with other routing algorithms is presented.

Manuscript received 26 April 2022; revised 26 August 2022; accepted 28 September 2022. Date of publication 4 October 2022; date of current version 6 January 2023. The research leading to the results obtained by Aleksandr Y. Romanov has received funding from the Basic Research Program at the National Research University Higher School of Economics (HSE University). The research leading to the results obtained by Emilia A. Monakhova and Oleg G. Monakhov was carried out within the framework of state assignment of ICMMG SB RAS N 0251-2021-0005. Recommended for acceptance by Dr. Sudip Misra. *(Corresponding author: Aleksandr Y. Romanov.)*

Emilia A. Monakhova and Oleg G. Monakhov are with the Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch of the Russian Academy of Sciences, 630090 Novosibirsk, Russia (e-mail: emilia@rav.sscc.ru; monakhov@rav.sscc.ru).

Aleksandr Y. Romanov is with Higher School of Economics (HSE University), 101000 Moscow, Russia (e-mail: a.romanov@hse.ru).

Digital Object Identifier 10.1109/TNSE.2022.3211985

II. Background

A. Related Work

The two-dimensional circulants are studied in connection with various practical applications as communication models of complex systems, in graph theory and cryptography. It has been proved in [29] that for any number of vertices, circulant graphs of the form (1) simultaneously have the minimum diameter and average distance coinciding with their exact lower bounds, that is, they are optimal:

Theorem 1. For any integer $N \geq 5$, the optimal circulant is

$$C(N;\ d,\ d+1), \tag{1}$$

where d is the nearest integer to $(-1+\sqrt{2N-1})/2$.

The family of circulants (1) was independently discovered in [30] as a family of graphs of the form $C(N;\ b-1,\ b)$, where $b = \lceil \sqrt{N/2} \rceil$, and called Midimew networks. The family of circulant graphs $C(N_D;\ D,\ D+1)$, where $N_D = 2D^2 + 2D + 1$, was proposed in [31] and called dense Gaussian networks in [26].

The shortest path problem for two-dimensional circulants has been studied in many works, and it remains (as before) actual (see, for example, [32], [33]). There are routing algorithms proposed for two-dimensional circulant networks of general form $C(N;\ s_1, s_2)$ (for example, in [25], [34], [35], [36], [37], [38]) and specialized routing algorithms for graphs of family (1) (in [23], [24], [30], [39]). The algorithms from [23], [24], [25] have been implemented in NoCs.

In [24] an analytical routing algorithm (called Pair exchange algorithm, PEA) was developed to determine the shortest paths in circulants of family (1) between any two nodes given by absolute addressing, i.e., labeled with integers from 0 to $N-1$. It was shown that in the PEA algorithm, the number of operations does not depend on the number of nodes in the network, in contrast to [34] with an estimate of $O(2\log N)$; [35] with an estimate of $O(\sqrt[2]{N})$; [36], [37], and [38] with a common estimate $O(\log N)$; [39] with an estimate of $O(D)$, and also to [40] with quadratic complexity $O(N^2)$. In [24] HDL modeling of a NoC shows that the PEA algorithm has also a number of advantages over the routing algorithms from [23], [25], proposed for a NoC, including less consumption of chip resources. But the PEA algorithm requires the division operations, which are very expensive to implement in fixed-point format.

On the other hand, in [26] and [41], for dense Gaussian graphs, the authors proposed a different type of routing algorithm based on using integer additions and comparisons. Dense Gaussian graphs, for which development of routing algorithms is an urgent problem in recent times (see, for example, [33], [42], [43]), are circulant graphs of family (1) with N_D vertices. The authors of [26], [41] note that routing algorithms of this type do not use the implementation of the Euclidean division algorithm with large computational costs, unlike, for example, the algorithms from [37], [44]. The essence of the considered routing algorithm is to use a set of neighboring zeros in a dense tessellation of graphs on the Z^2 plane to determine the shortest paths between vertices. It should be noted that in [45] the author also applies the method of using neighboring zeros for routing in double-loop graphs with $N = 2D^2$ vertices.

We extended the method from [26], [41] to all graphs of the family (1) with any number of vertices by analytically determining the coordinates of the necessary neighboring zeros. We also developed a routing algorithm on this basis and studied its applicability in NoCs.

The novelty of the study, conducted in this article, is as follows:

- we introduced a relative addressing of network nodes and proposed an optimal adaptive routing for any number of nodes in the optimal family of circulant graphs (1);
- we reduced to $O(\log N)$ the required number of operations, when number of nodes N grows, compared to the routing algorithm from [26];
- with a view to a reduction the consumed memory in a NoC, we developed a routing algorithm for a subclass (special case) of two-dimensional circulants;
- we conducted a comparison of the new routing algorithms with other algorithms, including the PEA algorithm, in terms of the main criteria of suitability for NoCs (consumed memory and logical resources, and operation time).

The rest of the paper is organized as follows. In Section II, we present a family of degree four optimal circulant graphs, as well as a scheme of transformations on the Z^2 plane of the geometric images of its graphs. In Section III, we define the relative addressing of the vertices of graphs of the family (1) in terms of the shortest path vectors and consider some of their properties; we propose an $O(1)$ algorithm for optimal routing in circulant networks using relative node addressing. Section IV contains some estimates of the reduction in the number of operations in the routing algorithm. Also, for a subclass of the family of graphs (1), a modification of the routing algorithm that does not require an additional memory to store parameters during the operation of the algorithm is described. Section V presents the results of implementation of the new routing algorithms in a NoC with optimal circulant topology and their comparison with other routing algorithms implemented in NoCs.

B. A Family of Optimal Degree Four Circulant Networks

The following description of the family of graphs (1) was obtained with the generators as functions of diameter $D > 1$:

$$C(N;\ s_1,\ s_2) = \begin{cases} C(N;\ D-1,\ D), & if\ N_{D-1} < N \leq 2D^2, \\ C(N;\ D,\ D+1), & if\ 2D^2 < N \leq N_D, \end{cases} \tag{2}$$

where $N_D = 2D^2 + 2D + 1$ corresponds to the maximum possible number of vertices of degree four circulant of diameter D.

In this work, we will use the description (2) for graphs of family (1). To understand the structure of graphs of the family (2), we consider their plane tessellation on the Z^2 plane in the

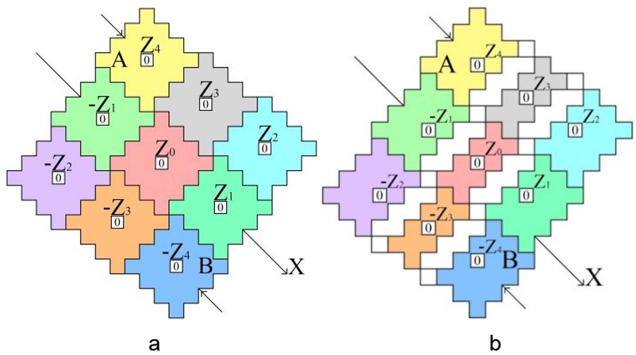

Fig. 1. Plane tessellation of graphs: a) $C(N;\ D,\ D+1)$, $N = N_D$, $D = 3$; b) $C(N;\ D-1,\ D)$, $N = 2D^2$, $D = 3$.

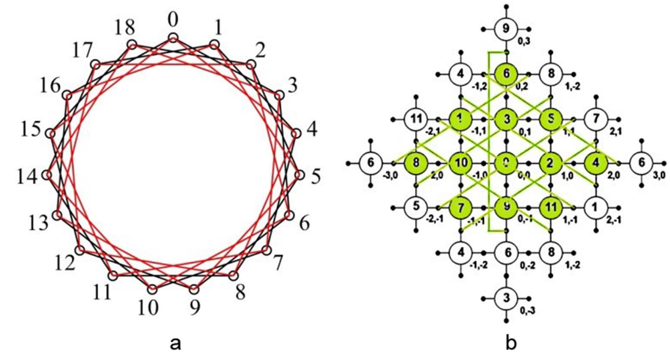

Fig. 2. (a) Circulant graph $C(19;\ 3,\ 4)$: an absolute addressing; (b) A packing on the plane of graph $C(12;\ 2,\ 3)$.

form of a rhomboid configuration of unit squares of an integer lattice. To sequentially obtain all graphs of the family (2), we describe briefly the scheme of transformations on the plane of their geometric representations presented in [28].

According to (2), there are two ranges of N variation: $Q_1(D) = [N_{D-1}+1,\ \ldots,\ 2D^2]$, where $s_1 = D-1$, $s_2 = D$, and $Q_2(D) = [2D^2+1,\ \ldots,\ N_D]$, where $s_1 = D$, $s_2 = D+1$.

The way to obtain optimal graphs of the family (2) with any N is as follows. We successively decrease the number of vertices N in a graph, starting from $N = N_D$ to $N = N_{D-1}+1$. Fig. 1 (a) shows the initial position of plane tessellation on Z^2 of the circulants with $N \in Q_2(D)$ (here, $D = 3$). An initial position for graphs from $Q_1(D)$ is the position shown in Fig. 1(b), which demonstrates an example of the graph $C(18;\ 2,\ 3)$ of diameter $D = 3$ (a fragment is shown; unshaded cells belong simultaneously to adjacent rhombuses). In Fig. 1, a diagram of possible displacements of the corresponding rhombuses on the plane is also presented. The result is achieved by simultaneously moving the packing layers A and B in two opposite directions along the X-axis by an equal number of steps. As a result of the given transformation, any circulant of the family (2) of diameter $D > 1$ and number of nodes $N_{D-1}+1 \le N < N_D$ forms a dense packing of rhombuses on Z^2 with a possible symmetric overlap of vertices of neighboring rhombuses at most on the last two levels of vertices numbered D and $D-1$.

In Fig. 1(a), the coordinates of a central zero at the plane are indicated by $z_0 = (0,\ 0)$; the coordinates of zeros (zero nodes) of neighboring regions with the central rhombus are indicated by $\pm z_i$, $i = \overline{1,\ 4}$. As shown in [26], the neighboring zeros participate in determining the shortest paths between the two vertices of the graph.

III. Study Area

A. Search for the Shortest Paths in Degree Four Circulants

Fig. 2(a) shows an example of a circulant graph of the family (2) with description $C(19;\ 3,\ 4)$ of diameter 3. In Fig. 2 (b), a packing on the plane of graph $C(12;\ 2,\ 3)$ is presented. Here we show two methods of addressing of vertices – absolute and relative – determined through the number of steps (with a "+" or "-" sign) along with generators $s_1 = 2$ and $s_2 = 3$ from the central (zero) vertex. The formula for a translation from relative addressing to the absolute one is presented: absolute number of a vertex m, $0 \le m \le 11$, $m = 2x + 3y (mod\ 12)$, where $(x,\ y)$ is the relative address of m. Missing edges between the vertices, located at distance (diameter) D from vertex 0, are shown in green. The formal definition of relative vertex addressing for circulants is given below.

For two nodes u, v of a circulant graph $C(N;\ s_1,\ s_2)$, we define the shortest path vector from u to v as vector $P(u,\ v) = (p_1,\ p_2)$ having two coordinates, where $|p_k|$, with $k = 1,\ 2$, specifies the number of hops along the edges corresponding to generator s_k (or $-s_k$) in a shortest path from u to v; the "+" ("-") sign determines a movement along s_k (or $-s_k$). Note that the shortest path vector in general can define multiple shortest paths between two nodes at once. The shortest path problem for a graph is formulated as follows: given two nodes u and v of a graph, find a path (paths) between u and v having a minimum length.

Let $D > 1$ be a positive integer, and $(x,\ y)$ and $(x',\ y')$ be the shortest path vectors (or relative addressing from node 0) of two nodes u and v, that is $P(0,\ u) = (x,\ y)$, $P(0,\ v) = (x',\ y')$. In a circulant graph of diameter D we have $|x| + |y| \le D$ and $|x'| + |y'| \le D$. The vector $P(u,\ v) = (x'-x,\ y'-y)$ defines a path from u to v of length $l = |x'-x| + |y'-y|$ called by $l_1 - norm$ [46]. The $l_1 - norm$ has another notation: $\|\ P(u,\ v)\ \| = |x'-x| + |y'-y|$.

But $P(u,\ v) = (x'-x,\ y'-y)$ not always is the shortest path vector in a circulant by closure of connections: if $l > D$, then $P(u,\ v)$ does not define the shortest path from u to v; if $l = D$, then $P(u,\ v)$ can define the shortest path in some cases. The following two theorems allow to reduce the amount of required operations when developing a routing algorithm.

Theorem 2. Let $(x,\ y)$ *and* $(x',\ y')$ be shortest path vectors from vertex 0 to vertices u and v, accordingly, in a graph of the family (2) of diameter $D > 1$. If

$$|x'-x| + |y'-y| < D, \tag{3}$$

then the vector $P(u,\ v) = (x'-x,\ y'-y)$ defines the shortest path from u to v.

Proof: By the symmetry of circulant graphs, if $v - u = v' - u' (mod\ N)$, then $P(u,\ v) = P(u',\ v')$. Therefore,

TABLE I
COORDINATES OF ZEROS FOR CIRCULANTS $\{C(N;\ D,\ D+1) \mid 2D^2+1 \le N \le N_D,\ D\rangle 1\}$

$k = N_D - N$	Zeros of 4 neighboring regions			
	$z_1 = (a_1,\ b_1)$	$z_2 = (a_2,\ b_2)$	$z_3 = (a_3,\ b_3)$	$z_4 = (a_4,\ b_4)$
$0 \le k \le \lfloor D/2 \rfloor$	$(D+1, -D)$	$z_3 + z_1$	$(D+k, D+1-k)$	$z_3 - z_1$
$k = \lfloor D/2 \rfloor + 1$	above	$\emptyset$	above	above
$\lfloor D/2 \rfloor + 1 < k \le D + \lceil D/2 \rceil$	above	$z_3 - 2z_1$	above	above
$k = D + \lceil D/2 \rceil + 1$	above	above	$\emptyset$	above
$D + \lceil D/2 \rceil + 1 < k \le 2D$	above	above	$z_3 - 3z_1$	above

$$P(u,\ v) = \begin{cases} P(0,\ v-u) & \textit{for } v > u, \\ P(0,\ N+v-u) & \textit{for } v < u, \end{cases}$$

that is, $P(u,\ v) = P(0,\ k)$, and the sum of modules of its coordinates is less than D for the vertex k. To all vertices k for which this inequality holds, $P(0,\ k)$ sets the shortest path from 0 to k. Then $P(u,\ v)$, calculated by the formula $P(u,\ v) = P(0,\ v) - P(0,\ u)$, specifies the shortest path between u and v, that is, it is a relative address. Q.E.D.

Theorem 3. Let $(x,\ y)$ and $(x',\ y')$ be shortest path vectors from vertex 0 to vertices u and v, accordingly, in a graph of the family (2) of diameter $D > 1$ and the number of nodes $2D^2 + D \le N \le N_D$. If

$$|x' - x| + |y' - y| \le D, \tag{4}$$

then the vector $P(u,\ v) = (x' - x,\ y' - y)$ defines the shortest path from u to v.

Proof: It follows from the property of graphs of the family (2) that only for the indicated number of vertices of graphs, all vertices located at distance D from vertex 0 do not have paths from vertex 0 with the length less than D. Q.E.D.

Note that in dense Gaussian graphs with $N = N_D$, the property (4) holds, which allow to reduce the amount of required operations when developing a routing algorithm for them compared to the algorithm from [26].

B. Routing Algorithm for Degree Four Circulant Networks

The essence of the method of routing, using relative addressing of nodes in a network, is as follows. To send a packet from node $u = (x,\ y)$ to node $v = (x',\ y')$, we need to obtain the minimum of $|\Delta X| + |\Delta Y|$ from the values $(\Delta X,\ \Delta Y) = (x' - x,\ y' - y)$. Adding the corresponding coordinates of $(\Delta X,\ \Delta Y)$ term by term with the coordinates of neighboring zeros from a set of minimum required zeros and then comparing the resulting sums of their modules, we choose the minimum, which gives us the shortest path. The obtained minimal $l_1$norm values of $(\Delta X,\ \Delta Y)$ will determine the shortest path between the two vertices. In the general case, as shown by us by an experimental test in Wolfram Mathematica 10 for any number of nodes (and shown in [26] at the example of dense Gaussian graphs), it is enough to get the coordinates of nine nearest zeros, including the central zero.

Thus, two values $(\Delta X,\ \Delta Y)$ are the inputs of the router. And then, with each next transition to a neighboring node along the shortest path, one of the coordinates decreases by one (in absolute value). The packet reaches the destination node v, when $\Delta X = 0$ and $\Delta Y = 0$. The estimate of the time complexity of this routing procedure is $O(1)$. By being able to use the entire set of the shortest paths to a destination node, the routing can be made adaptive and fault-tolerant to node and link failures.

The main problem lies in determining the set of neighboring zeros needed to calculate $(\Delta X,\ \Delta Y)$ and the coordinates of neighboring zeros relative to the central zero. We determined them analytically for any number of nodes in graphs of the family (2) using the proposed scheme of transformations of their patterns on the plane Z^2 (Fig. 1). For graphs of the family (2), the following property takes place: when the number of nodes in a graph decreases by one, then the coordinates of zeros z_2, z_3, and z_4 also change by one – the coordinate x increases, and the coordinate y decreases. For symmetrically located zeros $-z_2$, $-z_3$, and $-z_4$, vice versa, the coordinate y increases, and the coordinate x decreases by one (in absolute value).

The coordinates of zeros of neighboring regions $z_1,\ z_2,\ z_3, z_4$ can be expressed in terms of coordinates z_1 and any of $z_2,\ z_3,\ z_4$. Take, for example, z_3. Then we have the following result shown in Table I and Table II. Note that the conducted analysis allowed us to provide a complete characterization of the properties of neighboring zeros to construct the optimal routing for any number of nodes in a NoC.

Tables I and II give the formulas for calculation of coordinates of four neighboring zeros $z_1,\ z_2,\ z_3$, and z_4. For graphs with description $C(N;\ D,\ D+1)$, they are shown in Table I; for graphs with description $C(N;\ D-1,\ D)$ – in Table II. The coordinates of the other four zeros, symmetrically located at Z^2, differ only in signs. In Table I, the parameter k specifies the difference between N_D and N; in Table II – between $2D^2$ and N. Thus, for a network with any number of nodes, we need to store no more than 8 memory blocks for the algorithm to execute.

At the stage of forming a communication subsystem in a NoC with the circulant topology under consideration, we must form the necessary parameters for the routing algorithm to work, that is, to perform preprocessing stage.

Preprocessing (created at the stage of forming a network topology):

Input: N – number of nodes in a NoC topology;
Output: D – diameter; mapping the labels of nodes $0,\ 1,\ \ldots,\ N-1$ into their vectors of the shortest path from node 0 (using any algorithm from [12], [24], [40]); forming parameters from Table I (or Table II).

TABLE II
COORDINATES OF ZEROS FOR CIRCULANTS $\{C(N; D-1,\ D) \mid N_{D-1}+1 \le N \le 2D^2,\ D\rangle 2\}$

$k = 2D^2 - N$	Zeros of 4 neighboring regions			
	$z_1 = (a_1,\ b_1)$	$z_2 = (a_2,\ b_2)$	$z_3 = (a_3,\ b_3)$	$z_4 = (a_4,\ b_4)$
$0 \le k < \lceil D/2 \rceil$	$(D,\ -D+1)$	$z_3 + z_1$	$(D+k,\ D+1-k)$	$z_3 - z_1$
$k = \lceil D/2 \rceil$	above	$z_3 - 2z_1$ for odd D, else $\emptyset$	above	above
$\lceil D/2 \rceil < k < D + \lfloor D/2 \rfloor$	above	$z_3 - 2z_1$	above	above
$k = D + \lfloor D/2 \rfloor$	above	above	$\emptyset$	above
$D + \lfloor D/2 \rfloor < k \le 2D - 2$	above	above	$z_3 - 3z_1$	above

TABLE III
TABLE OF MAPPING THE LABELS OF NODES TO THEIR VECTORS OF A SHORTEST PATH FROM NODE 0

i	0	1	2	3	4	5	6	7
$P(0,\ i)$	(0,0)	(-1, 1)	(1, 0)	(0, 1)	(2, 0)	(1, 1)	(0, 2)	(2, 1)

TABLE IV
PARAMETERS (a_i, b_i) FOR THE ROUTING ALGORITHM

a_1	b_1	a_2	b_2	a_3	b_3	a_4	b_4
3	-2	0	5	6	1	3	3

Consider the results of preprocessing stage by the example of forming NoC topology modelled by the circulant graph $C(15;\ 2,\ 3)$.

The graph diameter is $D = 3$; $k = 3$. In Table III, the mapping of the labels of nodes $i \le \lfloor N/2 \rfloor$ into their vectors of the shortest path from the vertex 0 (the relative addressing) is presented. In Table IV, parameters a_i and b_i , where $i = \overline{1,\ 4}$, are computed from Table II for the $k = 3$.

Note that at the preparation stage, we will only need to determine (using Table I or Table II) the coordinates of the zeros $z_1 = (a_1,\ b_1)$ and $z_3 = (a_3,\ b_3)$, which will reduce the required memory for the routing algorithm to work.

Below we describe the routing algorithm itself (we named it the GRBT algorithm – General Routing Based on Tessellation algorithm), which is executed on each router, in which a request to transmit a packet arrives. This routing algorithm is applicable to circulant networks of the family (2) with any arbitrary number of nodes. The main items of the algorithm are the following: on the first step, check the condition (3) for the l_1-norm of relative source-destination addresses (lines 1–2); on the second step, calculate the shortest path vectors using all the neighboring zeros (lines 4–11); on the third step, find the minimal shortest path vector by comparing among all neighboring zeros (line 12).

In the GRBT algorithm, the condition (4) can be used instead of (3). In this case, the algorithm determines the shortest paths in a graph with any $2D^2 + D \le N \le N_D$. For $N_{D-1} < N < 2D^2 + D$, it determines suboptimal paths (one hop more than the shortest path) only for those source-destination pairs, for which the length between them is equal to the diameter.

The structure of the GRBT algorithm for graphs of the family (2) allows us to parallel execution of operations of addition/subtraction and comparison, which, being implemented in NoCs, will reduce the execution time of the algorithm by several times.

Above presented version of the routing algorithm uses all nine nearest zeros for its operation. But for some infinite subfamilies of values of N, the number of zeros needed to execute the routing algorithm is less than 9. The following result is a generalization of experimental data obtained.

Algorithm 1. General routing algorithm.

Input: $(x,\ y)$ – source node; $(x',\ y')$ – destination node; D – diameter; a_i , b_i – parameters from Table I (or Table II).
Output: $(\Delta X,\ \Delta Y)$ – the shortest path vector from $(x,\ y)$ to $(x',\ y')$.
1: **begin** $\Delta X := x' - x$; $\Delta Y := y' - y$; $\Delta x_o := \Delta X$; $\Delta y_o := \Delta Y$;
2: **If** $|\Delta X| + |\Delta Y| \ge D$ **then**
3: **begin**
4: $\Delta x_1 := \Delta X + a_1$; $\Delta y_1 := \Delta Y + b_1$;
5: $\Delta x_2 := \Delta X + a_2$; $\Delta y_2 := \Delta Y + b_2$;
6: $\Delta x_3 := \Delta X + a_3$; $\Delta y_3 := \Delta Y + b_3$;
7: $\Delta x_4 := \Delta X + a_4$; $\Delta y_4 := \Delta Y + b_4$;
8: $\Delta x_5 := \Delta X - a_1$; $\Delta y_5 := \Delta Y - b_1$;
9: $\Delta x_6 := \Delta X - a_2$; $\Delta y_6 := \Delta Y - b_2$;
10: $\Delta x_7 := \Delta X - a_3$; $\Delta y_7 := \Delta Y - b_3$;
11 $\Delta x_8 := \Delta X - a_4$; $\Delta y_8 := \Delta Y - b_4$;
12 $(\Delta X,\ \Delta Y) := (\Delta x_i,\ \Delta y_i)$ such that $|\Delta x_i| + |\Delta y_i|$ is minimum;
13 **end**
14 **end**

Theorem 4: Let $C(N;\ s_1,\ s_2)$ be a graph of the family (2) of diameter $D > 2$, and N be any value from the following ones

$$\begin{cases} 2D^2 - D - D/2;\ 2D^2 - D/2; \\ 2D^2 + D/2;\ 2D^2 + D + D/2. \end{cases}$$

Then the number of zeros sufficient for the GRBT algorithm to be realized correctly is 7.

Proof: As it was shown in [26], [41], the minimal path from a node $(x,\ y)$ to a node $(x',\ y')$ is one of the nine path alternatives considering the destination node image in the nine tessellations at the plane, including the central region, as it is illustrated in Fig. 1. For proof of the theorem, it is sufficient to consider the two nodes as source nodes in the routing with maximum distance from vertex 0 in the central region: $u_1 = Ds_1$ and $u_2 = Ds_2$.

Next, it is necessary to determine which neighboring zeros from the set $\{z_i,\ i = \overline{0,\ 4}\}$ are sufficient for the correct operation of the routing algorithm, that is, at least one of $u_1,\ u_2$ must satisfy the following conditions: for the subfamily of circulants $C(N;\ D,\ D+1)$ – either $\|\ z_i - u_1\ \| \leq 2D$ or $\|\ z_i - u_2\ \| \leq 2D$; for the subfamily of circulants $C(N;\ D-1,\ D)$ – either $\|\ z_i - u_1\ \| \leq 2D-2$ or $\|\ z_i - u_2\ \| \leq 2D-2$.

1) Consider circulants of the form $C(N;\ D,\ D+1)$, where $2D^2+1 \leq N \leq N_D$.
   Let $k = \lfloor D/2 \rfloor + 1$, where $D > 1$, then $N = 2D^2 + D + \lceil D/2 \rceil$. We have: $z_0 - u_1 = (-D,\ 0)$, $\|\ z_0 - u_1\ \| = D$, $z_1 - u_1 = (1,\ -D)$, $\|\ z_1 - u_1\ \| = D+1$, $z_3 - u_1 = (\lfloor D/2 \rfloor + 1,\ \lceil D/2 \rceil)$, $\|\ z_3 - u_1\ \| = D+1 < 2D$, $z_4 - u_2 = (\lfloor D/2 \rfloor,\ \lceil D/2 \rceil)$, $\|\ z_4 - u_2\ \| = D$.
   But $z_2 - u_1 = (D + 2 + \lfloor D/2 \rfloor, -\lceil D/2 \rceil)$, $\|\ z_2 - u_1\ \| = D + 2 + 2\lfloor D/2 \rfloor + 2D$, $z_2 - u_2 = (2D + 2 + \lfloor D/2 \rfloor,\ -\lfloor D/2 \rfloor - D)$, $\|\ z_2 - u_2\ \| = 3D + 2 + 2\lfloor D/2 \rfloor > 2D$.
   Therefore, for $N = 2D^2 + D + \lceil D/2 \rceil$, it is necessary only 7 zeros: $z_0, \pm z_1,\ \pm z_3,\ \pm z_4$.
   Let $k = D + \lceil D/2 \rceil + 1$, where $D > 1$, then $N = 2D^2 + \lfloor D/2 \rfloor$. We have: $z_0 - u_1 = (-D, 0)$, $\|\ z_0 - u_1\ \| = D$, $z_1 - u_1 = (1, -D)$, $\|\ z_1 - u_1\ \| = D+1$, $z_2 - u_2 = (-1 + \lceil D/2 \rceil,\ D - \lceil D/2 \rceil)$, $\|\ z_2 - u_2\ \| = D + 1 < 2D$, $z_4 - u_1 = (\lceil D/2 \rceil,\ D - \lceil D/2 \rceil)$, $\|\ z_4 - u_1\ \| = D$.
   But $z_3 - u_1 = (D + 1 + D/2,\ -\lceil D/2 \rceil)$, $\|\ z_3 - u_1\ \| = D + 1 + 2\lceil D/2 \rceil > 2D$, $z_3 - u_2 = (2D + 1 + \lceil D/2 \rceil, -\lceil D/2 \rceil - D)$, $\|\ z_3 - u_2\ \| = 3D + 1 + 2\lceil D/2 \rceil > 2D$.
   Thus, for $N = 2D^2 + \lfloor D/2 \rfloor$, it is necessary only 7 zeros: $z_0, \pm z_1,\ \pm z_2,\ \pm z_4$.
2) Consider circulants of the form $C(N; D-1,\ D)$, where $2D^2 - 2D + 2 \leq N \leq 2D^2$.
   If $k = \lceil D/2 \rceil$, where $D > 2$, then $N = 2D^2 - \lceil D/2 \rceil$. If $k = D + \lfloor D/2 \rfloor$, where $D > 2$, then $N = 2D^2 - D - \lfloor D/2 \rfloor$. The proofs for these values of N are similar to the proofs for case 1. Q.E.D.

This result corresponds to empty sets in Table I and Table II and gives the possibility to reduce the required memory under routing in a NoC. Further in the article, we will denote by z the number of neighboring zeros required for the operation of the GRBT algorithm.

IV. Design

A. Analysis of the Routing Algorithm

We have the following reductions in the number of operations required for the GRBT algorithm to be fulfilled:

1) For networks with the number of nodes, indicated in Theorem 4, when 7 zeros are used instead of 9 ones, the reduction in operations is 22%.
2) For many source-destination pairs – due to the fulfillment of condition (3) or (4).

The plot of reductions in the number of operations, required for the GRBT algorithm due to the fulfillment of condition (3), is presented in Fig. 3(a). The parameter λ means the ratio of the number of source-destination pairs with a l_1-norm, smaller than graph diameter D, to the total number of node pairs. This parameter gives a payoff function from 32 % to 56 % in the running time of the GRBT algorithm for graphs with a different number of nodes, $14 \leq N \leq 313$. Sections of the falling curve for values of λ at a given diameter D, where $3 \leq D \leq 12$, correspond to a maximum of λ for $N = N_{D-1} + 1$ and to a minimum for $N = N_D$.

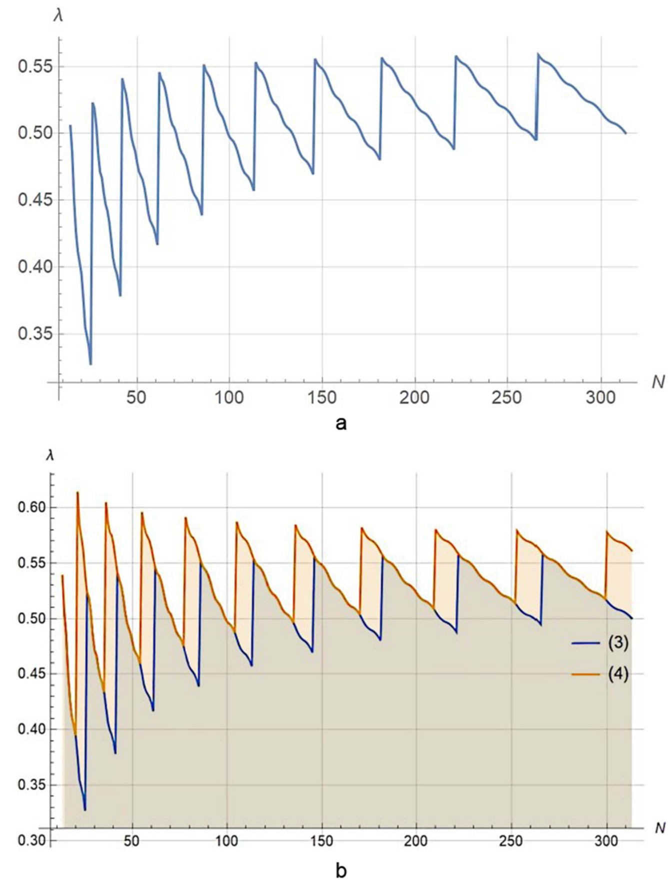

Fig. 3. Relative reduction of the operations number of the GRBT algorithm for graphs of the family (2): a) by using (3); b) by using (4).

The lowest points on the plot correspond to a payoff function in operations for dense Gaussian graphs in comparison with the routing algorithm proposed for them in [26]. This advantage (in comparison with the algorithm from [26]) increases as $O(\log N)$ when N grows.

The plot of reductions in the number of operations, required for the GRBT algorithm due to the fulfillment of condition (4), is presented in Fig. 3(b). Here the parameter λ is the same as the parameter λ at ranges $N_{D-1} < N < 2D^2 + D$ of Fig. 3(a) and increases at ranges $2D^2 + D \leq N \leq N_D$. This parameter gives a payoff function from 52 % to 56 % in the running time of the GRBT algorithm in comparison with the algorithm from [26] for dense Gaussian graphs with a different number of nodes $N = N_D, 3 \leq D \leq 12$.

The data presented in Fig. 3 is obtained by computer simulation of the GRBT algorithm by enumeration of all possible source-destination pairs for each graph from the family (2) in a given range of N.

Now we give an example of the GRBT algorithm execution.

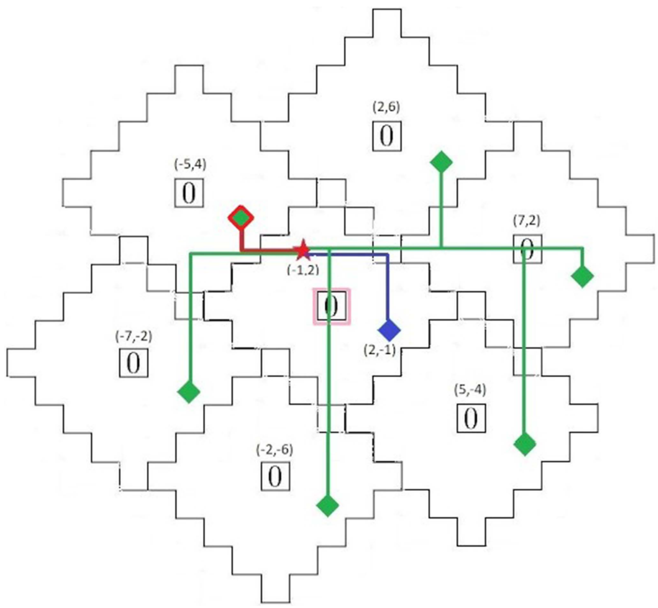

Fig. 4. An example of possible paths to a destination node calculated by the GRBT algorithm for neighboring zeros in circulant $C(38;\ 4,\ 5)$.

Example 1. Consider a NoC with a circulant topology with signature $C(38;\ 4,\ 5)$ of diameter $D = 4$. We have three zeros from Table I, plus their symmetrical ones, plus $(0,\ 0)$. Thus, there is a set of 7 zeros: $\{(5,-4),\ (7,\ 2),\ (2,\ 6),\ (-5,\ 4), (-7,\ -2), (-2,-6),\ (0,\ 0)\}$.

Let a source node be 6 with $(x,\ y) = (-1,\ \ 2)$, and a destination node be 3 with $(x',\ y') = (2,\ -1)$. As $(x'-x,\ y'-y) = (3,\ -3)$, and $l = 6 > D$, the following set of pairs of coordinates can be obtained by adding coordinates of seven zeros to $(3,-3)$: $\{(8,-7),\ (10,-1),\ (5,\ \ 3),\ (-2,\ \ 1),\ (-4,-5),\ (1,-9),\ (3,-3)\}$.

The pair with the minimum l_1-norm is $(-2,\ \ 1)$, which is the shortest path vector from node 6 to node 3.

Now, let a source node be 6 with $(x,\ y) = (-1,\ 2)$, and a destination node be 4 with $(x',\ y') = (1,\ 0)$. As $(x'-x,\ y'-y) = (2,\ -2)$, and $l = 4 = D$, then the vector of the shortest path from node 6 to node 4 is $(2,\ -2)$.

The example of the GRBT algorithm execution for circulant $C(38;\ 4,\ 5)$ is presented in Fig. 4. Here the source node (red) is 6 with $(x,\ y) = (-1,\ 2)$, and destination node (blue) is 3 with $(x',\ y') = (2,\ -1)$. The vector of the shortest paths from node 6 to node 3 will be $(-2,\ 1)$ (shown by the red line).

B. A Reduction of the Routing Algorithm Parameters

Using Table I and Table II and changing N and, accordingly, obtaining the parameter k, we can have different versions of the GRBT algorithm for any number of nodes in graphs of the family (2) with two times less number of parameters obtained at the stage of pre-preparation. Moreover, we can get routing algorithms, for which the coordinates of the required neighboring zeros can be calculated in dynamics.

Algorithm 2. SRBT – a modification of the GRBT algorithm for the given subfamily of circulants.

Input: $(x,\ y)$ – source node; $(x',\ y')$ – destination node; D – diameter.
Output: $(X,\ Y)$ – the shortest path vector from $(x,\ y)$ to $(x',\ y')$.

begin $X := x' - x; Y := y' - y;$
  If $|X| + |Y| > D$ **then**
    begin
      $4:\ x_0 := X + D + 1;\ \ y_0 := Y - D;\ \ x_1 := X - 1;\ y_1 := Y + 2D;\ x_2 := X + 2D + 1;\ y_2 := Y;\ \ x_3 := X + D;\ y_3 := Y + D;\ x_4 := X - D - 1;\ y_4 := Y + D;\ x_5 := X + 1;\ y_5 := Y - 2D;\ \ x_6 := X - 2D - 1;\ y_6 := Y;\ \ x_7 := X - D;\ y_7 := Y - D;\ \ X := x_0;\ Y := y_0;\ a := |X| + |Y|;$
      For $(i = 1; i < 8; i++)$
        begin
          $b := |x_i| + |y_i|;$
          If $b < a$ **then** $\{a := b;\ X := x_i;\ Y := y_i;\}$
        end
      end
end

For a specific NoC with a certain number of nodes, we can go even further and not pre-calculate the coordinates of neighboring zeros but use the formulas for them from Table I and Table II in the routing algorithm itself. This is possible when we consider subclasses of the general set of graphs of the given family of circulants. In such specialized routing algorithms, there is no memory to store additional parameters for the algorithm to work.

We propose a modification of the routing algorithm applicable to an infinite subfamily of the number of nodes in graphs of the family (2). Consider this option, for example, for graphs of the form $C(N;\ D,\ D+1)$ with $N = 2D^2 + D$, $D > 1$. In this case, the parameter $k = D + 1$. Using Table I, we obtain the desired algorithm.

Below we describe the routing algorithm (we named it the SRBT algorithm – Specialized Routing Based on Tessellation algorithm), which is executed on each router, in which a request to transmit a packet arrives. The SRBT algorithm is applicable to circulant networks with the number of nodes $N = 2D^2 + D$ for any diameter $D > 1$. Note that for such values of N, condition (4) is satisfied, as well as for dense Gaussian graphs.

Note that a parallel implementation of this algorithm in NoCs may provide a faster solution, but it will increase the required memory and hardware overhead in NoCs.

When implementing the routing algorithm in NoCs, we used both versions of the routing algorithms: the full one – for any number of nodes in graphs of the family (2) and its modification – for the considered subclass of graphs. The estimates of our algorithms for the required memory resources and time execution are presented in the next sections. We implemented only sequential versions of the GRBT and SRBT algorithms in order not to increase the required memory and to be able to compare them with other algorithms earlier developed for NoCs, which are also sequential.

V. Experimental Results

A. Routing Algorithm Implementation in NoCs

The GRBT algorithm was implemented in NoCs with a two-dimensional circulant topology of optimal graphs of the family (2).

For the GRBT algorithm to work, the router must store the following parameters: 1) the diameter of a graph; 2) the coordinates of a node in the network; 3) the coordinates of zeros from Table I (or Table II) for the corresponding family of graphs; 4) the results of intermediate calculations, that is, the values of x_i and y_i, where $0 \leq i \leq z-1$.

As a more accurate estimate of the required memory for these parameters, we do not use half of the number of graph nodes ($N/2$) but a function of its diameter D:

1) Since the diameter of a graph $C(N; D,\ D+1)$ is $D = \lfloor\sqrt{N/2}\rfloor$, it takes $\lceil\log_2\lfloor\sqrt{N/2}\rfloor\rceil$ bits to be stored in the router.
2) As $|x|,\ |y| \leq D$, only $2\lceil\log_2 D\rceil + 2$ bits are required to store the coordinates of a node $(x,\ y)$ (where 2 bits are needed for a sign).
3) The upper bounds for the coordinates of zeros $(a_i,\ b_i)$ follow from Table I and Table II: $|a_i|, |b_i| \leq 2D + \lfloor D/2\rfloor + 1$. Note that of the total number of coordinates of neighboring zeros, only half of them need to be stored in memory. Hence, to store them, $(z-1)\lceil\log_2(2D + \lfloor D/2\rfloor + 1)\rceil$ bits are enough, where $z \in \{7,\ 9\}$.
4) The upper bounds for results of intermediate calculations (i. e., for variables x_i and y_i, $i = \overline{0,\ z-1}$) are as follows: $|x_i|, |y_i| \leq 4D + \lfloor D/2\rfloor + 1$. Accordingly, the number of bits for storing them is $2z\lceil\log_2(4D + \lfloor D/2\rfloor + 1)\rceil + 2z - 2$, where $z \in \{7,\ 9\}$. Here we take into account the possibility of addition of variables with the maximum value.

Thus, the total amount of memory (in bits) for the operation of the GRBT algorithm in Verilog can be calculated using the following formula:

$$\begin{aligned} M_{all} = {} & 2z\lceil\log_2(4\lfloor\sqrt{N/2}\rfloor + \lfloor\lfloor\sqrt{N/2}\rfloor/2\rfloor + 1)\rceil \\ & + (z-1)\lceil\log_2(2\lfloor\sqrt{N/2}\rfloor + \lfloor\lfloor\sqrt{N/2}\rfloor/2\rfloor + 1)\rceil, \\ & + 3\lceil\log_2\lfloor\sqrt{N/2}\rfloor\rceil + 2z \end{aligned}$$

where $z \in \{7,\ 9\}$.

It should be emphasized that due to the peculiarity of graphs of the family (2) and relative addressing of their vertices, it is possible to use not $M_{all} = 3z * \lceil\log_2 N/2\rceil + 2z - 2$ but the above function $M_{all}(\sqrt{N/2})$ as an estimate of the required memory, which gives significant savings with an increase in the number of nodes in a graph (for example, for $N = 4900$, it is already more than 30 %).

We simulated NoCs with the number of nodes from 9 to 100, where $N = n^2$, $3 \leq n \leq 10$. Circulant topologies from the family (2) were selected. Table V shows the results of the theoretical calculation of occupied memory in bits, consumed memory resources in REG, and logical blocks in ALM obtained after modeling of the GRBT algorithm.

TABLE V
The Results Obtained for the GRBT Algorithm

Circulant	z	Memory theoretical	1 router REG	1 router ALM	Network REG	Network ALM
C(9; 2, 3)	7	94	57	116	494	784
C(16; 2, 3)	9	120	111	183	1288	2411
C(25; 3, 4)	9	146	113	187	2163	4088
C(36; 4, 5)	9	149	117	120	3649	6843
C(49; 4, 5)	9	149	149	234	6434	10403
C(64; 5, 6)	9	149	157	302	9903	17162
C(81; 6, 7)	7	123	135	270	10732	19581
C(100; 7, 8)	9	175	151	303	15173	23038

The following conclusions can be stated from the data obtained. With a pronounced increase in the used logical and memory resources due to the increase in the bits of depth of numbers, the number of coordinates of the used zeros for the network makes a valuable impact. The network $C(81;\ 6,\ 7)$ is clearly distinguished; its resource consumption is lower than that of networks with 64 and 100 nodes. This is due to the fact that the number of zero coordinates for it is 7 instead of 9, which leads to a decrease in the number of comparisons to determine the shortest path vector $(\Delta X,\ \Delta Y)$.

B. Memory Consumption Estimations for Routing Algorithms in NoCs

We made a comparative analysis of the hardware resources consumption for the GRBT algorithm and other routing algorithms from [23], [24], [25] previously implemented in NoCs with two-dimensional circulant topology. Table VI and Table VII show the results of NoC modeling with the following five routing algorithms implemented in them: the PEA [24], Table routing algorithm (TRA) [25], Clockwise algorithm (CA) [25], Adaptive algorithm (AA) [25], Algorithm for circulants of type $C(N;\ d,\ d+1)$ (AC) [23]. The last columns of Table VI and Table VII show the results for the proposed GRBT algorithm. Table VI shows the required cost of memory resources (REG), Table VII shows the consumption of logical resources (ALM) of the FPGA chip.

Comparing the obtained data for the required memory resources in the network with the number of nodes not exceeding $N = 100$, we obtain the following conclusions. The GRBT algorithm in using registers REG of the FPGA chip is better only than the TRA for the number of nodes starting from $N = 36$, and it is almost 2 times better for $N = 81$ and $N = 100$. The GRBT algorithm in using ALM of the FPGA chip is better than the AA and AC by more than 3 and 2 times, respectively, and slightly worse than the PEA. Note that it was previously shown in [25] that logical resources for NoCs are more critical than memory resources. Therefore, the GRBT algorithm is more preferable than the AA and AC.

It should be emphasized that due to the peculiarity of graphs of the family (2), the GRBT algorithm may use the function $M_{all}(\sqrt{N/2})$ as an estimate of the required memory but not

TABLE VI
DEPENDENCE OF MEMORY RESOURCES IN REG ON THE NUMBER OF NODES

N	PEA	TRA	CA	AA	AC	GRBT
9	371	274	102	81	322	494
16	757	820	315	210	617	1288
25	1158	1976	674	301	1284	2163
36	1962	4097	1048	546	2215	3649
49	2654	7442	1738	664	3186	6434
64	3965	12476	2419	988	4506	9903
81	4904	20057	3548	1356	6312	10732
100	6044	30429	4857	1662	8327	15173

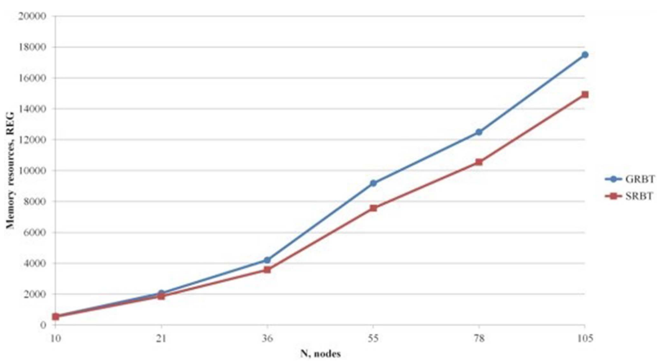

Fig. 5. Dependence of memory resources in REG on number of nodes in the network for the GRBT and SRBT algorithms.

the function of $N/2$ or N, which has been used for all the algorithms realized in NoCs (the CA, AA, AC, and PEA).

We also implemented in a NoC the SRBT agorithm, which is obtained from the GRBT algorithm (suitable for any number of nodes in the network) for the subfamily of graphs with $N = 2D^2 + D$ nodes. The difference of the SRBT algorithm from the GRBT is the absence of the coefficients a_i, b_i, where $i = \overline{1,\ 4}$, required for storage in the network, and also a lower upper bound for the results of intermediate calculations: $|x_i|$, $|y_i| \leq 3D + 1$, $i = \overline{0,\ 8}$. As a result, the amount of the memory, required for the operation of the SRBT algorithm decreased; it can be calculated by the formula: $M2_{all} = 3\lceil \log_2 |\sqrt{N/2}|\rceil + 18\lceil \log_2 (3|\sqrt{N/2}\,| + 1)\rceil + 18$.

A decrease in the use of chip resources, when implementing the SRBT algorithm in a NoC, was obtained by up to 17 % in comparison with the GRBT (Figs. 5 and 6).

C. Performance Comparison of Routing Algorithms in NoCs

We also measured the time spent on calculating paths in the network and calculating one path in the high-level C# language for all the algorithms under consideration, except for the TRA.

Previously implemented in NoCs the routing algorithms PEA, CA, AA, and AC were not compared in terms of the execution time. But we had the configurations of their implementations in a NoC with the number of nodes $N = n^2$ for $3 \leq n \leq 10$. For the aforementioned comparison of algorithms in time, we added the GRBT and SRBT algorithms,

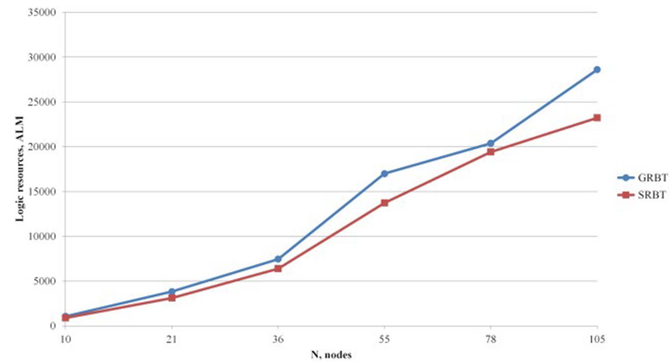

Fig. 6. Dependence of logical resources in ALM on number of nodes in the network for the GRBT and SRBT algorithms.

although the SRBT is applicable only for $N = 2D^2 + D$, $D > 1$. When calculating certain paths, the SRBT gives a non-optimal route for the number of nodes $N \neq 2D^2 + D$. In this case, the route is still correct, but not always optimal. However, this does not affect the calculation of the execution time, since regardless of the length of the route, the number of operations for its calculation does not change.

The node addresses in the GRBT and SRBT algorithms were calculated using Dijkstra's algorithm [40]. Dijkstra's algorithm gives the optimal path from the zero node to all the other nodes. Then, taking into account the sign, the number of hops along s_1 and s_2 generators at these paths is counted. These are the $(x,\ y)$ and $(x',\ \ y')$ coordinates for the nodes. All the other algorithms tested use absolute addressing of network nodes.

When calculating the time of search for the shortest path in the algorithms, a choice of source and destination nodes is the following. For the CA, AA, AC, and PEA, the paths from the zero node to all the others are calculated. For the GRBT and SRBT algorithms, a random source and destination are specified, and 10^6 calculations are performed so that the results are averaged.

The calculation was carried out on a computer with processor Intel Core i5-6500, 3.20 GHz, 32 GB RAM. For each network, 10^6 calculations of the execution time (T) of the algorithms were carried out, and the average values of T (in ms) were calculated (Figs. 7 and 8).

From the data in Figs. 7 and 8, we can conclude that the dependencies in the implementation of the GRBT and SRBT algorithms in the Verilog are also traced when they are executed in a high-level language. The computation time of a path for the SRBT is more linear than for the GRBT, which is explained by the same number of operations performed in the SRBT algorithm; which does not depend on the number z of the required zeros involved in calculation of the shortest paths. An increase in path calculation time with an increase in the number of nodes in the network is due to the large sizes of variables. The execution time of the SRBT algorithm is up to 46 % less than that of the GRBT.

When comparing the SRBT algorithm with the PEA, CA, AA, and AC, the following conclusions can be obtained. The SRBT algorithm (when calculating one path and paths in the

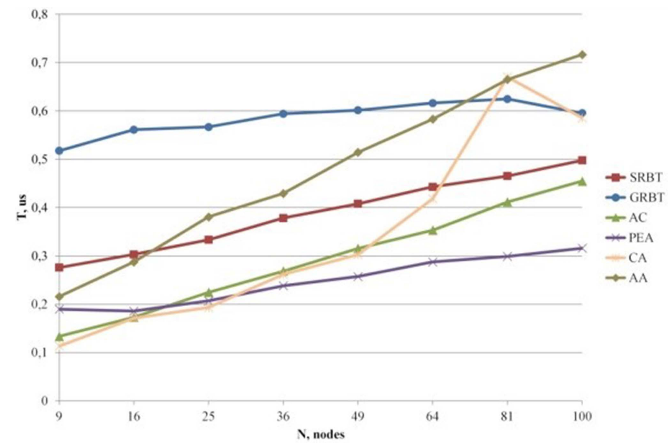

Fig. 7. Dependence of the execution time on the routing algorithm for one router $N = n^2$.

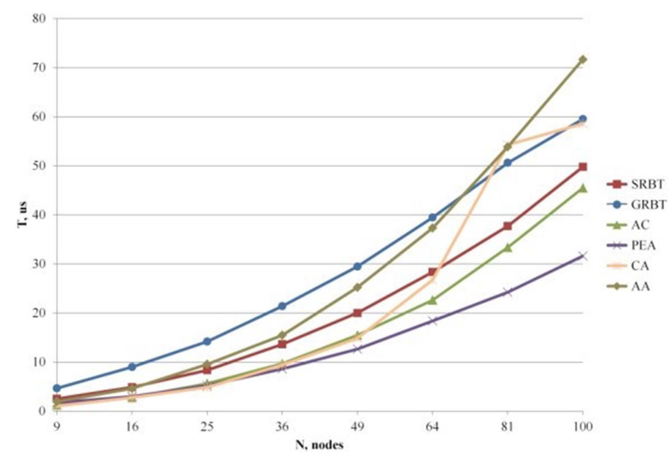

Fig. 8. Dependence of the execution time on the routing algorithm for the entire network $N = n^2$.

entire network) turns out to be faster than AA. This is explained by the fact that in the AA, there are more assignment and division operations than in the SRBT. Other algorithms have more operations than the SRBT, but they are mostly comparison operations that are faster than assignment operations.

The SRBT algorithm, when the number of nodes in the network is more than 64, starts calculating routes faster than the CA. The CA is not optimal, and when the number of nodes $N > 49$, its efficiency drops to 50 %. Although the CA on the small numbers of nodes shows the best speed for calculating the path; its peculiarity leads to the fact that the path becomes much longer than the optimal one, which results in a sharp increase in computation time. The results of search for the paths in different routing algorithms are determined by the number of different types of operations used and their execution time.

To substantiate the obtained execution times of the compared routing algorithms in C#, we determined the speed of individual operations in integer calculations. The calculation was implemented on a computer with the following characteristics: processor Intel Core i5-6500, 3.20 GHz, 32 GB RAM on the following platform: OS Windows 8.1 for 64-bit system, development environment – MS Visual Studio. The calculation was made by performing operations on one-dimensional

TABLE VII
DEPENDENCE OF LOGIC RESOURCES IN ALM ON THE NUMBER OF NODES

N	PEA	TRA	CA	AA	AC	GRBT
9	732	79	99	1933	1511	784
16	1825	187	301	4737	3192	2411
25	2903	536	742	11491	6257	4088
36	5473	1019	1354	22025	10959	6843
49	7899	1707	2042	30709	18318	10403
64	12909	2788	3230	48671	28157	17162
81	16755	4622	4839	65825	42333	19581
100	19226	6106	6324	83938	60516	23038

TABLE VIII
OPERATION TIME AND THE NUMBER OF OPERATIONS USED IN THE PEA AND GRBT ALGORITHMS

Operation	Time (ns)	PEA worst / best	GRBT worst / best
>=	0.586	1 / 1	1 / 0
<=	0.633	2 / 2	0 / 0
<	0.744	1 / 0	1 / 0
>	0.807	1 / 1	0 / 0
&	0.655	1 / 1	0 / 0
–	0.672	5 / 3	8 / 2
*	0.722	2 / 2	0 / 0
+	0.770	2 / 0	16 / 1
%	2.296	1 / 1	0 / 0
/	2.530	2 / 2	0 / 0
:=	2.978	12 / 10	34 / 5
$\lvert\cdot\rvert$	6.436	1 / 1	16 / 2

arrays with an array length equal to 10^9 elements. The calculation results are shown in Table VIII. The most time consuming operations are the modulus or absolute value ($|\cdot|$) and assignment ($:=$). Also, the operations of calculating division and remainder are done for a long time. For the PEA and GRBT algorithms, the number of operations used was calculated and presented in Table VIII. It should be noted that Table VIII shows the exact values for the best case of the number of operations in the GRBT calculation. For the worst case, approximate values are given, since the averaged exact values can be determined only by running the algorithm experimentally.

From Table VIII, it follows that the GRBT algorithm in the worst case uses significantly more assignments and modulus computations than the PEA, but in its best case (which is 32 % to 56 % of all the source-destination pairs), it is better than the PEA. Based on the data obtained, a theoretical calculation of the total time, required to compute one route for the compared algorithms together with a practical test in a NoC, was obtained (Table IX). Note that the running time of the algorithms in practical implementation is longer than the theoretical one, since additional operations are required to store the results for further use in the program. Comparing the difference between theoretical calculation and practical implementation for each of versions of the PEA and GRBT,

TABLE IX
TOTAL TIME FOR THE PEA AND GRBT ALGORITHMS

	Theory (us)	Test (us)
PEA worst	0.0599	0.1900
PEA best	0.0503	0.1900
GRBT worst	0.2344	0.5176
GRBT best	0.0305	0.5176

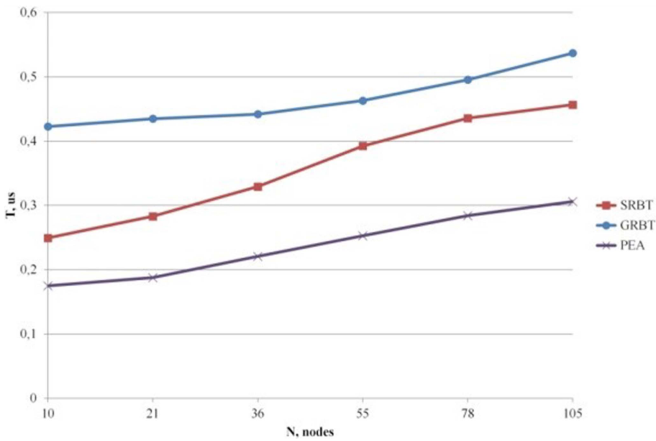

Fig. 9. Dependence of the execution time on the GRBT, SRBT and PEA algorithms for one router $N = 2D^2 + D$.

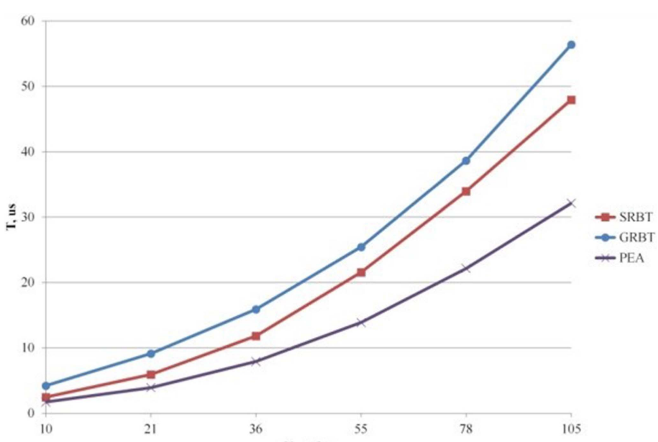

Fig. 10. Dependence of the execution time on the GRBT, SRBT and PEA algorithms for the entire network, $N = 2D^2 + D$.

it can be concluded that the influence of the type of operations on the speed of the algorithms occurs both as a result of the values obtained in theoretical calculations and in the data obtained in practical implementation. This explains the fact that, despite the absence of time-consuming division operations, the GRBT algorithm in the worst case is somewhat worse than the PEA.

To correctly compare the SRBT algorithm with the GRBT and PEA algorithms, we recalculated the metric for the number of nodes using the formula $N = 2D^2 + D$ and recalculated the results. The results are presented in Figs. 9 and 10. We compare the GRBT and SRBT with the PEA, since the PEA, as it was shown in [24], is the most efficient among other algorithms in a number of checked parameters. When comparing for such numbers of nodes, the difference in execution time between the GRBT and SRBT is on average 12–20 % in favor of the SRBT. Compared to the PEA, the GRBT algorithm runs 40–50 % longer, and almost so does the SRBT algorithm (30–35 % longer).

VI. CONCLUSION

For a family of optimal circulant networks of degree four with any number of nodes, a $O(1)$ pair routing algorithm (based on using relative addressing of nodes) was proposed. The new algorithm is an analytical extension of the approach presented in [26], [41] for dense Gaussian networks. The complete characterization of neighboring zeros in the plane tessellation made it possible to generate a series of routing algorithms for various subclasses of the optimal two-dimensional circulants. The routing algorithm in general form and its modification for a separate subclass of circulants are implemented in the HDL NoC model with a circulant topology. We obtained the comparison results of the new algorithms with other routing algorithms in terms of memory consumption and execution time. We showed that the new routing algorithms (when implemented in a NoC) do not have clear advantages over the analytical algorithm from [24], but in some cases (from 32 % to 56 % of all the source-destination pairs) show better performance. They also proved to be better than a number of algorithms based on other design principles. In further work, when developing such routing algorithms for NoCs, we assume the possibility of successfully reducing the number of neighboring zeros used and the number of time-consuming operations, as well as the usage of a parallel implementation, which is inherent in algorithms of this type.

ACKNOWLEDGMENT

The authors wish to thank E.V. Lezhnev and A.A. Amerikanov from HSE University for technical assistance and some calculations provided.

REFERENCES

[1] N. E. Jerger, T. Krishna, and L. S. Peh, *On-Chip Networks: Second Edition*. San Rafael, CA (USA): Morgan and Claypool Publishers, 2017.
[2] S. Hesham, J. Rettkowski, D. Göhringer, and M. A. Abd El Ghany, "Survey on real-time Network-on-Chip architectures," in *Proc. Int. Symp. Appl. Reconfigurable Comput.*, 2015, vol. 9040, pp. 191–202, doi: 10.1007/978-3-319-16214-0_16.
[3] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*. Amsterdam, Netherlands: Elsevier, 2003.
[4] D. Deb, J. Jose, S. Das, and H. K. Kapoor, "Cost effective routing techniques in 2D mesh NoC using on-chip transmission lines," *J. Parallel Distrib. Comput.*, vol. 123, pp. 118–129, Jan. 2019, doi: 10.1016/j.jpdc.2018.09.009.
[5] A. Touzene and K. Day, "All-to-all broadcasting in torus network on chip," *J. Supercomput.*, vol. 71, no. 7, pp. 2585–2596, Jul. 2015, doi: 10.1007/S11227-015-1406-Z/FIGURES/5.
[6] E. A. Monakhova, "A survey on undirected circulant graphs," *Discret. Math. Algorithms Appl.*, vol. 4, no. 1, 2012, Art. no. 1250002, doi: 10.1142/S1793830912500024.
[7] F. K. Hwang, "A survey on multi-loop networks," *Theor. Comput. Sci.*, vol. 299, no. 1–3, pp. 107–121, Apr. 2003, doi: 10.1016/S0304-3975(01)00341-3.

[8] J.-C. Bermond, F. Comellas, and D. F. Hsu, "Distributed loop computer-networks: A survey," *J. Parallel Distrib. Comput.*, vol. 24, no. 1, pp. 2–10, Jan. 1995, doi: 10.1006/jpdc.1995.1002.

[9] M. A. Fiol, "On congruence in Zn and the dimension of a multidimensional circulant," *Discrete Math.,* vol. 141, no. 1–3, pp. 123–134, Jun. 1995, doi: 10.1016/0012-365X(94)00361-L.

[10] X. Huang, A. F. Ramos, and Y. Deng, "Optimal circulant graphs as low-latency network topologies," *J. Supercomputing*, vol. 78, pp. 13 491–13 510, Mar. 2022, doi: 10.1007/s11227-022-04396-5.

[11] C. Dalfó, M. A. Fiol, N. López, and J. Ryan, "An improved Moore bound and some new optimal families of mixed Abelian Cayley graphs," *Discrete Math.,* vol. 343, no. 10, Oct. 2020, Art. no. 112034, doi: 10.1016/J.DISC.2020.112034.

[12] O. G. Monakhov, E. A. Monakhova, A. Y. Romanov, A. M. Sukhov, and E. V. Lezhnev, "Adaptive dynamic shortest path search algorithm in Networks-on-Chip based on circulant topologies," *IEEE Access*, vol. 9, pp. 160 836–160 846, 2021, doi: 10.1109/ACCESS.2021.3131635.

[13] S. Bujnowski, B. Marciniak, O. O. Oyerinde, Z. Lutowski, A. Flizikowski, and S. G. Galan, "The possibility of equalizing the transmission properties of networks described by chordal rings," in *Proc. IEEE 15th Int. Conf. Signal Process. Commun. Syst.*, 2021, pp. 1–8, doi: 10.1109/ICSPCS53099.2021.9660228.

[14] A. Erickson, I. A. Stewart, J. Navaridas, and A. E. Kiasari, "The stellar transformation: From interconnection networks to datacenter networks," *Comput. Netw.*, vol. 113, pp. 29–45, Feb. 2017, doi: 10.1016/J.COMNET.2016.12.001.

[15] R. R. Lewis, "Analysis and construction of extremal circulant and other Abelian Cayley graphs," Open Univ., London, U.K., Ph.D. dessertation, 2021, Art. no. 390, doi: 10.21954/ou.ro.00013612.

[16] M. Nabi-Abdolyousefi and M. Mesbahi, "On the controllability properties of circulant networks," *IEEE Trans. Automat. Control*, vol. 58, no. 12, pp. 3179–3184, Dec. 2013, doi: 10.1109/TAC.2013.2259992.

[17] K. Reji Kumar and G. MacGillivray, "Efficient domination in circulant graphs," *Discrete Math.,* vol. 313, no. 6, pp. 767–771, 2013, doi: 10.1016/j.disc.2012.12.003.

[18] R. R. Lewis, "The degree-diameter problem for circulant graphs of degrees 10 and 11," *Discrete Math.,* vol. 341, no. 9, pp. 2553–2566, Sep. 2018, doi: 10.1016/J.DISC.2018.05.024.

[19] R. El-Shanawany and A. El-Mesady, "On orthogonal labelling for the orthogonal covering of the circulant graphs," *Malaysian J. Math. Sci.*, vol. 12, no. 2, pp. 161–173, 2018.

[20] A. El-Mesady, Y. S. Hamed, and H. Shabana, "On the decomposition of circulant graphs using algorithmic approaches," *Alexandria Eng. J.*, vol. 61, no. 10, pp. 8263–8275, Oct. 2022, doi: 10.1016/j.aej.2022.01.049.

[21] A. El-Mesady, O. Bazighifan, and Q. Al-Mdallal, "On infinite circulant-balanced complete multipartite graphs decompositions based on generalized algorithmic approaches," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 11 267–11 275, 2022, doi: 10.1016/j.aej.2022.04.022.

[22] R. Hoffman, D. Diserable, and F. Seredynski, "Cellular automata rules solving the wireless sensor network coverage problem," *Natural Comput.*, vol. 21, pp. 417–447, Jun. 2022, doi: 10.1007/s11047-022-09888-0.

[23] A. Y. Romanov, E. V. Lezhnev, A. Y. Glukhikh, and A. A. Amerikanov, "Development of routing algorithms in networks-on-chip based on two-dimensional optimal circulant topologies," *Heliyon*, vol. 6, no. 1, Jan. 2020, Art. no. e03183, doi: 10.1016/j.heliyon.2020.e03183.

[24] E. A. Monakhova, A. Y. Romanov, and E. V. Lezhnev, "Shortest path search algorithm in optimal two-dimensional circulant networks: Implementation for networks-on-chip," *IEEE Access*, vol. 8, pp. 215 010–215 019, 2020, doi: 10.1109/ACCESS.2020.3040323.

[25] A. Y. Romanov, "Development of routing algorithms in networks-on-chip based on ring circulant topologies," *Heliyon*, vol. 5, no. 4, Apr. 2019, Art. no. e01516, doi: 10.1016/J.HELIYON.2019.E01516.

[26] C. Martínez, E. Vallejo, R. Beivide, C. Izu, and M. Moretó, "Dense Gaussian networks: Suitable topologies for on-chip multiprocessors," *Int. J. Parallel Program.*, vol. 34, no. 3, pp. 193–211, 2006, doi: 10.1007/s10766-006-0014-1.

[27] R. Lu, "Fast methods for designing circulant network topology with high connectivity and survivability," *J. Cloud Comput.*, vol. 5, no. 1, 2016, Art. no. 5, doi: 10.1186/s13677-016-0056-x.

[28] E. Monakhova and O. Monakhov, "A generalized routing algorithm for a family of optimal 2D circulant networks based on relative addressing," in *Proc. IEEE 17th Int. Asian Sch.-Seminar Optim. Problems Complex Syst.*, 2021, pp. 55–59, doi: 10.1109/OPCS53376.2021.9588737.

[29] E. A. Monakhova, "On the analytical description of the optimal two-dimensional diophantine structures of homogeneous computing systems," *Comput. Syst. Quest. Theory Constr. Comput. Syst.*, vol. 90, pp. 81–91, 1981.

[30] R. Beivide, A. Arruabarrena, E. Herrada, and J. L. Balcazar, "Optimal distance networks of low degree for parallel computers," *IEEE Trans. Comput.*, vol. 40, no. 10, pp. 1109–1124, Oct. 1991, doi: 10.1109/12.93744.

[31] J. L. A. Yebra, M. A. Fiol, P. Morillo, and I. Alegre, "The diameter of undirected graphs associated to plane tessellations," *Ars Comb.*, vol. 20-B, pp. 159–171, 1985.

[32] H. Perez-Roses, M. Bras-Amoros, and J. M. Seradilla-Merinero, "Greedy routing in circulant networks," *Graphs Combinatorics*, vol. 38, no. 86, Apr. 2022, doi: 10.1007/s00373-022-02489-9.

[33] B. AIBdaiwi, Z. Hussain, A. Cerny, and R. Aldred, "Edge-disjoint node-independent spanning trees in dense Gaussian networks," *J. Supercomputing*, vol. 72, pp. 4718–4736, Jun. 2016, doi: 10.1007/s11227-016-1768-x.

[34] B.-X. Chen, J.-X. Meng, and W.-J. Xiao, "A constant time optimal routing algorithm for undirected double-loop networks," in *Proc. Int. Conf. Mobile Ad-Hoc Sensor Netw.*, 2005, vol. 3794, pp. 308–316, doi: 10.1007/11599463_31.

[35] B. Robic, "Optimal routing in 2-jump circulant networks," Univ. Cambridge, Cambridge, U.K., Tech. Rep. N397, 1996.

[36] T. Dobravec, J. Žerovnik, and B. Robič, "An optimal message routing algorithm for circulant networks," *J. Syst. Archit.*, vol. 52, no. 5, pp. 298–306, 2006, doi: 10.1016/j.sysarc.2005.12.003.

[37] D. Gómez, J. Gutierrez, Á. Ibeas, C. Martínez, and R. Beivide, "On finding a shortest path in circulant graphs with two jumps," in *Proc. Int.Comput. Combinatorics Conf.*, 2005, vol. 3595, pp. 777–786, doi: 10.1007/11533719_79.

[38] J. Žerovnik, B. Robič, and T. Dobravec, "Optimal permutation routing in 2-jump circulant networks," in *Proc. 1st Int. Conf. Softv. Eng. Appl. Netw. Parallel/Distrib. Comput. (SNPD)*, 2000, pp. 175–180.

[39] C. Martínez, R. Beivide, E. Stafford, M. Moretó, and E. M. Gabidulin, "Modeling toroidal networks with the Gaussian integers," *IEEE Trans. Comput.*, vol. 57, no. 8, pp. 1046–1056, Aug. 2008, doi: 10.1109/TC.2008.57.

[40] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959, doi: 10.1007/BF01386390.

[41] C. Martínez, E. Vallejo, M. Moretó, R. Beivide, and M. Valero, "Hierarchical topologies for large-scale two-level networks," *XVI Jornadas Paralelismo*, pp. 133–140, 2005.

[42] O. Alsaleh, B. Bose, and B. Hamdaoui, "On-to-many node-disjoint paths routing in dense Gaussian networks," *Comput. J.*, vol. 58, no. 2, pp. 173–187, Feb. 2015, doi: 10.1093/comjnl/bxt142.

[43] K.-J. Pai, J.-S. Yang, G.-Y. Chen, and J.-M. Chang, "Configuring protection routing via completely independent spanning trees in dense Gaussian on-chip networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 2, pp. 932–946, Mar./Apr. 2022, doi: 10.1109/tnse.2022.3140329.

[44] J. Y. Cai, G. Havas, B. Mans, A. Nerurkar, J. P. Seifert, and I. Shparlinski, "On routing in circulant graphs," in *Proc. Int. Comput. Combinatorics Conf.*, 1999, vol. 1627, pp. 360–369, doi: 10.1007/3-540-48686-0_36.

[45] P. K. Jha, "Dimension-order routing algorithms for a family of minimal-diameter circulants," *J. Interconnect. Netw.*, vol. 14, no. 1, Mar. 2013, Art. no. 1350002, doi: 10.1142/S0219265913500023.

[46] J. Žerovnik and T. Pisanski, "Computing the diameter in multiple-loop networks," *J. Algorithms*, vol. 14, no. 2, pp. 226–243, 1993, doi: 10.1006/JAGM.1993.1011.

Emilia A. Monakhova received the M.S. degree in mathematics and applied mathematics from Novosibirsk State University, Novosibirsk, Russia, and the Ph.D. degree in computer science from Leningrad Electrotechnical Institute, Saint Petersburg,, Russia, in 1988. She is currently a Leading Scientific Researcher with the System Modeling and Optimization Laboratory, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, Novosibirsk, Russia. She is the author of two books and more than 100 articles. Her research interests include network topologies, parallel computer architecture, routing algorithms, and evolutionary computation.

Oleg G. Monakhov received the M.S. degree in system analysis from Novosibirsk Electrotechnical Institute, Novosibirsk, Russia, in 1974, and the Ph.D. degree in computer science from Taganrog State Radio Technical University, Taganrog, Russia in 1987. From 2000 to 2002, he was a Visiting Professor with the University of Aizu, Aizu-Wakamatsu, Japan. He is currently a Leading Scientific Researcher with the System Modeling and Optimization Laboratory, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, Novosibirsk, Russia. He is the author of two books and 120 articles. His research interests include parallel computer architecture, network topologies, routing algorithms, genetic programming, metaheuristic algorithms.

Aleksandr Y. Romanov (Member, IEEE) received the B.S. and M.S. degrees in computer systems and networks from National Technical University Kharkov Polytechnic Institute, Kharkov, Ukraine, in 2007 and 2009, respectively, and the Ph.D. degree in technical sciences from the Federal State-Funded Institution of Science Institute for Design Problems in Microelectronics of Russian Academy of Sciences (IPPM RAS), Zelenograd, Russia, in 2015. He is currently an Associate Professor with National Research University Higher School of Economics (HSE University), Moscow. He is a University Teacher of the following subjects: SoC Development, System Design of Electronic Devices. He is the author of the book *Digital Design: Practical Course*, he has also authored or coauthored more than 100 research articles. His basic scientific research interests include SoCs, NoCs, embedded systems, FPGA, and neural networks. He is an IEEE Industrial Electronics Society Member.