KHARKOV NATIONAL UNIVERSITY OF RADIOELECTRONICS

Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2012)

Technically Co-Sponsored by

Kharkov, Ukraine, September 14 – 17, 2012

IEEE EAST-WEST DESIGN AND TEST SYMPOSIUM 2012 COMMITTEE

General Chairs
V. Hahanov – Ukraine
Y. Zorian – USA

General Vice-Chairs
R. Ubar - Estonia
P. Prinetto - Italy

Program Chairs
S. Shoukourian – Armenia
D. Speranskiy – Russia

Program Vice-Chairs
Z. Navabi – Iran
M. Renovell – France

Publicity Chair's
G. Markosyan - Armenia
S. Mosin - Russia

Public Relation Chair
V. Djigan - Russia

Program Committee
E. J. Aas - Norway
J. Abraham - USA
M. Adamski - Poland
A.E.Mohamed Mohamed - Egypt
A . Barkalov - Poland
R. Bazylevych - Ukraine
A. Chaterjee - USA
V. Djigan - Russia
A. Drozd - Ukraine
E. Evdokimov - Ukraine
E. Gramatova - Slovakia
A. Ivanov - Canada
M. Karavay - Russia
V. Kharchenko - Ukraine
K. Kuchukjan - Armenia
W. Kuzmicz - Poland
A. Matrosova - Russia
V. Melikyan - Armenia
L. Miklea - Romania
O. Novak - Czech Republic
Z. Peng - Sweden
A. Petrenko - Ukraine
J. Raik - Estonia
A. Romankevich - Ukraine
A. Ryjov - Russia
R. Seinauskas - Lithuania
S. Sharshunov - Russia
A. Singh - USA
J. Skobtsov - Ukraine
V. Tverdokhlebov - Russia
V. Vardanian - Armenia
V. Yarmolik - Byelorussia

Steering Committee
M. Bondarenko - Ukraine
V. Hahanov - Ukraine
R. Ubar - Estonia
Y. Zorian - USA

Organizing Committee
S. Chumachenko - Ukraine
E. Litvinova – Ukraine
V. Kharchenko – Ukraine

EWDTS 2012 CONTACT INFORMATION

Prof. Vladimir Hahanov
Design Automation Department
Kharkov National University of Radio Electronics,
14 Lenin ave,
Kharkov, 61166, Ukraine.

Tel.: +380 (57)-702-13-26
E-mail: hahanov@kture.kharkov.ua
Web: www.ewdtest.com/conf/

10th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS 2012)

Kharkov, Ukraine, September 14-17, 2012

The main target of the **IEEE East-West Design & Test Symposium** (EWDTS) is to exchange experiences between scientists and technologies of Eastern and Western Europe, as well as North America and other parts of the world, in the field of design, design automation and test of electronic circuits and systems. The symposium is typically held in countries around the Black Sea, the Baltic Sea and Central Asia region. We cordially invite you to participate and submit your contributions to EWDTS'12 which covers (but is not limited to) the following topics:

- Analog, Mixed-Signal and RF Test
- Analysis and Optimization
- ATPG and High-Level Test
- Built-In Self Test
- Debug and Diagnosis
- Defect/Fault Tolerance and Reliability
- Design for Testability
- Design Verification and Validation
- EDA Tools for Design and Test
- Embedded Software Performance
- Failure Analysis, Defect and Fault
- FPGA Test
- HDL in test and test languages
- High-level Synthesis
- High-Performance Networks and Systems on a Chip
- Low-power Design
- Memory and Processor Test
- Modeling & Fault Simulation
- Network-on-Chip Design & Test
- Modeling and Synthesis of Embedded Systems
- Object-Oriented System Specification and Design
- On-Line Testing
- Power Issues in Design & Test
- Real Time Embedded Systems
- Reliability of Digital Systems
- Scan-Based Techniques
- Self-Repair and Reconfigurable Architectures
- Signal and Information Processing in Radio and Communication Engineering
- System Level Modeling, Simulation & Test Generation
- System-in-Package and 3D Design & Test
- Using UML for Embedded System Specification
- CAD and EDA Tools, Methods and Algorithms
- Design and Process Engineering
- Logic, Schematic and System Synthesis
- Place and Route
- Thermal, Timing and Electrostatic Analysis of SoCs and Systems on Board
- Wireless and RFID Systems Synthesis
- Digital Satellite Television

The Symposium will take place in Kharkov, Ukraine, one of the biggest scientific and industrial center. Venue of EWDTS 2012 is Kharkov National University of Radioelectronics was founded 81 years ago. It was one of the best University of Soviet Union during 60th - 90th in the field of Radioelectronics. Today University is the leader among technical universities in Ukraine.

The symposium is organized by Kharkov National University of Radio Electronics and Science Academy of Applied Radio Electronics http://anpre.org.ua/ in cooperation with Tallinn University of Technology. It is technically co-sponsored by the IEEE Computer Society Test Technology Technical Council (TTTC) and financially supported by Trades Committee of Kharkov National University of Radioelectronics and Trades Committee of Students, Aldec, Synopsys, Kaspersky Lab, DataArt Lab, Tallinn Technical University.

Technically Co-Sponsored by

CONTENTS

Models for Quality Analysis of Computer Structures

Murad Ali Abbas (Iraq), Chumachenko S.V., Hahanova A.V., Gorobets A.A., Priymak A.
Computer Engineering Faculty, Kharkov National University of Radioelectronics, Kharkov, Ukraine, hahanov@kture.kharkov.ua

Abstract

The methods for estimating computational structures and searching the shortest paths between the pair of nodes are presented. A criterion for evaluating the effectiveness of computational structures based on the graph model of the functional blocks of digital systems-on-chips is developed. A modified Dijkstra's algorithm to determine the average cost of interconnections in computing architecture for every pair of nodes is proposed. Verification of the criterion, when evaluating the effectiveness of different topologies of computational structures is performed.

1. Introduction

Creating effective computational structures is related not only to increasing the speed of primitives, but also with the topology of interconnections between them, which can significantly improve the performance of parallel processing data due to additional expensive connections. It is therefore necessary to have criteria for evaluating performance, taking into account not only transaction time between the nodes, but the hardware redundancy, which considerably reduces the average time of receiving and transmission of information between the primitive computing components. Such criteria can be used to evaluate the effectiveness of graph models of local and global computer networks, urban infrastructure of road communications, as well as the traffic flows in order to identify bottlenecks affecting the traffic. The problem of finding such criteria is related to the minimization of the computational cost for determination of all possible minimal paths between nodes of pairs.

The aim of research is development of criteria for evaluating the effectiveness of computational structures, based on graph model of interconnections of functional blocks, which make it possible to determine the quality of the topological architectures of digital systems-on-chips.

The objectives: 1) Analysis of methods for estimating the computational structures and finding the shortest paths between nodes of pair [5-9]. 2) Development of criteria for evaluating the effectiveness of computational structures, based on graph model of the functional blocks of digital systems-on-chips [1-4]. 3) Modification of Dijkstra's algorithm to determine the average cost of interconnections of computing architecture for a node pair [5-8]. 4) Verification of the criteria when evaluating the effectiveness of different topologies of computational structures [1-4].

2. Estimating the connection topology of digital system components

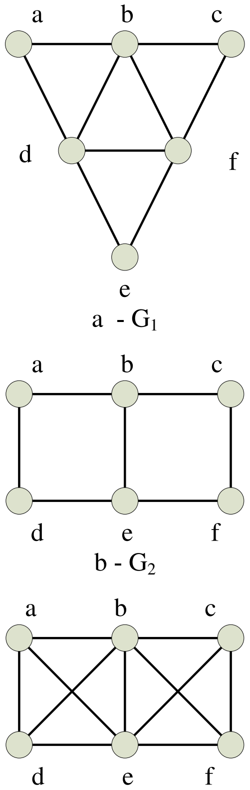
The distance between the components of a digital system is the main parameter that affects the speed of executing (for functionality or service) transactions between the components or structure elements. When considering two implementation variants of a multiprocessor system, for example, it is necessary to determine the integral characteristic as the sum of all distances between each pair of components or nodes of the corresponding graph. Due to the existence of such estimation, a natural question is appeared: which geometric (topological) primitive shapes have to be used to minimize the integral estimation of distances between each pair of nodes? Here, three variants of shapes are interested: rectangle (the metric of «Manhattan»), triangle and tetrahedron. The last one has unique feature – each node of the tetrahedron has three neighbors, while the triangle has only two adjacent nodes.

The feasibility attractiveness of analysis of structure performance is important not only for digital systems, networks, telecommunications, but also urban infrastructure in the existence of congestion.

The criterion is related to processing graph structure, which has E arcs and n nodes. Its feature lies in the calculation of absolute value that is not reduced to the interval and formed by the cost of connections multiplied by the quality of transactions between all pairs of nodes:

$$Q_4 = \frac{E}{n} \times \sum_{i=1}^{n} \min_{j} (p_{ij}) \cdot$$

Use this formula for the estimation of three graph structures with six nodes and different connecting topologies is shown in Fig. 1.



a - G_1

b - G_2

c - G_3

Fig. 1. Structures of connections for processor primitives

Here are three graphs, which have 9, 7 and 11 arcs, respectively. Defining criterion in accordance with the last formula gives the following results:

$$Q_4(G_1) = \frac{E}{n} \times \sum_{i=1}^{n} \min_{j}(p_{ij}) = \frac{9}{6} \times (9 \times 1 + 6 \times 2) = 31{,}5;$$

$$Q_4(G_2) = \frac{7}{6} \times (7 \times 1 + 6 \times 2 + 2 \times 3) = 29{,}2;$$

$$Q_4(G_3) = \frac{11}{6} \times (11 \times 1 + 4 \times 2) = 34{,}8$$

Modification of effectiveness estimation of the topology is related to reduction of real costs (number of arcs E) to the maximum possible number $V = \frac{n^2 - n}{2}$ of pair connections in the graph $\frac{E}{V}$, which provide the quality of communication properties $\frac{V}{\sum\limits_{i=1}^{V} \min\limits_{j}(p_{ij})}$:

$$Q_4 = \frac{E}{\frac{n^2 - n}{2}} \times \frac{\frac{n^2 - n}{2}}{\sum\limits_{i=1}^{\frac{n^2-n}{2}} \min\limits_{j}(p_{ij})} = \frac{E}{\sum\limits_{i=1}^{\frac{n^2-n}{2}} \min\limits_{j}(p_{ij})} = \frac{E}{\sum\limits_{i=1}^{V} \min\limits_{j}(p_{ij})}$$

The estimation is equal to one if the numerator and denominator are equal $V = \frac{n^2 - n}{2}$. In this case, the graph structure has all possible pair wise connections between the graph nodes, defined by a half of the Cartesian square of the power of a node set minus n nodes. Subtraction is defined by the nodes of a graph, which don’t have idempotent closures. At that each pair of nodes has the path length equal to one. Recalculation of efficiency criteria for interconnections of processor primitives gives the following result:

$$Q_5(G_1) = \frac{E}{\sum\limits_{i=1}^{V=12} \min\limits_{j}(p_{ij})} = \frac{9}{9 \times 1 + 6 \times 2} = 0{,}428;$$

$$Q_5(G_2) = \frac{7}{7 \times 1 + 6 \times 2 + 2 \times 3} = 0{,}28;$$

$$Q_5(G_3) = \frac{11}{11 \times 1 + 4 \times 2} = 0{,}578\ .$$

At the same time pay for the quality of communication is the power of connections, reduced to the maximum possible number of arcs:

$$H_5(G_1) = \frac{E}{\frac{n^2 - n}{2}} = \frac{9}{15} = 0{,}60;$$

$$H_5(G_2) = \frac{7}{15} = 0{,}46; H_5(G_3) = \frac{11}{15} = 0{,}73.$$

It is advisable to have two estimates: the integral criterion of communication quality, which implicitly defines the time of average reachability between every node pair of the graph structure, as well as the power of connections reduced to the maximum possible number and determining the cost of infrastructure quality, objective function of which minimizes the average reachability (path length or time) between node pair of the graph structure. Depending on the number of connections the first criterion tends to increase from 0 to 1, the second one also increases as the number of arcs in the graph. Therefore, multiplexing of two criteria does not give a new property when estimating the reachability infrastructure for each pair of nodes. Conclusions: 1) It is necessary to use both criteria for the estimation of structural design. 2) Dijkstra's algorithm should be modified to calculate the average value of the reachability between a pair of nodes in the graph as shown below.

3. Modification of Dijkstra's algorithm

To solve the problem of finding the shortest paths between all pairs of nodes in a weighted graph Johnson's and Floyd-Uorshell's algorithms are used. Both algorithms are applied to directed graphs. Johnson's algorithm is implemented in the presence in the graph of arcs with a positive or negative weight, but in the absence of cycles with negative weight. It is known that the complexity of Floyd's (Floyd-Uorshell's) algorithm for finding the shortest paths between all pairs of nodes is $O(n^3)$.

When finding the shortest paths from one node of the graph to all other nodes the Danzig's, Levit's, Bellman-Ford's, Dijkstra's algorithms are used.

Danzig's algorithm is applied to planar directed graphs, similar to Floyd-Uorshell's algorithm, but differs from it in a different order of executing the same instructions.

Levit's algorithm is applied for oriented/nonoriented graphs without arcs with negative weight.

Bellman-Ford's algorithm is designed to find the shortest path in a weighted oriented or undirected graph and permits for the presence of arcs with negative weight, in contrast to Dijkstra's algorithm. During the time of $O(|V|\times|E|)$ algorithm finds the shortest paths from one node to all others.

Dijkstra's algorithm finds the shortest distances from one of the nodes of the graph to all the others. It applies only to graphs with arcs of positive weight. The algorithm is widely used in programming and technologies. For example, in a dynamic routing protocol Open Shortest Path First (OSPF), which is based on the technology of tracking the status of the channel (link-state technology), to eliminate the circular routes.

Dijkstra proposed a modification of the algorithm for constructing the shortest paths from a given node of the graph to all others nodes, which reduced the number of operations (additions and comparisons), retaining the information at one stage for the next ones. This is achieved by procedure for placing Dijkstra's labels that reduces the complexity of the algorithm to $O(n^2)$ [5-9].

In some cases, when the configuration of the graph permits it, for finding the shortest paths between all pairs of nodes it is advisable to apply Dijkstra's algorithm for each of n possible initial nodes.

Problem. Find the shortest paths between all pairs of nodes for the graph shown in Fig. 1a with unit weights for each arc.

Solution. Considered undirected graph involves two equal nodes groups: 1) a, c, e; 2) b, d, f. The term "equal" here means the invariance of the representation of the shortest paths and their trees when finding from the nodes a, c, e (the first group of paths), or from the nodes b, d, f (a second group of paths). Thus, for a given problem, instead of Floyd's algorithm Dijkstra's algorithm can be applied twice to find the shortest paths between all pairs of nodes. Then the problem is divided in two subproblems: 1) find the shortest distances and specify all the shortest paths from the node a to all other nodes; 2) find the shortest distances and specify all the shortest paths from the node b to all other nodes.

The adjacency matrix of graph G_1 has the form:

$$G_1 = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & . & 1 & . & . & . & 1 \\ b & 1 & . & 1 & 1 & . & 1 \\ c & . & 1 & . & 1 & . & . \\ d & . & 1 & 1 & . & 1 & 1 \\ e & . & . & . & 1 & . & 1 \\ f & 1 & 1 & . & 1 & 1 & . \end{array} \qquad (1)$$

Subproblem 1.1. Find the shortest distances and specify all the shortest paths from the node a to all other nodes.

During the implementation of Dijkstra's algorithm Table 1 is filled, the number of rows and columns is determined by the power of the graph node set (6x6). In Table 1, row headers indicate the nodes, to which the shortest distance have to be found.

Table 1

	1	**2**	**3**
	u=a **r=0**	**u=b** **r=1**	**u=f** **r=1**
b	a, 1		
c	a, ∞	b,2	b,2
d	a, ∞	b,2	b,2
e	a, ∞	a, ∞	f,2
f	a,1	a,1	

Comments: 1) If a column includes two nodes with the same minimum numeric labels then any of them is selected. This means that there may be two different chains of the same minimal length. 2) If when calculating the current numerical labels a new total distance is equal to the previous one, then the old numeric label is saved. 3) Permanently labeled nodes in the column headers are not repeated. 4) The distances in the column headers do not decrease ($\leq$). 5) Current labels in lines do not increase ($\geq$).

Thus, Table 1 contains information about all the shortest chains and their lengths.

For example, it is necessary to find the shortest chain from the node a to the node e. The sequence of nodes of the chain is written from the end; the last filled cell of the row e contains information about the length of a shortest path r = 2 and last but one node of the chain - f. Information about the previous node is in the last cell of the row f, in this case – the node a, which is the beginning of the route:

$$a \xrightarrow{1} f \xrightarrow{1} e \Rightarrow \text{dist} = 1+1=2\,.$$

The data for all the shortest paths are presented in Tab. 2.

Table 2

Chain	Length
$a \xrightarrow{1} b$	$r(a,b)=1$
$a \xrightarrow{1} b \xrightarrow{1} c$	$r(a,c)=1+1=2$
$a \xrightarrow{1} b \xrightarrow{1} d$	$r(a,d)=1+1=2$
$a \xrightarrow{1} f \xrightarrow{1} e$	$r(a,e)=1+1=2$
$a \xrightarrow{1} f$	$r(a,f)=1$

A graph demonstrating the shortest paths tree from the node a is shown in Fig. 2

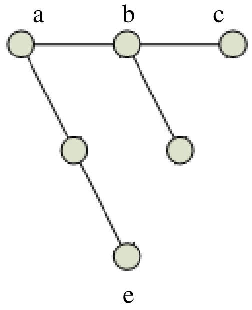


Fig. 2. The shortest paths tree from the node a

Since all the arcs of the graph (see Fig. 1a) have a weight of 1, Tab. 1 shows that when calculating the distances they can be increased only by 1. So, in fact, as soon as the infinite label is changed to finite numerical label, later it has not been changed. This means that the corresponding shortest distance between the nodes has been determined. Then the number of additions and comparisons in Dijkstra’s algorithm can be reduced (Tab. 3).

Table 3

	1	**2**	**3**
	u=a r=0	**u=b r=1**	**u=f r=1**
b	a, 1		
c	a, ∞	b,2	
d	a, ∞	b,2	
e	a, ∞	a, ∞	f,2
f	a,1	a,1	

It should be noted that the shortest routes from the nodes a, c, e, and the trees of the shortest paths will be identical.

Subproblem 1.2. For the graph shown in Fig. 1a it is necessary to find the shortest distances and specify all the shortest paths from the node b to all other nodes.

During the implementation of Dijkstra's algorithm the table is filled, where the number of rows and columns is determined by the power of the graph node set (6x6). In the headers of table rows the nodes, to which is necessary to find the shortest distance are indicated (Tab. 4). Below a modified table is considered.

Table 4

	1	**2**
	u=b r=0	**u=f r=1**
a	b,1	b,1
c	b,1	b,1
d	b,1	b,1
e	b, ∞	f,2
f	b,1	

Table 4 presents the calculation subject to the modification of Dijkstra's algorithm. At the initial stage after the placement of labels in the first column the shortest distances from the node b to the other graph nodes are defined, except the node e. They are equal to the length of arc 1; and themselves the shortest chains coincide with the arcs that connect node b with the nodes a, c, d, f. It remains to determine the shortest distance and the chain from b to f. To do this, we continue the algorithm step by step – choose the minimum label from all finite numerical labels in the first column. Since they are all equal to 1, select anyone. The choice of nodes d or f will complete the process of finding for a single pass.

The data for all the shortest paths are shown in Tab. 5. A graph demonstrating the tree of the shortest paths from the node b is shown in Fig. 3.

It should be noted that the shortest chains from the nodes b, d, f and the trees of the shortest paths will be identical.

Table 5

Chain	Length
$b \xrightarrow{1} a$	$r(b,a) = 1$
$b \xrightarrow{1} c$	$r(b,c) = 1$
$b \xrightarrow{1} d$	$r(b,d) = 1$
$b \xrightarrow{1} d \xrightarrow{1} e$	$r(b,e) = 1 + 1 = 2$
$b \xrightarrow{1} f$	$r(b,f) = 1$

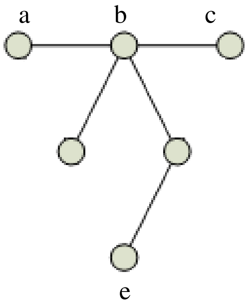


Fig. 3. The shortest paths graph from the node b

The matrix of shortest distances between all pairs of nodes of the graph G_1 is presented below:

$$Dist_1 = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & . & 1 & 2 & 2 & 2 & 1 \\ b & 1 & . & 1 & 1 & 2 & 1 \\ c & 2 & 1 & . & 1 & 2 & 2 \\ d & 2 & 1 & 1 & . & 1 & 1 \\ e & 2 & 2 & 2 & 1 & . & 1 \\ f & 1 & 1 & 2 & 1 & 1 & . \end{array} \qquad (2)$$

4. Description of the modified Dijkstra's algorithm

Each node of the set of nodes V is associated with a label that specifies the minimum well-known distance from it to the initial node a. The algorithm is executed step by step. At each step it «visited» one node and tries to reduce the labels. The implementation of the algorithm terminates when all nodes are visited.

Initialization. The label of the node a is equal to 0, a temporary label – infinity is assigned to other nodes. This means that the distances from the node a to other ones yet unknown. All nodes are marked as unvisited.

Step of the algorithm. If all the nodes are visited, the algorithm terminates. Otherwise, from the nodes have not yet visited the node u with a minimum label is chosen. At that all possible routes, where u is last but one point, are considered. Nodes, to which the arcs from u are directed, are called adjacent with respect to u. For each neighbor of the node u, except ones marked as visited, a new path length is considered that is equal to the sum of the values of the current label of the node u and the length of the arc connecting u with this neighbor.

In the traditional Dijkstra's algorithm such step is performed further: if obtained value of the length is less than value of neighbor's label, neighbor's label is replaced by obtained value of length.

For graphs with arcs of unit length (weight) each time the sum of the distances can be increased only by 1. So in the above step only infinite neighbor's labels can be changed on the finite numerical label, which subsequently are not changed (they cannot be reduced). The corresponding distances are natural numbers. For this reason, the comparison should be carried out to determine the finite number labels for those nodes, which do not have them, that is, their temporary labels are infinite. If there is no arc connecting permanently labeled node with the node having infinite label, as the next step the permanently labeled node with minimal label in given column is chosen (as above), which allows making attempt to find for minimal route through another node. At that addition and comparison of labels with existing in the column finite labels is not carried out, which reduces the time of finding.

After consideration of all the neighbors the node u is marked as visited and the step of the algorithm is repeated.

LABEL is array to store the current node labels. PERM is an array to specify the permanently labeled nodes (nodes are permanently labeled when they are equal to u_i for some i). If the PERM(v)=1, v is permanently labeled node and its label is equal to d(s, v). First, PERM(s)=1 and PERM(v)=0 for v≠s. PRED is an array of pointers to the nodes from which passing to the nodes with permanent labels is performed. If the node v is permanent labeled then the sequence v, PRED(v), PRED(PRED(v)),..., s involves the nodes which are shortest directed path from s to v.

1. Begin. Let LABEL(s)=0, PERM(s)=1, PRED(s)=s; ∀v≠s let LABEL(v)=∞, PERM(v)=0, PRED(v)=v.

2. Let i=0, u=s (u is last of the nodes with constant label. Now it is the node s).

3. Calculating LABEL and changing the elements of the array PRED. Let i=i+1.

For each node v with infinite label it is necessary to carry out the following actions (in the traditional Dijkstra's algorithm this point was performed for all nodes v, except the nodes with the same label; and in the modified algorithm it applies only to the nodes with temporary labels LABEL(v)=∞, because another labels will not be modified):

3.1. Let M=min{LABEL(v), LABEL(u)+w(u,v)}, where w(u,v)=1 is the length of an arc, connecting the nodes u and v if it exists, in other case (when the arc (u,v) is not exist) the node with minimum finite numerical label is chosen as permanent labeled node; if there are several such nodes one of them is chosen;

3.2. If M<LABEL(v) then LABEL(v)=M, PRED(v)=u.

4. Choosing the node u_i. Among all the nodes which are not labeled, find the node w with the minimum label (if there are several nodes, the choice can be made arbitrarily.) Let PERM(w)=1, u_i=w (it is the last node with invariable label).

5. If i<n-1, go to item 3, in other case – end (all shortest paths are found).

Labels of nodes are the lengths of shortest paths; v, PRED(v), PRED(PRED(v)), …, s are the nodes of the shortest directed s-t path.

4. Conclusion

The scientific novelty. The criteria of evaluating the quality of the topological connections of digital systems components, which are focused on the evaluation of projects, from the standpoint of operational and strategic minimizing of connecting routes for two nodes are proposed. They make it possible to modify the structures by introducing additional costs on some connections in the corresponding graph in order to improve the system performance. Modification of Dijkstra's algorithm for undirected weighted graphs with unit arc length is developed. It reduces the number of additions and comparisons due to the exclusion from this process the finite numerical labels, found in the previous step, which cannot be decreased hereafter and remain constant, as well as the possibility of transformation of infinite neighbor's labels only to the finite numerical ones.

The practical value. The feasibility attractiveness of analyzing the effectiveness of the topology for the component connections in the structures is important not only for digital systems, networks, telecommunications, but urban infrastructure in the existence of congestion. Areas for further research in this area are associated with the following problems: development of analytical models for estimating the efficiency of the infrastructure IP for embedded repairing and servicing the combinational digital systems with different complexity levels of the primitives and structures.

10. References

[1] *Reliability* of Technical Systems / Ed. I.A.Ushakova. Moscow, 1985. 512 (in Russian)

[2] *Hahanov V.I., Litvinova E.I., Chumachenko S.V., Guz O.O.* Logical associative computer // J. Electronic simulation. .№ 1. 2011. C. 73-90 (in Russian).

[3] *Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S.* Information analysis infrastructure for diagnosis. Information. An international interdisciplinary journal. 2011. Japan. Vol. 14. No 7. P. 2419-2433.

[4] *Hahanov V.I.* and others. Infrastructure of intellectual property for SoC simulation and diagnosis service. Springer, Germany, 2011. P. 289-330.

[5] *Dijkstra E. W.* A note on two problems in connexion with graphs // Numerische Mathematik. Vol. 1. 1959. P. 269-271.

[6] *Stern T.H, Leiserson Charles I., Rivest R.L, Stein C.* Algorithms: The Design and Analysis = Introduction to Algorithms. M. Williams, 2006. 1296 p/ (in Russian)

[7] *Anany V. Levitin.* Introduction to Design and Analysis of Algorithms. Villanova University. 2003. 576 p.

[8] *Kuznetsov N.A., Fetisov V.N.* Dijkstra's algorithm with improved robustness to control the routing of IP-based networks // Automation and Remote Control. 2008. № 2. C. 80–85 (in Russian).

[9] *Thomas T.* OSPF Network Design Solutions. Cisco Press, 2004. 816 p.

Camera-ready was prepared in Kharkov National University of Radio Electronics
Lenin Ave, 14, KNURE, Kharkov, 61166, Ukraine

Approved for publication: 27.08.2012. Format 60×841/8.
Relative printer’s sheets: 49. Circulation: 200 copies.
Published by SPD FL Stepanov V.V.
Ukraine, 61168, Kharkov, Ak. Pavlova st., 311

Матеріали симпозіуму «Схід-Захід Проектування та Діагностування – 2012»
Макет підготовлено у Харківському національному університеті радіоелектроніки
Редактори: Володимир Хаханов, Світлана Чумаченко, Євгенія Литвинова
Пр. Леніна, 14, ХНУРЕ, Харків, 61166, Україна

Підписано до публікації: 27.08.2012. Формат $60\times84{}^{1}/_{8}$.
Умов. друк. Арк. 49. Тираж: 200 прим.
Видано: СПД ФЛ Степанов В.В.
Вул. Ак. Павлова, 311, Харків, 61168, Україна