Computer Science Journal of Moldova, vol.**6**, no.2(17), 1998

High level of synthesis parallel problem-oriented TS — coordinated digital computing systems

V. Kondrat'ev Y. Bazhanov A. Balabanov

Abstract

Exemple abstact. C'est une place pour remplacer notre abstract

1 Introduction

The development of parallel digital computing systems (DCS) is a main direction in development of architecture of the computers which succeets the on change of traditional principles of machines designing of consecutive action machines based on a model offered by von Neumann.

Creation of DCS with parallel structures now moves along the direction of design both universal DCS used for solution of a broad circle of tasks, and problem-oriented DCS, intended for more effective solution of the narrow class of tasks or even one specific task.

Parallel universal DCS allow to use only universal kind of parallelism. Parallel problem-oriented DCS, unlike universal one, in principle allow to establish full coordination between potential parallelism contained in the task and its realization in the real computing environment.

Therefore in process of designing of such systems there is rather actual task of structural-temporal coordination (we shall name such coordination the TS-coordination) of all subsystems to supply balance of separate parts of DCS, at which each part has sufficient complexity and speed (neither more then this) to perform the job laid on DCS with minimum apparatus expenditure in the time that does not exceed the beforehand determined valne.

In [1,2] for TS-concordance of operational elements and devices the approach in offered, based on optimization of the network graphs with the help of effective Ford-Fulkerson-Kelley (FFC) algorithm. In the present article this approach is spreaded to the more complex class of systems,hamely parallel problem-oriented DCS. This approach allows to execute synthesis implemeiting the DCS beginning from the usual sequential flowchart of algorithm (GSA) and finishing with an parallel TS- coordinated structure of given GSA. GSA determines some mapping of inputs onto outputs and by virtue of this it is the abstract behavioral specification of DCS, that allows to consider the offered approach as a method of high level synthesis of parallel problem – oriented DCS [3].

2 Objects of TS — coordination

Let's consider the generalized block diagram of parallel problem-oriented DCS, represented at fig.1.

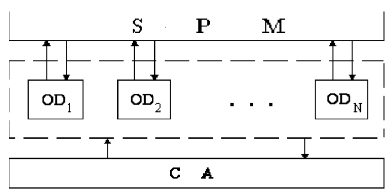

Fig.1. The generalized block-diagram parallel problem-oriented DCS

The scheme contains three main components: the system of parallel memory (SPM), operational devices $OD_1, OD_2 \ldots, OD_N$ and controlling automaton (CA).

SPM can simultaneously interact with N operational devices. In this case all conflicts concerning with installation of physical link between CA and the memory generated by necessity of simultaneous access to arbitrarily addressed cells of memory several OD, are solved inside the memory. The methods of design of such SPM are known and described, for example, in [4].

The memory capacity E necessary for algorithm implementation, can be defined according to a technique from [5] by maximum number of various variables of algorithm, considering their coincidence. The access time t_{SPM} depends E and N. Hereinafter we shall come from that dependence $t_{SPM}(E, N)$ is known.

When values of E are small SPM can contain E independent registers, and the necessary circuit of the registers choice, writing/reading of an information in/from the registers are included in appropriate OD.

Each operational device will implement a group of defined operations, specified by GSA, and represents of composition of an operational automaton and a controlling one. The methods of synthesis such OD with beforehand specified lists of operations are enough well developed and are described, for example, in [6- 8]. Problem of TS – coordination of separate OD is considered in [2]. Let's note, that each OD and CA, represented in fig.1, has the autonomous of clock frequencise generator. The frequency of these sequences can be various and is subject to definition in a process of synthesis of OD and CA.

CA will implement the parallel control process flow in the system, which only initializes the certain OD for fulfillment of specific operations. Properly execution of operation in OD happens in an autonomous mode under the control of its own controlling automaton included in the structure of OD.

The increment of switching time CA practically does not influence the execution time of algorithm. Therefore we shall consider the task of synthesis CA as the independent task, for solution which it is enough to formalize DCS the parallel controlling process flow, for example, as parallel GSA (PGSA). For logical synthesis of CA, implementing beforehand specifed PGSA, there are detailed manuals, ensuring synthesis of CA on a criterion of the apparatus costs minimum without limitations on period of clock pulses [8,9].

3 Sequential GSA

Let's define the sequential flowchart of algorithm G (as quintuple $< G, Q, W, Z, u >$):

1. Z – is the set all variables (cells in parallel memory), on values of which the algorithm G represented by its own GSA. And also $Z = X \cup Y \cup A \cup B$, where X – is the set of input variables of operator and conditional vertices of GSA; Y – is the set of output variables of operator vertices of GSA; $A(B)$ – is the set of an input (output) variables of GSA of G. The sets X, Y, A and B can intersecteach ofher.

   Let $x_i \in X, y_i \in Y, a_i \in A, b_i \in B$, then $\mathbf{X}_i, \mathbf{Y}_i, \mathbf{A}_i, \mathbf{B}_i$ are the sets of values (states) correspondent to this variables (memory cells).

2. u – is the controlling variable with a value from the set $\mathbf{U} = \{0, 1\}$.

3. Q – is the set of operator vertices (operators). The operator $q_i \in Q$ implements the mapping $\mathbf{V}_1 \times \mathbf{V}_2 \times \ldots \times \mathbf{V}_n \to \mathbf{H}$, which implements the function $h = \varphi_i(v_1, v_2, \ldots v_n)$, where $v_i \in X, i = \overline{1, n}, h \in Y, \mathbf{V}_i(\mathbf{H})$ – is the set of variables $v_i(h), \times,$ – is the Cartesian product, φ_i – are function (mapping realized by an operator) names.

   The various operators with the same function name (equivalent) are possible. Let $\varphi(i) \in F_1$ where F_1– is the set of different operators).

4. W – is the set of conditional vertices (identifiers). The identifier $w_i \in W$ implements the mapping $\mathbf{V}_1 \times \mathbf{V}_2 \times \ldots \times \mathbf{V}_n \to \mathbf{U}$, which implements the logical function $u = \psi_i(v_1, v_2, \ldots v_n)$, where $v_i \in X, i = \overline{1, n}, \psi_i$ is the name of the logical function (mapping implemented by the identifier). Identifiers also may be equivalent, i.e. have an identical name of logical function. Let $\psi_i \in F_2$ (F_2 – is the set different identifiers).

Let's assume, that F_1 and F_2 in aggregate will form functionally full set of operations $F(F = F_1 \cup F_2)$. In [7] the sets of different operations forming a functionally full set of the operations F is given. For example, $F = \{:=, +, -, *, /, =, >\}$, where $F_1 = \{:=, +, -, *, /\}, F_2 = \{=, >\}$; $:=$ – is the operation of assignment; $+, -, *, /$ are operations of addition, subtraction, multiplying and division accordingly; $>$ $(=)$ fre comparison operations – operations.

5. G – is a digraph specifying order relation on the sets Q and W. This graph should satisfy the known properties of a sequential GSA, listed, for example, in [8]. Let's remark also, that along with use at a construction of the graph G of all main structures of a type: the sequence, ifthen and IF-THEN-ELSE the organization of cycles is possible only on the "do until" type. The introduction of this limitation on organization of cycles facilitates obtaining later the acyclic form of the graph G without pendant points of graph, necessary for performance of the algorithm of the TS - coordination. At the same time, from the theory of structured programming it is known, that the structures listed abo veare anough to present any sequential GSA [10].

Any point in the graph G we shall designate as $g_i, i = \overline{0,k}$, where g_0 and g_k – are initial and final point correspondingly. Let mapping (operator) $A \to Z$, corresponds the point g_0 establishing equivalence of variables from Z and A, and mapping (operator) $Z \to B$ corresponds to point g_k.

To the other edges of a graph, outgoing from conditional points, there correspond conditions, i.e. values of variable $\mathbf{u}$, which are determined by identifiers of these points. To the remaining edges there correspond unconditional sequences of operators.

As a whole sequential GSA G describes some mapping $\mathbf{A}_1 \times \mathbf{A}_2 \times \ldots \times \mathbf{A}_{|A|} \to \mathbf{B}_1 \times \mathbf{B}_2 \times \ldots \mathbf{B}_{|B|}$, where $|X|$ – is the power of set X.

As an example we shall consider the sequential GSA, intended to calculate of the following relations:

$$\begin{cases} b_1 & = \sum\limits_{i=2}^{11} a_i, & if \quad a_1 \le 0.6, \\ b_{2,3} & = \frac{-a_{13} \pm \sqrt{a_{13}^2 - 4a_{12}a_{14}}}{2a_{12}} & if \quad a_1 > 0.6 \end{cases}$$

where $b_{2,3}$ – are the radicals of a quadratic equation:

$$a_{12}x^2 + a_{13}x + a_{14} = 0.$$

The sets Z, X, Y, A, B, Q and W for this example have the form:

$Z = \{z_i / i = \overline{1,35}\}$, $X = \{z_i / i = \overline{1,25}\} \cup \{z_i / i = \overline{28,35}\}$,

$Y = \{z_2, z_3\} \cup \{z_i / i = \overline{17,27}\}$,

$A = \{a_i / i = \overline{1,22}\}$, $B = \{b_i / i = \overline{1,3}\}$,

$Q = \{q_i / i = \overline{1,15}\}$, $W = \{w_1, W_2\}$.

The variables $z_i, i = \overline{28,35}$ accept constant values 0; 0; 6; 1; 2; 3; 4; 10; -1 accordingly.

The graph G is represented at fig.2, the operators and identifiers of which have the following interpretation:

$$g_0 : (a_1 \equiv z_1, a_2 \equiv z_4, a_3 \equiv z_5, \ldots, a_14 \equiv z_{16}, \\ a_{15} \equiv z_{28}, a_{16} \equiv z_{29}, \ldots, a_{22} \equiv z_{35}),$$

$g1 : u = \begin{cases} 1, z_1 & \le & z_{29} \\ 0, z_1 & > & z_{29} \end{cases}$ $g_7 : z_{17} = z_{31} * z_{14}$, $g_{13} : z_{23} = z_{35} * z_{15}$,

$g_2 : z_2 = z_{30}$, $g_8 : z_{18} = z_{14} * z_{16}$, $g_{14} : z_{24} = z_{22} + z_{23}$,

$g_3 : z_3 = z_{28}$, $g_9 : z_{19} = z_{15} * z_{15}$, $g_{15} : z_{25} = z_{23} - z_{22}$,

$g_4 : z_3 = z_3 + z_{z_2} + z_{32}$, $g_{10} : z_{20} = z_{33} * z_{18}$, $g_{16} : z_{26} = z_{24} / z_{17}$,

$g_5 : z_2 = z_2 + z_{30}$, $g_{11} : z_{21} = z_{19} - z_{20}$, $g_{17} : z_{27} = z_{25} / z_{17}$,

$g6 : u = \begin{cases} 1, z_2 & \le & z_{34} \\ 0, z_2 & > & z_{34} \end{cases}$ $g_{12} : z_{22} = \sqrt{z_{21}}$, $g_k : (z_3 \equiv b_1, z_{26} \equiv b_2, z_{27} \equiv b_3)$.

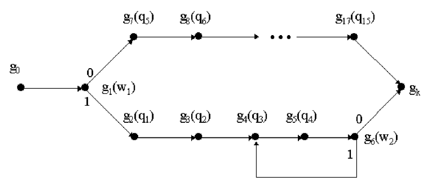

Fig.2. The example of graph G

4 Maximal Case of Parallelized GSA

Let us select in sequential GSA the set G of linear parts (blocks) $L = \{L_i\}$, $i = \overline{1, \nu}$ where i – is the ordinal number of the block; ν – is the number of linear parts of GSA G. Each block L_i represents sequential connection of some number of operators $q_j \in Q$.

Sequential GSA of G gives on set $M = Q \cup W \cup \{g_0, g_k\}$ the binary relation of direct order of operators and identifiers. For $a, b \in M_i$ $a\beta^* b$ is executed $a\beta^* b$, if the points **a** and **b** in GSA G are connected by an edge directed from **a** to **b**.

Let M_i relation – set of all operators $q_j \in Q$ of a linear part L_i. We shall define on M_i the following four relations $\beta_i^*, \beta_i, \varepsilon_i^*$ and α_i^*:

- β_i^* – is the relation of direct order of operators of the block L_i. For $a, b \in M_i, a\beta_i^*, b$ is executed if and only if $a\beta_i^* b$, and the operator **a** is not an output operator of the block;
- β_i – is the relation of oeder of operators in the block L_i, being the transitive closure of β_i^*;
- ϵ_i^* – is the relation of information dependence of operators of the block L_i. For $a, b \in M_i, a\epsilon_i^* b$, is executed if the output variable s of an operator **a** is one of input variables of an operator **b** and one of the two conditions is executed:

1) $a\beta_i^* b$ is executed,

2) there is a chain of operators $q_{i_1} q_{i_2} \ldots q_{i_k}$, for which the conjunction $(a\beta_i^* q_{i_1}) \wedge (q_{i_1}\beta_i^* q_{i_2}) \wedge \ldots \wedge (q_{i_k}\beta_i^* b)$ is executed and any of operators of this chain has not a variable s as an output;

α_i^* – is the relation of direct dependence of operators of the block L_i. For $a, b \in M_i$ $\alpha_i^* b$ is executed if one of the two conditions is executed:

1) $a\epsilon_i^* b$ is executed,

2) the operator **a** has as an input/output variable some variable s, which the operator b has as an output variable, and $a\beta_i b$ is executed.

The relation α_i^* establishes on set M_i the sequence of operator execution of the block L_i in time. So if $a_1, \ldots, a_k, b \in M_i$ are all those and only those operators, for which $(a_1 Large\alpha_i^* b) \wedge (a_2 Large\alpha_i^* b) \wedge \ldots \wedge (a_k Large\alpha_i^* b)$ is executed, the operator **b** may to begin to execute only after execution of all operators $a_1, \ldots, a_k$.

We form of the set M_i of operators of the block L_i two subsets $M_i^H \subseteq M_i$ and $M_i^K \subseteq M_i$ (M_i^H and M_i^K can intersect). We shall include operators **a**, for which is executed $\rceil \exists b(b\alpha_i^* a)$, to a subset M_i^H. The subset M_i^K will contain by operators **a**, satisfying to a condition $\rceil \exists b(a\alpha_i^* b)$.

The operators $a \in M_i^H$ have only input variables of the block as input ones. The beginning of execution of operators $a \in M_i$ in parallel GSA depends on the simultaneous beginning of execution of operators $a \in M_i^H$, and their termination depends on the termination of execution of all operators $a \in M_i^K$.

Let G_i is the subgraph of the graph G of sequential GSA G, defined on set M_i, and I_i is the subgraph of the graph I of parallel GSA P, defined on set M_i and reflecting the relation α_i^*. Then the parallelization of the block L_i will mean transition from the graph G_i to the graph I_i' (fig.3), where I_i' differs from I_i only by addition of fictitious operators of multisequencing R_i and connection S_i.

Thus, parallel flowchart of algorithm P can be defined as quintuple $< I', Q, W, Z, u >$, where Q, W, Z, u are determined by sequential GSA G, and I' – is a digraph defined on set of the operators $M \cup \{R_1, \ldots, R_v\} \cup \{S_1, \ldots, S_v\}$ by parallelization of all linear parts L_i of the sequential GSA G.

The relation α_i^* determines the maximal degree of parallelization of operators from set M_i therefore parallel GSA P we shall name a GSA with vaximal parallelization.

For sequential GSA, the graph G of which is represented at fig.2, we shall obtain: $v = 3, L = \{L_i/i = \overline{1,3}\}, M_1 = \{g_2, g_3\}, M_2 = \{g_4, g_5\}, M_3 = \{g_i/i = \overline{7,17}\}$. The graph Iof GSA P with maximal parallelization for this example is shown at fig.4.

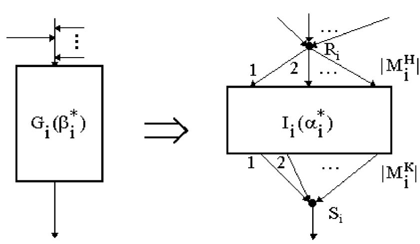

Fig.3. The transition from the graph G_i to the graph I_i'

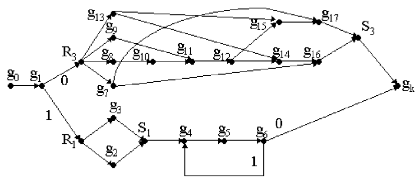

Fig.4. The graph I_i' of GSA P with maximal parallelization

5 Execution time of algorithm

Let for each point $g \in Q \cup W$ of sequential GSA G the set of acceptable its implementations (set of operational devices are equivalent in the functional) $R_g, g = 1, 2, \ldots, |Q| + |W|$ is known. And for equivalent points the corresponding sets R_g coincide. For points g_0 and $g_k, Rg = 0$.

Let's designate as $r_{g/c} = (s_{g/c}, t_{g/c})$ c-th implementation of point g, where $c = 1, 2, \ldots, |R_g|, s_{g/c}$ and $t_{g/c}$ are accordingly complexity and the average execution time of point g under condition that the operational device constructed following the scheme c is used. Let's assume, that all objects which are included in the set R_g, will form group of Pareto optimal devices. Thus $s_{g/1} < s_{g/2} < s_{g/|R_g|}$, and $t_{g/1} > t_{g/2} > t_{g/|R_g|}, \forall g \in Q \cup W$.

Let's consider sequential GSA G with selected linear parts L_i. Let input variables $a_i \in A$ of algorithm are random variables with given probability distribution. Then the statistical simulation of GSA G allows, using techniques from [7,11] to determine an average n_k of repetitions of k-th cycle. It enables to reduce the graph G GSA G to an acyclic G_a, which the same execution time.

The graph G_a is obtained from the graph G by deleting all edge that close cycles. Thus the average time $t_{g/c}$ of execution of all implementations $r_{g/c}$ of each point g, that alongs at least to one of cycles, is increased in n_g time, where $n_g = n_{i_1} \bullet \ldots \bullet n_{i_m}$, and $i_1, \ldots, i_m$ are numbers of cycles, which to point g belongs.

By parallelization of all linear parts L_i of sequential GSA G, represented by the acyclic graph G_a, we shall obtain the acyclic form I'_a of graph I' of the parallel GSA P.

We shall evaluate the execution time of algorithm using the networks constructed because of the graph I'_a using the following rules:

1. Each point $g \in I'_a$ with number of outgoing edges $\rho^+(g) \geq 2$ is doubled according to fig.5. And the points $g - g'$ are connected by continuous arrows, and $g' - g_1, g' - g_2, \ldots, g' - g_l$ by dashedones.

2. To the point $g_0 \in I'_a$ in a network corresponds the initial point, and point $g_k \in I'_a$ is final.

3. All points of a network graph are numbered by positive integer numbers. Number 1 is assigned to the initial point. The final point obtains maximum number.

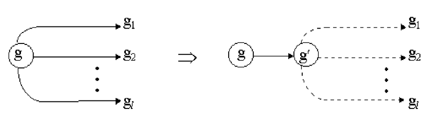

Fig.5. The doubbing of the point $g \in I'_a$

Each continuous arrow of a network corresponds to computing job, fulfilled by OD, implementing one or another vertex of the graph I'_a. Dashed arrows of the network correspond to fictitious operations having the zero duration and cost. Besides the operations correspoding to the vertives g_0, R_i, and S_i yave the zero cost (on a network these operations are designated by continuous arrows).

Each point of network means some event, nameey the termination of all operations, which arrows ender the selected vertex, and the beginning of those operations, which arrows goesout the same vertex.

The earliest time ti of a beginning for all operations, for which i-th point of the network will precede, is calculated under the formula [12]:

$$t_i = max\{t_k + t_{ki}\}, i = \overline{2,n}; t_1 = 0; t_n = T,$$

where t_{ki} is the execution time of operation, beginning in point with number k and finished in point with number i; T – execution time of algorithm.

Let the arc (k, i) of the network corresponds to the top $g \in Q \cup W$. Then

$$t_{ki} = t'_{g/c} = n_g((m_g + \alpha)t_{SPM}(E, N) + t_{g/c} + t_{CA}), (1) \qquad (1)$$

where m_g is an amount of necessary arguments of an operator / recognizer $g, \alpha = 1$ for operator points, $= \alpha = 0$ for conditional points, t_{CA} is the time of switching CA.

The clause $(m_g + \alpha)t_{SPM}(E, N)$ in (1) takes into account of the acces input data and write the of result in SPM. The writing of a variable u in SPM is not done, therefore for conditional points $\alpha = 0$. The time $t_{g/c}$ should be calculated in the supposition, that the input data and result of operation are stored in the registers OD.

If the point $g \in \{g_0, R_i, S_i\}$ corresponding to an arc (k, i) of the network,

$$t_{ki} = t'_g = n_g \bullet t_{CA}. \qquad (2)$$

The network allowns to interpret the graph I'_a using the language of computing operations made by operational devices when executing operator and conditional points of the graph I'_a. And such interpretation as a matter of fact ignores an alternate character of distribution of the computing process through conditional points.

It allows automatically to take into account all possible paths of the computing process without their explicit construction. The value of T thus will be efnal to the average execution time of the longest path.

6 MC-classes

Let's introduce a sequence of necessary definitions.

Definition 1 *Two operators $a, b \in M_i$ are compatible, if their execution on one OD does not lead to increase of T calculated under the condition, that each operator is executed on separate OD.*

Definition 2 *Some set of operators from M_i will form the class of compatible operators, if all operators, included in it, in pair are compatible.*

Definition 3 *The class of compatible operators is named as maximal $(MC - class)$, if in the class of addition to it of any operator from M_i, not belonging to the class, it ceases to be compatible.*

Definition 4 *the subgraph generated by a sequence of points g_{i_1}, g_{i_2}, $\ldots$, g_{i_n} $(g_{i_j} \in M_i \cup \{R_i, S_i\})$, in which each two adjacent g_{i_k} and $g_{i_{k+1}}$ are connected by arc $(g_{i_k}, g_{i_{k+1}})$ is called a path $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_n})$ of the graph I_i'. The first point of path g_{i_1} we shall name the initial point of path, and the last g_{i_n} the final point of path.*

Definition 5 *The path $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_n})$ in the graph I_i' we name a route, if it begins with point $R_i(g_{i_1} = R_i)$ and is finished in point $S_i(g_{i_n} = S_i)$.*

Let's consider any path $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_n})$ in the graph I_i'. Each vertex g_{i_j} of this path can start executing only then, when all vertices $g_{i_k}, k < j$ are executed. From here follows, that intervals of time of execution of any two operators $a, b \in \{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$ are not intersected. It means, that the execution time of all points of the path is the same and does not depend on an amount OD, necessary for implementation of these points. The following lemma therefore is correct.

Lemma 1 *Set of all vertices $\{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$ of any path $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_n})$, $g_{i_j} \in M_i \cup \{R_i, S_i\}$ of the graph I_i' will form the class of compatible operators.*

This lemma allows to prove the following theorem a bout the relation of the chains of edges to MC-classes.

Theorema 1 *Tye set of all points* $\{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$ *of any route* $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_n}), g_{i_j} \in M_i,\ j = \overline{2, n-1}, g_{i_1} = R_i,\ g_{i_n} = S_i$ *of the graph* I_i' *will form the MC - class.*

▷ Set of verties $\{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$ of some route $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_n})$, will form a path. Therefore, starting from the Lemma 1, set $\{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$ is the class of compatible operators.

Any other operator $a \in M_i \backslash \{g_{i_1}, g_{i_2}, \ldots, g_{i_{n-1}}\}$ belongs to other rout $\mu' = (g'_{i_1}, g'_{i_2}, \ldots, g'_{i_m})$. The routes μ and μ' of parallel graph I_i' can intersect, but it necessaryly contain parallel (not intersected) perts. The operator **a** belongs to one of such parts. It leads to the intersection of intervals of execution time of an operator **a** and even one of operators $b \in \{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$. Therefore execution of operators **a** and **b** cannot be combined on one OD without loss of speed, so they are incompatible. Thus, it is impossible to expand the class of compatible operators $\{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$ not breaking a property of compatibility of operators. Therefore, set of points $\{g_{i_1}, g_{i_2}, \ldots, g_{i_n}\}$ of route $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_n})$ is MC -class.◁

Let's consider the procedure of a construction of all routes of the acyclic graph I_i'. Let's assume, that all points of the graph I_i' are ranged so, that the rank 1 is assigned to point R_i and any other point can be referred to a rank l, if it is connected with its entering arcs to the points of a rank $l-1, l-2, \ldots, 1$. The maximum rank corresponds to point S_i. To each point g_i of the graph I_i' we shall assign the set μ_{g_i} of all different routes, beginning in point Ri and finished in point g_i.

Let point g_{i_n} of graph I_i' is connected with its input arcs to points $g_{i_1}, g_{i_2}, \ldots, g_{i_k}$, for which the correspondent sets $\mu_{j_{i_1}}, \mu_{j_{i_2}}, \ldots, \mu_{j_{i_k}}$ are already defined. Then the rule of finding of set $\mu_{j_{i_n}}$ includes the two items:

1. To calculate intermediate set $\mu'_{j_{i_n}}$ by association of sets, $\mu_{j_{i_1}}$, $\mu_{j_{i_2}}$, $\ldots$, $\mu_{j_{i_k}}$ or $\mu'_{j_{i_n}} = \bigcup\limits_{i=1}^{\nu} \mu_{g_{i_j}}$;
2. To each route of set $\mu'_{g_{i_j}}$ to add the vertex g_{i_n} at the right-hand site. As the result we shall obtain set μ_{gi_n}.

According to this rule, assuming $\mu_{R_i} = \{(R_i)\}$, we shall find sets μ_{g_j} for all points g_j of the graph I'_i, sequentially looking through points of a 2-nd rank, then 3-rd rank etc. The set μ_{S_i} will contain all various routes of the graph I'_i.

By virtue of the theorem 1 set μ_{S_i} also determines all MC-classes of operators from set M_i.

As an example we shall consider the subgraph I'_3, of the graph I', represented on fig.4. The set of all routes $\overline{\mu_{S_3}}$ has view:

$$\overline{\mu_{S_3}} = \left\{ \begin{array}{cc} \overline{\mu}^1 = (8,10,11,12,14,16) & \overline{\mu}^5 = (8,10,11,12,15,17), \\ \overline{\mu}^2 = (9,11,12,14,16) & \overline{\mu}^6 = (9,11,12,15,17), \\ \overline{\mu}^3 = (13,14,16) & \overline{\mu}^7 = (13,15,17), \\ \overline{\mu}^4 = (7,16) & \overline{\mu}^8 = (7,17) \end{array} \right\}$$

The set $\overline{\mu}_g$ is obtained from set μ_g by deleting from each route the points R_i and S_i, and the i-th route $\overline{\mu}^i = q_{i_1}, q_{i_2}, \ldots, q_{i_n}$ in set $\overline{\mu}_{S_3}$ is denoted more compactly as $\overline{\mu}^i = (i_1, i_2, \ldots, i_n)$.

7 Minimal Partitionings of Operators

At first we shall consider the task of covering of set of operators M_i with the minimal number of classes of compatible operators $Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k}$.

Definition 6 *The system of sets $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k})$ is named the covering of set M_i, if the following conditions hold:*

1. $Q_{i,j} \subset M_i, j = \overline{1,k}$,
2. $Q_{i,j}, j = \overline{1,k}$ *is the class of compatible operators,*
3. $\bigcup_{j=1}^{k} Q_{i,j} = M_i$

Definition 7 *Covering $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k})$ is named minimal, if other covering $Q'_i = (Q'_{i,1}, Q'_{i,2}, \ldots, Q'_{i,m})$, such,that $m < k$, does not exist.*

The following theorem considerably simplifies the task of search of minimal covering.

Theorema 2 *In minimfl covering $Q_i = Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k}$ all classes of compatible operators $Q_{i,j}, j = \overline{1,k}$, are maximal.*

$\triangleright$ Let us assume, that there is a minimal covering $Q_i = Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k}$, which has at least one class of compatible operators $Q_{i,j}$ theat is not maximal. Then there is a MC-class $Q_{i,j} \supset Q'_{i,j}$ and the substitution Q'_i on $Q_{i,j}$ will leave the system of sets $Q_i = Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k}$ be the covering and will not lead to increase of number of classes of compatible operators. The repetition of such substitution for all classes which are not being maximal, will allow to obtain the covering $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k})$ with number of classes not greater than starting minimal covering Q'. And each class of compatible operators $Q_{i,j}, j = \overline{1,k}$ will be maximal. $\triangleleft$

Corollary 1 *For a construction of minimal covering $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k})$ of the sets of operators M_i it is necessary to analyze only MC-classes of set of operators M_i.*

The technique of finding of minimal covering Q_i is connected with a construction of the table of coverings P_i. The lines of such table correspond to MC-classes of set M_i (routes of the graph I_i), while columns to operators $q_j \in M_i$. On an intersection of line i and column j the label is put, if the MC-class appropriate to i-th line, is covered with an operator appropriate to j-th column. The MC-class i is covered with an operator j, if the j-th operator belongs to the i-th MC-class, e.g. the route, corresponding to the i-th MC-class, passes through the j-th operator.

To find minimal covering in the table of coverings it is necessary to select the least number of table lines, such, that they by units have covered all columns with labels.

The similar task arises at minimization of Boolean functions by Kwain method with the help of the implicant tables, and is now enough well investigated. Therefore the all exact (for the tables of small dimensionality) and approximate (for the tables of large dimensionality) the methods of searching of minimal covering of the implicant tables described, for example, in [13], can be used and for construction of minimal coverings Q_i of sets of operators M_i without any changes.

Let's consider now task of partitioning of set of operators M_i into minimal number of mutually not intersected classes of compatible operators $\tilde{Q}_{i,1}, \tilde{Q}_{i,2}, \ldots, \tilde{Q}_{i,k}$.

Definition 8 *Covering* $\tilde{Q}_i = (\tilde{Q}_{i,1}, \tilde{Q}_{i,2}, \ldots, \tilde{Q}_{i,k})$ *of set of operators* M_i *is named a partition of set* M_i*, if* $\tilde{Q}_{i,j} \cap \tilde{Q}_{i,m} = 0, \;\; j \neq m$.

Definition 9 *Partitioning* $\tilde{Q}_i = (\tilde{Q}_{i,1}, \tilde{Q}_{i,2}, \ldots, \tilde{Q}_{i,k})$ *is named minimal, if other partitioning* $\tilde{Q}'_i = (\tilde{Q}'_{i,1}, \tilde{Q}'_{i,2}, \ldots, \tilde{Q}'_{i,m})$*, such, that* $m < k$*, does not exists.*

Let's prove the following theorem, which considerably simplifies the task of a construction of minimal partitioning.

Theorema 3 *Minimum covering* $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k})$ *becomes minimal partitioning* $\tilde{Q}_i = (\tilde{Q}_{i,1}, \tilde{Q}_{i,2}, \ldots, \tilde{Q}_{i,k})$ *with the same value* k*,if every time, when* $Q_{i,j} \cap Q_{i,m} \neq 0, j \neq m$*, one of the classes of compatible operators, for example,* $Q_{i,m}$*, will be transformed under the formula:* $Q_{i,m} \setminus (Q_{i,j} \cap Q_{i,m})$.

▷ At first we shall show, that initial covering $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k})$ as a result of conversions, described in the theorem, becomes a partition.

For this we shall mark, that the set obtained by deletiong of some number of operators, from the class of compatible operators remains the class of compatible operators. Therefore conversions being made connected to deletion of some operators from sets $Q_{i,j}, j = \overline{1,k}$, leave thein to be the classes of compatible operators. Besides the deletion of common operators of two sets $Q_{i,j}, j and Q_{i,m}$ only from one set $(Q_{i,m})$ leaves the set $Q_{i,j} \cup Q_{i,m}$ without changes. From here follows, tyat after conversions $\bigcup\limits_{j=1}^{k} Q_{i,j} = M_i$ and this by definition 6, means, that the system of $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,k})$ after conversions remains the covering.

Taking into account, that conversions are directed to obtaining of not mutually intersected sets, we come to the conclusion, that as a result of conversions the starting covering really becomes the partitionibg (see definition 8).

Now it is necessary to show, that the minimal covering in a course of conversion passes into the minimal partitioning.

For it we shall assume, that the conversions lead to partition $\tilde{Q}'_i = (\tilde{Q}'_{i,1}, \tilde{Q}'_{i,2}, \ldots, \tilde{Q}'_{i,l})$ and $l < k$. Let $\tilde{Q}'_{i,j}$ be not the MC-class. Then there is a MC- class $\tilde{Q}_{i,j} \subset \tilde{Q}'_{i,j}$ and the substitution $\tilde{Q}_{i,j'}$ on $\tilde{Q}_{i,j}$ keeps the system of sets $(\tilde{Q}'_{i,1}, \ldots, \tilde{Q}'_{i,j-1}, \tilde{Q}'_{i,j}, \tilde{Q}'_{i,j+1}, \ldots, \tilde{Q}'_{i,l})$ to be the covering and also does not lead to increase of number of classes of compatible operators. The repetition of such substitution for all classes which are not being MC-classes, allows to obtain the covering $Q_i = (Q_{i,1}, Q_{i,2}, \ldots, Q_{i,j})$ with number of classes $l < k$, that is the contradiction, because minimal covering was transformed according to a condition of the theorem. ◃

The minimal partitionings $\tilde{Q}_i$ of sets M_i allows to define minimal number N_max OD, necessary for implementation maximal parallelized GSA without loss of speed.

Let N_{I_i} be the minimal number OD, necessary for implementation of the graphI_i without loss of speed. The value N_{I_i} is determined by powerof minimal partitioning of set M_i, i.e. $N_{I_i} = |\tilde{Q}_i|$. Then $N_{max} = \max_i N_{I_i}$

Let's consider set of operators M_3. The table of coverings P_3 is represented as table 1.

Table 1

	7	8	9	10	11	12	13	14	15	16	17
$\overline{\mu}^1$		1		1	1	1		1			1
$\overline{\mu}^2$			1		1	1		1		1	
$\overline{\mu}^3$							1	1		1	
$\overline{\mu}^4$	1									1	
$\overline{\mu}^5$		1		1	1	1			1		1
$\overline{\mu}^6$			1		1	1			1		1
$\overline{\mu}^7$							1		1		1
$\overline{\mu}^8$	1										1

With the table P_3, using [13], we find the minimal covering Q_3:

$$Q_3 = (\{7, 16\}, \{8, 10, 11, 12, 14, 16\}, \{9, 11, 12, 15, 17\}, \{13, 15, 17\}).$$

By the theorem 3 from minimal covering Q_3 we shall obtain the minimal partitioning $\tilde{Q}_3$:

$$\tilde{Q}_3 = (\{7, 16\}, \{8, 10, 11, 12, 14\}, \{9, 15, 17\}, \{13\}).$$

From here it is easy to see, that $N_{I_3} = 4$. By doing similar procedures for M_1 and M_2, we shall obtain:

$$Q_1 = (\{2\}, \{3\}), \quad \tilde{Q}_1 = Q_1, \quad Q_2 = (\{4, 5\}),$$

$$\tilde{Q}_2 = Q_2, \quad N_{I_1} = 2,$$

$$N_{I_2} = 1, \ N_max = 4$$

8 GSA with a necessary and sufficient degree of a parallelism

The degree of a parallelism GSA we shall evaluate by minimal number of operational devices necessary for implementation of the GSA without loss of speed. Obviously, $N = 1$ corresponds to the sequential GSA, and $N = N_max$ – maximal parallelized one. To intermediate values N correspond to GSA distinct from of the extreme cases, which are the sequential and themaximal parallelized GSA

Among these intermediate meanings N there is such $N = N_{HD}$, to which corresponds GSA with a necessary and sufficient degree of a parallelism. It means, that at $N = N_{HD}$ there exists a GSA, for which $T \leq T_{giv}$ (condition of sufficiency of a degree of a parallelism GSA), and already at $N = N_{HD} - 1$ for all GSA $T > T_{giv}$ (condition of necessity). Thus the calculation T should be produced in the supposition, that the most high-speed implementations of operator and conditional points of the GSA are used, and the time of switching of CA is determined by the time of switching$t_{CA/N_{max}}$ CA, constructed for extreme parallelized GSA, i.e. in the formulas (1) and (2) $t_CA = t_{YA/N_{max}}, t_{g/c} = t_{g/|R_g|}, \forall g$.

Let's consider two methods of lowering of a degree of a parallelism of the GSA: sequential and logarithmic. The first method represents the multistage process, which allows, starting from the maximal parallelized GSA ($N = N_{max}$) to reduce a degree of a parallelism N at each step per 1, modifying thus in appropriate way the GSA constructed on previous step. The process proceeds until the GSA will be found with a necessary and sufficient degree of a parallelism.

We shall accept the following denotations: $I_{i,k}$ is the modification of the graph I_i on k-th step ($k = 0$, $I_{i,0} = I_i$; at $k = (N_{max} - 1)N_{HD} = 1,$ and $I_{i,N_{max}-1}$ is the graph of the relation $\beta_i^*; k = \overline{0,(N_{max} - N_{HD})}$), $\tilde{Q}_j^k$ is the partitioning of set of operators M_i at k-th step, $N_{I(i,k)} \equiv N_{I_{i,k}}$ is the minimal number OD, necessary for implementation of the graph $I_{i,k}$ without loss of speed.

Let at $0 - i$ step of the sequential method $N = N_{max}, I_{i,0} = I_i, \tilde{Q}_i^0 = (\tilde{Q}_{i,1}^0, \tilde{Q}_{i,2}^0, \dots, \tilde{Q}_{i,N_{I(i,0)}}^0) = \tilde{Q}_i = (\tilde{Q}_{i,1}, \tilde{Q}_{i,2}, \dots, \tilde{Q}_{i,N_{I(i)}})$ $i = \overline{1,\nu}, N_{I_i} \equiv N_{I_i}$.

On k-th step $N = N_{max} - k$, it is necessary to change the graphs $I_{i,k-1}$ and partitioning $\tilde{Q}_j^{k-1}$, for which $NI_{i,k-1} > N_{max} - k$ only. For the remaining graphs and partitionings $I_{i,k} = I_{i,k-1}, \tilde{Q}_j^k = \tilde{Q}_j^{k-1}$.

The partitionings $\tilde{Q}_j^{k-1}$, being subject to change at k-th step, are modified at first in to intermediate partitionings $\tilde{Q}_j^k(j_1, j_2)$ under the following formulas:

$$\tilde{Q}_i^k(j_1, j_2) = \left(\tilde{Q}_{i,1}^k(j_1, j_2), \tilde{Q}_{i,2}^k(j_1, j_2), \ldots, \tilde{Q}_{i,N_{I_{(i,k)}}}^k(j_1, j_2)\right)$$

where

$$\tilde{Q}_{i,1}^k(j_1, j_2) = \tilde{Q}_{i,j_1}^{k-1} \cup \tilde{Q}_{i,j_2}^{k-1},$$

$$j_1, j_2 \in \{1, 2, \ldots, N_{I_{i,k-1}}, \quad j_1 \neq j_2,$$

$$\tilde{Q}_{i,j}^k(j_1, j_2) = \tilde{Q}_{i,j_3}^{k-1}, j = \overline{2, N_{I_{i,k-1}-1}},\ j_3 \in \{1, 2, \ldots, N_{I_{i,k-1}-1}\},$$

$$j_3 \neq j_1, \quad j_3 \neq j_2.$$

According to these relations the set $\tilde{Q}_{i,1}^k(j1, j2)$ is determined by association of two classes: $\tilde{Q}_{i,j_1}^{k-1} and \tilde{Q}_{i,j_2}^{k-1}$, and the remaining classes $\tilde{Q}_{i,j_3}^{k-1}$ pass in to partitioning $\tilde{Q}_i^k(j_1, j_2)$ without changes.

Let we have set of operators $\tilde{Q}_{i,j_1}^k(j_1, j_2) = \{g_{i_1}, g_{i_2}, \ldots, g_{i_m}\},\ g_{i_j} \in M_i,\ i_1 > i_2 \ldots < i_m$. To the intermediate partitions $\tilde{Q}_i^k(j_1, j_2)$ there correspond the $I_{i,k}(j_1, j_2)$, which are created under the following rules:

a) In the graph $I_{i,k-1}$ the path $\mu = (g_{i_1}, g_{i_2}, \ldots, g_{i_m})$ is be formed with the help of additions of missing arcs;

b) In the graph obtained after execution of an item a), from each route $\mu = (g_{j_1}, g_{j_2}, \ldots, g_{j_n})$ all arcs of a type (g_{j_p}, g_{j_m}) are deleted provided the condition $m \neq p + 1$.

Average execution time of the longest route of the $I_{i,k}(j1, j2)$ and $I_{i,k}$ we shall designate by $T_{i,k}(j1, j2)$ and $T_{i,k}$ accordingly.

It is convient to represent the values $T_{i,k}(j1, j2)$ as the triangular table, cells of correspond to all different unnordered pairs of classes of compatible operators $\tilde{Q}_{i,j_1}^{k-1}$and $\tilde{Q}_{i,j_2}^{k-1}$ of partitioning $\tilde{Q}_i^{k-1}$. The cell located a the intersection of j_1-th column ($j_1 = 1, 2, \ldots, N_{I_{i,k-1}} - 1$) and of j_2-th line ($j_2 = 2, 3, \ldots, N_{I_{i,k-1}}$), is filled by the value $T_{i,k}(j1, j2)$.

The triangular table allows finally to determine partitionings $\tilde{Q}_i^k$ and graphs $I_{i,k}$ as follows:

$$j_1^*, j_2^* = arg \min_{(j_1,j_2)} (T_{i,k}(g_1, g_2) - N_{i,k-1})$$

$$\tilde{Q}_i^k = \tilde{Q}_i^k(j_1^*, j_2^*), \quad I_{i,k} = I_{i,k}(j_1^*, j_2^*).$$

Time T_k of execution of algorithm as a whole on k-th step we shall evaluate using the graph $I'a, k$. This graph is obtained from the graph I'_a by substitution of its subgraphs I_i by $I_{i,k}, i = \overline{1,\nu}$.

With great values N_{max} and/or complex GSA the sequential method can require excessive temporary expenditures and storage space. In this case for lowering the degree of parallelism ofGSA it is possible to take advantage of a logarithmic method (method of half division). This approximate method is much less labour-consuming in comparison with a sequential method, but also it is less exact, as comes from smaller amount of a necessary information.

The maximal execution time of method of half division is $log_2(N_{max} - 1)$ steps. We shall describe the first step explicitly. For this we shall designate by N^- and N^+ accordingly the left and right boundaries of an interval ($N^- < N^+$), containing the N_{HD} value. All the first step $N^- = 1, N^+ = N_{max}$.

The middle of an interval N_{mid} is calculated under the formula:

$$N_{mid} = N^- + [\frac{N^+ - N^-}{2}]$$

where $[x]$ is the integral part of x.

The N_{mid} value allows to determine the amount of iterations $k = N_{max} - N_{mid}$, which are necessary to execute for obtaining the partitionings $\tilde{Q}_i^{k^*} = (\tilde{Q}_{i_1}^{k^*}, \tilde{Q}_{i_2}^{k^*}, \ldots, \tilde{Q}_{i,n(i,k^*)}^{k^*}), n(i,k^*) \leq (N_{mid}, \quad i = \overline{1,\nu}$.

The partitionings $\tilde{Q}_i^{k^*}$ are determined mainly in the game way as in a sequential method. The exception makes only that:

1) The graphs $I_{i,k}, \quad k = \overline{1,k^*}, i = \overline{1,\nu}$ are not created;

2) The values j_i^* and j_2^* on k-th of iteration ($k = \overline{1, k^*}$) are determined as follows:

$$j_i^* = arg \min_{j \in \{1,2,\dots,n(i,k-1)\}} t_{i,j}^{k-1},$$

$$j_2^* = arg \min_{j \in \{1,2,\dots,n(i,k-1)\} \setminus \{j_1^*\}} t_{i,j}^{k-1},$$

where

$$t_{i,j}^{k-1} = \sum_{g \in \tilde{Q}_{i,j}^{k-1}} t'_{g/|R_g|}.$$

Let the set of operators $\tilde{Q}_{i,l}^{k^*} = \{g_{j_1}, g_{j_2}, \dots g_{j_m}\}, \quad g_{g_p}\} \in M_i, \quad j_1 < j_2 < \dots < j_m, \quad l = \overline{1, n(i,k^*)}, \quad j = \overline{1,\nu}$. The partitionings $\tilde{Q}_i^{k^*}$ allow to construct the graphs $I_{i,k^*}, i = \overline{1,\nu}$ by the following rules:

a) In the graph I_i the paths $\mu_{i,l}^{k^*} = \{g_{j_1}, g_{j_2}, \dots g_{j_m}\}, l = \overline{1, n(i,k^*)}$. Are formed by addition of missing arcs;

b) In the graph obtained the after the item a), from each route $\mu = g_{j_1}, g_{j_2}, \dots g_{j_p}$, all arcs of a type (g_{j_r}, g_{j_s}) are deleted provided that $s \neq (r+1)$.

Let T be the average execution time of the longest path of the graph I'_{a,k^*}, which is obtained from the graph I (a by substitution of all subgraphs I_i on $I_{i,k^*}, i = \overline{1,\nu}$).

After calculation of the of value T the transition to the following step is implemented by determination of new boundaries N^- and N^+ by the rule:

$$\begin{array}{lll} if & T \geq T_{giv} & N_- = N_{aver}, \\ if & T \leq T_{giv} & N^+ = N_{aver} \end{array}$$

The process is finished, when at the next step the length of an interval $N^+ - N^-$ equals 0 or 1. In this case $N_{HD} = N$, and the necessary graph $I'_{a,HD} = I'_{a,k^*}$ at $k^* = N_{max} - N_{HD}$.

Let's consider as an example a sequential method of decrease of the degree of the parallelism of GSA in connection with maximal parallelized GSA P, the graph I' of which is represented at fig.4.

The execution time of all operator and conditional points with the help of separate OD of given GSA with regard of extraction of operands from SPM and writing of results in memory using conventional time units is reflected in tab.2.

Table 2

points	$t'_{g/\|R\|}/t'_g$
g_0, R_i, S_i	0.1
g_1, g_6	4
g_2, g_3	1
g_4	9
g_5, g_{14}	5
$g_7 \dots g_{10}, g_{13}$	150
g_{11}, g_{15}	5
g_{12}	200
g_{16}, g_{17}	170

The sequence of steps at $T_{giv} = 700$ the form:

Step 0.

$$\tilde{Q}_1^0 = \tilde{Q}_1 = Q_1 = (\{2\},\{3\}), I_{1,0} = I_1, N_{i_1} = 2, T_{1,0} = 1;$$

$$\tilde{Q}_2^0 = \tilde{Q}_2 = Q_2 = (\{4,5\}), I_{2,0} = I_2, N_{i_2} = 1, T_{2,0} = 15$$

$$\tilde{Q}_3^0 = \tilde{Q}_3 = (\{7,16\},\{8,10,11,12,14\},\{9,15,17\},\{13\}),$$

$$I_{3,0} = I_3, N_{i_3} = 4, T_{3,0} = 680;$$

$$N = N_{max} = 4, \quad I'_{a,0} = I'_a, \quad T_0 = 684.3$$

The graphs $I_{1,0}, I_{2,0}, I_{3,0}$ and $I_{a,0}$ are represented at fig.6, a-d accordingly. The numbers at vertices of the graphs designate execution time of these vertices.

Step 1.

$N = N_{max} - 1 = 3;$

$\tilde{Q}_1^1 = \tilde{Q}_1^0, I_{1,1} = I_{1,0}, N_{i_{1,1}} = 2, T_{1,1} = T_{1,0};$

$\tilde{Q}_2^1 = \tilde{Q}_2^0, I_{2,1} = I_{2,0}, N_{i_{2,1}} = 1, T_{2,1} = T_{2,0};$

$\tilde{Q}_3^1(1,2) = (\{7,8,10,11,12,14,16\},\{9,15,17\},\{13\}),$

$\tilde{Q}_3^1(1,3) = (\{7,9,15,16,17\},\{8,10,11,12,14\},\{13\}),$

$\tilde{Q}_3^1(1,4) = (\{7,13,16\},\{8,10,11,12,14\},\{9,15,17\}),$

$\tilde{Q}_3^1(2,3) = (\{9,9,10,11,12,14,15,17\},\{7,16\},\{13\}),$

$\tilde{Q}_3^1(2,4) = (\{8,10,11,12,13,14\},\{7,16\},\{9,15,17\}),$

$\tilde{Q}_3^1(3,4) = (\{9,13,15,17\},\{7,16\},\{8,10,11,12,14\}),$

The graphs $I_{3,1}(1,2)$, $I_{3,1}(1,3)$, $I_{3,1}(1,4)$, $I_{3,1}(2,3)$, $I_{3,1}(2,4)$ and $I_{3,1}(3,4)$ are represented at fig.7, a-f accordingly.

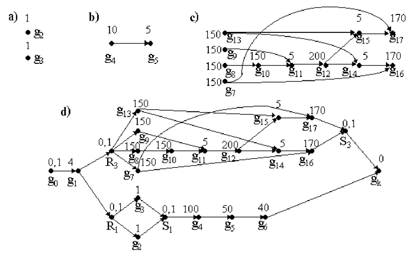

Fig.6. The graphs $I_{1,0}, I_{2,0}, I_{3,0}, I'_{a,0}$

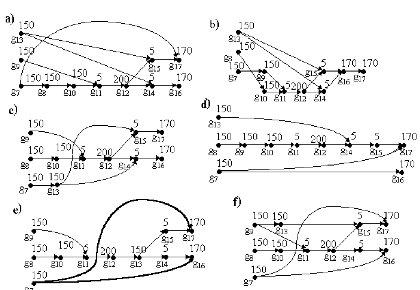

Fig.7. The graphs $I_{3,1}(1,2)$, $I_{3,1}(1,3)$, $I_{3,1}(1,4)$, $I_{3,1}(2,3)$, $I_{3,1}(2,4)$, and $I_{3,1}(3,4)$

Values $T_{3,1}(j_1, j_2)$ are shown in table 3. With the help of the table one can find:$(j_1^*, j_2^*) = (1, 4)$.

j_2				
2	830			
3	850	835		
4	680	730	680	
	1	2	3	j_1

Therefore

$$\tilde{Q}_3^1 = \tilde{Q}_3^1(1, 4),\ I_{3,1} = I_{3,1}(1, 4),\ T_{3,1} = 680$$

The graph $I_{a,1}$ is represented at fig.8, from which it follows that $T_1 = 684.3$.

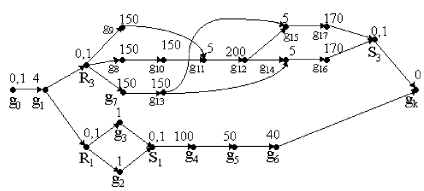

Fig.8. The graph $I'_{a,1}$

Step 2. The calculations at this step show, that T_2=839.3.

The value T_{giv} =700. Therefore N_{HD}=3, and the graph $I_{a,1}$ represents the acyclic form $I'_{a,HD}$ of the graph I'_{HD} of the parallel GSA with a necessary and sufficient degree of a parallelism. This graph is shown at fig.9.

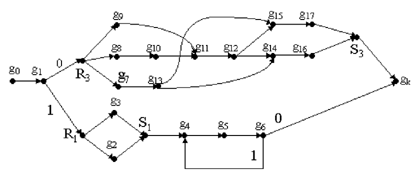

Fig.9. The acyclic form $I'_{a,HD}$ of the graph I'_{HD} parallel GSA

9 Agorithm of TS - coordination of the parallel DCS

All above-stated allows us to formulate the algorithm of TS - coordination parallel of the DCS as a sequence of the following items:

1) Selection in the sequential GSA (of the set of linear parts $L = \{L_i\}$ and determination for each linear part L_i of the sets of operators M_i.

2) Determination of an average number n_k of repetitions of each cycle. Account for each point g of the graph G of an average number n_g of repetitions with regard to the belonging of point g in a various cycles (taking account of nested cycles).

3) Construction of the acyclic form G_a of the graph G GSA G.

4) Maximal parallezitation of all linear parts L_i of acyclic graphs G_a and obtaining of the acyclic form I'_a of the graph I of parallel GSA P.

5) Construction for each set M_i of minimum partitionings $\tilde{Q}_i, i = \overline{1,\nu}$ and definition of minimal number N_{max} of operational devices

necessary for implementation of GSA without loss of speed. Synthesis of CA for maximal parallelizied GSA and implementation $t_{CA/N_{max}}$.

6) Construction of the parallel GSA with a necessary and sufficient degree of parallelism N_{HD} by finding minimal partitioning $\tilde{Q}_i^{k^*} = (\tilde{Q}_{i,1}^{k^*}, \tilde{Q}_{i,2}^{k^*}, \ldots, \tilde{Q}_{i,n_{(i,k^*)}}^{k^*}), i = \overline{1,\nu}, k = (N_{max} - N_{HD})$, and also graphs $I'_{a,HD}$ and I'_{HD}, using for this a sequential method or a method of half division. Synthesis of CA for PGSA with a necessary and sufficient degree of a parallelism and determination of $t_{CA/N_{HD}}$.

7) Distribution of the set of operator and conditional points $Q \cup W$ on to operational devices $OD_1, OD_2, \ldots, OD_{N_{HD}}$.

   The conditional points are compatible one with another. Besides each conditional point is compatible with any operator point, that enables to generate sets Q_{OD_j} of points executed on j-th OD, as follows:

$$\begin{cases} Q_{OD_1} = & (\bigcup\limits_{i=1}^{\nu} \tilde{Q}_{i,1}^{k^*}) \bigcup W, \\ Q_{OD_j} = & (\bigcup\limits_{i=1}^{\nu} \tilde{Q}_{i,j}^{k^*}), \quad j = \overline{2, N_{HD}} \end{cases}$$

8) First stage of the TS - coordination of parallel DCS (rough approximation).

   The general course of fulfillment of the first stage is shown at fig.10. This stage can be divited into he three folowing stages:

   a) At the first phase the transition from a point 1 to point 2 is fulfilled. Preliminary optimization of the network constructed on the graph Ia,HD, with the help of algorithm FFK here is made. And for each point of the graph $I'_{a,HD}$, accepting alternate construction $r_{g/c}$, the piecewise linear convex approximation of the set of possible implementations R_g, Pareto optimal, in a plane of complexity and delay is used;

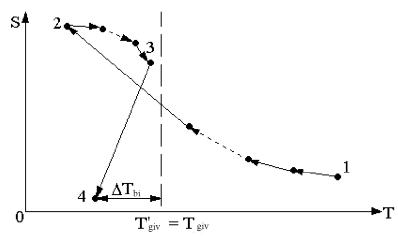

Fig.10. The general course of fulfillment of the first stage

b) The second phase provides transition from point 2 to point 3. At this transition the discrete optimization supposing the analysis of all works (points), lying on critical paths and choice (is long as it is possible) more simple implementation for that work (point), which provides maximum decrease of complexity per unit of delay at the next step.

Tye result of two first phases of this stage are the implementations $r_{g/c'(g)}, \forall g \in Q \cup W$, ensuring the minimum of the total instrument costs

$$S = \sum_{g \in Q \cup W} S_{g/c}\,,$$

recognizing that each point of the graph $I'_{a,HD}$ will be implemented on separate OD, in this case in the formule (1) and (2) $t_{CA} = t_{CA/N_{HD}}$. Let's remark, that the detail algorithm of fulfillment of first two phases practically coincides algorithm A_1 from [1];

c) The third phase is used for transition from point 3 to point 4 and provides the synthesis of TS- coordinated operational devices $OD_1, OD_2, \ldots, OD_{N_{HD}}$.

Let be the set of points implemented on the i-th operational device, $Q_{OD_i} = \{g_{j_1}, g_{j_2}, \ldots, g_{j_{OD_i}}\}$ The time of execution g_{j_k} of i-th point by separate OD $t_{g_{j(k)}/c'(g_{j(k)})}$ is determined by the following

$$t_{g_{j(k)}/c'(g_{j(k)})} = \vartheta_{g_{j(k)}} \tau_{g_{j(k)}}, \tag{3}$$

where $\vartheta_{g_{j(k)}}$ is the average amount of clock cycles necessary for execution of point g_{j_k} with the help particular OD; $\tau_{g_{j(k)}}$- duration of the clock sycle for this OD.

From the formula (3) it follows, that

$$\tau_{g_{j(k)}} = t_{g_{j(k)}/c'(g_{j(k)})} / \vartheta_{g_{j(k)}}$$

The joint execution of set of points $\{g_{j_1}, g_{j_2}, \ldots, g_{j_{OD_i}}\}$ on one OD results in necessity of choice of duration of the clock cycle τ_{OD_i} for the i-th OD under the formula:

$$\tau_{OD_i} = \min g_{j(k) \in Q_{OD_i}} \tau_{g_{j(k)}}$$

τ_{OD_i} being known, it is possible to synthesize TS-coordinated OD_i with duration of the clock cycle $\tau^*_{OD_i}$, and $\tau^*_{OD_i} \leq \tau_{OD_i}$. The algorithm of synthesis of similar objects described in more detail is given in [2].

The average execution time $t_{g_{j(k)}/OD_i}$ of the point g_{j_k} on i-th operational device is determined by the expression:

$$t_{g_{j(k)}/OD_i} = \vartheta_{g_{j(k)}} \tau^*_{ODy_i}$$

As $\tau^*_{OD_i} \leq \tau_{OD_i}$, then and $t_{g_{j(k)}/OD_i} \leq t_{g_{j(k)}/c'(g_{j(k)})}$. Therefore average time T of execution of the longest route of the graph $I_{a,HD}$, calculating using the of durations $t_{g_{j(k)}/OD_i}$ does not exceed $T_{giv}, i.e. T \leq T_{giv}$. At the first stage $T'_{giv} = T_{giv}$.

9) The second stage of TS- coordination of the parallel DCS (clarification).

   At this stage the bias $\triangle T_{bi} = T_{giv} - T$ is calculated, and if $\triangle T_{bi} \neq 0$ all operations of the first stage (item 8) with $T'_{giv} = T_{giv} + \triangle T_{bi}$ are repeated. Otherwise (if $\triangle T_{bi} = 0$) the transition to item 11 is fulfilled.

10) The third stage of the TS- coordination of the parallel DCS (finishing).

   After the second stage of the TS- coordination of DCS the value of T can be both more, and less than of T_{giv}. In this case the repetition of operations described at of item 8 is necessary, with the single purpose: by minimal (± 1) changes of T'_{giv} to approach maximally T to T_{giv} providing simultaneous fulfillment of an inequality $T \leq T_{giv}$.

11) The end.

References

[1] Bajanov Iu. Strukturno-vremennoe soglasovanie operatsionnyh elementov kombinatsionnogo tipa./v kn. Modelirovanie i optimizatsia slojnyh sisitem Trudy VGAVT, vyp.273, part 1, N.Novgorod: 1996. (Russian)

[2] Bajanov Iu. Structurno-vremennoe soglasovanie operationnyh ustroistv v kn. Sistema jbrabotki informatsii i upravlenia. Mejvuz. nauthn. sb., N.Novgorod: NGTU, 1997. (Russian)

[3] cfarland M., Parker E., Camposano R. Vysokourovnevyj sintez tsifrovyh sistem. -TIIER, v.78, No.2, 1990. (Russian)

[4] Metlitskij T., Kaverznev V. Sistemy parallelnoj pamiati: Theoria, proektirovanie, primenenie /Red.Timohin V. L.:LGU, 1989. (Russian)

[5] Ilovaiskij I., Sidristyj B. Osnovy Theorii proektirovania tsifrovyh mashin i sistem. Novosibirsk: Nauka, 1976.

[6] Majorov S., Novikov G. Principy organizacii tsifrovyh sistem. - L.: Mashinostroenie, 1974.

[7] Majorov S., Novikov G. Struktura electronnyh vychislitelinyh mashin - L.: Mashinostroenie, 1979.

[8] Baranov S. Sintez mikroprogrammnyh avtomatov (graf-shemy i avtomaty). L.: Energia, 1979.

[9] Lazarev V., Pijli E. Sintez upravleaushih avtomatov. .: Energoizdat, 1989.

[10] Linger R., Mills H., Uitt B. Teoria i praktika strukturnogo programmirovania. M.: Mir, 1982.

[11] Mishenko V., Gorodeckij L., Gurskij L. i dr.; Red. Mishenko V. Intelektualinye sistemy avtomatizirovannogo proektirovania bolishih i sverhbolishih integralinyh mikroshem. M.: Radio i sveazi, 1988.

[12] Red. Moudera J., Elmagrabi S. Issledovanie operacij: v 2-h tomah. T.2. Modeli i primenenia. M.: Mir, 1981.

[13] Fridman A., Menon P. Teoria i proektirovanie perekluchatelnyh shem. M.: Mir, 1978.

V.Kondrat'ev, V.Utrobin, A.Balabanov
Received 2 Novembre, 1997

V.Kondrat'ev, V.Utrobin,
State Technical University of
Nizhniy Novgorod (NSU),
Computing mashinery Departament,
str. Minin, 24,
603000, N.Novgorod, Russia,
phone: (8312) 36-82-28; Fax 360-569
e–mail: *nstu@nnpi.nnov.su*

A.Balabanov,
Technical University of Moldova,
Automatica Departament,
Stefan cel Mare 168,
2012, Kishinev, Moldova,
phone: (373-2) 497694, 497014;
fax: (373-2) 497014, 497022
e–mail: *besliu@cc.acad.md* or *ast@utm.usm.md*