

Fault-tolerant on-board computer systems designing based on symmetry groups modeling

Liubov.S. Lomakina, Maksim.A. Nadezhkin, Nikolay.V. Sil'yanov

Computer systems and technologies department
Nizhny Novgorod State Technical University n. a. R.E. Alexeev
Nizhny Novgorod, Russian Federation
llomakina@list.ru

Abstract — Structure of on-board computer system based on a high-speed switching network is shown. Tolerance for single fault of processing modules is provided. Fault-tolerance based on models of symmetry groups is considered. Algorithm for generating reconfiguration tables is developed. The example is shown applying to four-node structure with a dihedral symmetry group.

Keywords—fault-tolerance, redundant, reconfiguration, automorphism, symmetry groups

I. INTRODUCTION

Problems of ensuring stability of computing systems to failures had been known for a long time [1]. Despite the fact, these problems are still actual, because of continuously increasing degree of automation both in high-responsible applications and in non-serviced objects. Areas of responsible use are those where the failure of the system threatens people's lives or those where unpredictable breakdowns and downtime will entail heavy economic consequences. Non-serviced objects are systems where maintenance is very seldom performed or there is no possibility of conducting it.

At the same time, issues of ensuring the fail-safety of on-board computer systems acquire special urgency, since several of the listed categories can classify the controlled object. Additionally, the system must be operable in harsh conditions (climatic, mechanical, under the influence of various kinds of radiation, natural and artificial interference).

Fault tolerance can mean high reliability indicators, which are provided by the use of elements and circuits with a large margin of reliability, as well as increased attention to manufacturing and assembly technologies. Another approach to fault tolerance does not exclude the occurrence of malfunctions, but their adverse effects are prevented or minimized by introducing a certain form of redundancy [2, 3].

II. ON-BOARD COMPUTER SYSTEM'S STRUCTURE

For further analysis, the simplified structure of the on-board computer system is shown in Fig. 1. The system is built on the basis of a high-speed SpaceWire switching network. Detailed structural diagram and components of the system are considered in [4].

The switch-router module has spare ports that can be used for scaling or redundancy. While each processor has its own memory (both ROM and RAM), this structure can be classified as a multi-machine [5]. Multi-machine structures allow simultaneous increasing in performance and fault tolerance. In this case, of course, the control algorithms must have the possibility of parallelization.

In this case, redundancy is introduced only for processing modules, assuming that they are more likely to suffer from failures than other modules. The assumption is based on the

traditional idea that the least reliable element of computer technology is RAM [6]. Microchips used in the module of specialized RAM for storing intermediate results have an increased level of fault-tolerance, but their volume is low for applications in processing modules. The interface module and switch-router module are built on the basis of application specific integrated circuits, which (in comparison with RAM) are also less prone to failures. Therefore, redundancy is further considered only with regard to processing modules.

It should also be noted that in this paper the minimum redundancy is considered, namely, it has fault tolerance when only one module fails. Accordingly, in the structure diagram only one backup processing module is added, as shown in Fig. 2. Fault tolerance in general case, i.e. when k modules are refused, is considered in [7].

According to Fig. 2, the hardware of the on-board computer system has a fully connected structure. The logical connections structure, i.e. a distribution of data streams and control information in parallel processing is less obvious and can be implemented in different ways when developing the system software. Therefore, the subject of further consideration is searching of general recommendations for logical level system designing that allow the considered level of fault tolerance.

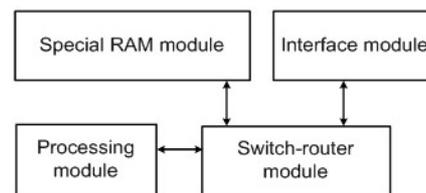


Fig. 1. The initial structure of the on-board computer system.

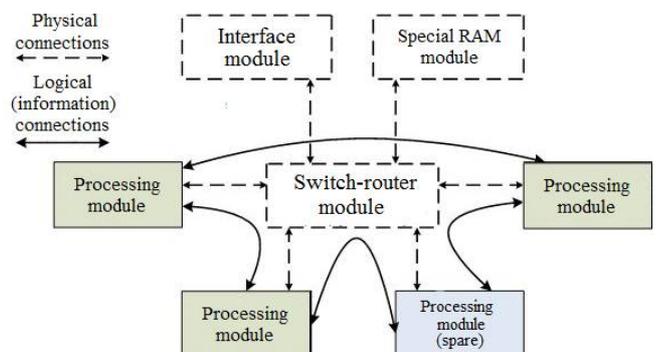


Fig. 2. Multi-machine structure with backup module.

III. COMMUNICATION MODELS FOR PROCESSING MODULES

The structure models as a graph whose vertices correspond to processing modules, and arcs to logical (informational) links between them. Further supposed, that the redundant processing module corresponds to the vertex V_0 . The applied model assumes that only the vertices of the graph are subjects to failures. The switching network is not subject to failures, i.e. if the processing module fails, the system can be reconfigured in accordance with Fig. 2 due to reconfiguring the switch-router module and replacing the failed processing module with a redundant. In the simplest case, the system has two modules, while the redundant module replaces the failed module, as shown in Fig. 3a.

When organizing logical connections in a structure with three modules, one should take into account that in the chains in Fig. 3b and Fig. 3c the redundant module can only replace one of the main modules. Greater fault tolerance will be achieved in structures with the organization of logical connections, shown in Fig. 3d and Fig. 3e. In these cases, the redundant module is able to replace any of the main.

When a system dimension increases, the number of connection options between modules increases correspondingly, and the organization of logical connections becomes less obvious. As an example, Fig. 4 shows structures with four processing modules. In Fig. 4a-4d, the vertex V_0 is adjacent to only one of vertices of the graph. The redundant module is able to replace only one of the main. The best features to ensure fault tolerance in accordance with Fig. 4g are in the fully connected graph. With a further increase in the dimensionality of the system, the organization of such a structure will become more complex and more expensive.

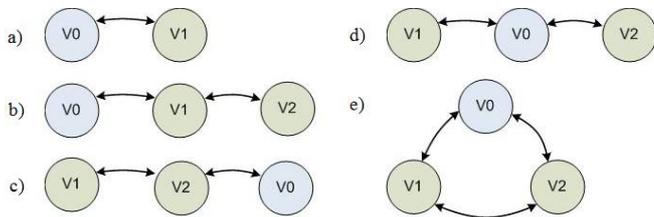


Fig. 3. Structures with two and three processing modules.

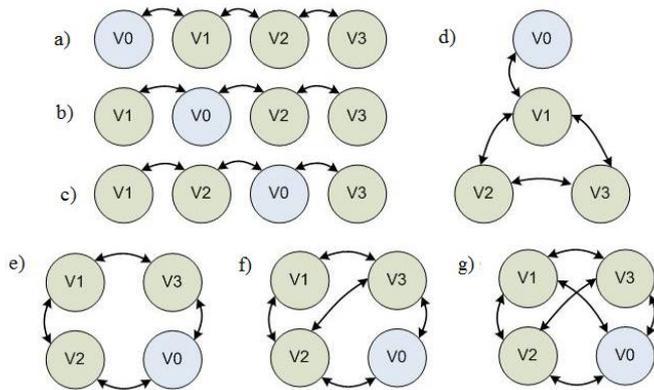


Fig. 4. Structures with four modules.

Analyzing the remaining two examples, it should be noted that it is possible to replace any of the main modules by a redundant one, without violating the incidence, only for the case in Fig. 4g.

In the general case, it was shown in [8] that fault tolerance with minimum possible redundancy is realizable only in structures possessing a certain kind of mathematical symmetry. The discussed model supposes that replacing a failed module with a redundant equals renaming the logical names of the graph vertices. In the process, the redundant vertex replaces the failed one without changing the incidence between the new logical names. Such a transformation of a graph is invariant and is also called an automorphism. The set of automorphisms forms a symmetry group. Further, cyclic, dihedral, and tetrahedral symmetry groups are discussed.

Knowing the symmetry group that the system possesses, we can create a table of group transformations. A number of transformations is equal to the order of a symmetry group. A specified table will uniquely determine a reconfiguration process after failure. For the system shown in Fig. 2, the module of the switch-router must perform diagnostics and reconfiguration, respectively, the reconfiguration tables should be stored in its memory. The following is the algorithm of generating reconfiguration tables.

IV. THE ALGORITHM OF GENERATING RECONFIGURATION TABLES

Simplified, the reconfiguration algorithm can be represented as a linear and written in the text form:

- 1) *Data input.*
- 2) *Transformation of an input data into a form of an original automorphism.*
- 3) *Compilation of a set of possible automorphisms of an original graph.*
- 4) *Transformation of all automorphisms of an original graph into a set of possible reconfigurations.*
- 5) *Generating for each vertex a set of reconfigurations in case of failure.*
- 6) *Output results.*

Further, each stage is considered in more detail.

At the first stage, as an input data, the algorithm takes a list of vertices of a structure, obtained by traversing the original graph clockwise, and the type of symmetry group possessed by a structure under study.

At the second stage, based on a list of graph vertices, a structure is formed, which is an initial automorphism of a graph in the form used to represent a graph in the process of the algorithm. In the developed algorithm, any automorphism of a graph is a set of mappings between logical names of vertices of a graph and a numbers of their positions relative to the origin. The origin of the count is the position of the vertex of the graph in space, the logical name of which is the first in a list of vertices. The numbering of the vertices is clockwise. The initial graph is formed as follows: each of the vertices in a list is associated with its ordinal number in a list of vertices of the graph (the numbering starts from zero).

In the third stage, a set of possible automorphisms of the original graph is compiled, in the process of searching for a set over a graph, rotation and reflection operations can be performed.

The operation of rotating the graph by k positions relative to the axis passing through the center begins with the calculation of the value $m = k \bmod n$, where n is the number of vertices of the graph. In other words, m characterizes the value of the “pure rotation” of the graph and does not take into account the complete rotations of the graph about its axis for $k \geq n$. Then for each of vertices the following sequence of actions is performed:

- A new position in the space $g = p + m$ is determined, where p is the number of the vertex position before the rotation.
- To ensure the cyclicity (next to the last position of the vertex of a graph follows the first), it is necessary to control the value of the new position of the vertex. If it is negative or exceeds the value $n - 1$, then n or $-n$ must be added to the position obtained.
- The vertex is assigned its new position in space.

The operation of rotation of a graph relative to the second-order axis (reflection relative to the axis of symmetry passing through the center) is realized by pairwise permutation of the vertices placed to the right and left of an axis of symmetry:

- A list of pairs of vertices is generated, which must be interchanged to achieve a reflection effect on the axis.
- For each of a pairs of the formed list, the values of the vertex position are exchanged. Thus, each of vertices varies in places with an opposite axis.

The process of forming a set of automorphisms is based on the knowledge of the symmetry group to which the structure under consideration corresponds. In other words, for each of a possible symmetry groups an obtaining of the set of possible automorphisms performs in a special way.

The cyclic symmetry group (C_n) is a group of cyclic rotations through an angle $\alpha = 360^\circ / N$ about an axis passing through the center O of the system. Its order is $|C_n| = N = n$ and it contains n automorphisms.

The algorithm for searching automorphisms for structures possessing a given symmetry group looks as follows. For each of the numbers of the sequence $0 \dots n - 1$, iteration is performed:

- The original graph is rotated by i (in the positive direction, a clockwise rotation is adopted), where i is the number relative to which the iteration is performed. This operation is equivalent to rotating the graph around the point O by an angle $\alpha = 360^\circ \cdot i / N$;
- The obtained automorphism is preserved in the list of automorphisms of the original graph.

The dihedral symmetry group (D_n) consists of proper rotations in space at angles $\alpha = 360^\circ / (N/2)$ around the axis passing through the center O , and 180° around $N/2$ axes of the

second order. Its order is $|D_n| = N = 2n$ and it contains $2n$ automorphisms: identical automorphism; $n - 1$ rotations by an angle $2\pi k / n$, where $k = 1 \dots n - 1$; and n reflections with respect to an axes passing through the center O and forming angles multiples of $2\pi / n$.

The algorithm for searching automorphisms for structures possessing a given symmetry group looks as follows. For each of the numbers of the sequence $0 \dots n - 1$, iteration is performed:

- The original graph is rotated by i (in the positive direction, a clockwise rotation is adopted), where i is the number relative to which the iteration is performed. This operation is equivalent to rotating the graph around the point O by an angle $\alpha = 360^\circ \cdot i / N$.
- The obtained automorphism is preserved in the list of automorphisms of the original graph.
- For the resulting automorphism, a reflection operation is performed with respect to an axis of symmetry. For structures possessing the symmetry D_n , an axis passing through the edge between vertices with numbers 0 and 1. For structures possessing the symmetry $D_n/2$, two types of symmetry axes are possible: analogous to the structures of the group D_n , or an axis passing through vertices with numbers 0 and $(n / 2) - 1$.
- The obtained automorphism is preserved in the list of automorphisms of the original graph.

The tetrahedral symmetry group (T) is closely related to the representation of a planar graph in the form of a spatial. The tetrahedron has four axes of symmetry passing through its vertices and the centers of the opposite faces. Around each axis, in addition to the identity, two more rotations are possible. Thus, the tetrahedron rotation group has twelve permutations including the identity.

The algorithm for searching automorphisms for structures possessing a given symmetry group looks as follows. For each of the numbers of the sequence $0 \dots n - 1$, iteration is performed:

- The original graph is rotated by i (in the positive direction, a clockwise rotation is adopted), where i is the number relative to which the iteration is performed.
- From the obtained graph a subgraph is allocated, by removing the vertex of the graph with the position $n - 1$.
- For the obtained subgraph there are all automorphisms of the cyclic symmetry group.
- To each of the obtained subgraphs, a previously deleted vertex with position $n - 1$.
- All the obtained graphs are stored in the list of automorphisms of the original graph.

It should be explained that the result of each iteration is the automorphisms of the original graph obtained by rotating around the fixed symmetry axis of the tetrahedron passing through the vertex with the position $n - 1$. A complete group of automorphisms is obtained by rotating a tetrahedron around the axes passing through each vertex of a tetrahedron, so in the

case of a fixed axis it is necessary that at each iteration the axis pass through a unique vertex, which is achieved by rotating the original automorphism at each iteration.

Also, symmetry groups of a octahedron (W) and an icosahedron (P) can be used to search for automorphisms [8], however, the current implementation of the developed algorithm does not support the work with them.

At the fourth stage, a transformation of automorphisms into reconfiguration tables is carried out. To obtain reconfiguration tables of the system in the event of a failure, it is necessary to form a set of possible reconfigurations of the system based on the set of automorphisms of the graph. The transformation of each automorphism into the reconfiguration table is as follows: for each of the vertices of the automorphism, a vertex of the original automorphism is put in correspondence, having the same position relative to the origin. Thus, for each vertex a unique correspondence of the “old” (before reconfiguration) of the logical name to the “new” (after reconfiguration) is formed.

At the fifth stage, reconfiguration tables are constructed for the failed vertices. To form a reconfiguration tables, each of a vertices is alternately considered as a failed. Further, from the set of all possible reconfigurations, only those in which the failed vertex has changed its logical name to the name of the spare vertex are selected (the condition for replacing the failed worker vertex).

The sixth stage is the output of the algorithm results. As a result of the algorithm, a set of reconfiguration tables for each of vertices is generated, assuming that this vertex failed. Each group transformation represents a substitution in the form

$$\begin{pmatrix} V_0 & V_1 & \dots & V_i & \dots & V_n \\ V_i & V_j & \dots & V_n & \dots & V_k \end{pmatrix}$$

Such a record means that for this substitution the vertex V_0 replaces V_i , V_1 replaces V_j , etc.

V. THE EXAMPLE OF GENERATING RECONFIGURATION TABLES

The algorithm results in the form of reconfiguration tables for the discussed structure shown in Fig. 4g. This structure has a symmetry group D_4 and has eight automorphisms. The vertex V_0 is adopted as a redundant. The result of the algorithm will be sets of reconfiguration tables in case of failure of vertices V_1 , V_2 , V_3 (tables 1–3).

TABLE I. RECONFIGURATION IN CASE OF A VERTEX V_1 FAILURE

	Reconfiguration variant							
	a				b			
Logical name before	V_0	V_1	V_2	V_3	V_0	V_1	V_2	V_3
Logical name after	V_1	V_0	V_2	V_3	V_1	V_0	V_3	V_2

TABLE II. RECONFIGURATION IN CASE OF A VERTEX V_2 FAILURE

	Reconfiguration variant							
	a				b			
Logical name before	V_0	V_1	V_2	V_3	V_0	V_1	V_2	V_3
Logical name after	V_2	V_3	V_0	V_1	V_3	V_2	V_0	V_1

TABLE III. RECONFIGURATION IN CASE OF A VERTEX V_3 FAILURE

	Reconfiguration variant							
	a				b			
Logical name before	V_0	V_1	V_2	V_3	V_0	V_1	V_2	V_3
Logical name after	V_2	V_3	V_1	V_0	V_3	V_2	V_1	V_0

For example, in Table 1, the reconfiguration option specifies a unique and optimal match of the logical vertices names before and after reconfiguration, namely: the vertex V_0 replaces V_1 , V_1 – V_0 , and the vertices V_2 and V_3 do not change their logical names during the reconfiguration.

VI. CONCLUSION

For on-board computer systems fault tolerance with minimal redundancy is an actual task. Taking into account limitations on mass and dimensions, it is advisable to reserve the least reliable elements of a system. In this case, an organization of logical connections between processing modules must be performed on the basis of models of symmetry groups.

The developed algorithm allows generating reconfiguration tables for systems whose structure has cyclic, dihedral or tetrahedral symmetry. Reconfiguration tables should be stored in a memory of the switch-router module and used by it when reconfiguring the system.

Development of algorithms for verifying system functioning and determining the reliability of such inspections are the subjects for further researches.

REFERENCES

- [1] Avizienis A., “Fault-tolerant computing: an overview”, IEEE Computer, vol. 4, no. 1, pp. 5-8, February 1971.
- [2] Sil’yanov N.V., “Fault-tolerant on-board computers design problems”, in Modern informatization problems in the technological and telecommunication systems analysis and synthesis, Proc. of the XXI International Open Science Conference. Yelm: Science Book Publishing House, 2016, pp. 283–288.
- [3] Sil’yanov N.V., Lomakina L.S., “Voprosy razrabotki otkazoustoichivoy bortovoy EVM”, Proc. of XV International science-technical conference Budushee Tekhnicheskoy Nauki. N. Novgorod: NSTU, 2016, p. 120.
- [4] Sil’yanov N.V., “Razrabotka apparatnogo obespecheniya mnogofunkcionalnoy bortovoy vychislitelnoy sistemy”, Sensors and Systems, 2017, in press.

- [5] Larionov A.M., Vychislitelniye kompleksy, sistemy i seti, Leningrad: Energoatomizdat, 1987.
- [6] Gorshkov V.N., Nadezhnost operativnykh zapominaushikh ustroystv EVM, Leningrad: Energoatomizdat, 1987.
- [7] Karavay M.F., “Invariantno-gruppovoy podkhod k issledovaniyu k-otkazoustoychivyykh struktur”, *Avtomatika i telemekhanika*, no. 1, 2000, pp. 144-156.
- [8] Karavay M.F., “Primeneniye teorii simmetrii k analizu i sintezu otkazoustoychivyykh sistem”, *Avtomatika i telemekhanika*, no. 6, pp. 159-173, 1996.