# Graph-Logic Models of Hierarchical Fault-Tolerant Multiprocessor Systems

**Alexei M. Romankevich, Kostiantyn V. Morozov, Vitaliy A. Romankevich**

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

**Summary**
The work discusses the further development of graph-logic models (GL-models), which reflect the behavior of fault-tolerant multiprocessor systems (FTMS) in the component failover stream. The main attention is paid to the method of combining GL-models of different FTMSs into a single model. Constructing approach is proposed for graph-logic models of hierarchical failover multiprocessor systems consisting of several subsystems. Primarily, it concerns systems for managing complex objects, where the principle of partitioning a complex task of management into a set of simpler ones is often used. The approach is to construct individual models for each of the subsystems and combine them into a single model, also possessing a hierarchical property. Each of these submodels can be constructed by any convenient method. One of the advantages of this approach is the relative model transformation simplicity in case of individual subsystems modification. Such FTMSs are called k-out-of-n systems.

***Keywords:***
*fault-tolerant multiprocessor systems, graph-logic GL model models, reliability calculation*

## 1. Introduction

Fault-tolerant multiprocessor systems (FTMS) are widely used in the modern technical world, in particular, as control systems for various objects. The reliability of the complex (object + control system) as a whole, as well as its management systems in particular, is especially important for the aviation, space, rail, defense, industrial, energy and other industries, i.e. where the failure of the system can lead to significant economic losses, threaten the life or health of people, etc.

In the course of the development of FTMS, it is necessary to calculate its reliability both for confirming compliance with the set requirements, and in order to find the most unreliable nodes to be further modified. One of the methods for calculating the reliability of FTMS [1] is based on carrying out statistical experiments with so-called graph-logic models (GL models).

The GL-model is an undirected graph, each of whose edges corresponds to the boolean edge function [2]. The arguments of such functions are the values of the state vector system components, i.e. a vector whose components correspond to the state of the system's processors and assume a value of 1 if the corresponding processor is valid and 0 if it fails. If the edge function takes the null value, the corresponding edge is excluded from the graph. The connection of the graph corresponds to the regularity of the system as a whole.

In [3, 4], methods for constructing GL models with different properties were proposed. In particular, the model described in [3] is convenient in that it loses the minimum of edges when the number of failures is allowed. The proposed methods allow the models construction of so-called basic systems, i.e. those that remain valid until the number of failures does not exceed the specified value (k-out-of-n system) [3, 5]. The basic GL models will be K (m, n), where n is the number of system processors, and m is the maximum number of failures. For other types of systems, which are called non-base ones, modification methods of basic GL models [6-9] are proposed. GL models are used to calculate the parameters such as reliability and security of multiprocessor systems.

Using the above methods of constructing GL-models, one can construct a graph-logical model of any system, however, in certain cases (usually for fairly complex systems), such a model may turn out to be very cumbersome. Therefore, it is logical to consider some systems separately, taking into account those or other properties, for example, architecture. In particular, for hierarchical systems, i.e. those that consist of a plurality of separate subsystem modules distributed over several levels of the hierarchy, it is logical to assume that the corresponding model should also be constructed in the form of a hierarchy of several submodels that need to be combined into a single model. This task is solved in this work.

## 2. Building a Model for a Hierarchical System

Large systems usually solve complex tasks that can be broken into many individual subtasks. Each such subtask, which in its turn is also quite complex, can be solved by a separate group of processors - a subsystem. Complex subsystems, in turn, can also consist of subsystems of a lower level, etc.

Manuscript received July 5, 2019
Manuscript revised July 20, 2019

Thus, the system is divided into several levels of the organization (hierarchy): from the highest - the system as a whole, to the lowest - individual processors. This approach can greatly simplify both the construction of systems and models, as well as their maintenance.
Consider a system that has several levels of the hierarchy, where each of the subsystems can consist of subsystems of the next hierarchy levels and can be stable to fail some of them. Such a system in the general case is non-base and, naturally, a non-hierarchical model can be constructed for it by means of one of the known methods. However, such a model will be very complicated, since the number of processors in the system, and, consequently, variables, is large enough.
On the other hand, the model of the described hierarchical system can also be hierarchical and consist of a corresponding number of submodels. As components of the input vector for a submodel of a higher level, the values obtained with the submodel of the lower level of the hierarchy and corresponding to the state of the subsystems associated with it are used. These values are in fact the variables of the higher-level sub-model sub-model edges. Statistical experiments are initially performed with submodels of the lowest level of the hierarchy and end with experiments with the submodel of the highest level.
Indeed, knowing the states of the components of the subsystem at each instant of time, one can construct the state vector of this subsystem, and to determine its behavior, a corresponding GL model can be constructed. Values obtained using such models that actually describe the state of the subsystems can in turn be used to form a state of the higher-level subsystem state. Taking into account the nature of the interrelationships between subsystems of the lower levels, including the reconfiguration possibility and ensuring the subsystem failures stability, it is possible to construct a higher subsystem model, etc. By constructing models of all subsystems of all levels, we will eventually get a model describing the system state as a whole. We note that the models of each of the subsystems can be non-base and, generally speaking, for each of them, one can choose its most convenient way of construction.
As we see, the resulting model is represented as a hierarchy of several more simple submodels, which in this case corresponds to the architecture of the system. We note that generally speaking such an agreement is optional. As we have already noted for a hierarchical system, a non-hierarchical model can also be constructed. In turn, for a non-hierarchical system it is possible to construct both hierarchical and non-hierarchical models. In addition, the structure, i.e. The number and size of hierarchy levels in the system and model may vary.
Thus, the proposed approach follows the principle of partitioning a complex task into a set of simpler ones that has several advantages. In addition to simplifying the process of building a model of the system as a whole, the process of its modification may also be simplified in the event of a subsequent refinement of the system. So, in the case of changing its individual nodes, it is sufficient to modify only the corresponding submodels.
For clarity, consider examples of constructing models of hierarchical systems by the proposed method.

## Example 1

Let there be a hierarchical system consisting of five subsystems and which is stable to the failure of two of them (Fig. 1). The first subsystem contains 11 processors and is resistant to failure of 4 of them, the second subsystem contains 9 processors and is resistant to failure 3 of them, the third - contains 12 processors and is resistant to failure of 5 of them, the fourth - contains 7 processors and is resistant to failure 2 of and fifth - contains 15 processors and is resistant to failure of 4 of them.
Note that the subsystems are chosen as basic only for simplicity. At the same time, if any of the subsystems are non-base, to construct adequate models, one could use one of the modification methods [7-9] to include them in the general model.

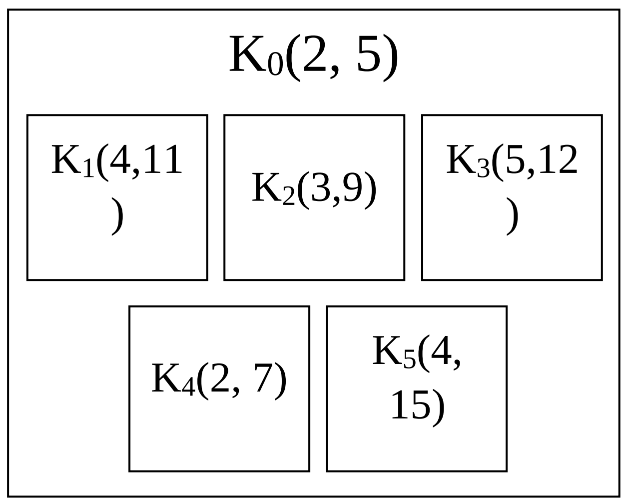


Fig. 1 Structure of the system

We will construct a model of such a system. First we will construct models of subsystems. The first subsystem will correspond to the model K (4, 11). The method of forming GL-models is presented in [3]. Here and thereafter, we present the model's graph edge functions, freeing the reader from the need for a detailed acquaintance with [3]. The model K (4, 11) has 8 edges with functions

$$f_1^1 = x_1 \vee x_2 \vee x_3 \vee x_4 x_5 x_6$$

$$f_2^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6)$$

$$f_3^1 = x_1 x_2 x_3 \vee x_4 \vee x_5 \vee x_6$$

$$f_4^1 = (x_1 \vee x_2 \vee x_3)((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5 x_6) \wedge$$
$$\wedge (x_1 x_2 x_3 \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6))(x_4 \vee x_5 \vee x_6) \vee x_7 x_8 x_9 x_{10} x_{11}$$

$$f_5^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_4 x_5 x_6)(x_4 \vee x_5)(x_4 x_5 \vee x_6) \vee$$
$$\vee (x_7 \vee x_8)(x_7 x_8 \vee x_9)(x_7 x_8 x_9 \vee x_{10} x_{11})(x_{10} \vee x_{11})$$

$$f_6^1 = x_1 x_2 x_3 x_4 x_5 x_6 \vee (x_7 \vee x_8 \vee x_9)((x_7 \vee x_8)(x_7 x_8 \vee x_9) \vee x_{10} x_{11}) \wedge$$
$$\wedge (x_7 x_8 x_9 \vee x_{10} \vee x_{11})$$

$$f_7^1 = x_7 \vee x_8 \vee x_9 \vee x_{10} x_{11}$$

$$f_8^1 = (x_7 \vee x_8)(x_7 x_8 \vee x_9) \vee x_{10} \vee x_{11}$$

The graph of this model will have the form shown in Fig. 2

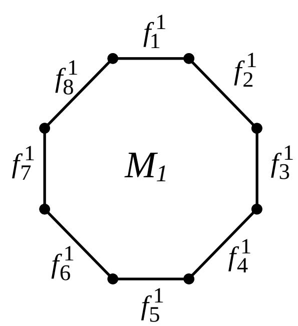


Fig. 2 Model K (4, 11)

The reader can verify the system model behavior in the flow of failures. For example, if the state of the subsystem appears <x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11> = <0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1>, containing four zeros (which means that there is a failure of the processors x1, x4, x6 and x10 ), only one function is reset $f_4^1$.

Then only one edge is excluded from the graph, and its connection is not lost, which corresponds to the normal state of the system. At the same time, with theappearanceofthevector<x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11>=<1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1>, containing five zeros, the zero value will already have two functions $f_5^1$ and $f_6^1$, and two edges will be excluded from the graph. The graph will lose connectivity, which corresponds to the failure of the system.

The second subsystem will correspond to the model K (3, 9) with the edge functions:

$$f_1^2 = x_{12} \vee x_{13} \vee x_{14}$$

$$f_2^2 = (x_{12} \vee x_{13})(x_{12} x_{13} \vee x_{14}) \vee x_{15} x_{16}$$

$$f_3^2 = x_{12} x_{13} x_{14} \vee x_{15} \vee x_{16}$$

$$f_4^2 = (x_{12} \vee x_{13})(x_{12} x_{13} \vee x_{14})(x_{12} x_{13} x_{14} \vee x_{15} x_{16})(x_{15} \vee x_{16}) \vee$$
$$\vee x_{17} x_{18} x_{19} x_{20}$$

$$f_5^2 = x_{12} x_{13} x_{14} x_{15} x_{16} \vee (x_{17} \vee x_{18})(x_{19} \vee x_{20})$$

$$f_6^2 = x_{17} \vee x_{18} \vee x_{19} x_{20}$$

The third subsystem will correspond to the model K (5, 12) with the edge functions:

$$f_1^3 = x_{21} \vee x_{22} \vee x_{23} \vee (x_{24} \vee x_{25})(x_{24} x_{25} \vee x_{26})$$

$$f_2^3 = (x_{21} \vee x_{22})(x_{21} x_{22} \vee x_{23}) \vee x_{24} \vee x_{25} \vee x_{26}$$

$$f_3^3 = (x_{21} \vee x_{22} \vee x_{23} \vee x_{24} x_{25} x_{26}) \wedge$$
$$\wedge ((x_{21} \vee x_{22})(x_{21} x_{22} \vee x_{23}) \vee (x_{24} \vee x_{25})(x_{24} x_{25} \vee x_{26})) \wedge$$
$$\wedge (x_{21} x_{22} x_{23} \vee x_{24} \vee x_{25} \vee x_{26}) \vee x_{27} x_{28} x_{29} x_{30} x_{31} x_{32}$$

$$f_4^3 = (x_{21} \vee x_{22} \vee x_{23})((x_{21} \vee x_{22})(x_{21} x_{22} \vee x_{23}) \vee x_{24} x_{25} x_{26}) \wedge$$
$$\wedge (x_{21} x_{22} x_{23} \vee (x_{24} \vee x_{25})(x_{24} x_{25} \vee x_{26}))(x_{24} \vee x_{25} \vee x_{26}) \vee$$
$$\vee (x_{27} \vee x_{28})(x_{27} x_{28} \vee x_{29})(x_{27} x_{28} x_{29} \vee x_{30} x_{31} x_{32}) \wedge$$
$$\wedge (x_{30} \vee x_{31})(x_{30} x_{31} \vee x_{32})$$

$$f_5^3 = (x_{21} \vee x_{22})(x_{21} x_{22} \vee x_{23}) \wedge$$
$$\wedge (x_{21} x_{22} x_{23} \vee x_{24} x_{25} x_{26})(x_{24} \vee x_{25})(x_{24} x_{25} \vee x_{26}) \vee$$
$$\vee (x_{27} \vee x_{28} \vee x_{29})((x_{27} \vee x_{28})(x_{27} x_{28} \vee x_{29}) \vee x_{30} x_{31} x_{32}) \wedge$$
$$\wedge (x_{27} x_{28} x_{29} \vee (x_{30} \vee x_{31})(x_{30} x_{31} \vee x_{32}))(x_{30} \vee x_{31} \vee x_{32})$$

$$f_6^3 = x_{21} x_{22} x_{23} x_{24} x_{25} x_{26} \vee (x_{27} \vee x_{28} \vee x_{29} \vee x_{30} x_{31} x_{32}) \wedge$$
$$\wedge ((x_{27} \vee x_{28})(x_{27} x_{28} \vee x_{29}) \vee (x_{30} \vee x_{31})(x_{30} x_{31} \vee x_{32})) \wedge$$
$$\wedge (x_{27} x_{28} x_{29} \vee x_{30} \vee x_{31} \vee x_{32})$$

$$f_7^3 = x_{27} \vee x_{28} \vee x_{29} \vee (x_{30} \vee x_{31})(x_{30} x_{31} \vee x_{32})$$

$$f_8^3 = (x_{27} \vee x_{28})(x_{27} x_{28} \vee x_{29}) \vee x_{30} \vee x_{31} \vee x_{32}$$

The fourth subsystem will correspond to the model K (2, 7) with the edge functions:

$$f_1^4 = x_{33} \vee x_{34}$$

$$f_2^4 = x_{33}x_{34} \vee x_{35}x_{36}$$

$$f_3^4 = x_{35} \vee x_{36}$$

$$f_4^4 = x_{33}x_{34}x_{35}x_{36} \vee x_{37}x_{38}x_{39}$$

$$f_5^4 = x_{37} \vee x_{38}$$

$$f_6^4 = x_{37}x_{38} \vee x_{39}$$

The fifth subsystem will correspond to the model K (4, 15) with the edge functions:

$$f_1^5 = x_{40} \vee x_{41} \vee x_{42} \vee x_{43}$$

$$f_2^5 = (x_{40} \vee x_{41} \vee x_{42}x_{43})(x_{40}x_{41} \vee x_{42} \vee x_{43}) \vee x_{44}x_{45}x_{46}x_{47}$$

$$f_3^5 = (x_{40} \vee x_{41})(x_{40}x_{41} \vee x_{42}x_{43})(x_{42} \vee x_{43}) \vee$$
$$\vee (x_{44} \vee x_{45})(x_{44}x_{45} \vee x_{46}x_{47})(x_{46} \vee x_{47})$$

$$f_4^5 = x_{40}x_{41}x_{42}x_{43} \vee (x_{44} \vee x_{45} \vee x_{46}x_{47})(x_{44}x_{45} \vee x_{46} \vee x_{47})$$

$$f_5^5 = x_{44} \vee x_{45} \vee x_{46} \vee x_{47}$$

$$f_6^5 = (x_{40} \vee x_{41} \vee x_{42}x_{43})(x_{40}x_{41} \vee x_{42} \vee x_{43}) \wedge$$
$$\wedge ((x_{40} \vee x_{41})(x_{40}x_{41} \vee x_{42}x_{43})(x_{42} \vee x_{43}) \vee x_{44}x_{45}x_{46}x_{47}) \wedge$$
$$\wedge (x_{40}x_{41}x_{42}x_{43} \vee (x_{44} \vee x_{45})(x_{44}x_{45} \vee x_{46}x_{47})(x_{46} \vee x_{47})) \wedge$$
$$\wedge (x_{44} \vee x_{45} \vee x_{46}x_{47})(x_{44}x_{45} \vee x_{46} \vee x_{47}) \vee x_{48}x_{49}x_{50}x_{51}x_{52}x_{53}x_{54}$$

$$f_7^5 = (x_{40} \vee x_{41})(x_{40}x_{41} \vee x_{42}x_{43})(x_{42} \vee x_{43}) \wedge$$
$$\wedge (x_{40}x_{41}x_{42}x_{43} \vee x_{44}x_{45}x_{46}x_{47}) \wedge$$
$$\wedge (x_{44} \vee x_{45})(x_{44}x_{45} \vee x_{46}x_{47})(x_{46} \vee x_{47}) \vee$$
$$\vee (x_{48} \vee x_{49})(x_{48}x_{49} \vee x_{50}x_{51})(x_{50} \vee x_{51}) \wedge$$

$$f_8^5 = x_{40}x_{41}x_{42}x_{43}x_{44}x_{45}x_{46}x_{47} \vee$$
$$\vee (x_{48}x_{49} \vee x_{50} \vee x_{51})(x_{48} \vee x_{49} \vee x_{50}x_{51}) \wedge$$
$$\wedge ((x_{48} \vee x_{49})(x_{48}x_{49} \vee x_{50}x_{51})(x_{50} \vee x_{51}) \vee x_{52}x_{53}x_{54}) \wedge$$
$$\wedge (x_{48}x_{49}x_{50}x_{51} \vee (x_{52} \vee x_{53})(x_{52}x_{53} \vee x_{54}))(x_{52} \vee x_{53} \vee x_{54})$$

$$f_9^5 = x_{48} \vee x_{49} \vee x_{50} \vee x_{51}$$

$$f_{10}^5 = (x_{48} \vee x_{49} \vee x_{50}x_{51})(x_{48}x_{49} \vee x_{50} \vee x_{51}) \vee x_{52}x_{53}x_{54}$$

$$f_{11}^5 = (x_{48} \vee x_{49})(x_{48}x_{49} \vee x_{50}x_{51})(x_{50} \vee x_{51}) \vee (x_{52} \vee x_{53})(x_{52}x_{53} \vee x_{54})$$

$$f_{12}^5 = x_{48}x_{49}x_{50}x_{51} \vee x_{52} \vee x_{53} \vee x_{54}$$

In accordance with the states of the subsystems calculated using the above-described models, a state vector of the model M0 <y1, y2, y3, y4, y5>, where yj corresponds to the connectivity of the graph of the corresponding submodel: 1 if the graph is connected and 0 otherwise. The behavior of the system as a whole in the fail flow will reflect the model K (2, 5) with the edge functions:

$$f_1 = y_1 \vee y_2$$
$$f_2 = y_1y_2 \vee y_3$$
$$f_3 = y_1y_2y_3 \vee y_4y_5$$
$$f_4 = y_4 \vee y_5$$

Thus, the system model has two levels of hierarchy and consists of six submodels (Fig. 3).

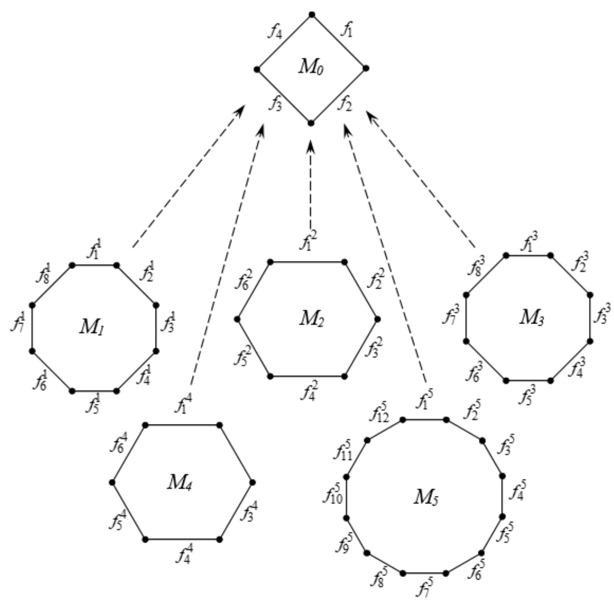


Fig. 3 Model of the hierarchical system

The reader has already paid attention to the fact that the number of subsystems in the example does not coincide with the number of submodels. It does not matter to calculate the reliability parameters of the FTMS.

## Example 2

Note that the number of hierarchy levels does not have to coincide for all subsystems. Assume that the developer has decided to change the system from Example 1, replacing the subsystem 5 with an entire complex consisting of six subsystems and stable to failure of two of them (Figure 4). These subsystems consist of 8, 4, 8, 5, 7 and 9 processors, and are resistant to failure, respectively, 3, 1, 2, 1,3 and 4 ofthem.

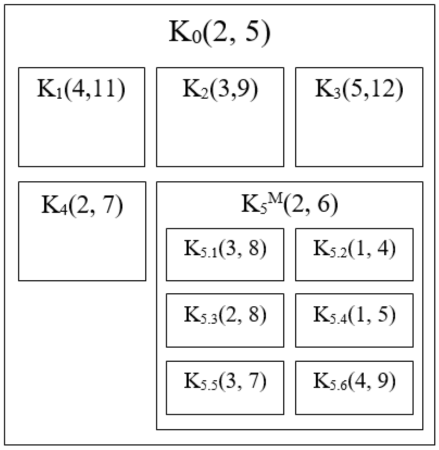


Fig. 4 Structure of the modified system

As already mentioned above, one of the advantages of a hierarchical approach to building models is the absence of the need to rebuild the entire model in the event of a change in a particular subsystem. Thus, it's enough to construct a new model of the fifth subsystem and replace it with the fifth submodel in the model from the previous example.

By constructing, as in the previous example, models of subsystems of the lower level, and then combining their model with M5, we will obtain a new GL model (Figure 5), which reflects the behavior of the modified system in the fail flow. The system model has three levels of hierarchy and consists of twelve submodels.

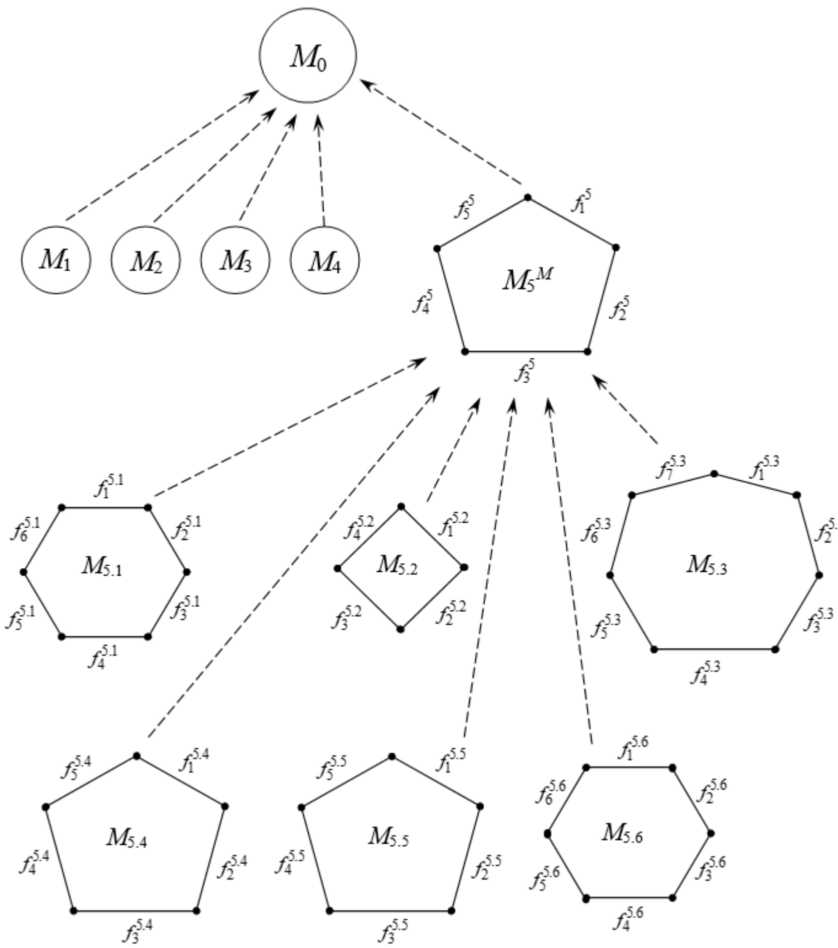


Fig. 5 Modified system model

## 3. Conclusions

The paper considers the further development of a universal approach to calculating the reliability of FTMS, based on the implementation of statistical experiments with GL models, which reflect the system's response to the occurrence of various multiplicities failures.

Large systems usually have a hierarchical structure, which simplifies their construction, modification and maintenance. Graph-logic models of such systems make sense also to make hierarchical ones. At the same time, for each subsystem, a separate, usually small, model is being built. The submodels of the lowest level are constructed on the basis of the corresponding fragments of the system state vector, and for the submodels of the upper levels, the input vectors are formed on the basis of the results obtained with the help of sub-modules of the lower level, while if the submodel graph remains connected, the corresponding component of the vector is set to 1, and if it loses connectivity, then 0. Thus, the statistical experiment is carried out in several stages, starting with submodels of the lower levels to the submodel of the highest level.

This approach does not limit the choice of how to construct graph-logic models: each of the submodels can be constructed in the most convenient way, while mixing different types of GL-models is allowed. In the case of non-base subsystems, the corresponding submodels can be modified by any known method.

In addition, one of the advantages of this approach is the absence of a need to rebuild the entire model, in the case of modification of some subsystem. In this case, the model changes will also be localized by the corresponding submodel, which can greatly simplify the process of constructing the modified system model.

## References

[1] Romankevich, A., Maidaniuk, I., Feseniuk, A., Romankevich, V. Complexity Estimation of GL-models for Calculation FTMS Reliability // Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham – Vol. 938, P. 369-377

[2] Romankevich, A., Feseniuk, A., Maidaniuk, I., Romankevich, V. Fault-tolerant multiprocessor systems reliability estimation using statistical experiments with GL-models // Advances in Intelligent Systems and Computing.– 2019.– Vol. 754, P. 186-193

[3] Romankevich V., Potapova K., Hedayatollah Bakhtari The some properties of model's of k-out-of-n system's behavior in the stream of faults // Proceedings of IEEE East-West Design & Test Symposium.- Armenia, Yerevan, September 2007.- p.763.

[4] Romankevych A., Romankevych V., Kononova A., Rabah Al Shboul GL-models of K(2,N) FTMpS // Proceeding of East-West Design & Test Workshop (EWDTW'05).– Kharkov National University of Radioelectronics.– Kharkov, 2005.– pp. 88-91

[5] Kuo W., Zuo M.J. Optimal Reliability Modeling: Principles and Applications. 1st ed. Hoboken, John Wiley & Sons, 2003, 544 p.

[6] Романкевич А.М., Романкевич В.А., Морозов К.В. Об одной GL-модели системы со скользящим резервом // Радіоелектронні і комп'ютерні системи.–№5, 2013.– С.333-336

[7] Романкевич А.М., Иванов В.В., Романкевич В.А. Анализ отказоустойчивых многомодульных систем со сложным распределением отказов на основе циклических GL-моделей // Электронное моделирование.–№5, т.26, 2004.– С.67-81

[8] Романкевич В.А., Морозов К.В., Фесенюк А.П. Об одном методе модификации рёберных функций GL-моделей // Радіоелектронні і комп'ютерні системи.– №6, 2014.– С.95-99

[9] Romankevych Oleksiy, Grol Volodymyr, Romankevych Vitaliy, Potapova Kateryna The Models reflecting reaction of the fault-tolerant multiprocessor systems to faults appearance // Abstracts of Scientific Information for society – from Today to the Future (CODATA-21).– Ukraine, Kyiv.– October 5-8.– 2008.– p.76