MULTILEVEL ABSTRACT AUTOMATIC MACHINES OF MARAKHOVSKY

Marakhovsky L.F.
Doctor of Technical Sciences, Professor, Professor of the State University of Infrastructure and Technology, Corresponding Member of RAE, Academician of the Interdisciplinary Academy of Sciences of Ukraine

Abstract

The article affirms one of the characteristic fundamental properties of modern research in new directions of their development on the basis of automatic discrete time, which limits the possibility of the appearance of new transitions in elementary memory.

The article discusses the principle of hierarchical program management, which extends the possibilities of simultaneous processing of hierarchical information.

On the basis of the principle of hierarchical program management, multi-level abstract automata working in automatic continuous time and capable of processing and storing hierarchical information in parallel are considered

Keywords: program management principle, program storage principle in machine memory, automatic discrete time, hierarchical program management principle, multi-level abstract automata, automatic continuous time

INTRODUCTION

For the first time the principle of program management was proposed by the British scientist Charles Babbage in the construction of an analytical machine in the 30s of the XIX century. It consisted in the fact that information was divided into two types: processing (data) and control (program).

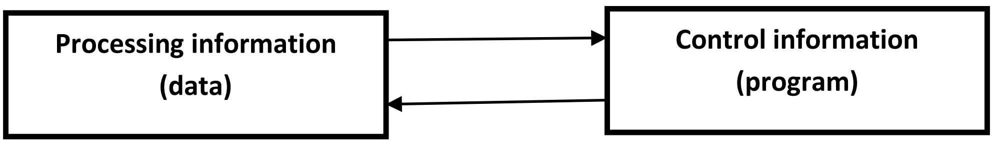

Fig. 1. The principle of program management

In 1945, the American scientist J. von Neumann supplemented it with the principle of a program stored in memory, thus completing the theoretical justification for the possibility of creating universal technical means for automating computational work [1].

It is important to note that the calculations made by the computer's processor are determined by the program's commands. It is the program commands that "configure" the computer to obtain the necessary results. Replacement of commands in the program can lead to changes in the functions of the computer. Consequently, the variety of subsets of commands and programs executed in the computer determines a class of functions that are capable of effectively implementing algorithms for a particular class of tasks.

At the present time, the capabilities of computers are growing [2]. They strive to find new capabilities of the element base for integrated circuits based on multifunctional memory circuits and analogs of a biological neuron [3-5]. They try to move away from the Neumann architecture for the purpose of compatibility of operations (pipeline processing of information), parallelization of algorithms (multiprocessor computers), development of interpretation systems (introduction of non-traditional means of addressing and operations over information), etc. [4].

One of the characteristic fundamental properties of these modern studies in new directions of their development is the preservation of automatic discrete time, which limits the possibility of the appearance of new transitions in elementary memory.

One of the proposals for expanding the possibilities for a qualitatively new basis for information processing was the principle of hierarchical program management, proposed by Prof. Marakhovsky LF in 1996 [6].

THE PRINCIPLE OF HIERARCHICAL PROGRAM MANAGEMENT

The principle of hierarchical program management is that the information processed and managed is divided into a local and a general, where the management information is vertically linked from the general information to the local information and processed simultaneously with it (in parallel).

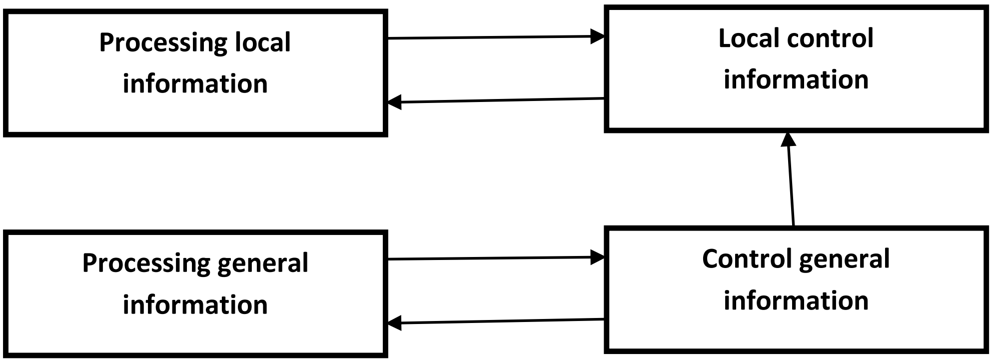

Fig. 2. Principle of hierarchical program management

One of the main temporal characteristics of processing hierarchical information in this case is faster processing of local information in relation to the general, and one of the functional characteristics is the change in the algorithm for processing the multifunctional local information, depending on the simultaneous change in the processing of general information.

General information can also be presented as local and general, increasing the number of levels of processing hierarchical information. Such a hierarchical separation of information is finite and possible up to a certain minimum amount of general information.

Local information can be processed unambiguously, probably or unclearly, in the hierarchical program management principle. General ("root") information should be processed unambiguously and determine the mode of processing of local information.

To process the hierarchical memory, we had to abandon the limitations of the automatic discrete time and proceed to automatic continuous time, which allowed us to expand the range of transitions in multi-level automata.

MULTILEVEL ABSTRACT AUTOMATA

Open multifunctional automata of Marakhovsky 1, 2 and 3, having two sets of input alphabets: X - information input alphabet and E - preserving the input alphabet, can be tuned by another open multifunction machine through a set of letters $e_j(e_j \in E)$ preserving the input alphabet to function in a certain block (subset) of its states. In this case, a multilevel (hierarchical) structure of the F-automaton from open multifunctional subautomata S_i (i = 1, 2, ..., n) is formed. The organization of such a combined structure of the hierarchical automaton (IA) is shown in Fig. 3.

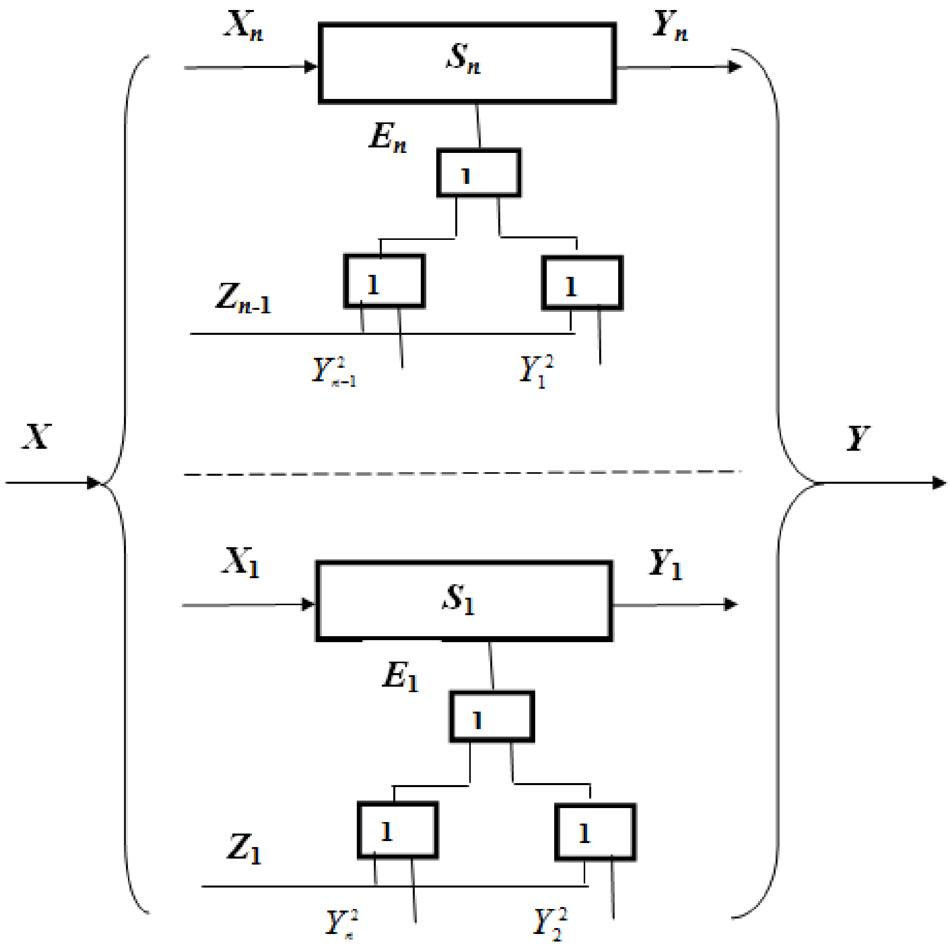

Fig. 3. Hierarchical abstract F_i-automaton

Each multifunctional subautomaton S_i (i = 1, 2, ..., n) of a hierarchical abstract F-automaton can pass from one state to another in parallel with other subautomata of IA (Figure 3). The functioning of the S_i subautomata in a particular subset of states can be changed by the effects of the input letters of the information alphabet X at the clock time t and the effects of the results of the operation of other subautomata S_i IA by the letters of the preserving input alphabet E at the moments of the internal cycle Δ of the automatic continuous time T.

Let us define a multi-level (hierarchical) abstract F-automaton.

Definition. The mathematical model of a hierarchical discrete device with a multifunctional system of memory organization is an abstract hierarchical *F*-automaton, defined as an *N*-component vector

$$F = (S_1, S_2, \ldots, S_N), \qquad (1)$$

whose S_i components are given by the sixteenth component vector

$$S_i = (X_i, E_i, Z_i, Y_i^1, Y_i^{II}, Y_i^{III}, Q_i, \pi_i, e_{i_0}, a_{i_0}, \delta_{i_0}, \delta_{i_e}, \delta_{i_y}, \lambda_{i_1}, \lambda_{i_2}, \lambda_{i_3}) \qquad (2)$$

which one

- X_i - a set of information input signals;
- E_i - the set of input signals that save;
- Z_i - the set of resolving input signals;
- Y_i^I - a set of output signals of type 1;
- Y_i^{II} - a set of output signals of type 2;
- Y_i^{III} - a set of output signals of type 3;
- Q_i is an arbitrary set of states;
- π_i - the set of blocks of states of the subautomata S_i;
- e_{i_0} - initial storing the input signal;
- a_{i_0} – is the initial state of the Si subautomaton;
- $\delta_{i_0} : Q_i \times X_i \to Q_i$ is a single-valued transition function;
- $\delta_{i_e} : Q_i \times e_{i_j} \to \pi_{i_j}$ - state blocks saving function;
- $\delta_{i_y} : Q_i \times E_i \to \pi_{i_j}$ - the function of the enlarged transition;
- $\lambda_{i_1} : Q_i \times X_i \to Y_i^I$ - function of outputs type 1;
- $\lambda_{i_2} : \pi_{i_j} \to Y_i^{II}$ – - function of outputs type 2;
- $\lambda_{i_3} : Q_i \times E_i \to Y_i^{III}$ – - output function type 3

and defined functionally, as well as the components of the S_i structure, by a sixteenth component vector

$$F_A = (X, E, Z, Y^{I}, Y^{II}, Y^{III}, Q, \pi, E_0, Q_0, F_1, F_2, F_3, \lambda_1, \lambda_2, \lambda_3) \quad (3)$$

which one

- $X=\{X_1, X_2, \ldots, X_N\}$ – the set of information input signals;
- $E=\{E_1, E_2, \ldots, E_N\}$– is the set of input signals that conserve;
- $Z_i=\{Z_1, Z_2, \ldots, Z_N\}$– the set of resolving input signals;
- $Y^{I}=\{Y_1^{I}, Y_2^{I}, \ldots, Y_N^{I}\}$– set of output signals of type 1;
- $Y^{II}=\{Y_1^{II}, Y_2^{II}, \ldots, Y_N^{II}\}$– set of output signals of type 2;
- $Y^{III}=\{Y_1^{III}, Y_2^{III}, \ldots, Y_N^{III}\}$– set of output signals of type 3;
- $Q=\{Q_1, Q_2, \ldots, Q_N\}$– is an arbitrary set of states;
- $\pi=\{\pi_1, \pi_2, \ldots, \pi_N\}$– the set of blocks of states of the subautomata S_i;
- $E_0=\{e_{1_0}, e_{2_0}, \ldots, e_{N_0}\}$ – initial saving the input signal;
- $a_0=\{a_{1_0}, a_{2_0}, \ldots, a_{N_0}\}$ – the initial state of the S_i subautomaton;
- F_1: $Q\times X \rightarrow Q$ – is a single-valued transition function realizing the mapping $D_{F1} \subseteq Q\times X$ to Q;
- F_2: $Q\times e_j \rightarrow \pi_j$ – is the conservation function of the state blocks, realizing the map $D_{F2} \subseteq Q\times e_j$ to π_j;
- F_3: $Q\times E \rightarrow \pi_j$ – is the coarse transition function realizing the mapping $D_{F3} \subseteq Q\times E$ to π_j;
- λ_1: $Q\times X \rightarrow Y^{I}$ – is a function of outputs of type 1 that implements the mapping $D_{\lambda 1} \subseteq Q\times X$ on Y^{I};
- λ_2: $\pi_j \rightarrow Y^{II}$ – is a function of outputs of type 2, realizing the map $D_{\lambda 2} \subseteq \pi_j$ on Y^{II};
- λ_3: $Q\times E \rightarrow Y^{III}$ – is a function of outputs of type 3 that implements the mapping $D_{\lambda 3} \subseteq Q\times E$ on Y^{III}.

Subautomata S_i, realizing their automaton memory on open-type registers, function in automatic continuous time T.

At the initial clock moment t_0, all the S_i subautomata are set to the initial state a_0 by the corresponding input signal $N_{i_0}(t_0) \in X_i$. During the subsequent internal clock cycle Δ_0, the initial state a_0 is preserved under the influence of the initial input e_{i_0} (Δ_0) input signals. The union of the states a_i of the S_i subautomata determines the states a_i of the hierarchical automaton at a given clock time t_i or Δ_i of the automatic continuous time T_i.

In the hierarchical automaton A, at the time t_i, all or only some S_i subautomata can perform single-pass functions δ_{i_0} , realizing the common function of single-valued transitions F_1 of the hierarchical automaton A to the new state $a_s(t) = \bigcup_i a_{i_j}(t)$. During the internal cycle Δ_i, the S_i subautomata can perform the coarse transitions δ_{i_y} , realizing the common transition function F_3 of the hierarchical automaton A to a new state $a_k(\Delta) = \bigcup_i a_{i_j}(\Delta)$.

If the S_i subautomata do not perform transitions to a new state during the entire outer cycle T, then, consequently, they realize the state conservation function δ_{ie}, realizing in IA A the joint function of conservation of states F_2.

Each of the S_i subautomata operates in a certain state block π_{i_j} of the whole set of its states Q_i. The blocks π_{i_j} of states of S_i subautomata form a definite block π_i of the set of state blocks π_{i_j} in IA A, in which the IA A is functioning at a given time.

A characteristic feature of IA is the possibility of interaction of S_i subautomata not only during the clock t_i, but also during the internal cycle Δ_i of the automatic continuous time T_i. The memory of the S_i subautomata is a matrix structure in which, during the time t_i, under the influence of the information signals $x_i(t)$ of the input signals, the S_i sub-automaton is able to go from one state to another in one state block π_{i_j} (the matrix structure line of the automatic memory circuit). And during the internal cycle Δ_i, under the influence of the input signals retaining $e_j(\Delta)$, the S_i sub-automaton is able to go from one state to another in one block μ_i (the column of the matrix structure of the automatic memory circuit), that is, from a certain state of one block π_{i_j} to a certain state of another block π_{i_j} states.

Thus, the mathematical model of a multilevel hierarchical automaton A is able to describe the functioning of not only parallel S_i, but also their inter-layer interaction by using the letters of the input preserving E alphabet of the S_i subautomata.

CONCLUSION

The article discusses the principle of hierarchical program management, which allows you to present multi-level control information with a vertical relationship between levels and the ability to process this information simultaneously.

Management of multi-level abstract automata is described on the basis of the principle of hierarchical program control.

LITERATURE

1. Справочник по цифровой вычислительной технике: (процессоры и память) / Б.Н.Малиновский, Е.И.Брюхович, Е.Л.Денисенко и др. / Под ред. Б.Н.Малиновского. – К.: «Техніка», 1979. – 366 с.
2. Татур В.Ю., 15 лет проекту «СКИФ»: история и итоги // «Академия Тринитаризма», М., Эл № 77-6567, публ.18063, 09.06.2013
3. A. S. Cassidy, P. Merolla, J. V. Arthur, S. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. Rubin, F. Akopyan, E. McQuinn, W. Risk, and D. S. Modha, “Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores,” in International Joint Conference on Neural Networks (IJCNN). IEEE, 2013.
4. Мараховский Л. Ф. Основы новой информационной технологии. Фундаментальные основы проектирования реконфигурируемых устройств компьютерных систем и искусственного нейрона: монография. Л. Ф. Мараховский, Н. Л. Михно – Germany: Saarbrcken, LAP LAMBERT, 2012. – 347 с.
5. Marachovsky L.F. Basic Concepts to Build the Next Generation of Reconfigurable Computing Systems – International Journal Of Applied And Fundamental Research. – 2013. – № 2 – URL: www.science-sd.com/455-24170 (20.11.2013). – 6 с.
6. Мараховский Л.Ф. Основы теории проектирования дискретных устройств. Логическое проектирование дискретных устройств на схемах автоматной памяти: монография. – Киев: КГЄУ, 1996.–128 с.