## MARAKHOVSKY L.F.

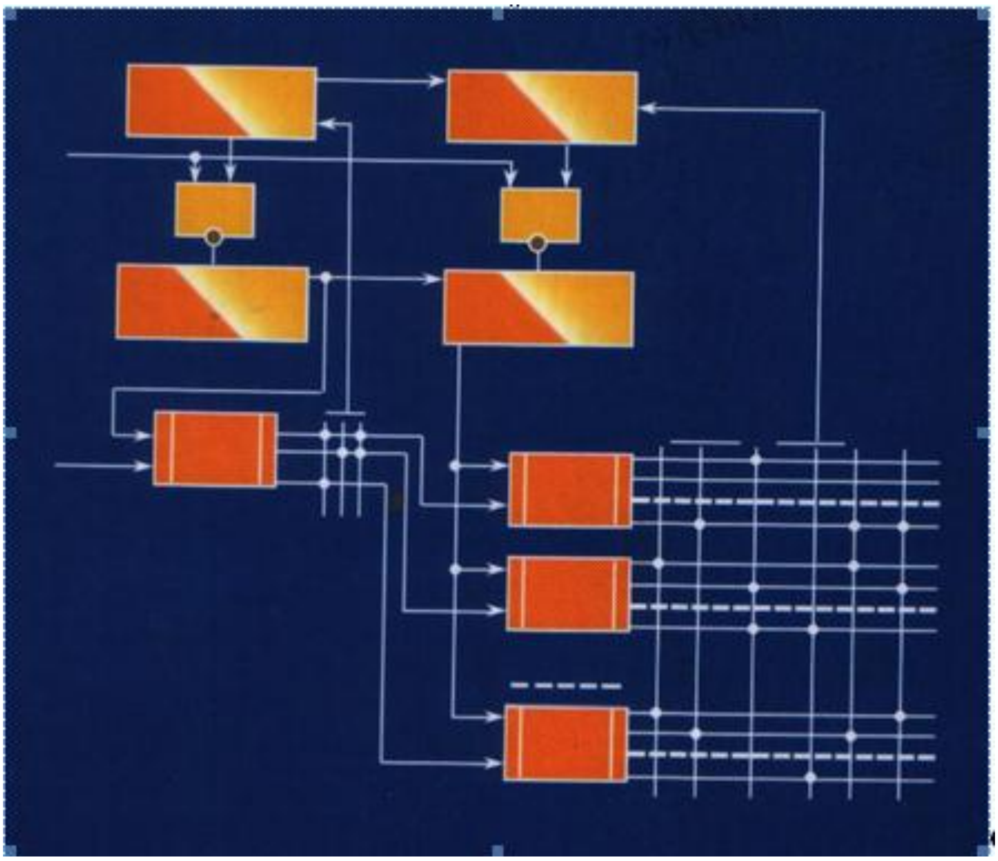
# EXPLANATORY DICTIONARY

# ON A NEW DIRECTION

# IN THE FIELD OF COMPUTING TECHNOLOGY

**(INFORMATION TECHNOLOGY PROCESSING**

**HIERARCHICAL INFORMATION)**



**Kyiv -2019**

УДК 004.258:252 + 519.716:713.1

**MARAKHOVSKY L.F.**
Explanatory dictionary on a new direction in the field of computing technology (Information technology processing hierarchical information). – K.: Interdisciplinary Academy of Sciences of Ukraine, 2019. - 163 p.

## Abstract

The dictionary provides a brief interpretation of the meaning of the concept of terms and words which are supported by quotations from the scientific literature on new interdisciplinary scientific direction. Terms and concepts discussed in the areas of: the theory of multi-functional and hierarchical automata theory, synthesis and analysis of circuits in automaton (multi-function and multi-level) memory, as well as the construction of the diagrams of typical machine-memory devices of computer technology and artificial neuron circuits and neural networks.

**Keywords:** the theory of multi-functional and hierarchical automata theory, synthesis and analysis of circuits in automaton memory, of typical machine-memory devices of computer technology, artificial neuron circuits and neural networks.

***Reviewers:***

**Verbitsky V.G.** Doctor of Technical Sciences, Professor of the Department of Microelectronics, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»
**Gavrilenko V.V.** Head of the Department of Information systems and technologies of the National Transport University, Doctor Physics and Mathematics, Professor
**Kozubtsev I.N.** Leading researcher of the research Department cyber security and information systems, Research center of communication and Informatization, Honored worker of science and education of RAE, the candidate of technical Sciences, Doctor of Science, Honoris Causa, Professor of RAE

## CONTENTS

## LIST OF ACRONYMS

**CT** - computer technology;

**AI** - artificial intelligence;

**SMP** – multi-stable memory scheme;

**MFIS** - multifunctional memory circuits;

**MUSP** - multi-level memory schemes;

**EA** - is an elementary automaton;

$\boldsymbol{x(t)}$ - setting the input signal;

$\boldsymbol{e(\Delta)}$ - preserving the input signal;

$\boldsymbol{p}(\boldsymbol{T})$ - is the input word;

$\boldsymbol{M}$ - is the number of stable memorized states of the memory scheme;

$\boldsymbol{r_x}$ - is the number of $x(t)$ input signals in the memory scheme;

$\boldsymbol{r_e}$ - the number of $e(\Delta)$ input signals, preserving the state in the memory scheme;

$\boldsymbol{F_p}$ - limiting working frequency of switching;

$\boldsymbol{P_Q}$ - load capacity on outputs;

$\boldsymbol{S_{cv}}$ - number of internal connections;

$\boldsymbol{S_{vc}}$ - is the number of external connections;

$\boldsymbol{L}$ - is the number of elements on one state;

***IEA-*** is a multifunctional elementary automaton;

***BA*** - basic automaton;

$\boldsymbol{Y_1}$ $(t)$ - is the output signal of an automaton of the first kind;

$\boldsymbol{Y_2}$ $(T)$ - is the output signal of an automaton of the second kind;

$\boldsymbol{Y_3}$ $(\Delta)$ - is the output signal of the automaton of the third kind;

$\boldsymbol{p_0}$ $(T)$ - is a single-valued input word;

$\boldsymbol{p_y}$ $(T)$ - is an enlarged input word;

***VLSI*** - (Very-large-scale integration);

***CIN*** - is a digital artificial neuron.

***Intellect is eliminating unnecessary options*.**

"Modern computer. Translation from English

magazine "In the world of science"

(Scientific American) "

## INTRODUCTION

Dictionary in a new direction in the field of computer science is a guide and reference material for students, teachers, researchers and developers in computer engineering. This determines the selection of concepts of terms and words. In the dictionary you can find answers to questions about the meaning of unfamiliar terms and words that reflect interdisciplinary scientific direction in the field of computer technology.

The dictionary includes the most common concepts of terms and words that reveal the contents of this direction. After each entered explanations of concepts and words provides a link to its source. Some explanations and formulas are entered in the drawings.

The dictionary consists of the following sections: automata theory, schema auto-maten-memory model of the device circuits automatic memory, firmware and software, artificial neuron and neural network.

The dictionary is the first experience making this type. The material of the dictionary is limited in its focus. The author is grateful for comments and suggestions for improving the dictionary.

## *THEORY AUTOMATA*

### ABSTRACT.

Abstract. Abstract concept. Abstract thinking [1].

### GENERAL THEORY AUTOMATA.

It is divided into two parts with the names: the abstract and structural theory of automata [2].

### ABSTRACT THEORY AUTOMATA.

It studies only those transitions that the automaton undergoes under the influence of input signals, and those output signals that it gives in this case [2].

### STRUCTURAL THEORY AUTOMATA.

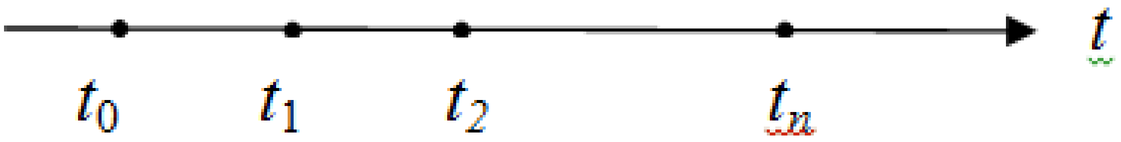
It studies methods of constructing automata from elementary automata, methods for encoding input and output signals by elementary signals transmitted over real input and output channels, etc. [2].

### DETERMINED.

Certain. An Unequivocal [1].

### AUTOMATIC DISCRETE TIME.

Accepts nonnegative integer values $t = 0, 1, 2, \dots n \dots$ [2].



### INPUT SIGNAL OF THE DETERMINED AUTOMATA.

The input signal $x(t)$ is an arbitrary letter of the input alphabet X [2].

## OUTPUT SIGNAL OF THE DETERMINED AUTOMATA.

The output signal $y(t)$ is an arbitrary letter of the output alphabet Y [2].

## THE LAW OF THE FUNCTIONING OF THE AUTOMATA OF THE 1-st KIND (Mealy).

Set by equations [2]:

$a(\mathrm{t}) = \delta(a\,(\mathrm{t} - 1), x(\mathrm{t}))$,

$y(\mathrm{t}) = \lambda(a\,(\mathrm{t} - 1), x(\mathrm{t}))$, $(\mathrm{t} = 1, 2, ...)$.

## THE LAW OF FUNCTIONING OF THE AUTOMATA OF THE 2-nd KIND (Moore).

Set by equations [2]:

$a(\mathrm{t}) = \delta(a(\mathrm{t} - 1), x(\mathrm{t}))$,

$y(\mathrm{t}) = \lambda(a(\mathrm{t}))$, $(\mathrm{t} = 1, 2, ...)$.

## EMPTY WORD OF ZERO LENGTH.

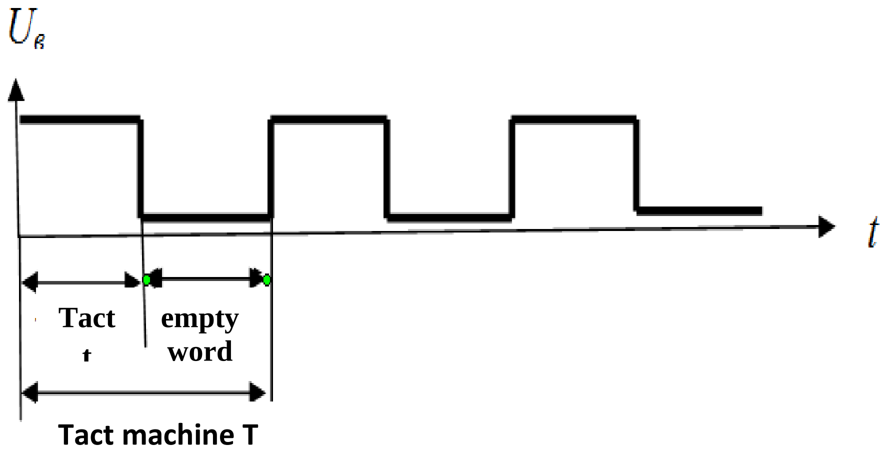
The empty word is the input signal $e(\Delta)$, which is not capable of performing transitions from one state to another in triggers [3]. The input signal $e(\Delta)$ is called the zero length because in automaton discrete time this signal was not allocated a length. This situation was justified due to the fact that the set of states in all triggers was preserved with a single input signal $e(\Delta)$, which was not considered in automatic time. With simultaneous exposure to the input signals $x(t)$ and $e(t)$, the signal $e(\mathrm{t})$ is absorbed by the signal $x(t)$ [4]:

$x(t) \cup e(t) = x(t)$

In the matrix structure of the multi-functional memory circuit, the input signal $e\ (\Delta)$ is capable of transferring the automaton to a new state, which caused the introduction of automaton continuous time.

## AUTOMATIC CONTINUOUS TIME.

It takes nonnegative integer values $T$ = 0, 1, 2, ... $n$ ..., consisting of two consecutive time intervals $t$ and Δ ($T_i = t_i + \Delta_i$) [5].



## DEFINITIONS OF AUTOMATIC CONTINUOUS TIME [5].

***Definition 1***. A tack is the time interval during which an arbitrary clock signal $\tau_i$ can be applied to the machine.

***Definition 2***. The inner beat is the shorter open time interval between the occurrences of the clock signals $\tau_i$.

***Definition 3.*** An external beat is a smaller open right-hand time interval, which corresponds to the period of the synchronized signal $\tau_i$.

Tacts that are defined can be expressed by the following formula:

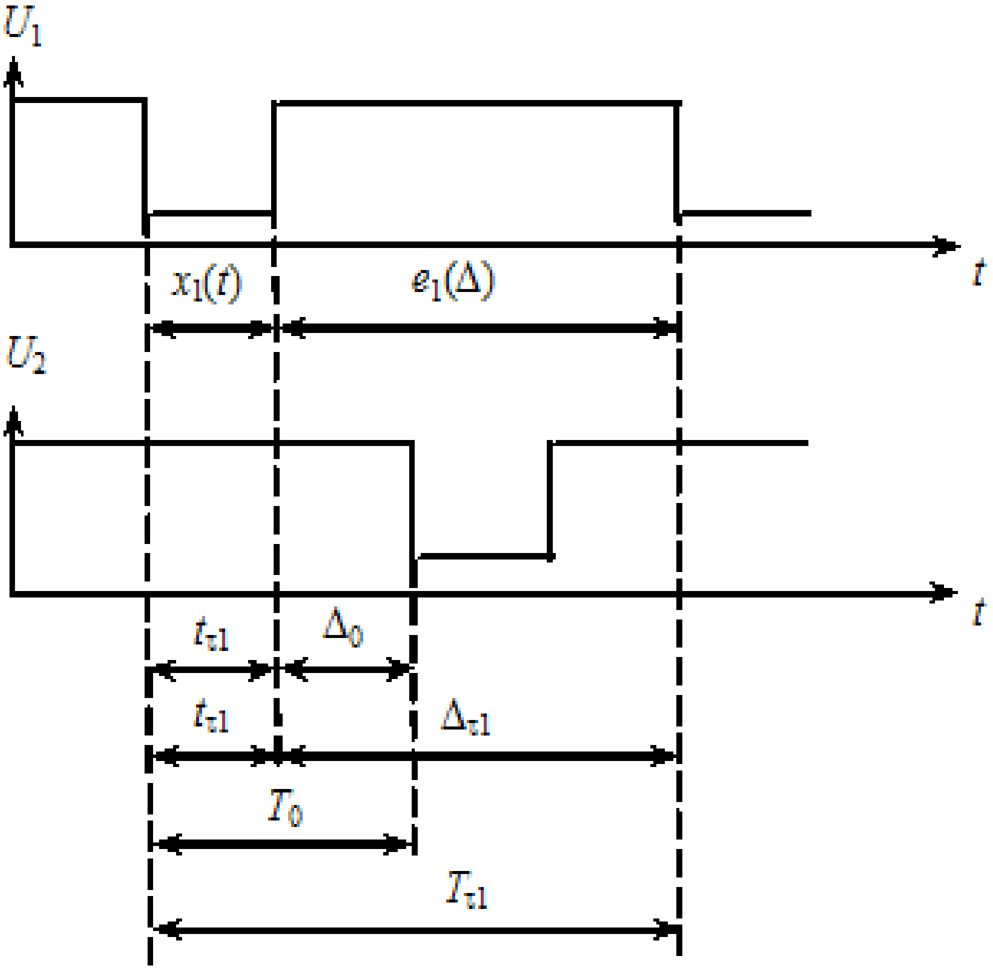
Information input signals x($t$) are given to the input channels of the complex automaton and are synchronized by the signal $\tau_j$. These input signals affect the automaton in the automaton continuous tact time.

***Definition 4.*** The inner unit cycle $\Delta_0$ of the automaton continuous time is called the shorter open time interval between the appearance of two arbitrary and consecutive synchronized signals $\tau_p$ and $\tau_{p+1}$.

***Definition 5***. The external unit clock cycle $T_0$ is the smaller open right time period, which corresponds to the time period between the appearance of two arbitrary and consecutive synchronizing signals $\tau_p$ and $\tau_{p+1}$.

The external unit clock cycle $T_0$, in accordance with time, is equal to the sum of the cycle time $t_{\tau_j}$ and the internal unit clock cycle $\Delta_0$. This dependence is considered by such equality:

$$T_0 = t_{\tau_j} + \Delta_0.$$



Timing ratios of sync and input signals

## INPUT WORD OF MULTIFUNCTIONAL AUTOMATA [6].

The input word *p*(*T*) consists of consecutive input signals x(*t*) and e(Δ), arbitrary letters of the corresponding input alphabets *X* and *E*.

## PARADIGM.

If possible, this is the minimum set of objects (units) from which we request such an object or unit that we want to give relevant meaning [7]. (from ancient Greek παράδειγμα, “example, model, sample” <παραδείκνυμι - “comparing”) in the philosophy of science means a combination of explicit and implicit (and often not realized) prerequisites that determine scientific research and are recognized at this stage of science development, as well as a universal method of making evolutionary decisions, an epistemological model of evolutionary activity [8].

## PARADIGM BASES OF INFORMATION ONE TECHNOLOGY ON AUTOMATIC MEMORY SCHEMES [4].

A new interdisciplinary direction combining the theory of automata, the theory of constructing memory circuits, the principles of analyzing basic memory circuits for their catastrophic failures, the methods of logical design of reconfigurable devices of computer equipment and the principles of constructing digital artificial neuron

**SYSTEM**.

Something whole, consisting of interconnected parts or elements [1]. This is a structurally organized object in which states, transitions, substructures and interactions of parts are distinguished [4].

**NOTION OF INFORMATION [9].**

Belongs to the number of fundamental concepts of modern science. The importance of this concept is due to its universality. Information is hierarchical ("information about information").

**ABSTRACT ALPHABET [3].**

This is any finite collection of objects (letters) of the alphabet. The nature of these objects in this case is not significant. It is only important that the considered alphabet is finite, that is, consists of a finite number of letters. In the abstract alphabet, whole words of a specific language can be considered as separate letters, if it is necessary for the convenience of converting information.

**LENGTH OF THE WORD [3].**

The number of letters in a particular word is usually called the length of the word. Along with words of positive length (consisting of at least one letter), the abstract alphabet also considers an empty word that does not contain a single letter, which is usually denoted by the Latin alphabet *e*.

## TRANSFORMATION OF INFORMATION [3].

From the abstract point of view, the transformation of information is the mapping of one class of phenomena into another class of phenomena in accordance with a specific law.

## ALPHABETIC DISPLAY (OPERATOR) [3].

$\varphi$ is any correspondence (function) that associates a word in a given alphabet with a word in the same or in some other fixed alphabet.

The first alphabet is called the input alphabet, and the second is the output alphabet of another operator. In case of coincidence of the input and output alphabet, it is said that the alphabetical operator is specified in the corresponding alphabet.

## ALPHABETIC OPERATOR [3].

This is a single-valued, generally speaking, partially defined mapping of a set of words in the input alphabet of an operator into a set of words in its output alphabet. The collection of all words on which the alphabetic operator is defined is called its scope.

## ENCODING MAPPING [3].

This is just coding.

## ALGORITHM [3].

Determines the sequence of actions of an object to achieve the goal. Algorithms can be unambiguous, probabilistic, fuzzy, self-improving, etc.

### SINGLE ALGORITHM [4].

An example would be a system of rules of addition of integers in an arbitrary positional number system.

### A PROBABILISTIC ALGORITHM [4].

An example can be a variety of games (craps, checkers, chess, etc.). With respect to the probabilistic algorithms are the 'best' stroke is not determined uniquely, and randomly, in accordance with some probabilistic criteria.

### FUZZY (BLURRED, VAGUE OR INDEFINITE) ALGORITHMS [11].

They allow the use of fuzzy instructions is widespread in various spheres of human activity.

### THE LEARNING (TRAINING) ALGORITHMS [3].

Usually defined in a special way organised hierarchical system of algorithms. In the simplest case, the system consists of two algorithms, one of which is a working (multi-function) and performs the processing of information (transformation of input words in a weekend) and the second supervising or training algorithm. The controlling algorithm affects the working of the algorithm, changing the mode of his work.

## RECURSION.

Under the recursion in a General sense to understand this way of organizing the system in which it is in certain moments of its development, as determined by its rules, may establish (cause to) change their own copies, interact with them and include them in its structure [3]. An example of recursion can be Fibonacci numbers, where the first two members is 1 and the others are calculated using the formula $a_i = a_{i-2} + a_{i-1}$. This formula is called recursive. The input for each subsequent step are the results of previous [10].

## NORMAL algorithms (ALGORITHMS A. A. Markov) [3].

Convert words in any finite alphabet A = ($a_1$, ..., $a_n$), words in the same alphabet, and, as a rule, the algorithm is determined by the strength of the only parts of words and sets, hence a partial mapping. V. M. Glushkov showed that any algorithm is equivalent to some normal algorithm. Therefore, without violating the generality, speaking of an arbitrary algorithm, having in mind only the normal algorithms.

## COMPUTERS [2; 12].

Computing machines (computers) are the hardware realization of finite state machines. They can be digital, analog, hybrid, and neural.

## A DEFINITE TYPE OF TRANSITION [4].

All existing machine algorithms in one way or another used only this transition in Mealy and Moore of automata — a deterministic transition per machine clock from one state to another, which depended on the previous state of the automaton $a(t_{i\text{-}1})$ and the input signal $x(t_i)$ at discrete time $t_i$. Only this transition is used in all devices with memory on triggers.

## THEOREM ON STRUCTURAL COMPLETENESS [2].

The theorem substantiates the functional requirements for the existing element base with memory on triggers and a functionally complete system of logic elements, with the help of which by means of a canonical method of structural synthesis to build successive automata Mealy, Moore and *C*-automata.

## LIMITATIONS OF ELEMENTAL BASE WITH MEMORY ON TRIGGERS [13].

Triggers are a closed, non-tunable structure, which makes it difficult to use them when building reconfigured devices. To build reconfigured devices, structures at the “automaton” level are used.

1. All devices with memory on triggers operate in automaton discrete time $t_i$ ($i$ = 1, 2. ..., $n$, ...).
2. All these devices are described by Mealy, Moore, and C-automata, which determine the sequential nature of the operation of the devices.
3. The transition in the memory circuits is only one-valued for one variable $x(t)$.
4. The principle of programmed control, proposed by Charles Babbage, allows only sequential processing of information.

## PRINCIPLE OF SOFTWARE MANAGEMENT [12].

This principle allows information to be divided into two types: processed and managing.  It served as a theoretical prerequisite for the automation of numerical calculations in the creation of universal technical means of computing automation.

## EXTENDED THEOREM OF STRUCTURAL COMPLETENESS [5].

The theorem says that any system of elementary automata that contains an elementary multifunctional automaton with a Marakhovsky memory, has a complete transition system, a complete output system and a system of state conservation functions, the number of which is at least two, and an arbitrary functionally complete system of logical elements (elementary automata without memory), is a structurally complete system.  There is a general constructive method, which in this case allows us to solve the problem of structural synthesis of arbitrary finite multifunctional automata of the 1st, 2nd and 3rd kind, as well as multilevel automata that store and process parallel hierarchical information.

## MULTIFUNCTIONAL AUTOMATS MEALY AND MOORE [14].

Multifunctional automats Mealy:

$a^{(i)}(t) = \delta^{(i)}[\, a^{(i)}(t\text{-}1), x^{(i)}(t), \cup_{\delta}^{(i)}(t)];$

$y^{(i)}(t) = \lambda^{(i)}[\, a^{(i)}(t\text{-}1), x^{(i)}(t), \cup_{\lambda}^{(i)}(t)];$

$(t = 1, 2, \ldots; i = 1, 2, \ldots, L)$

Moore's Multifunction automaton:

$a^{(i)}(t) = \delta^{(i)}[\, a^{(i)}(t\text{-}1), x^{(i)}(t), \cup_{\delta}^{(i)}(t)];$

$y^{(i)}(t) = \lambda^{(i)}[\, a^{(i)}(t),\ \cup_{\lambda}^{(i)}(t)];$

$(t = 1, 2, \ldots;\ i = 1, 2, \ldots, L)$

Multifunctional automats Mealy, Moore, and their combination C-automata realize memory on triggers and describe their functioning in automaton discrete time. In these triggers, the excitation and output functions are switched, as can be seen from their description.

## MULTIFUNCTIONAL AUTOMATS OF MARAKHOVSKY [4–6].

They gave an opportunity in the theory of algorithms and automata to expand with qualitatively new additional transitions in computer devices (hardware) and software (software). These automats are divided into multifunctional automats of the 1-st, 2-nd and 3-rd kind.

## INPUT WORD [4–6].

In Marakhovsky's automata, a new concept was given: instead of the input signal $x(t)$, the "input word $p(T)$" appeared, consisting of two successive input signals $p(T) = x(t), e(\Delta)$.

## THE LAW OF FUNCTIONING OF THE MULTIFUNCTIONAL AUTOMATA OF THE 1-st KIND [4–5].

$$\begin{cases} a(t) = \delta_0(a(\Delta-1), x(t)); \\ a(\Delta) = \delta_e(a(t), e(\Delta)); \\ y_L^1(t) = \lambda_1(a(\Delta-1), x(t)), \\ a(t), a(\Delta) \in \pi_j;\ i = 0, 1, 2, \ldots;\ \Delta = 0, 1, 2, \ldots. \end{cases}$$

## THE LAW OF FUNCTIONING THE MULTIFUNCTIONAL AUTOMATA OF THE 2-nd KIND [4–5].

$$\begin{cases} a(t) = \delta_0(a(\Delta-1), x(t)); \\ a(\Delta) = \delta_e(a(t), e(\Delta)); \\ y_L^2(T) = \lambda_2(a(t), a(\Delta)), \\ a(t), a(\Delta) \in \pi_j;\ i = 0, 1, 2, \ldots;\ \Delta = 0, 1, 2, \ldots. \end{cases}$$

## THE LAW OF FUNCTIONING THE MULTIFUNCTIONAL AUTOMATA OF THE 3-rd KIND [4–5; 15-18].

$$\begin{cases} a(t) = \delta_0(a(\Delta-1), x(t)); \\ a(\Delta) = \delta_y(a(t), e(\Delta)); \\ y_L^3(\Delta) = \lambda_3(a(\Delta), e(\Delta)), \\ a(t) \notin \pi_j, a(\Delta) \in \pi_j;\ i = 0, 1, 2, \ldots;\ \Delta = 0, 1, 2, \ldots. \end{cases}$$

## FUNCTIONS OF MULTIFUNCTIONAL AUTOMATA [4–5].

1. the function $\delta_0$ implements an unambiguous transition from the state $a(\Delta\text{-}1)$ under the action of the input signal x($t$) to the state $a(t)$;

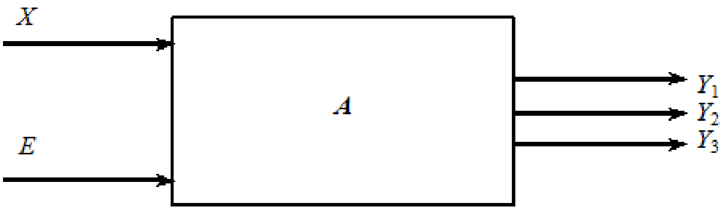
2. the function $\delta_e$ preserves the state $a(\Delta) = a(t)$ of one subset $\pi_j$ of states under the action of a certain input signal $e_i(\Delta)$;
3. the function $\delta_y$ performs an enlarged transition from the state $a(t)$ of the subset $\pi_j$ of the states to the state $a(\Delta)$ of the new subset $\pi_j$ of the states under the action of the input signal $e_i(\Delta)$. At the same time, the states $a(t) \neq a(\Delta)$.

## OUTPUT SIGNALS OF MULTI-FUNCTIONAL AUTOMATA OF MARAKHOVSKY [4–5].

The output signals in the 1st, 2nd, and 3rd kind of automata, respectively, differ from each other by the time of their appearance during one machine cycle $T$, thus, $y_L^1(t)$, $y_L^2(T)$, $y_L^3(\Delta)$, as indicated in the formulas for the assignment of the corresponding automata.

## THE MEANING OF THE CONCEPT OF ABSTRACT AUTOMATS OF MARAKHOVSKY [4–5].

It consists in implementing some mapping $\varphi$ of a sequential set of elementary $p$ words consisting of letters of the input information alphabet $X$ and letters of the input saving alphabet $E$ into the set of words of the output alphabet $Y$. Each sequence of input elementary words $p_i$ is associated with the corresponding sequence of the output word $q_i$, we get the mapping $\varphi_i$, which is called the mapping induced by the abstract automaton $M$, which is able to work as an automaton of the 1st, 2nd and 3rd kind.



Structural scheme abstract the *M*- automaton *A*

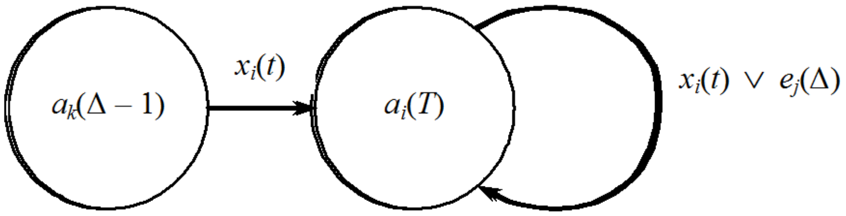
The operation of the *M*-automaton is investigated in automaton continuous time *T*.

## THEORY OF PROBABLE AUTOMATA [5].

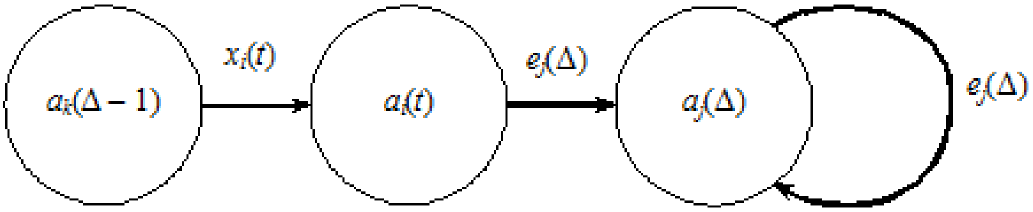
The meaning of the theory of probabilistic automata is to obtain a positive meaning of randomness in discrete data converters. A probabilistic automaton essentially appears only when probabilities are not postulated as computable.

## GRAPH SCHEMES OF TRANSITIONS IN MARAKHOVSKY AUTOMATA [4–5; 15].

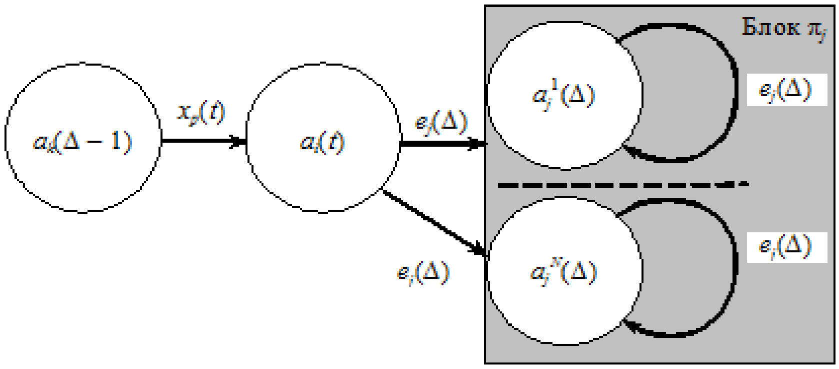
1. Unambiguous transition:



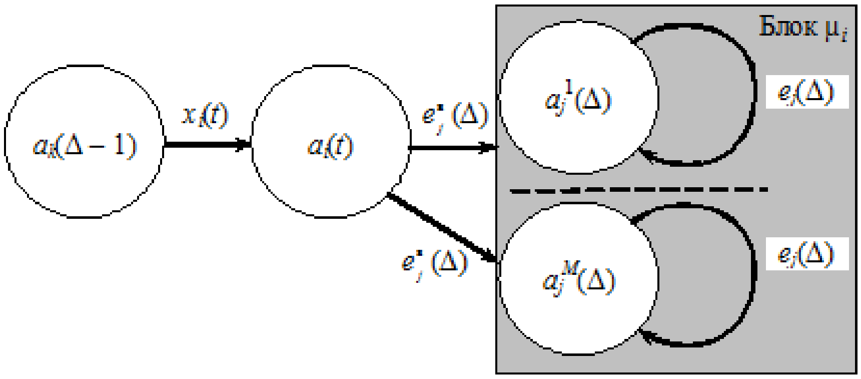
2. Enlarged transition



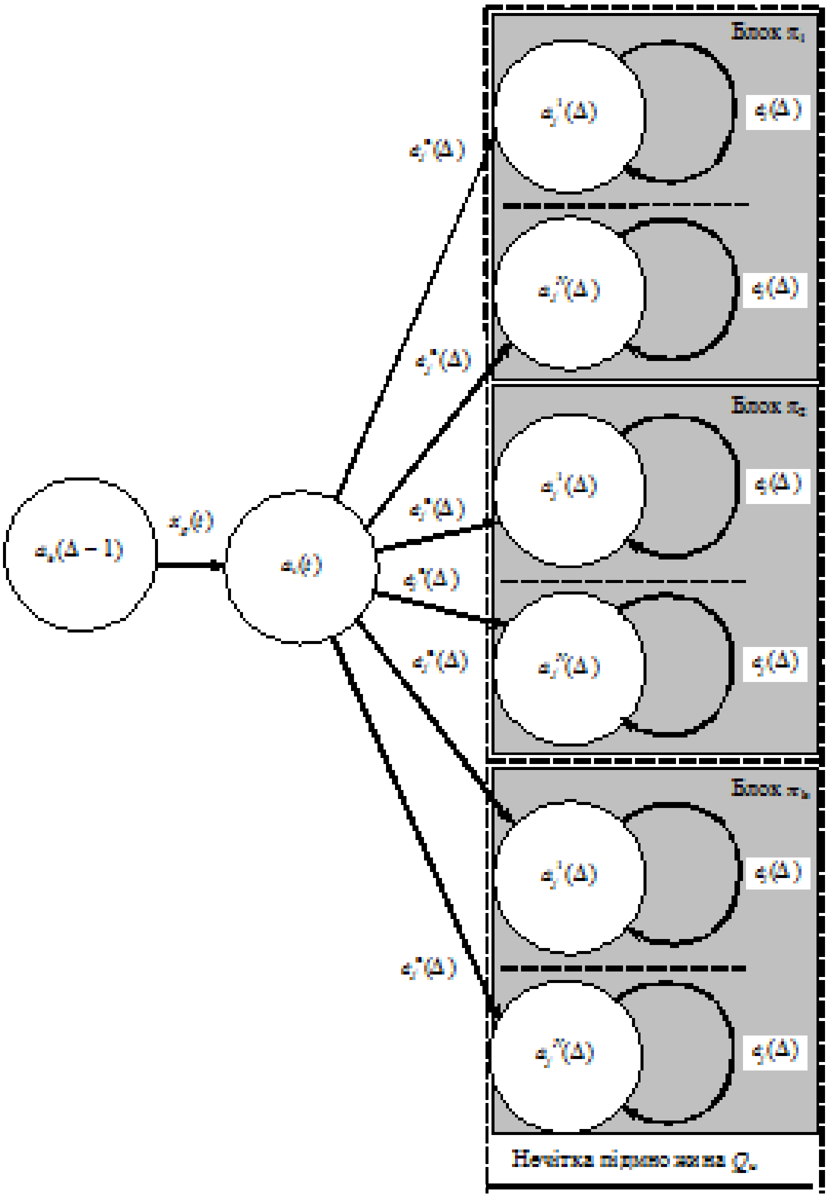
3. Probability transition type 1



4. Probability transition type 2



5. Fuzzy transition to the fuzzy region $Q_n$ of the states of the Marakhovsky automaton


$a_k(\Delta-1)$
$x_p(t)$
$a_i(t)$
Блок $\pi_1$
$a_j^1(\Delta)$
$e_j(\Delta)$
$e_j^*(\Delta)$
$a_j^{n_j}(\Delta)$
$e_j(\Delta)$
$e_j^*(\Delta)$
Блок $\pi_2$
$a_j^1(\Delta)$
$e_j(\Delta)$
$e_j^*(\Delta)$
$e_j^*(\Delta)$
$a_j^{n_j}(\Delta)$
$e_j(\Delta)$
$e_j^*(\Delta)$
Блок $\pi_n$
$a_j^1(\Delta)$
$e_j(\Delta)$
$e_j^*(\Delta)$
$a_j^{n_j}(\Delta)$
$e_j(\Delta)$
Нечітка підмножина $Q_*$

## PROBABILITY TRANSITIONS IN AUTOMATA OF THE 3-rd KIND [15].

They occur under the influence of probabilistic elementary $p_{v1}(T)$ of the input words of automatic continuous time of two types:

$$p_{v1}(T) = x_p(t), e_j(\Delta),$$

where $x_p(t)$ is an informational input signal ($x_p \in X$), which uniquely sets the state of the memory of the machine, which is not saved at any saving $e_j(\Delta)$ input signal;

$$p_{\text{v}2}(T) = x_i(t),\ e_j^{в}(\Delta),$$

where $e_j^{в}(\Delta)$ is a probabilistic preserving input signal ($e_j^{в} \in E$). Probabilistic transitions in machines of the 3rd kind are capable of making transitions from one state to a state of a certain block of states with a certain measure of probability.

## THE LAW OF FUNCTIONING OF A PROBABLE ABSTRACT AUTOMATA OF THE 3-rd KIND OF FIRST TYPE [15].

$$\begin{cases} a_p(t) = \delta_0(a(\Delta - 1), x_p(t)); \\ a(\Delta) = \delta_{B1}(a_p(t), e(\Delta), P_e); \\ y_L^{B1}(\Delta) = \lambda_3(a(\Delta), e(\Delta)), \\ a(t) \notin \pi_j, a(\Delta) \in \pi_j; \\ i = 0, 1, 2, \ldots; \Delta = 0, 1, 2, \ldots, 0 < P_e \le 1. \end{cases}$$

The system of conditional probabilities $P_e$, defined for each pair $((a_p(t), e(\Delta))$ on the set $\pi_j \times Y_{III}^{B1}$ ($\pi_j \in Q$; $Y_{III}^{B1} \in Y_{III}^{B1} \in Y_{III}$,, where $Y_{III}$ is the output signal of an automaton of the 3rd kind) characterizes the probability of the automaton going to the state $a_s(\Delta)$ with issuing a signal $y_k(\Delta)$, which displays the state $a_s(\Delta)$ of the automaton, provided that the automaton was previously set to the state $a_p(t)$, which is not memorized, and at its inputs is fed an $\mathrm{e}_\mathrm{j}(\Delta)$ input signal capable of storing the whole set of states of the $\pi_j$ state block ($a_s \in \pi_j$).

## THE LAW OF FUNCTIONING OF A PROBABLE ABSTRACT AUTOMATA OF THE 3rd KIND OF THE SECOND TYPE [15].

$$\begin{cases} a(t) = \delta_0(a(\Delta-1), x(t)); \\ a(\Delta) = \delta_{B2}(a(t), e_j^{B2}(\Delta), P_0, P_e); \\ y_L^{B2}(\Delta) = \lambda_3^{\text{в}2}(a(\Delta), e(\Delta)), \\ a(t) \notin \pi_j, a(\Delta) \in \pi_j; \\ i = 0, 1, 2, \ldots; \Delta = 0, 1, 2, \ldots, 0 < P_0 \leq 1; 0 < P_e \leq 1. \end{cases}$$

The system of conditional probabilities $P_e$ specified for each pair $(a(t), e^{B2}{}_j(\Delta))$ on the set $\pi_j \times Y_{III}^{B2}$ ( $\pi_j \in Q$; $Y_{III}^{B2} \in Y_{III}$, where $Y_{III}$ is the output signal of the 3rd automaton) characterizes the probability of the automaton going to the $a_s(\Delta)$ state issuing a signal $y^{\text{в}2}(\Delta)$, which displays the state $a_s(\Delta)$ of the automaton, provided that the automaton was previously set to the state $a(t)$ and its inputs were supplied with a probabilistic $e^{B2}{}_j(\Delta)$ input signal capable of storing the entire set of block states $\pi_j$ states (($a_s \in \pi_j$).

## THEORY OF FUZZY AUTOMATA [15].

The theory of fuzzy subsets best makes it possible to structure everything that is divided by not very precise boundaries, for example, thought, language, and people's perceptions. The merit of Professor L.A. Zadeh is to introduce the concept of weighted membership, which gives the concept of a fuzzy (fuzzy) subset [8].

## FUZZY TRANSITIONS IN AUTOMATA 3-rd KIND [15] .

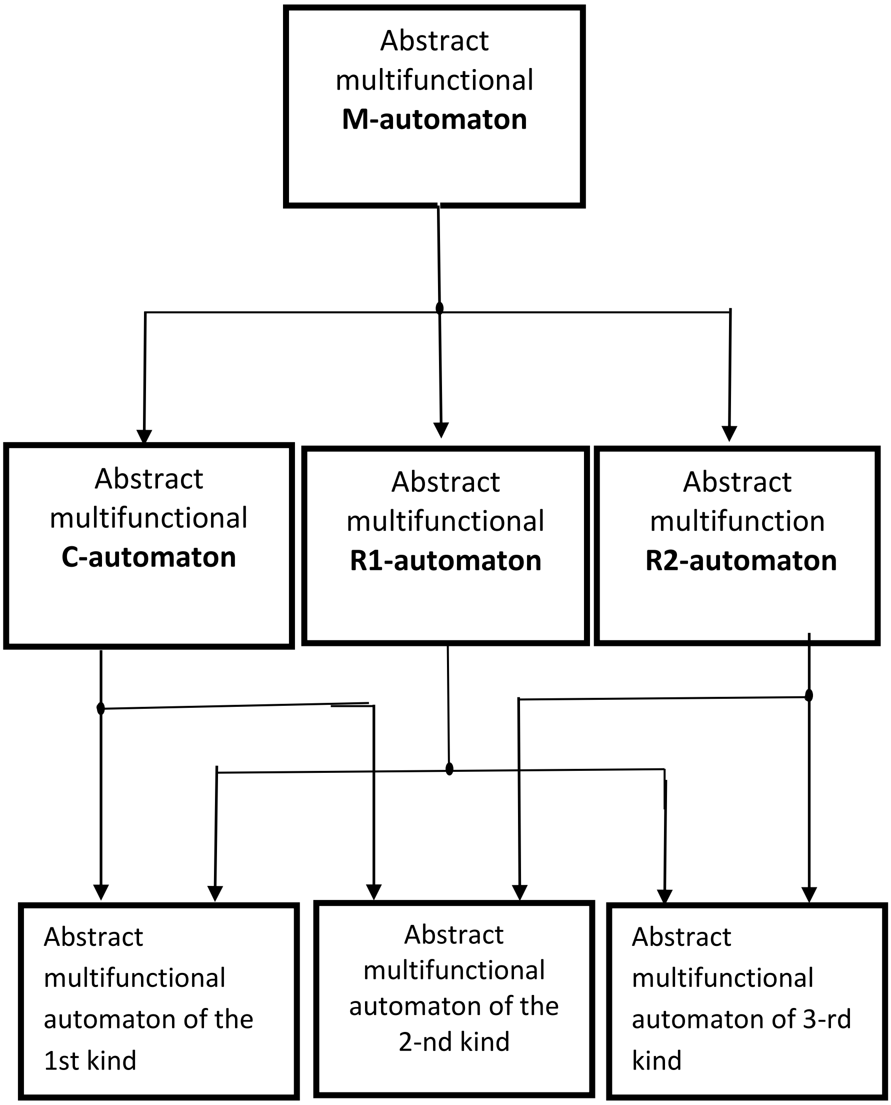
Abstract fuzzy automata of the 3rd kind are considered when incoming to their input nodes of elementary fuzzy $p_н(T)$ input words $(p_н(T) = x_p(t), e_в(\Delta))$ in automaton continuous time $T$.

## THE LAW OF FUNCTIONING OF THE FUZZY ABSTRACT AUTOMATA OF THE 3-rd KIND [15].

$$\begin{cases} a_p(t) = \delta_0(a(\Delta-1), x_p(t)); \\ a(\Delta) = \delta_н(a_p(t), e_j^н(\Delta), P_0, P_н); \\ y_L^н(\Delta) = \lambda_3^н(a(\Delta), e^н(\Delta)), \\ a_p(t) \notin \pi_j, a(\Delta) \in \pi_j; \\ i = 0, 1, 2, \ldots; \Delta = 0, 1, 2, \ldots, 0 < P_0 \le 1; 0 < P_н \le 1. \end{cases}$$

The system of conditional probabilities of $P_H$, set for each pair $(a_p(t), e^H(\Delta))$ on the set $Q_H \times Y_{III}^H$ ( $Q_H \in Q$; $Y_{III}^H \in Y_{iii}$ is $Y_{III}$ the output signal of the automaton of the 3rd kind) characterizes the probability of the automaton going to the state $a_s(\Delta)$ with issuing the signal $y_k(\Delta)$, which displays the state $a_s(\Delta)$ of the automaton, provided that the automaton was previously set to the $a_p(t)$ state and a fuzzy $e^H{}_j(\Delta)$ input signal capable of translating the automaton from the $a_p(t)$ states in $a(\Delta)$ state of the $\pi_j$ state block, which belongs to the fuzzy subset $Q_H$ of the state $(\pi_j \in Q_H)$.

## CLASSIFICATION OF DETERMINED AUTOMATA


Abstract multifunctional **M-automaton**
Abstract multifunctional **C-automaton**
Abstract multifunctional **R1-automaton**
Abstract multifunction **R2-automaton**
Abstract multifunctional automaton of the 1st kind
Abstract multifunctional automaton of the 2-nd kind
Abstract multifunctional automaton of 3-rd kind

Classification of abstract multifunctional deterministic automata on automatic memory circuits

**SYSTEMS [4].**

Marakhovsky automata are classified according to the types of output signals, as is the case in the classic Mealy and Moore automata. Thus, the combined abstract multifunctional M-automaton has the ability to generate three types (type 1, type 2 and type 3) of the output signals $y_1(t)$, $y_2(T)$ and $y_3(\Delta)$.The combination in the abstract multifunctional automaton of only two types (type 1 and type 2) of the output signals $y_1(t)$ and $y_2(T)$ is classified by the C-automaton, in the same way as this is done with the combined Mealy and Moore automata. The combination in the abstract multifunctional automaton of only two types (type 1 and type 3) of the output signals $y_1(t)$ and $y_3(\Delta)$ is called the multifunctional R1 automaton, and the combination in the abstract multifunctional automaton of only two types (type 2 and type 3) of the output signals $y_2(T)$ and $y_3(\Delta)$ is called a multifunctional R2 automaton. Abstract multifunctional automata using only one type of output signals $y_1(t)$, $y_2(T)$ or $y_3(\Delta)$ are called respectively abstract multifunctional machines of the 1st, 2nd, or 3rd kind.

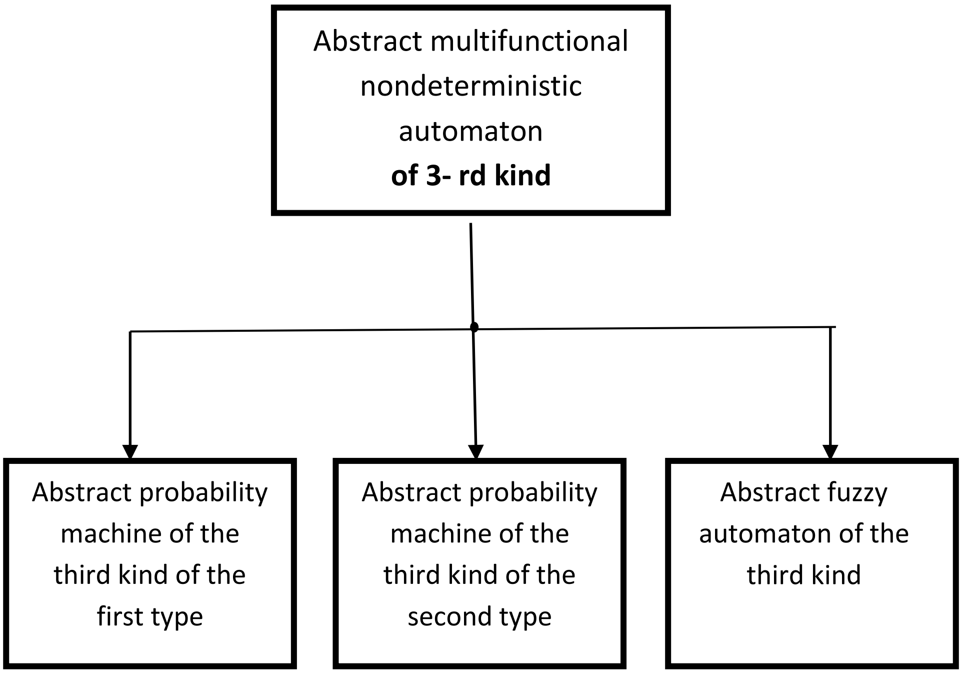
Note that abstract monofunctional machines Mealy (1st kind), Moore (2nd kind) and C-automaton are a special case of abstract multifunctional automata of 1st, 2nd kind and C-automaton.

Such a classification of automata is rather conditional, since the type of transition (deterministic, probabilistic, fuzzy), the type of memory used in the automaton (single-level, multi-level, hierarchical), etc.

## CLASSIFICATION OF ABSTRACT NONDETERMINISTIC AUTOMATA OF THE THIRD KIND [4].

An abstract nondeterministic automaton of the 3rd kind can be common when it uses elementary probabilistic input words of the first and second type, as well as elementary fuzzy input words.

The combined abstract non-deterministic automaton can be considered together with the deterministic abstract automaton of the 3rd kind. When this occurs, the use of the corresponding elementary input words: unambiguous, probabilistic first and second type or fuzzy.



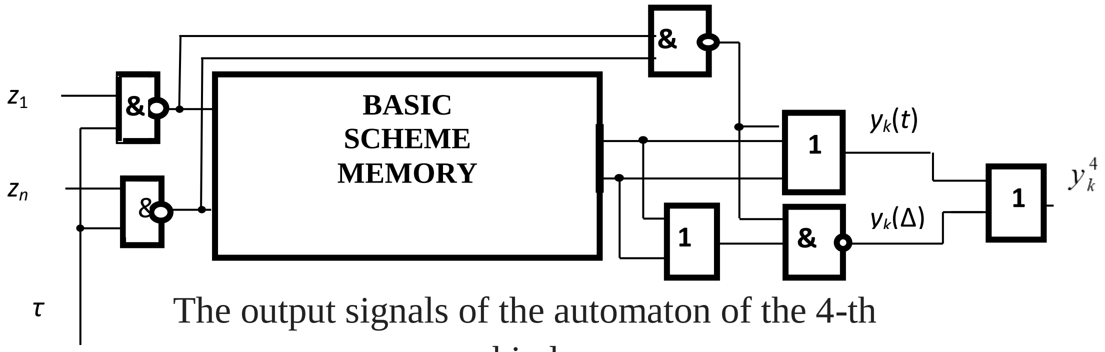
Abstract multifunctional nondeterministic automaton **of 3- rd kind**
Abstract probability machine of the third kind of the first type
Abstract probability machine of the third kind of the second type
Abstract fuzzy automaton of the third kind

Classification of abstract nondeterministic automata of the third kind

## CATASTROPHIC FAILURES IN ELEMENTARY AUTOMATA.

There are two types of catastrophic failures in elementary automata (memory circuits). The first type in the memory circuits on the OR-NOT (AND-NOT) logical elements occurs when an unchanged active output signal appears on at least one of the output nodes of the logical elements whose value is equal to the logical 1 (0). The second type in the basic memory circuits on the OR-NOT (AND-NOT) logical elements occurs when unchanged passive signals appear simultaneously on all the output nodes of the OR-NOT (AND-NOT) logical elements, the value of which is equal to logical 0 (1). The first type of catastrophic failures is the active constant output signal on the output node of the logic element, which along the circuits to the inputs of the logic elements of other groups of the memory circuit makes the memory circuit completely inoperative. The same memory circuit becomes inoperable with the second type of catastrophic failures, when the values are passive at all its output nodes of the basic memory circuit. Such a state of the memory circuit is not remembered with any saving $e(\Delta)$ input signal.

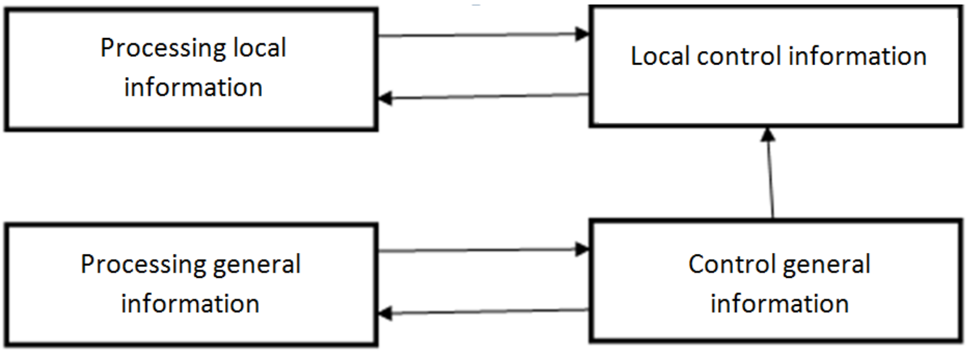
## CONTROLLING AUTOMATA 4-th KIND [4].



The output signals of the automaton of the 4-th kind

The $x_p(t)$ input signal is used to determine the output signals of catastrophic failures of the first type, when $y^4(t) = \lambda_4(x_p(t), a_p(t))$, and to determine the output signals of catastrophic failures of the second type, when the output signal of a probabilistic automaton of the 3rd kind is $y^{в1}(\Delta) = \lambda_3(a_p(\Delta), e(\Delta))$.  This is explained by the fact that in case of catastrophic failures of the first type in the memory circuit of the machine, when at least one logical element in the memory OR-NOT (AND- NOT) has a constant output signal equal to the logical 0 (1), then the output signal is $y^4(t)$ changes its value, which determines the failure of memory performance to the opposite value 1 (0). Catastrophic failures of the second type are also formed when there is no logical 1 on all output nodes of the circuit and when the input signal is saved on the input nodes.

## PRINCIPLE OF HIERARCHICAL PROGRAM MANAGEMENT [6;  16].


Processing local information
Local control information
Processing general information
Control general information

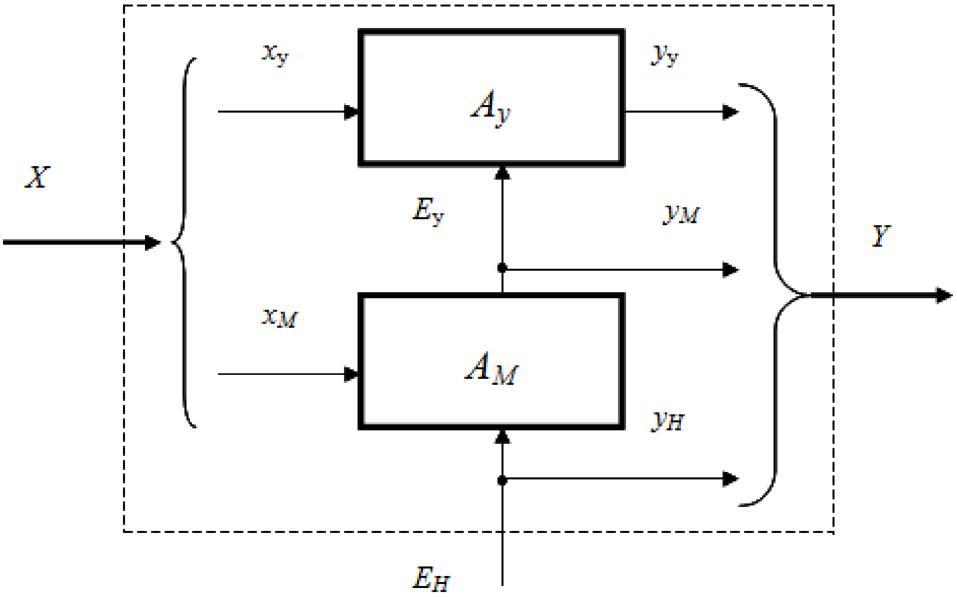
The principle of hierarchical program management

The principle of hierarchical program management, in contrast to the program management proposed by Charles Babbage, is that the information — which is processed, as well as the control — is divided into private and general, and the general control information selects the private control information

The interrelation of control information is carried out vertically from general control information to private information, and all levels of hierarchical information are processed in parallel with respect to each other.  The functional characteristic of this principle is that the change in the private information processing algorithm depends on the general information processing algorithm. Private information is processed unambiguously, probabilistically or indistinctly. Common, "root" information must be processed uniquely and determine the mode of processing of private information.

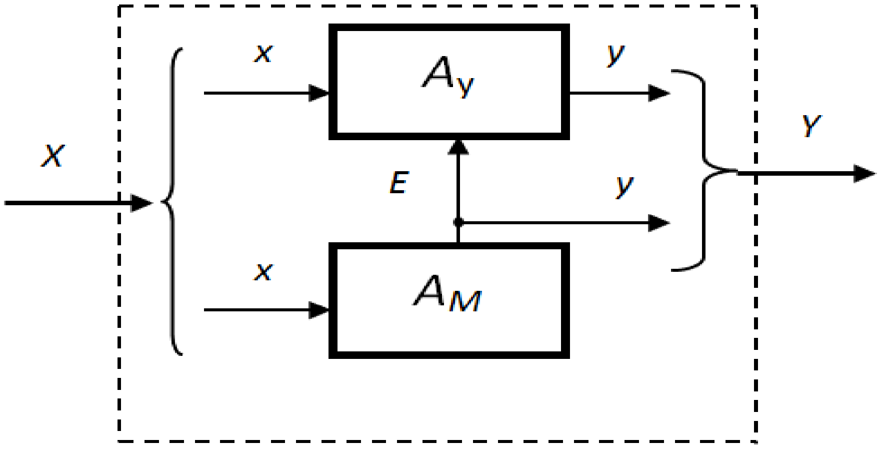
**OPEN STRUCTURE [4].**

The open multilevel memory device structure of an MUSP with a multi-functional organization system is called an automaton, which consists of two devices: a controlled MFIS $A_y$ and a multifunctional $A_M$ strategy, which have combined sets of states, input $X$ and output $Y$ alphabets and preserves the input alphabet $E$ of the $A_M$ strategy automaton .



**SEMI-CLOSED STRUCTURE [4].**

The semi-closed multilevel structure of a MUSP memory device with a multifunctional system of organization is called an automaton that consists of two devices: a managed MFIS $A_y$ and a multifunctional (or mono-functional) $A_M$ strategy that have combined sets of states, input $X$ and output $Y$ alphabets.



## HIERARCHICAL ABSTRACT AUTOMATON [4].

The hierarchical abstract automaton A with memory on MFIS is considered as a g-step (g > 1) device consisting of the registers $R_j$ (j = 1, 2, ..., g), which interact simultaneously, and two combinational circuits $KC1_i$, $KC2_i$ ($i$ = 1, 2, ..., $g$ ) at each step. At the g level (upper) hierarchical g-step automaton $A$ is able to function as a multifunctional automaton of the 1st, 2nd, as well as the 3rd kind, which has an (g – 1)-step $A_M$ strategy automaton used to save a certain block $\pi_i$ states of the corresponding automaton at the level $g$.

The canonical structure of a $g$-step abstract automaton of the 1st, 2nd, and 3rd kind on the MFIS simultaneously with the (g - 1)-level automaton of the strategy.

Unlike Mealy (1st kind) and Moore (2nd kind) automata with memory on triggers, whose operation is considered in automaton discrete time, Marakhovsky's automata on MFIS, whose operation is considered in automaton continuous time [5], are capable of different input signals $e_j(\Delta)$ memorize the state of certain blocks (subsets) $\pi_j$ ($\pi_j \in Q$). In accordance with this ability, in each of the $\pi_j$ ($\pi_j \in Q$) blocks, various mono-functional Mealy and Moor automata can be implemented, as well as using different g-level MFIS memory $\pi_j$ ($\pi_j \in Q$) memory blocks and implementing multifunctional 1st and 2nd automatic machines kind of.

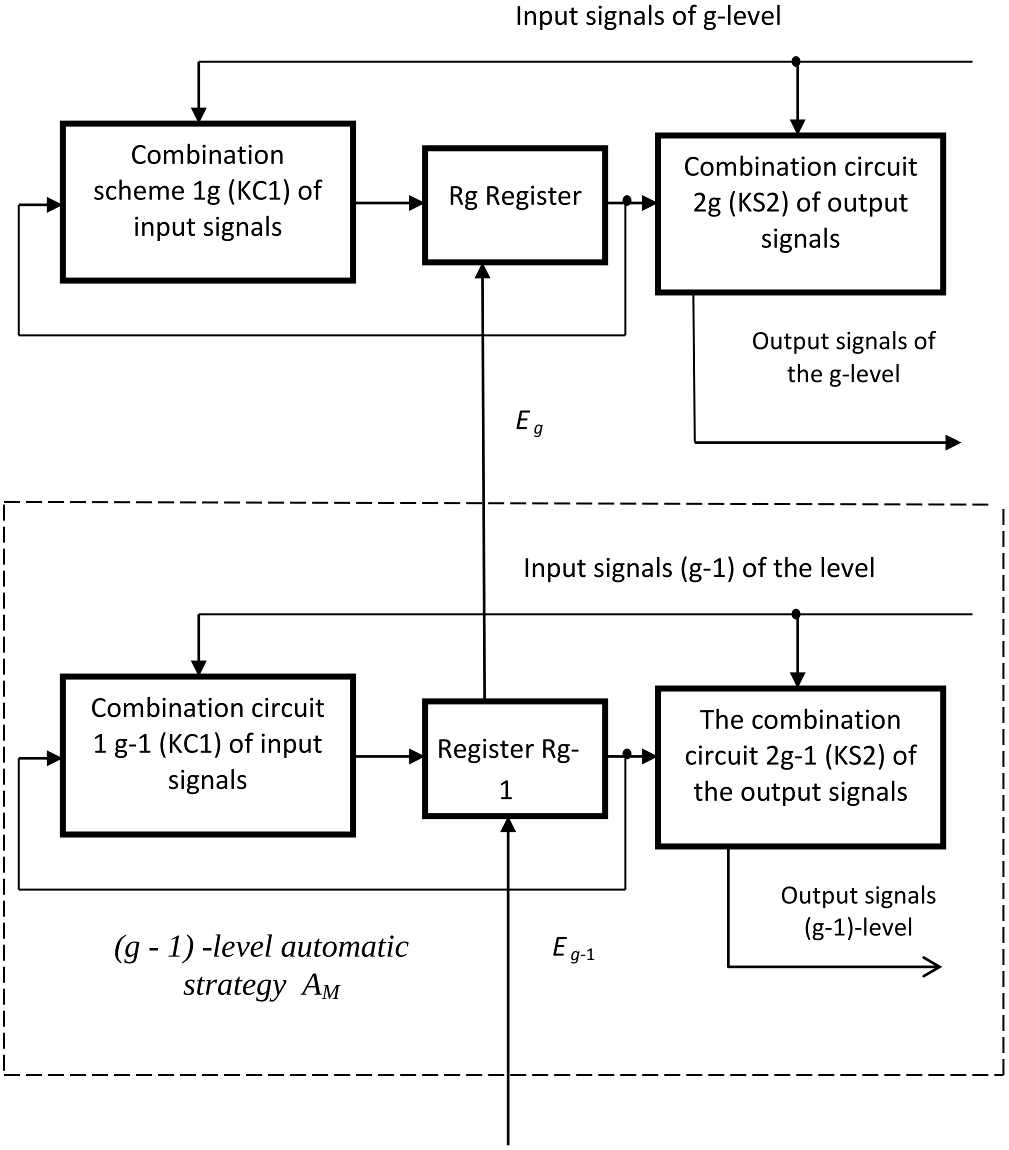
The $R_g$ register on the MFIS and combinational circuits (CS) are devices that are used to process private information of the upper *g*-level. Each such device can be described as an abstract automaton that processes the input information and which stores the $e_j(\Delta)$ input signals of which belong to the set $E_g$ ($e_j(\Delta) \in E_g$) come from the (g – 1)-level strategy automaton.

Thus, we come to the conclusion that the Mealy and Moore automata that use memory on triggers are a special case of *M*-automata that realize their memory on the MFIS.

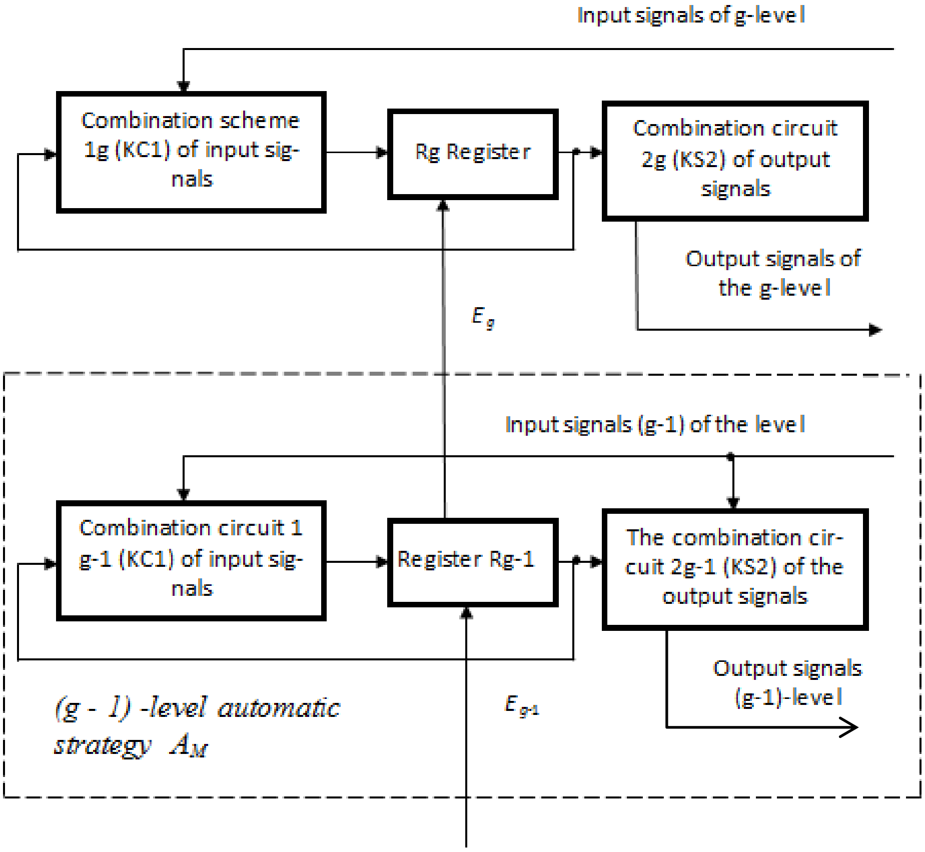
It is shown that automaton *M* g–level steps at the MFIS simultaneously with the (g – 1) level automata of the $A_M$ strategy have the unique ability to process local and general information in one cycle of machine time *T*.

It is shown that automata *M* of the g–level stage at MFIS have qualitatively new (as compared to automata on triggers) transitions from one state to another under the influence of $e_j(\Delta)$ input signals: enlarged, probabilistic and fuzzy.

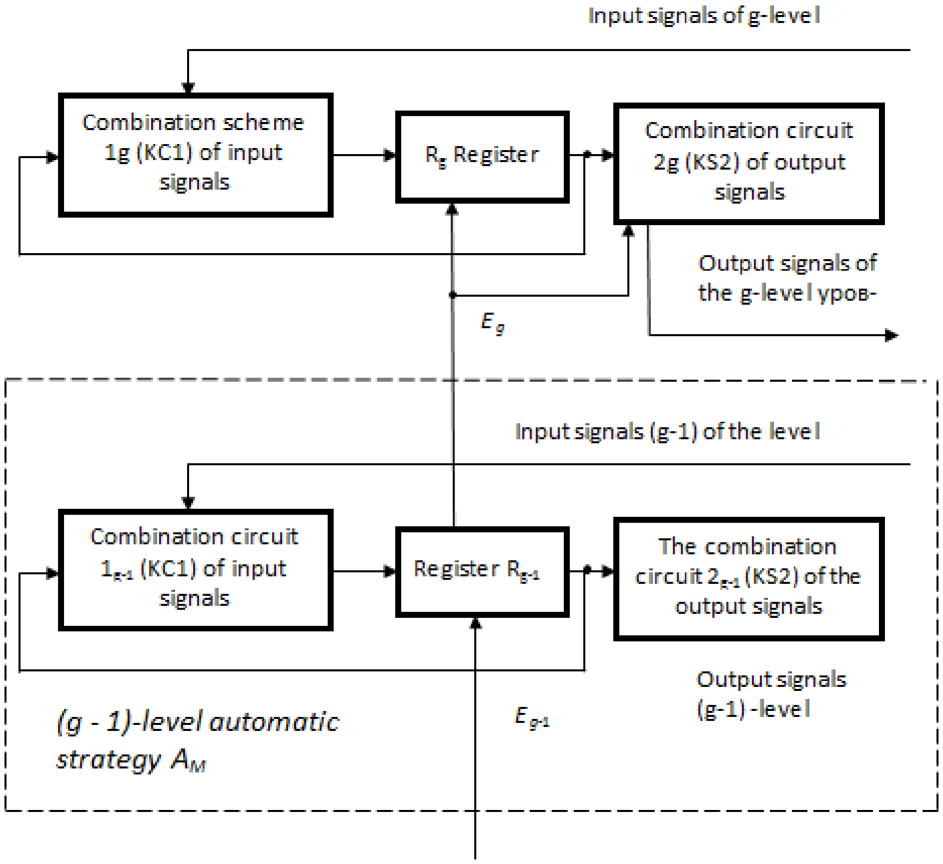
## STRUCTURAL SCHEME OF THE HIERARCHICAL AUTOMATA OF THE 1-st KIND (open structure) [4].


Input signals of g-level
Combination scheme 1g (KC1) of input signals
Rg Register
Combination circuit 2g (KS2) of output signals
Output signals of the g-level
E g
Input signals (g-1) of the level
Combination circuit 1 g-1 (KC1) of input signals
Register Rg-1
The combination circuit 2g-1 (KS2) of the output signals
Output signals (g-1)-level
(g - 1) -level automatic strategy A_M
E g-1

## STRUCTURAL SCHEME OF THE HIERARCHICAL AUTOMATA OF THE 2-nd KIND (open structure) [4].


Input signals of g-level
Combination scheme 1g (KC1) of input signals
Rg Register
Combination circuit 2g (KS2) of output signals
Output signals of the g-level
$E_g$
Input signals (g-1) of the level
Combination circuit 1 g-1 (KC1) of input signals
Register Rg-1
The combination circuit 2g-1 (KS2) of the output signals
Output signals (g-1)-level
(g - 1) -level automatic strategy $A_M$
$E_{g-1}$

## STRUCTURAL SCHEME OF THE HIERARCHICAL AUTOMATA OF THE 3-rd KIND (open structure) [4].


Input signals of g-level
Combination scheme 1g (KC1) of input signals
$R_g$ Register
Combination circuit 2g (KS2) of output signals
Output signals of the g-level уров-
$E_g$
Input signals (g-1) of the level
Combination circuit $1_{g-1}$ (KC1) of input signals
Register $R_{g-1}$
The combination circuit $2_{g-1}$ (KS2) of the output signals
Output signals (g-1) -level
(g - 1)-level automatic strategy $A_M$
$E_{g-1}$

## DEFINITION OF A MULTILEVEL (HIERARCHICAL) ABSTRACT FI-AUTOMATON [4].

***Definition*** A mathematical model of a hierarchical discrete device with a multi-functional memory organization system is an abstract hierarchical $F_i$ – automaton defined as an N-component vector

$F_u = (S_1, S_2, \dots, S_N)$,

whose $S_i$ components are given by a sixteen-component vector

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

which one

- $X_i$ is a set of information input signals;
- $E_i$ - a set of saving input signals;
- $Z_i$ - a set of allowing input signals;
- $Y_i^I$ – set of output signals of type 1;
- $Y_i^{II}$ – set of output signals of type 2;
- $Y_i^{III}$ – set of output signals of type 3;
- $Q_i$ – is an arbitrary set of states;
- $\pi_i$ – a set of state blocks $\pi_{i_j}$ of the $S_i$ sub-machine;
- $e_{i_0}$ – initial saving input signal;
- $a_{i_0}$ – the initial state of the $S_i$ sub-machine;
- $\delta_{i_0}$ : $Q_i \times X_i \longrightarrow Q_i$ – is a unique transition function;
- $\delta_{i_e}$ : $Q_i \times e_{i_j} \longrightarrow \pi_{i_j}$ – is a function of saving state blocks;
- $\delta_{i_y}$ : $Q_i \times E_i \longrightarrow \pi_{i_j}$ – function of the enlarged transition;
- $\lambda_{i_1} : Q_i \times X_i \rightarrow Y_i^I$ –function of type 1 outputs;
- $\lambda_{i_2}$ : $\pi_{i_j} \rightarrow Y_i^{II}$ – function of type 2 outputs;
- $\lambda_{i_3} : Q_i \times E_i \rightarrow Y_i^{III}$ – function of type 3 outputs

and functionally defined, like the components of the $S_i$ structure, by a sixteen-component vector

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

which one

- $X$={$X_1$, $X_2$, …, $X_N$} – is the set of information input signals;
- $E$ ={$E_1$, $E_2$, …, $E_N$} – a set of saving input signals;
- $Z_i$ ={$Z_1$, $Z_2$, …, $Z_N$} – is the set of enabling input signals;
- $Y^I=\{Y_1^I, Y_2^I, ..., Y_N^I\}$ – a set of output signals of type 1;
- $Y^{II}=\{Y_1^{II}, Y_2^{II}, ..., Y_N^{II}\}$ – a set of output signals of type 2;
- $Y^{III}=\{Y_1^{III}, Y_2^{III}, ..., Y_N^{III}\}$ – a set of output signals of type 3;
- $Q$ ={ $Q_1$, $Q_2$, …, $Q_N$}–is an arbitrary set of states;
- $\pi=\{\pi_1, \pi_2, \ldots, \pi_N\}$– many blocks $\pi$ of states of the $S_i$ sub-machine;
- $E_0=\{e_{1_0}, e_{2_0}, ..., e_{N_0}\}$ –initial saving input signal;
- $a_0=\{a_{1_0}, a_{2_0}, ..., a_{N_0}\}$ – is the initial state of the $S_i$ sub-machine;
- $F_1$: $Q\times X\rightarrow Q$ – is an unambiguous transition function that implements the mapping of $D_{F1} \subseteq Q\times X$ to $Q$;
- $F_2$: $Q\times e_j \rightarrow \pi_j$ – is a function of saving state blocks, realizing the mapping of

$D_{F2} \subseteq Q\times e_j$ to $\pi_j$;

- $F_3$: $Q\times E\rightarrow \pi_j$ – is an enlarged transition function that implements the mapping

$D_{F3} \subseteq Q\times E$ to $\pi_j$;

- $\lambda_1$: $Q\times X\rightarrow Y^I$ – is a type 1 output function that implements the mapping

$D_{\lambda 1} \subseteq Q\times X$ to $Y^I$;

- $\lambda_2$*:* $\pi_j \to Y^{II}$ – is a function of type 2 outputs that implements the mapping $D_{\lambda 2} \subseteq \pi_j$ to $Y^{II}$;

- $\boldsymbol{\lambda_3}$**:** $Q\times E \to Y^{III}$ – is a function of type 3 outputs that implements the mapping $D_{\lambda 3} \subseteq Q\times E$ onto $Y^{III}$ .\

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

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

## ASYNCHRONOUS ELEMENTARY SCHEMES OF MEMORY (elementary automata) [2; 4; 12].

They have a complete transition system, a complete exit system, and a system of state preservation functions. They consist of groups of NAND logical elements (OR-NOT), which are not connected to each other by the output, and the outputs of the logical elements of each group are connected to one input of the logical elements of all other groups, except for one. Monofunctional memory circuits have only one function of storing states in an asynchronous trigger and a multi-stable memory circuit (SMEs), which have a closed structure. Multifunctional memory circuits (MFIS) have more than one state saving function and an open structure, by connecting one input of each logic element from the input bus memory circuit, if there is more than one logical element in a group.

## COMPLETE TRANSITION SYSTEM [2].

The completeness of the transition system of the elementary automaton (memory circuit) determines that for each pair of internal states of the automaton that are saved with one input e(Δ) preserving signal, there will be an input signal x(t) that will translate the automaton from one state to another.

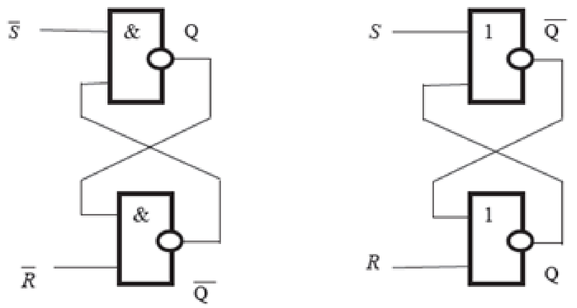
## COMPLETE OUTPUT SYSTEM [2].

The completeness of the output system of an elementary automaton (memory circuit) determines that in each state the automaton generates a signal $y_i$, which is different from signals that arise in other states of the automaton.

## MONO-FUNCTIONAL ELEMENTARY MEMORY CIRCUIT [4].

A monofunctional elementary memory circuit is called a single-level elementary automaton that has complete transitions and outputs when storing all the states of the automaton with only one *e*(Δ) preserving input signal.

## ASYNCHRONOUS *RS*-TRIGGER [4; 12].



*RS*-flip-flop patterns on the elements AND-NOT and OR-NOT

A feature of the *RS*-flip-flop is that the information is recorded in it directly with the arrival of the information x(*t*) (setting) signal to its inputs, which uniquely determines the values of the output signals of the *RS*-flip-flop. The states of the output signals are stored in the trigger circuits with one input preserving e(Δ), which operates between the setting input signals.

*RS*-trigger consists of two groups, each of which contains one logical element AND- NOT (OR-NOT). The outputs of each group of elements are connected to the inputs of the elements of another group. The free inputs of the elements are connected to the input bus of the memory circuit. It should be noted that all binary triggers use an asynchronous *RS*-trigger at the heart of their memory. Such as synchronous one-step and two-step triggers - *RS*-flip-flop, *D*-flip-flop, *JK*-flip-flop, *T*-flip-flop, etc.

## WORK RS-TRIGGER [9].

on elements AND-NOT

| $\bar{R}$ | $\bar{S}$ | $Q$ | $\bar{Q}$ |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
|  |  | 0 | 1 |

on elements OR-NOT

| $R$ | $S$ | $Q$ | $\bar{Q}$ |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |
|  |  | 0 | 1 |

The trigger on the elements OR-NOT has the following three set input signals: $x_1(t)$ (R = 1; S = 1); $x_2(t)$ (R = 1; S = 0); $x_3(t)$ (R = 0; S = 1); and one preserving the input signal: $e(\Delta)$ (R = 0; S = 0).

## MODELING OF WORK ON TACTS *RS*-TRIGGER [9].

Consider the action of the three input words in cycles in the tables with the input words $p_1(T)= x_1(t), e(\Delta); p_2(T)= x_2(t), e(\Delta); p_3(T)= x_3(t), e(\Delta)$.

The input word is $p_1(T)= x_1(t), e(\Delta)$

| Tacts | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $R$ | 1 | 1 | 1 | * | 0 | 0 |
| $S$ | 1 | 1 | 1 | * | 0 | 0 |
| $Q$ | * | 0 | 0 | 0 | * | * |
| $\bar{Q}$ | * | 0 | 0 | 0 | * | * |

The input word is $p_2(T)= x_2(t), e(\Delta)$

| Tacts | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $R$ | 1 | 1 | 1 | * | 0 | 0 |
| $S$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $Q$ | * | 0 | 0 | 0 | 0 | * |
| $\bar{Q}$ | * | * | 1 | 1 | 1 | 1 |

The input word is $p_3(T) = x_3(t), e(\Delta)$

| Такты | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| *R* | 0 | 0 | 0 | 0 | 0 | 0 |
| *S* | 1 | 1 | 1 | * | 0 | 0 |
| *Q* | * | * | 1 | 1 | 1 | 1 |
| $\bar{Q}$ | * | 0 | 0 | 0 | 0 | 0 |

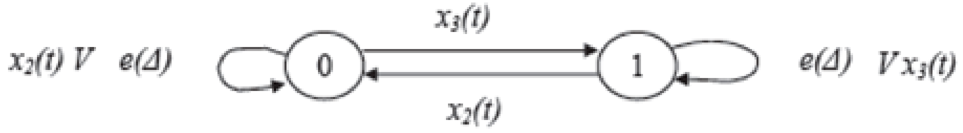
When considering the results of calculations under the action of the input word
$p_1(T)$, we see that the input signal setting $x_1(t)$ determines the value 0 on both output nodes, which is not preserved when the input signal preserving $e(\Delta)$ appears, because trigger nodes signals have an undefined value of 0.5 ($Q = 0.5$ and $\bar{\bar{Q}} = 0.5$). This explains the fact that when applying the word $p_1(T)$ the trigger does not have a stable output signal, and therefore, such an input word for the *RS*-trigger is prohibited when used. On the elements AND-NOT and on the elements OR-NOT the input word
$p_1(T)$ in the *RS*-flip-flop is prohibited when used in deterministic discrete devices.

When considering the results of calculations under the influence of the input word
$p_2(T)$. We see that the input signal setting $x_2(t)$ determines the output $Q = 0$ and
$\bar{\bar{Q}} = 1$ at the output nodes, which are saved when a preserving $e(\Delta)$ input signal appears.

When considering the results of calculations under the action of the input word
$p_3(T)$. We see that the input signal setting $x_3(t)$ determines at the output nodes the value $Q = 1$ and $\bar{\bar{Q}} = 0$ stored at the appearance of a saving $e(\Delta)$ input signal.

A trigger is considered to be at zero, when $Q = 0$, and at one, when $Q = 1$. This condition must be remembered so that it is always easy to determine the state in which the trigger is located, since it is symmetrical and easy to make a mistake.

## GRAPH DIAGRAM OF *RS*-TRIGGER WORK IN AUTOMATIC CONTINUOUS TIME [9].



## CALCULATION OF BASIC DIGITAL PARAMETERS OF RS-TRIGGERS ON LOGITAL ELEMENTS AND-NOT (OR-NOT) [9].

1. The number $M$ of storing states of binary or multistable triggers is calculated by the formula:

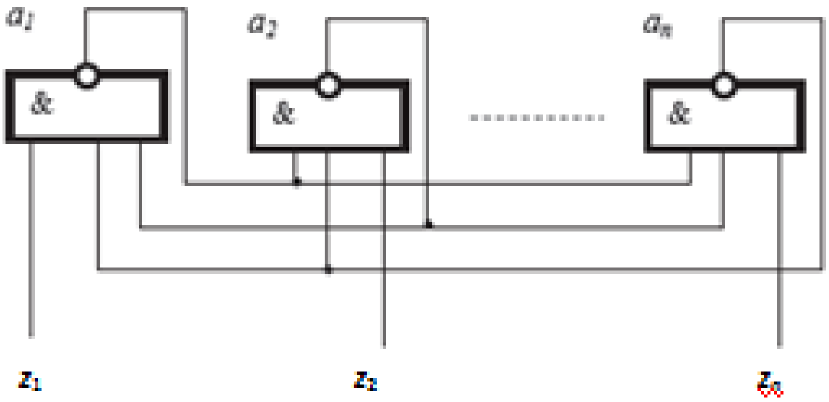
$$M = \sum_{i=1}^{n} (2-1)_i = n,$$

where n is the number of elements AND-NOT (OR-NOT) in the scheme of the trigger or SME.

2. The number $r_e$ of $e(\Delta)$ input signals of binary or multistable triggers is calculated by the formula:

$$r_e = \prod_{i=1}^{n} (2-1)_i = 1.$$

## GENERALIZED STRUCTURAL SCHEME OF MULTIPLE-STABLE MEMORY SCHEME (SME) [9].



An SME consists of *n* (*n*> 2) groups, each of which contains one logical AND-NOT (OR-NOT) element. The outputs of each group of elements are connected to the inputs of the elements of other groups. The free inputs of the elements are connected to the input bus of the memory circuit.

The calculation of the basic digital parameters of the SME on the elements AND-NOT (OR-NOT) is determined by the same formulas as the *RS*-trigger (see above).

## PRINCIPLE OF RECONFIGURABILITY [22]

means that the logical structure of the reconfigurable device can dynamically change both in preparation for solving the problem and during the computational process.

## STRUCTURAL VERSATILITY OF RECONFIGURED DEVICES (RD) [22].

Determined by the principle of reconfigurability, when it is possible by simple reprogramming the structure of the switchgear to adjust it to the effective implementation of a given algorithm.

## PRINCIPLE OF THE STRUCTURAL ORGANIZATION OF MULTIFUNCTIONAL MEMORY CIRCUITS (MFIS) [9].

It consists in the use of n logical elements OR-NOT (AND-NOT), which are divided into m (m <n) groups. The outputs of the elements of one group are not connected with the inputs of their group of logical elements. They are connected to the inputs of elements of other circuit groups. One of the free inputs of each i-th element is connected to the inputs of the establishing input bus, to which the input signal x(t) is received, and the second of the free inputs of each i-th element is connected to the inputs of the saving input bus of the memory circuit, to which the input signal goes e(Δ).

## DEFINITION OF MULTI-FUNCTIONAL MEMORY SCHEME (MFIS) [4; 9; 19].

MFIS is a single-level multifunctional elementary automaton with a complete transition system and a complete system of outputs when implementing each of re (re> 1) functions δe of saving states.

The MFIS can be functionally represented as re single-level elementary automata, each of which stores all its states only with one of the various corresponding sets of input signals that store $e_j(\Delta)$. MFIS has a matrix structure of memorized states.

State matrix of MFIS

| | $\mu_1$ | $\mu_2$ | ….. | $\mu_n$ |
|---|---|---|---|---|
| $\pi_0$ | $a_{10}$ | $a_{20}$ | … | $a_{n0}$ |
| $\pi_1$ | $a_{11}$ | $a_{21}$ | … | $a_{n1}$ |
| $\pi_2$ | $a_{12}$ | $a_{22}$ | … | $a_{n2}$ |
| … | … | … | … | … |
| $\pi_m$ | $a_{1m}$ | $a_{2m}$ | … | $a_{nm}$ |

The MFIS is called the automaton memory scheme, because the matrix scheme for storing states in which the transition in two variables can take place is similar to specifying the transition matrix in the Mealy and Moore automata [2].

## DIGITAL DESCRIPTION OF THE STRUCTURE OF MULTI-FUNCTIONAL MEMORY CIRCUITS WITH DIGITAL NUMBERS [4].

The digital description of the elementary memory device is represented as a position number (decimal or sixteen-year), which represents the structure of the MFIS. This number characterizes the structure so that using this number, it would be possible to formally compute the basic parameters of the memory circuit, on the basis of which to make the choice of the optimal, in the opinion of the designer, structure and build it on the logical elements in the form of a functional memory circuit. In the digital description of the MFIS, it is advisable to enter a decimal number, the number of digits of which corresponds to the number of groups of logical elements in the MFIS, and each digit - the number of logical elements in a particular group. The maximum number of decimal digits is 10, which corresponds to the restriction of the number of groups to 10 in the structure of the MFIS. These constraints are purely conditional, although they correspond to some extent by the restriction of many asynchronous elements of OR-NOT (AND-NO) integrated circuits.

The number that determines the structure of the MFIS in the decimal number system has limitations on the number of NAND logic elements (OR-NOT) in each group up to 9, which corresponds to the real limit of the integrated circuits. The number of digits characterizing the structure of the MFIS in the decimal number system has limitations on the number of possible inputs in the logic elements used (in this numerical description - up to 10). In the description, the structure of the MFIS is given by the number of logical elements in each i-th position of the number and the number of groups (bits) in the number itself.

To obtain the results that are obtained when using the digital description when selecting the MFIS, you can perform the following steps:

1. In the digital description of the MFIS, its main parameters can be determined: *M* - the number of memorized stable states; $r_x$ is the number of input signals that establish $x_i(t)$ and re is the number of input signals $e_j(\Delta)$, as well as select the necessary criterion (satisfying the obtained results of the main parameters) for the logical design of the MFIS, which is necessary to create promising devices for computers and networks.

2. Choosing the basic parameters of the MFIS, one can construct its functional diagram on the basis of logical elements (AND-NOT, OR-NOT, AND-OR-NOT), and also find its description in the form of a system of logical equations, when necessary, for simulation modeling memory circuits.

## DETERMINATION OF PARAMETERS OF MULTIFUNCTIONAL MEMORY SCHEMES [9].

The characteristic number of memorized states $K_i$ in the *i*-th group is determined by the formula:

$$K_i = 2^{R_i} - 1,$$

where $K_i$ is the number of logical elements in the *i*- th group.

The number *M* of steady states a(Δ) of the MFIS stored under the influence of the input signals that save e(Δ) is determined by the formula:

$$\boldsymbol{M} = \sum_{i=1}^{m} K_i \ ,$$

where $K_i$ is the characteristic number of the *i*-th group.

The total number of $r_x$ different sets of establishing $x(t)$ input signals MFIS is determined by the formula:

$$r_x = M + 1,$$

where $M$ is the number of steady states of MFIS, which are saved;

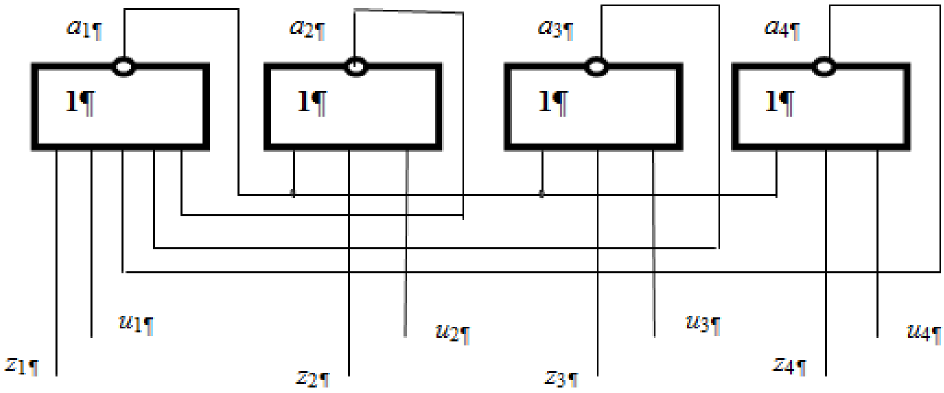
1 - additional set of input signal setting $x_p$(t), unambiguously setting the state $a_p$(t), which is not saved with any set of preservation e(Δ) input signal of the MFIS. Such a set of $x_p$(t) - installing input signal in deterministic devices is prohibited [12].

The total number re of different sets of saving $e$(Δ) input signals MFIS can be determined by the formula:

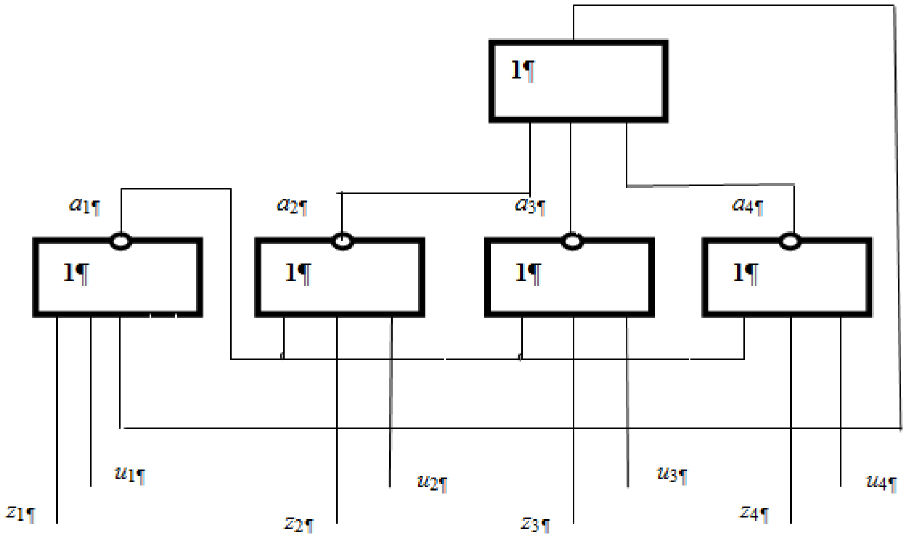
$$r_e = \prod_{i=1}^{m} K_i \ .$$

## STRUCTURES OF MULTIFUNCTIONAL CIRCUITS [4; 9; 19].

MFIS consider two classes of functional schemes: class $L$ and class $L^m$. For example, the multifunctional scheme of class $L$, described by the digital number 13 and denoted by $L13$, is represented by the inheriting scheme.

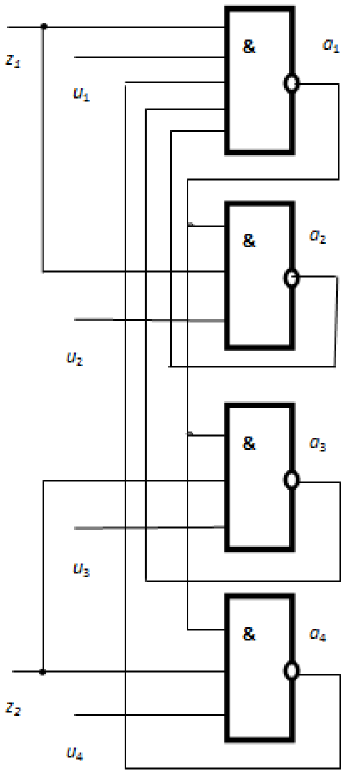


The MFIS of class $L^m$ on the elements OR NOT, described by the digital number 13 and denoted by *M13*, is represented by the inheriting scheme.



## DECREASE IN THE KNOTS OF THE INPUT SIGNAL x(*t*) IN THE MFIS [4].

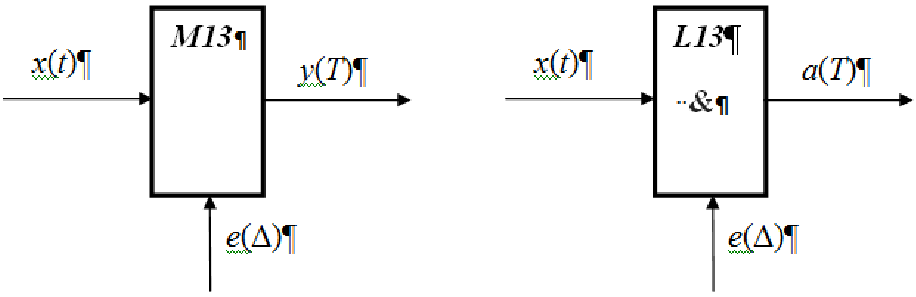
Reducing the number of input nodes in the MFISs of class $L^m$ and class *L*, where the groups of logic elements are more than 1, allows the setting inputs $z_i(t)$ of each i-th group of MFISs to combine into one input node. Input $u_i(\Delta)$ nodes in the MFIS are not specifically combined. At the same time, the algorithm of the MFISs operation does not change.



Thus, the set of the input signal x(t), which unites the setting nodes $z_i$(t) of all the *i*-th MFIS groups, reduces the number of input nodes to *m* groups.

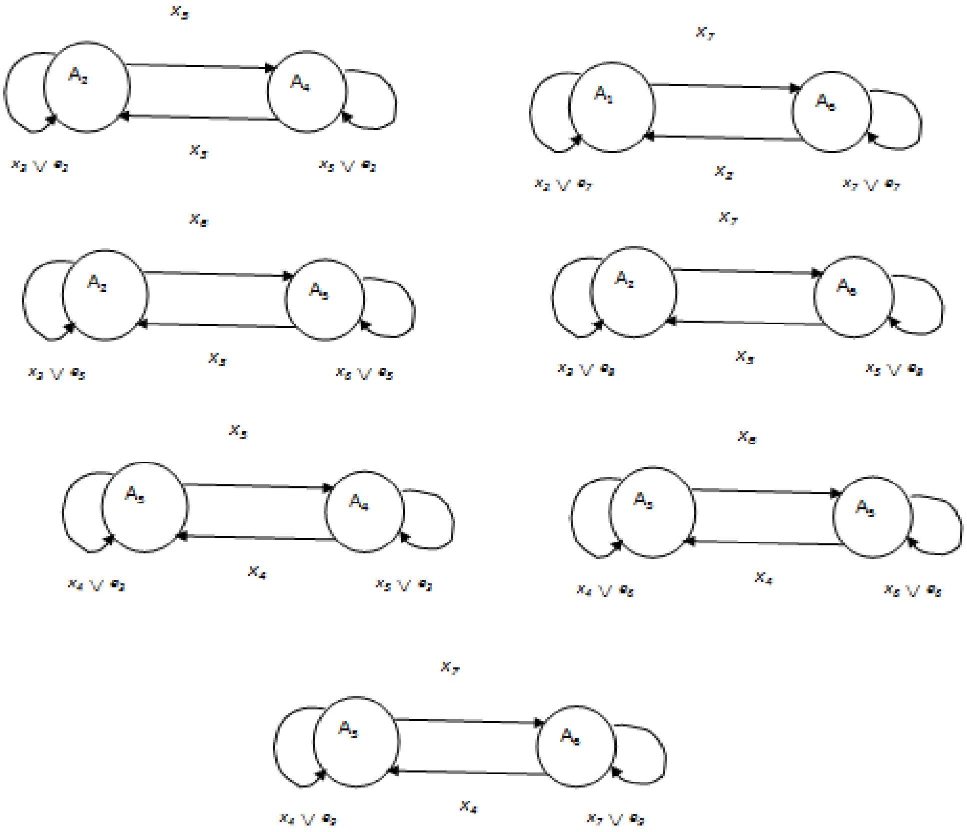
## REFERENCE SYMBOLS OF MULTIFUNCTIONAL MEMORY SCHEMES [4].

Inside the very symbol MFIS put the symbol *M13*, which reflects the structure of the MFIS class $L^m$. When class MFIS *L* symbol is replaced by *L13*. If necessary, the symbol must indicate on which logical elements the scheme of the MFIS is implemented, then you can put the & sign under the digital code when the AND-NOT logic elements are used, and if the & sign is not used, the OR-NOT logical elements are used.



Legend MFIS two classes *L* and $L^m$ reflect the structure, the number of input nodes of the input signal *x*(*t*), the number of which is equal to the number of digits of the number, the number of input nodes of the input signal e(Δ) and output signals *y(T)* (or *a(T)*), which are respectively equal to the sum of digits of the number, as well as what logical elements AND-NOT or OR-NOT are used in this scheme.

## DIFFERENCE OF MFIS FROM ASYNCHRONOUS *RS*-TRIGGER [4].



The fundamental difference between the MFIS and asynchronous *RS*-flip-flops during their operation is that the MFIS can function in different $\pi_j$ blocks of its own states, and the *RS*-flip-flops - only in one. With different sets of $e_j(\Delta)$ input signals in the $\pi_j$ blocks of their own states, the MFIS (*M22*) can function as nine different *RS*-flip-flops with different sets of their own states.

## DETERMINATION OF PARAMETERS OF ASYNCHRONIC MEMORY SCHEMES [4].

- ***The maximum number M of storage states with the limitations of the parameters of the logic elements from which the memory circuit is built.***

Single-phase multistable memory circuits, built on *K*-input elements AND-NOT (OR-NOT) with a load capacity at the outputs of *P*1, use *n* ($2 \leq n \leq K$; $2 \leq n \leq P1$) elements. The maximum possible number *M* of storing states of single-phase multistable memory circuits with $K \leq P1$ and $n = m = K$ is determined by the formula:

$$M_{\max} = \sum_{i=1}^{K} (2-1)_i = K.$$

For class *L* MFIS based on *K*-input elements AND-NOT (OR-NOT) with a loading capacity *P*1, the number of $R_i$ which is the same in all *m* groups, the maximum possible number $M_{max}$ of storage states for *n = K; m = 2; Ri* $\left( R_i = 2,\ 3,\ ...,\ \frac{K}{2} \right)$ which is the same in all *m* groups, the maximum number of $M_{max}$ storage conditions for $n = K;\ m = 2;\ R_i = \frac{K}{2}$ $(K \leq P_1)$ calculated by the formula:

$$M_{\max} = \sum_{i=1}^{m}\left(2^{R} - 1\right) = \sum_{i=1}^{2}\left(2^{\frac{K}{2}} - 1\right) = 2 \cdot \left(2^{\frac{K}{2}} - 1\right).$$

For MFIS class $L^{m}$ based on $K$-input elements AND-NOT (OR-NOT) with loading capacity $P1$, the number of $R_i$ ($R_i = 2, 3, ..., R$) of which is the same in all $m$ groups, and $R$ - input elements AND (OR) with a loading capacity of $P2$, the maximum possible number of $M_{max}$ of storage states at $n < P_2$; $m = K - 1$; $(K \le P_1)$; $R_i = R$; $n = m\,R$ is calculated by the formula:

$$M_{\max} = \sum_{i=1}^{m}\left(2^{R_i} - 1\right) = \sum_{i=1}^{K-1}\left(2^{R_i} - 1\right).$$

**• *Load capacity at outputs ($P_Q$)***

The parameter $P_Q$ indicates how many similar logic elements can be connected to the output of the device without disrupting its operation at the limiting operating frequency. This parameter depends on the load capacity on the outputs of the element $R_e$ and on the number of connections of the output of this element $S_e$ to other elements of the device. For single-phase multistable memory circuits, the smallest parameter $P_Q$ is determined by the formula:

$$P_Q = P_{э} - M_{\max} = P_{э} - K; \ \left(P_Q \ge 1; \ P_{э} > K\right).$$

where $M_{max}$ - the maximum number of memory states;

$K$ ($K = M_{max}$) - the number of inputs of elements AND-NOT (OR-NOT).

For class $L$ MFIS, the load capacity for the outputs $P_Q$ is determined by the formula:

$$P_Q = P_e - R_i = P_e - \frac{K}{2} \quad \left(P_Q \geq 1;\, P_1 \geq \frac{K}{2};\, R_i = \frac{K}{2}\right),$$

where $R_i$ is the number of elements in the $i$-th ($i$ = 1, 2) MFIS group.

For type MFIS $L^m$ load capacity for the outputs of $P_Q$ is determined by the formula:

$P_Q = n_e - 1$

and does not depend on the parameters $P_Q$, used elements.

- **Number of internal connections ($S_{cb}$)**

The $S_{cb}$ parameter characterizes the total number of links between the elements that need to be organized in a memory circuit for its operation. The parameter $S_{cb}$ for single-phase multistable memory circuits is defined as:

$S_{cb} = n(n-1)$

where $n$ is the number of AND-NOT elements (OR-NOT) that are used in the memory circuit.

The parameter $S_{cb}$ for class $L$ MFIS is calculated by the formula:

$S_{cb} = mR_i\,(n – Ri),$

where $n$ is the number of elements AND-NOT (OR-NOT) used in MFIS;

$m$ ($m$ < $n$) is the number of groups of such elements in the MFIS;

$R_i$ - the number of elements AND-NOT (OR-NOT) in the $_i$-th group of MFIS.

For MFIS class $L^m$ parameter $S_{cb}$ is calculated by the formula:

$S_{cb} = N + mR_i\,(m - 1),$

where the designations are the same as in the class $L$.

**• *Number of external links ($S_{vc}$)***

The $S_{vc}$ parameter characterizes the total number of external outputs that must be organized in the memory circuit for its further correct functioning. For a single-phase multistable memory circuit, the $S_{vc}$ parameter is defined as:

$S_{vc}$ = 2 n,

where *n* is the number of AND-NOT elements (OR-NOT) that are used in the memory circuit.
For MFISs of different classes with the same value of *M* (the number of memory states)
$S_{vc}$ <2n.

**• *Number of elements per state (L)***

The parameter *L* refers to the important characteristics of the memory circuits, since it allows one to estimate the power consumption and, to some extent, the hardware cost of organizing the memory. For an asynchronous single-level *RS* trigger and all single-phase multistable triggers, the value of the parameter *L* is equal to one, since *M* = *n* [12].

For class *L* MFIS we have:

$$L = \frac{\sum_{i=1}^{m} R_i}{\sum_{i=1}^{m} \left( 2^{R_i} - 1 \right)},$$

and for MFIS class $L^m$:

$$L = \frac{m + \sum_{i=1}^{m} R_i}{\sum_{i=1}^{m} \left( 2^{R} - 1 \right)}.$$

- ***Maximum number of alternative mappings ($r_e$).***

The number of memorized states *M* can be considered as entropy, which serves as a measure of the freedom of the system: the more entropy, the more states available to the system, the more degrees of freedom it has [22].

Multistable memory circuits [12] function only in one block of their states, which are remembered under the influence of only one set of the preserving $e_0(\Delta)$ input signal, which determines the zero degree of freedom.

MFIS function in different blocks $\pi_j$ of their own states, which are memorized under the influence of the corresponding sets of input signals that save $e_j(\Delta)$. Due to this, the possibilities of monofunctional memory circuits expand, which determines the re degrees of freedom. In this case, it is possible to implement alternative maps

$\{X\} \xrightarrow{e_j} \{a\}$ with different saving input signals $e_j(\Delta)$in the memory circuit itself without corresponding switching of the input and output signals.

## COMPARISON OF PARAMETERS ASYNCHRONOUS SCHEMES OF MEMORY [4].

| Parameter | A multistable memory circuit (SMEs) | MFIS class *L* | MFIS class $L^M$ | Class advantage MFIS |
|---|---|---|---|---|
| $M_{max}$ | 8 | 30 | 90 | $L^M$ |
| $F_p$ | 125 МГц | 125 МГц | 100 МГц | *L* |
| $P_Q$ | 3 | 6 | 9 | $L^M$ |
| $S_{внут\cdot c.}$($M$ = 28) | 756 | 18 | 12 | $L^M$ |
| *L* (*M* = 28) | 1 | 0,2 | 0,3 | *L* |
| $r_e$ | 1 | >3 | >3 | *L*<br>$L^M$ |

Multifunctional memory circuits have the advantage over the binary memory scheme of the *RS*-flip-flop and SMEs. MFIS reduce hardware costs for one memorized state; Increase the functionality by realigning the structure of remembering states in one machine cycle *T*.

## IMPROVING THE RELIABILITY OF MULTI-FUNCTIONAL MEMORY SCHEMES [4].

MFIS have increased reliability and survivability, which is very important for their use in building computer systems for important and important objects, such as nuclear power plants, air and railway transport, astronautics, which is important for the security of any country.

1. With an increase in the number of elements in groups, the average time between failures grows, which indicates an increase in the reliability of the MFIS as a memory circuit compared to multistable and binary memory circuits.

2. As the number of groups with the same number of elements increases, the mean time between failures decreases, which indicates the most preferable in terms of increasing the reliability of using MFIS with two groups with $R_i > 1$ elements in each of them.

3. If a logical element in a group fails, the output of which is not active, the MFIS can operate as a memory circuit, but with limited capabilities, which reflects its survivability as a memory ..

## MULTILEVEL SCHEMES OF MEMORY [4; 10–21].

The MFIS are an open structure that has the ability to rebuild the state memory structure, and, therefore, change the direction of the active output information on certain output nodes. In addition, MFIS has two sets of input signals: establishing and preserving. These two sets of input signals do not overlap in time. To use saving input signals, it is necessary to generate them from an additional source. Such an additional source of input signals may be output signals of another memory circuit. At the same time, such a combined memory scheme becomes hierarchical. The entropy of such a hierarchical memory scheme naturally increases.

When creating a multi-level memory scheme, its levels naturally exist. The maximum number of levels when creating a half-closed memory scheme is limited to the last (lower) level, which retains its state only in one saving state, like a trigger or a multi-stable memory scheme. If the lower level of the hierarchical memory scheme has the ability to use several sets of saving input signals ($r_e$> 1), then this memory structure is open.

Multifunctional (MFIS) and multilevel (MUSP) memory circuits are called automatic memory circuits, since they are based on the matrix structure of state memorization.

MUSP with a multi-functional organization system determines such a structure in which the multi-functional mode of operation of one device is determined by another device, the so-called $A_M$ strategy automaton. The $A_M$ strategy automaton in MUSP can be a mono- and multi-functional memory device. Depending on the capabilities of the $A_M$ MUSP strategy automaton with a multi-functional organization system, the structure of the MUSP can be defined as an open, half-open or closed structure.

## PRINCIPLE OF STRUCTURAL ORGANIZATION OF ELEMENTARY MULTILEVEL MEMORY SCHEMES [4].

It consists in their division into $A_M$ controllers and $A_y$-controlled multifunctional memory circuits interconnected accordingly. Multilevel memory circuits of a class $L_N$, with a common automaton of $A_M$ strategy, and a class $L_N^B$, with automata of $A_M$ strategy for each group, are considered in automaton continuous time. In the MFIS, the input signal $e_j(\Delta)$ retains certain subsets of πj states that allow you to create new transition functions, which expands the functionality of the MUSP, which uses in its structure of the MUSP. Single-stage MUSPs use several vertically connected MFIS that can be synchronized with a single signal.

## SYMBOL LANGUAGE OF DESCRIPTION OF MUSP [4].

The symbolic description of the structure of the MUSP class $L_N^B$ is presented in the following order:

1. $(K$-1$)b$ - reflects the $K$-level of the MUSP structure;

2. $A_{k-1}$, $A_{k-2}$, $A_0$ - reflects the digital description of each structure of the MFIS, starting from the upper level of the MUSP. When using SMP, one digit can be used that indicates the number of logical elements;

3. $\tau_i$ ($i$ = 0, 1, 2) - reflects the first (or second) stage of the MUSP, which is synchronized by the signal $\tau_1$ (or $\tau_2$) or asynchronous MUSP - with the symbol *Ø*;

4. *Rc* (R = 1, 2) - reflects a one - or two-layer structure of MUSP. Consider the symbolic description asynchronous memory circuits, MUSP, which is on the upper level MFIS (*M22*), and the lower level for each group *M22* three-digit triggers *T111* (or *MCП3*) and describes the record:

1*b*, 22, 111, 111, Ø, 1c

where 1*b* - reflects the two levels of MUSM;

22 - reflects the structure of the MFIS (*M22*) of the upper level, with two groups of two elements in each group;

111 - displays the structure of the trigger (*T111*) of the second level, consisting of three logical elements, divided into three

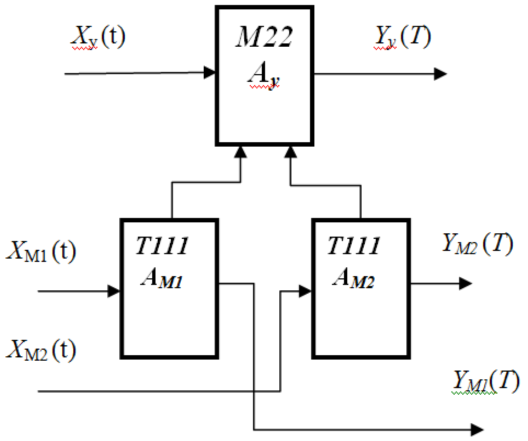
groups of one element each, and the connection of their outputs, respectively, with the inputs of elements of one group *M22*;

111 - displays the structure of the trigger (*T111*) of the second level, consisting of three logical elements, divided into three groups of one element each, and the connection of their outputs, respectively, with the inputs of elements of one group *M22;*

Ø - indicates that the MUSP is asynchronous;

1c - indicates that the MUSP is a single-stage memory circuit.

Symbol of an asynchronous two-level class memory circuit



In the symbolic description, the structure of the class $L_N$ MUSP is depicted in the following order:

1. (*K-1*) or (*K-1*)b - displays the *K*-level structures of the MUSP;

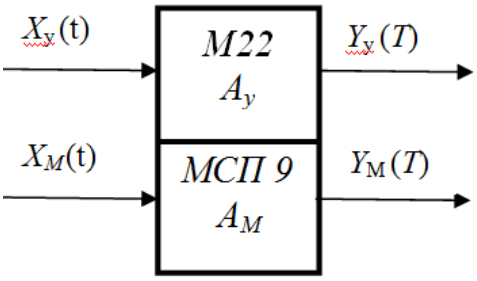
2. $A_{k-1}$, $A_{k\ -2}$, $A_0$ - reflects the symbolic description of each structure of the MFIS, starting from the upper level of the MUSP;

3. $\tau_i$ (*i* = 0, 1, 2) - reflects the first (or second) stage of the synchronizing MUSP signal $\tau_1$ (or $\tau_2$) or asynchronous MUSP - with the symbol *Ø*;

4. Rc (R = 1, 2) - reflects one-or two-stage structures of MUSP.

The symbolic descriptions of the MFIS and the MUSP ($n$ > 2) are key to the description of any class of MUSP. For class $L_N$ MUSPs, each character number $A_i$ MFIS reflects the connection of output signals from MFIS of all lower levels, which generate sets of $e_j(\Delta)$ input signals, with incoming nodes of preserving tires of upper MFIS. For MUSPs of a class $L_N^B$, the symbolic number $A_i$ MFIS reflects the connection of the output signals with MFIS of all lower levels. At the same time, the sets of saving $e_j(\Delta)$ input signals with input nodes are generated for one group of logical elements of the upper MFIS, in which the number of logical elements has a value greater than one ($q$ > 1).

The symbol MUSP class $L_N$ can be represented in one rectangle, divided into two parts: MFIS and strategy automaton ($A_M$), which includes the corresponding input signals x(t).



## DETERMINATION OF THE PARAMETERS OF THE MUSP BY THE SYMBOLS DESCRIPTION [4].

Determining the number of states

$$M = m_k \cdot r_e,$$

where $m_k$ is the number of bits in a character description MFIS, which is controlled by the;

$r_e$ - the number of sets preserving $e_j(\Delta)$ of the input signals to MFIS, which is controlled. The number of states MUSP that are remembered, it is also possible to determine the number $m_k$ of bits in a character description for each of MFIS.

$$M = \prod_{i=1}^{K} m_i$$

where $m_i$ ($i$ = 1, 2., $K$) is the number of digits in the symbolic description of the $i$-th MFIS of the whole MUSP;

$M$ - the number of states MUSP, which are remembered.

## COMPLIANCE WITH RELATIONSHIPS [4].

When creating a symbolic description of the strategy automaton (lower MFISs), the following relation should be observed:

$$r_e \le M,$$

where $r_e$ is the number of sets preserving $e_j(\Delta)$ of input signals MFIS top-level MUSP;

$M$ - the number of states of the MFIS lower of the levels MUSP that are remembered.

## THE LIMITATIONS OF LOGIC ELEMENTS IN THE SYNTHESIS OF MUSP [4].

For the synthesis of the symbolic description of MUSP need to check first the validity of the synthesis of MFIS, which is part of the MUSP, subject to the restrictions of logic elements, and then to execute the design of MFIS with the appropriate character description. Given the connection between the levels of MFIS in MUSP, we must ensure that the load capacity of the lower *P*I MFIS satisfy this relation:

$$\boldsymbol{P_1} \geq \sum_{i=2}^{m} R_i + K,$$

where $R_i$ is the value of the digits (except for one minimum digit) in the symbolic description of the lower MFISs;

*K* - the number of levels MUSP.

The number of permissible input nodes of the logical elements of the upper MFISs K*b* should satisfy the following relation:

$$K_b \geq \sum_{i=2}^{m} R_i + K,$$

where $R_i$ is the value of the digits (except one minimum digit) in the symbolic description of the upper MFIS;

*K* - the number of levels MUSP;

*m* is the number of digits of the symbol number of the upper MFIS.

For MUSP class $L_N^B$ when using MFIS class $L^M$, the number of input nodes of the logical elements of the upper MFIS K*b* should satisfy the following relation:

$$\mathrm{K}b = m + k.$$

The essence of the synthesis of MUSP at MFIS lies in finding the connections between the output nodes of the lower MFIS and the input nodes of the upper MFIS.

**ADVANTAGES OF MUSP [4].**

***According to the hardware characteristics*** of the MUSP, in comparison with multistable triggers, they use:

- fewer logical elements per memorized state by at least 40% (instead of 1 logical element 0.56 elements);

- less internal links by 20 times (instead of 306 links, it uses 14 internal links).

***According to the functional characteristics*** of the MUSP, in comparison with multistable triggers, the following features are used:

- MISP is a tunable structure, but no triggers;

- MUSP is capable of using deterministic transitions - unambiguous and enlarged, as well as probabilistic and fuzzy transitions for a three-stage structure, which enhances the computer's computer intelligence, and triggers - only one unambiguous transition;

- MISP is capable of perceiving hierarchical information for memorization in one machine tact T, which it is in principle not able to implement triggers.

***In the deterministic*** MUSP ***mode***, elementary $p_0(T)$ and $p_y(T)$ input words are implemented. Analyzing the elementary $p_0(T)$ and $p_y(T)$ input words, we arrive at the following conclusions:

- to transfer the MFIS from one state $A_i$ to another $A_k$ unambiguous $p_0(T)$ the input word can within one machine cycle $T$ make the transition by one variable $x_i(t)$ of the input signal in one block $\pi_j$ ($A_i$, $A_k \in \pi_j$) of its states, which is characteristic for triggers;

- to go from one state $A_i$ of block $\pi_s$ ($A_i \in \pi_s$) to another state $A_k$ of another block $\pi_j$ ($A_k \in \pi_j$) provided that $A_i$, $A_k$ belong to the block $\mu_m$ of their states, the input word sufficiently enlarged to the correct elementary $p_y(T)$, which is able machine cycle $T$ to make the transition in two variables $x_i(t)$ and $e_j(\Delta)$ input signals, which is typical only for MFIS.

The use of such deterministic elementary $p_0(T)$ and $p_y(T)$ input words extends the functionality of the triggers. This allows the implementation of transitions within one machine cycle $T$ by two $x_i(t)$ and $e_j(\Delta)$ input signals, which in principle can not be done in the triggers.

The use of different sets of saving $e_j(\Delta)$ input signals allows changing the structure of the storage states in the MFIS in one machine cycle *T*, which is very important when accelerating the restructuring of information processing algorithms.

## TWO-LEVEL SCHEMES OF MEMORY OF MUSP CLASS $L_N$ [4].

When implementing an $A_M$ strategy automaton on structures of multistable memory circuits (SMP) [17], it is sufficient to know $r_e$ the necessary number of re sets of input-saving $e_j(\Delta)$ input signals for the controlled MFIS $A_y$ [9] in order to apply the re states in the MFIS used as an automaton of the $A_M$ strategy.

The number of links between the managed MFIS $A_y$ and the $A_M$ strategy automaton is determined by the formula:

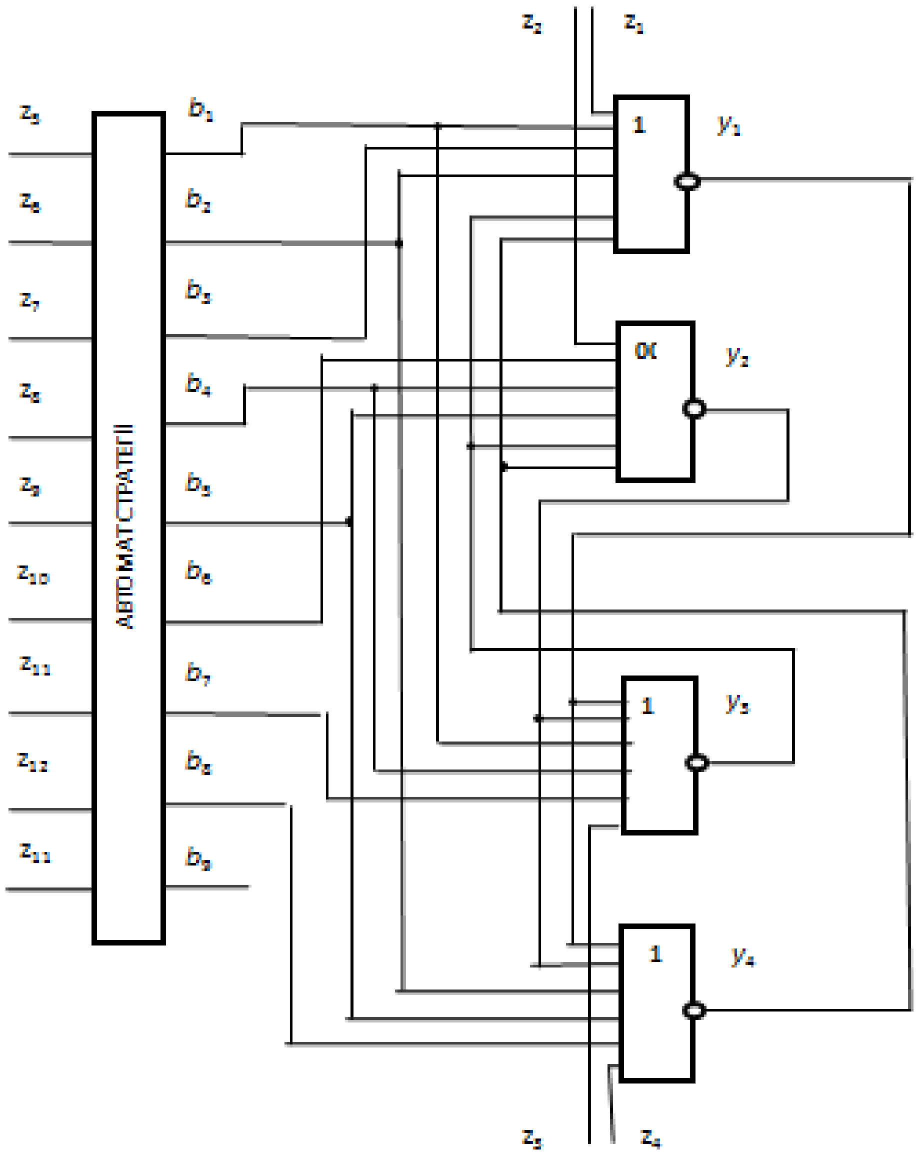
$$r_c = \prod_{i=2}^{m} (2^{R_i} - 1) - 1 = r_e - 1,$$

where *i* is the *i*-th *BA* group;

*m* is the number of $BA_j$ groups in the controlled MFIS $A_y$;

$R_i$ is the amount of $BA_j$ in the *i*-th group of the controlled MFIS ;

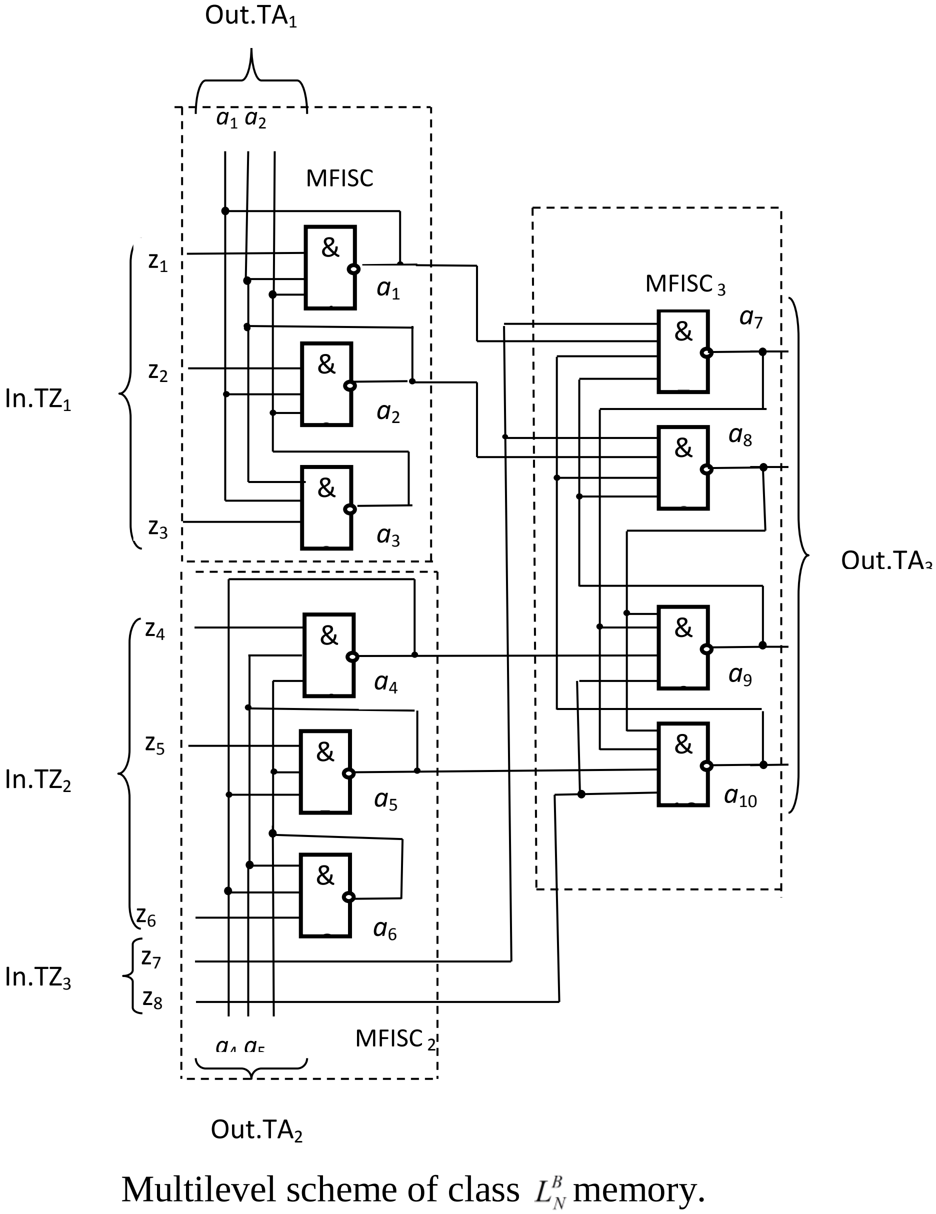
$r_e$ is the number of sets of preserving $e_j(\Delta)$ input signals in MFIS $A_y$.



## TWO-LEVEL SCHEMES OF MEMORY OF MUSP CLASS $L_N^B$ [4].

The total number *M* of memory states of a multi-level memory scheme is calculated by the formula:

$$M = \prod_{i=1}^{J} m_i$$



Multilevel scheme of class $L_N^B$ memory.

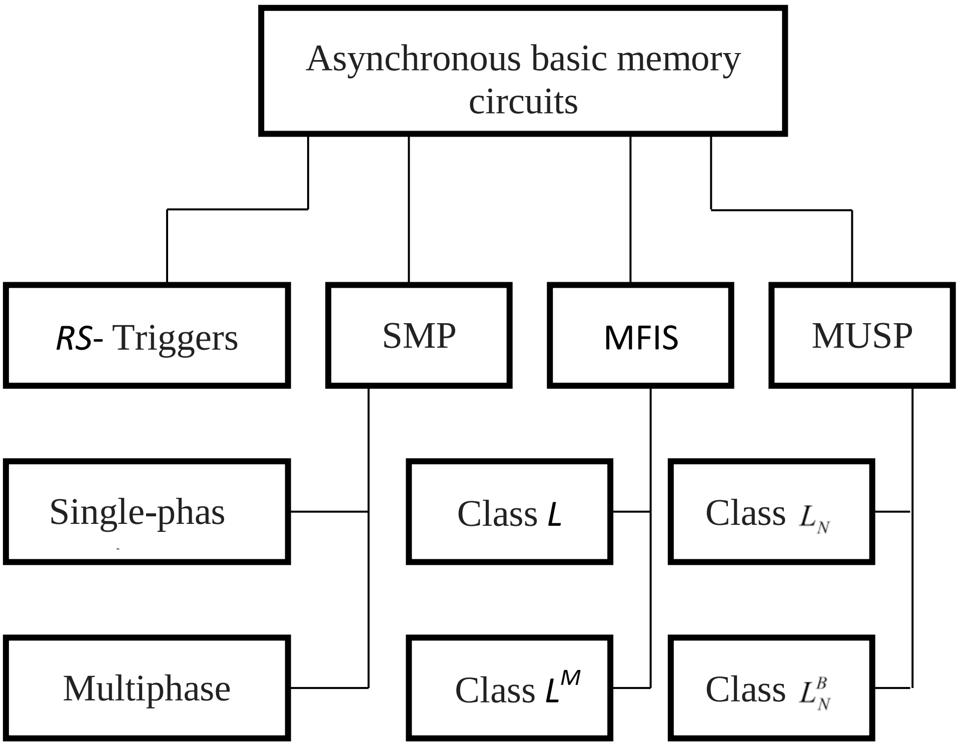
.

Total number $S_{cv}$. external relations is determined by the ratio:

$S_{cv}$ .. < 2*n*.

The total number of $S_{vs}$ internal links between elements is determined by the ratio: $S_{vs}$.< $n$×($n$-1).

## CLASSIFICATION OF MEMORY SCHEMES [4].



## COMPARISON OF THE PARAMETERS OF THE MEMORY SCHEMES, MEMORIZING 18 STATES [4].

Parameters of basic memory circuits that store 18 states

| Parameters | Single-phase triggers | MUSP |
|---|---|---|
| $F_p$ | 12,5 МГц | 12,5 МГц |
| $n_Q$ | 1 | 10 |
| $S_{cv.}$ | 306 | 24 |
| $S_{vc..}$ | 36 | 18 |
| $L$ | 1 | 0,56 |

## ADVANTAGES OF MFIS AND MUSP [4].

***According to the hardware characteristics*** of the MUSP, in comparison with multistable triggers, they use:

- fewer logical elements per memorized state by at least 40% (instead of 1 logical element 0.56 elements);
- less internal links by 20 times (instead of 306 links, it uses 14 internal links).

***According to the functional characteristics*** of the MUSP, in comparison with multistable triggers, the following possibilities are used:

- MUSP is a tunable structure, but no triggers;
- MUSP is capable of using deterministic transitions - unambiguous and enlarged, as well as probabilistic and fuzzy transitions in a three-stage structure, which increases the computer's "machine" intelligence, and triggers - only one unambiguous deterministic transition ;.
- MISP is able to memorize hierarchical information in one machine tact T, which is fundamentally unable to trigger triggers.

***In the deterministic MUSP mode***, elementary $p_0(T)$ and $p_y(T)$ input words are implemented. Analyzing the elementary $p_0(T)$ and $p_y(T)$ input words, we arrive at the following conclusions:

- to transfer the MFIS from one state $A_i$ to another $A_k$ unambiguous $p_0(T)$ the input word can within one machine cycle

$T$ make the transition by one variable $x_i(t)$ of the input signal in one block $\pi_j(A_i,\ A_k \in \pi_j)$ of its states, which is characteristic for triggers;

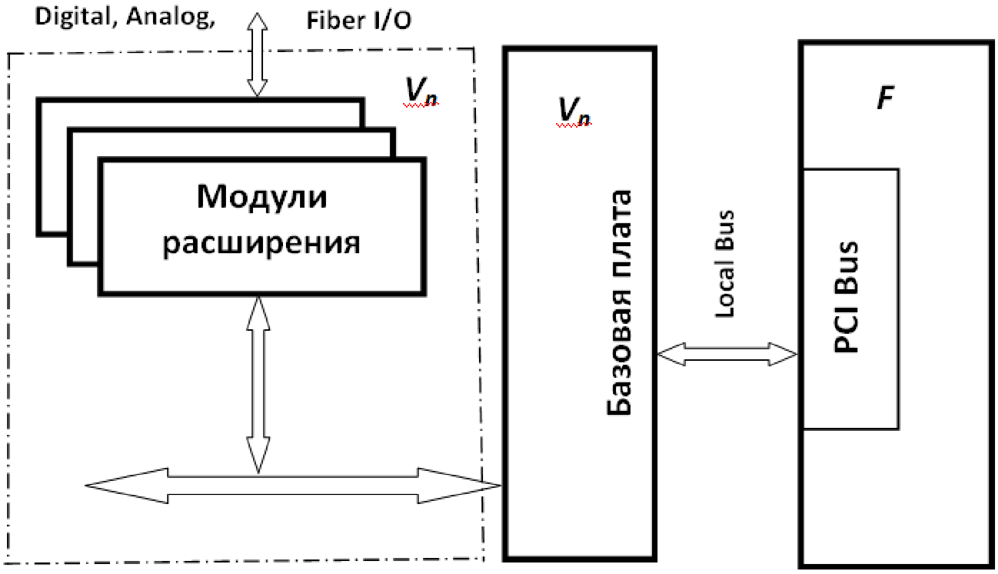
- to go from one state $A_i$ of block $\pi_s(A_i \in \pi_s)$ to another state $A_k$ of another block $\pi_j(A_k \in \pi_j)$ provided that $A_i$, $A_k$ belong to the block μm of their states, the input word sufficiently enlarged to the correct elementary $p_y(T)$, which is able machine cycle $T$ to make the transition in two variables $x_i(t)$ and $e_j(\Delta)$ input signals, which is typical only for MFIS.

The use of different sets of -saving $e_j(\Delta)$ input signals allows changing the structure of the storage states in the MFIS in one machine cycle $T$, which is very important when accelerating the restructuring of information processing algorithms.

## *SYSTEM APPROACH TO THE CONSTRUCTION OF RECONFIGURABLE COMPUTER DEVICES*

## A RECONFIGURABLE STRUCTURE WITH MEMORY ON TRIGGERS [23].

The structure of a computer system with memory on triggers, which changes the system configuration, consists of two parts: a permanent (or fixed) part F - the host computer and a part V - a multifunctional device, which is transformed into various system configurations [23].



The architecture of reconfigurable systems depends on the power of the set of algorithms (*N*) that are executed.

$$N = N_F + N_V,$$

where $N_F$ is the power of the set of algorithms running on devices *F*;

$N_V$ is the power of many algorithms running on devices *V*.

An important factor in a computer with a flexible structure is the level of flexibility and the level of components that are programmable. In this direction of work "Reconfigurable Computing" the level of multifunctional components that are programmable falls to the functional units of the computer control system, that is, to the "automatic" level.

The existing elemental base of modern integrated circuits with memory on triggers embedded in FPGA of the FPGA type with a large logical capacity (up to 10 million logic gates) created opportunities for building flexible devices and systems with an architecture that is programmable.

Triggers are fundamentally not able to change the structure of remembering states in the course of their functioning. This situation does not allow to lower the "automaton" level of flexibility of computer components and computers themselves to the "elemental" level of integrated circuits (IC).

## REGISTERS.

A register is a node that provides for receiving, storing, and issuing information, as well as performing a series of logical operations on information that is stored in the register. According to the main functional purpose distinguish between memory registers and shift registers. In the literature, memory registers are found under the name of static registers on triggers [12].

## ONE STEP REGISTERS WITH MEMORY ON MUSP [4].

The main purpose: parallel reception of hierarchical information and storing it for a long time. Information signals x(*t*) for the

MUSP can be clocked by the signal $\tau$. In the absence of a signal $\tau$ at the input nodes of the MUSP, only one e(Δ) preserving input signal is fed to the input nodes. Clocked, by one signal $\tau$, MUSP, are called single-stage.

## INPUT SIGNALS OF INPUT WORD MUSP [4].

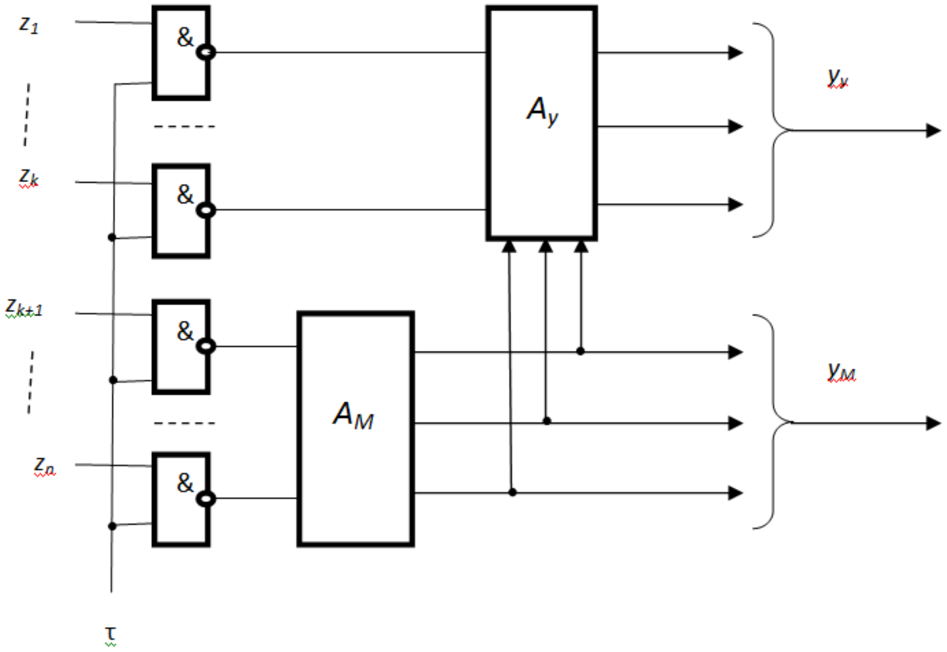
The input signals of the input word MUSP, which store parallel hierarchical information, are of two types: input signals x($z_y$) controlled by MFIS $A_y$ and input signals x ($z_M$) MFIS of the $A_M$ strategy automaton.

## OUTPUT SIGNALS MUSP [4].

They exist of two types: output signals $y_y$ controlled by MFIS $A_y$ and output signals from the $A_M$ MFIS automaton of the $A_M$ strategy, which controls the change in the structure of storing the states of MFIS $A_y$.

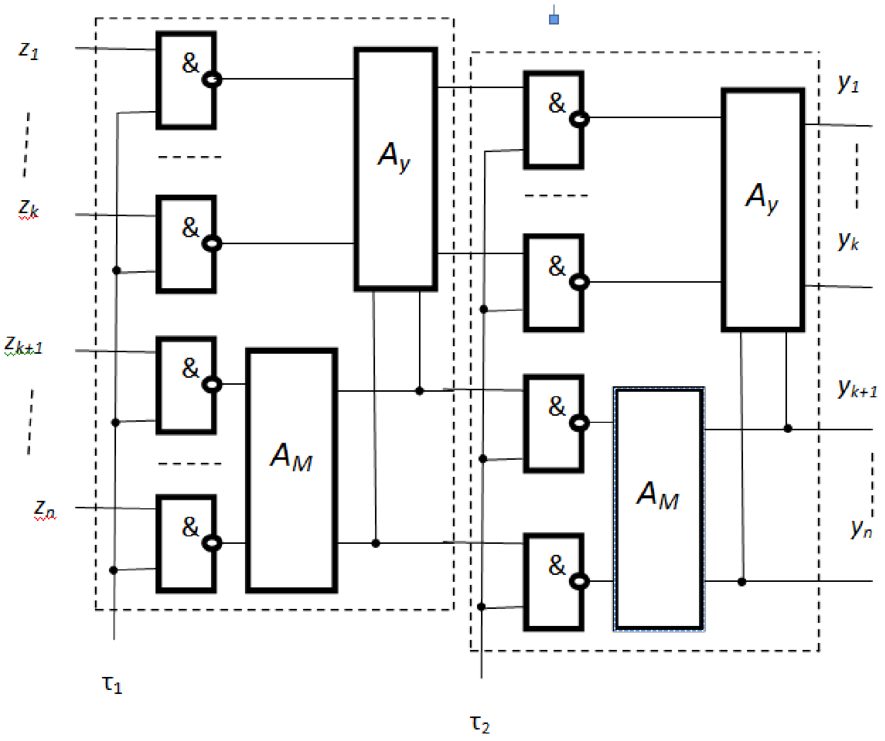
## FUNCTIONING OF SYNCHRONOUS MUSP [4].

MUSP can function as an automaton of the second kind, having a transition in tact t from one state to another, producing an output signal y($T$), or as an automaton of the third kind, having a transition in tact Δ from one state to another and producing an output signal y(Δ). The output signals of the MUSP (or other memory circuits) can be perceived by other memory circuits only with the appearance of the next clock signal $\tau_{i\ +\ 1}$. For this purpose, the output signals of the MUSP must have stable values after the machine clock $T$, which reflects the period between the two clock signals $\tau_i$ and $\tau_{i+1}$, for reliable removal of information. The output signals y($T$) of automata of the second kind as well as the output signals y(Δ) of automata of the third kind can be used when the next clock pulse $\tau_{i+1}$ appears. Receiving information from the source nodes of the memory circuits can be done in the usual ways: asynchronously or synchronously. Removing information from the output nodes of the memory circuits can be performed in parallel with all nodes or sequentially.



**STRUCTURAL SCHEME OF TWO-STEP SYNCHRONOUS MUSP [4].**

A synchronous MUSP is characterized by the fact that each transition occurs when a clock (synchronous) signal τ is applied to the input or when a clock (synchronous) signal τ ends with a minimum delay of one logical element, which at the output nodes of the MUSP, to the appearance of a clock sync pulse $\tau_{i+1}$, has stable sets of $y_j(\Delta)$ output signals.



**REGISTERS OF MEMORY ON MUSP [4].**

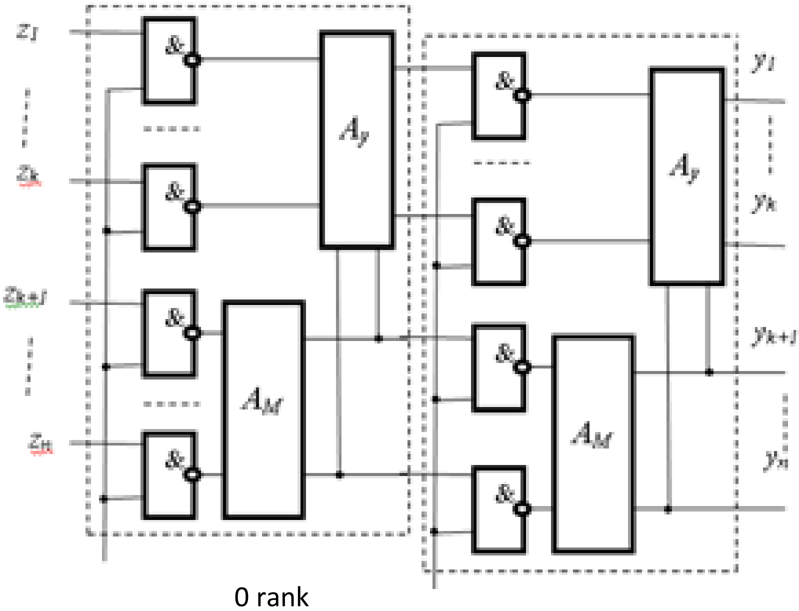
By repeating the structure of the *i*-th digit of the structural scheme of a two-stage synchronous MUSP *n* times, one can make up a general scheme of an *n*-bit parallel memory register. The block diagram of the parallel memory register of *n* bits is:

The number of $M_N$ memory states MUSP is determined by the formula:

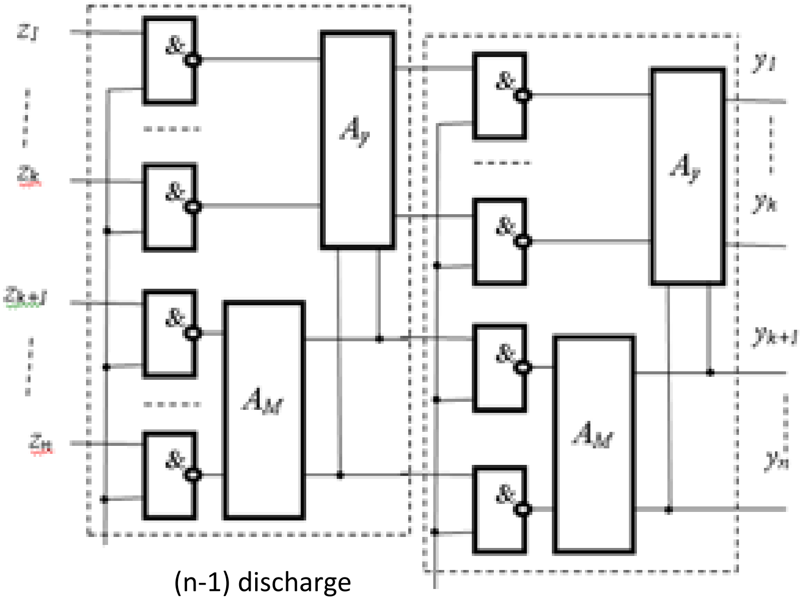
$$M_N = \prod_{j=1}^{N} m_j,$$

where *N* is the number of levels in the MUSP;

$m_j$ is the number of BA groups in the $j$-th level MUSP.



0 rank



(n-1) discharge

The number of memory states $M_N$ of the memory register is determined by the formula:

$$M_N = \prod_{j=1}^{N} m_j,$$

where *n* is the number of bits of the memory register;

*M<sub>N</sub>* is the number of memory states of the MUSP.

The range of integers that are identified with the number of register states has the following form:

$$0 \le \lfloor A \rfloor < M_{p.}$$

## PARALLEL REGISTER OF MEMORY ON THE MFIS WITH ONE AUTOMATIC STRATEGY [4].

In a multilevel parallel register, it is possible to use the $A_M$ strategy automaton not in every MUSP *m* digit, but one for all MFSPs $A_y$ of the entire register.

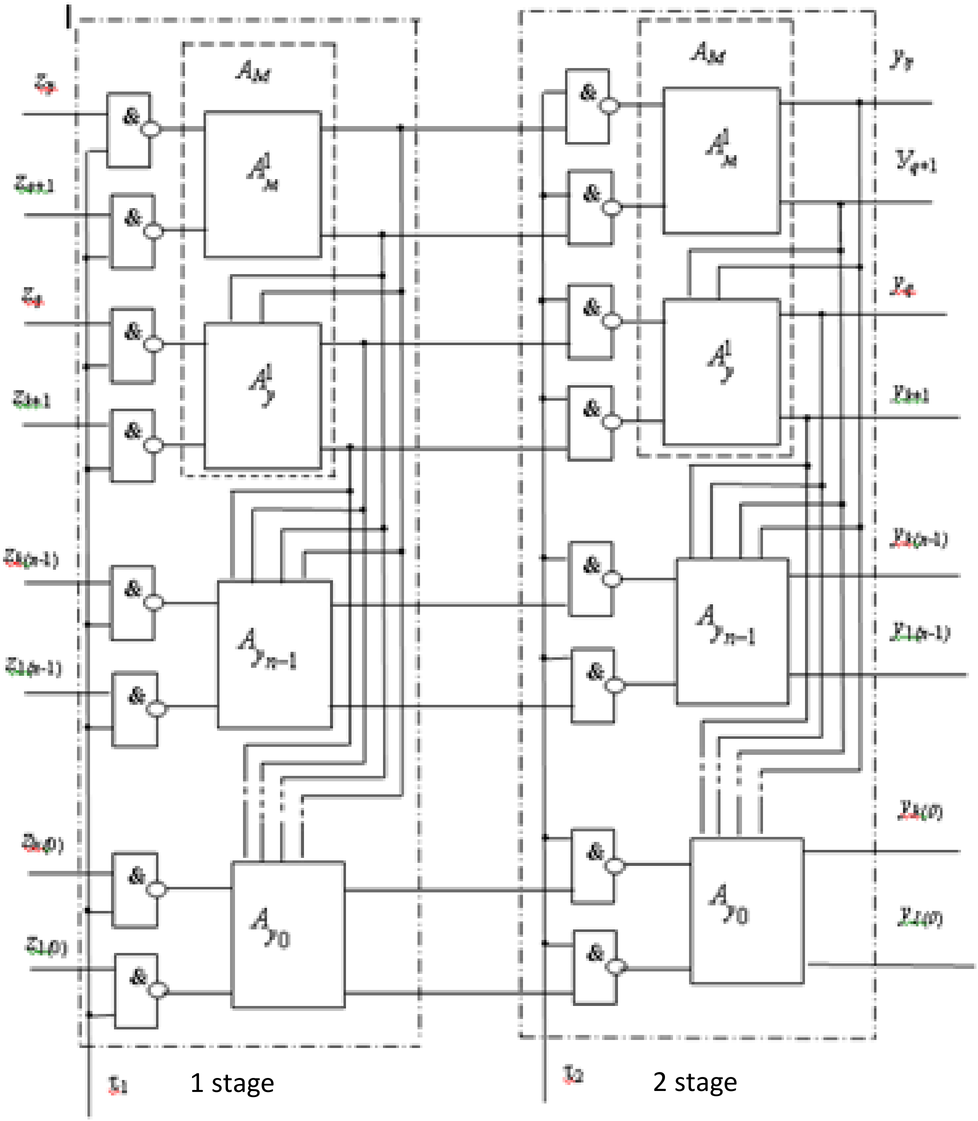
The range of integers that are identified with the number of register states with one $A_M$ strategy automaton looks like this:

$$0 \le [A] < M_{м} \times M_y^n,$$

where $M_M$ is the number of memorizing states of the $A_M$ strategy automaton;

$M_y$ is the number of memory states of the MFIS $A_y$;

*n* is the number of register bits.



The ranges of the register numbers on the MFIS with one automaton of the strategy are divided into numerical segments determined by the states of the automaton of the $A_M$ strategy. The properties of the memory registers on the MUSP allow the two-stage synchronous memory devices to rebuild the algorithm of their functioning without losing speed, while memorizing general information in the $A_M$ strategy automaton and private information in the MFIS Ay.

## COMPARISON OF MEMORY REGISTERS [4].

| Options | Register on Triggers | Register on MUSP |
|---|---|---|
| Number of logic elements per state | *L*=2 | *L*=1 |
| Number of internal links per state | $S_{int.s} = 2$ | $S_{int.s} < 1{,}9$ |
| Number of external links per state | $S_{vntr.s} = 2{,}5$ | $S_{vntr.s} < 1{,}06$ |
| Functions for implementing different mappings | NO | How in MFIS |
| Implementation functions of large-scale transitions | NO | How in MFIS |
| Border Operating Frequency | the same | the same |

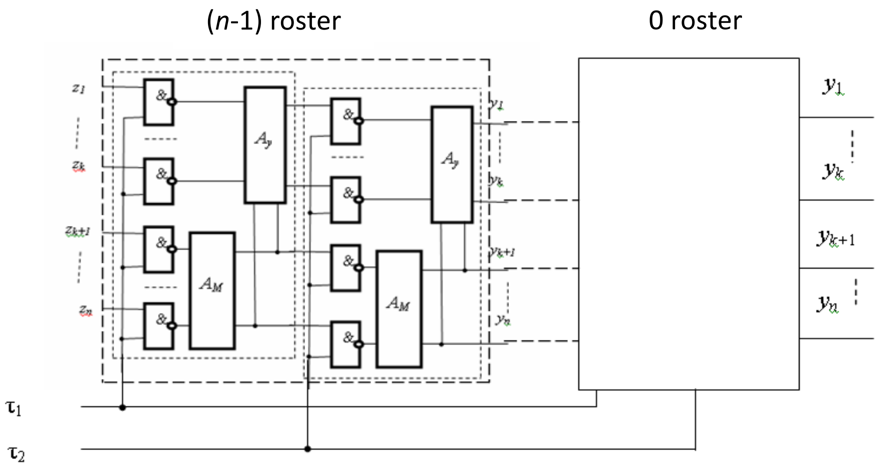
## REGISTERS SHIFT [12].

Shift registers are typical computer nodes. Allocate registers with a shift to the right, left, or reversible bits of the shift register in two-stage memory circuits.

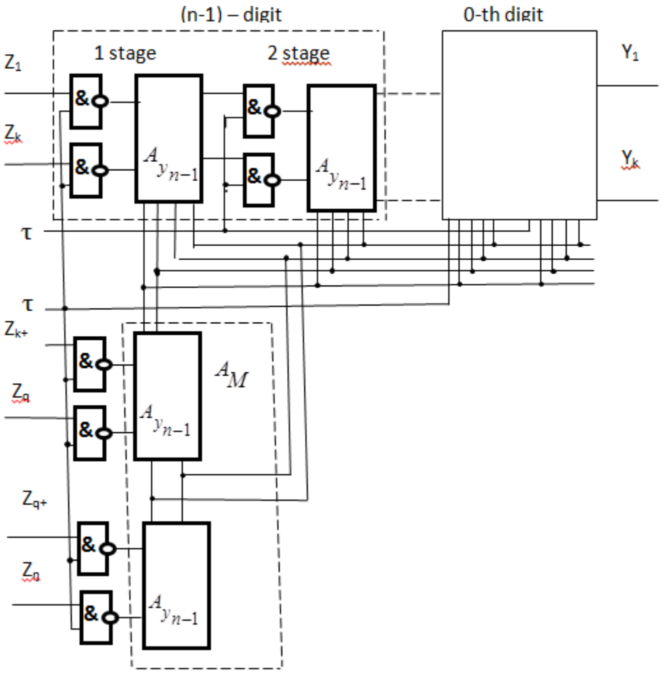
## REGISTERS SHIFT ON MUSP [4].

The shift registers on the MUSP have a higher power of storage states in each *i*-th digit than with binary ones (shift registers on triggers), which allows for a shift immediately modulo $M_N$ ($M_N$ is the number of memory states in one register bit).

## STRUCTURAL SCHEME OF REGISTER OF SHIFT ON MUSP [4].



## STRUCTURAL SCHEME OF REGISTER OF A SHIFT ON MFIS WITH A SINGLE AUTOMATON STRATEGY $A_M$



## WORK OF THE REGISTER ON MFIS AND MUSP [4].

The operation of the shift register on the MFIS and MUSP is carried out in certain blocks $\pi_j$ of its states, which are preserved under the corresponding states of the automaton of the $A_M$ strategy. When only the states of the $A_M$ strategy automaton change in the shift register, an enlarged transition is performed in the $\mu_i$ block of register states.

By the same principle, it is possible to implement left shift registers and reverse registers.

Applying parallel registers and shift registers, we see that they are able to memorize both common and private information, make transitions on two variable input signals *x* and *e*, rebuild the algorithm of their work, which is fundamentally impossible to do on trigger registers.

## COUNTER [12].

A counter is an automaton that, according to certain rules, calculates input signals (pulses), which generates and remembers the result of the counting in a certain code. Counters can be positive, negative or reversible.

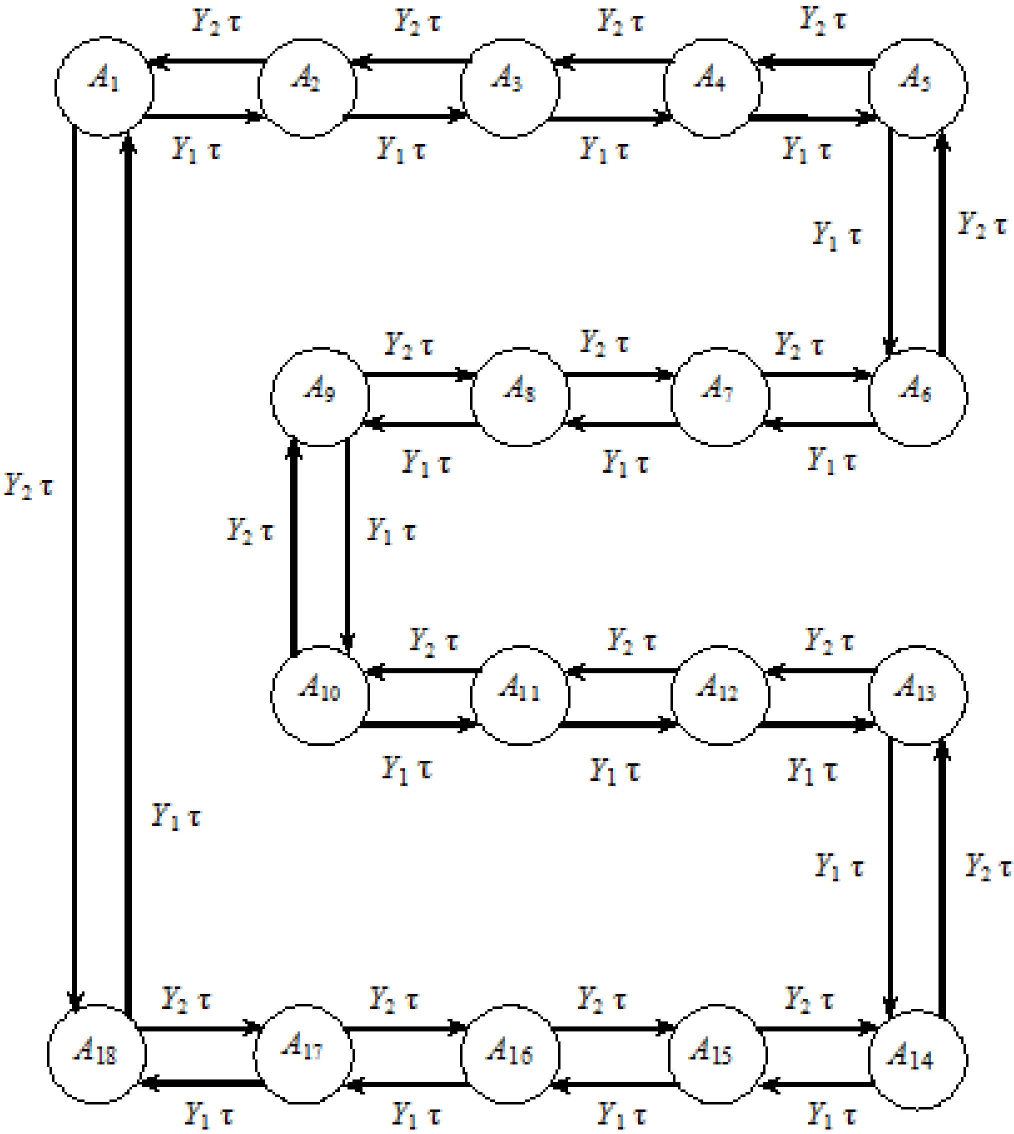
An important characteristic of the counter is the conversion factor *K* (module, period) of the counter - the maximum number of input signals that the counter can calculate**.**

## COUNTERS ON MUSP [4].

On the basis of the extended structural completeness theorem, it is possible to build counters, as well as any discrete devices that, besides the transition in *t*, also have transitions during the internal tact Δ of automatic continuous time *T*.

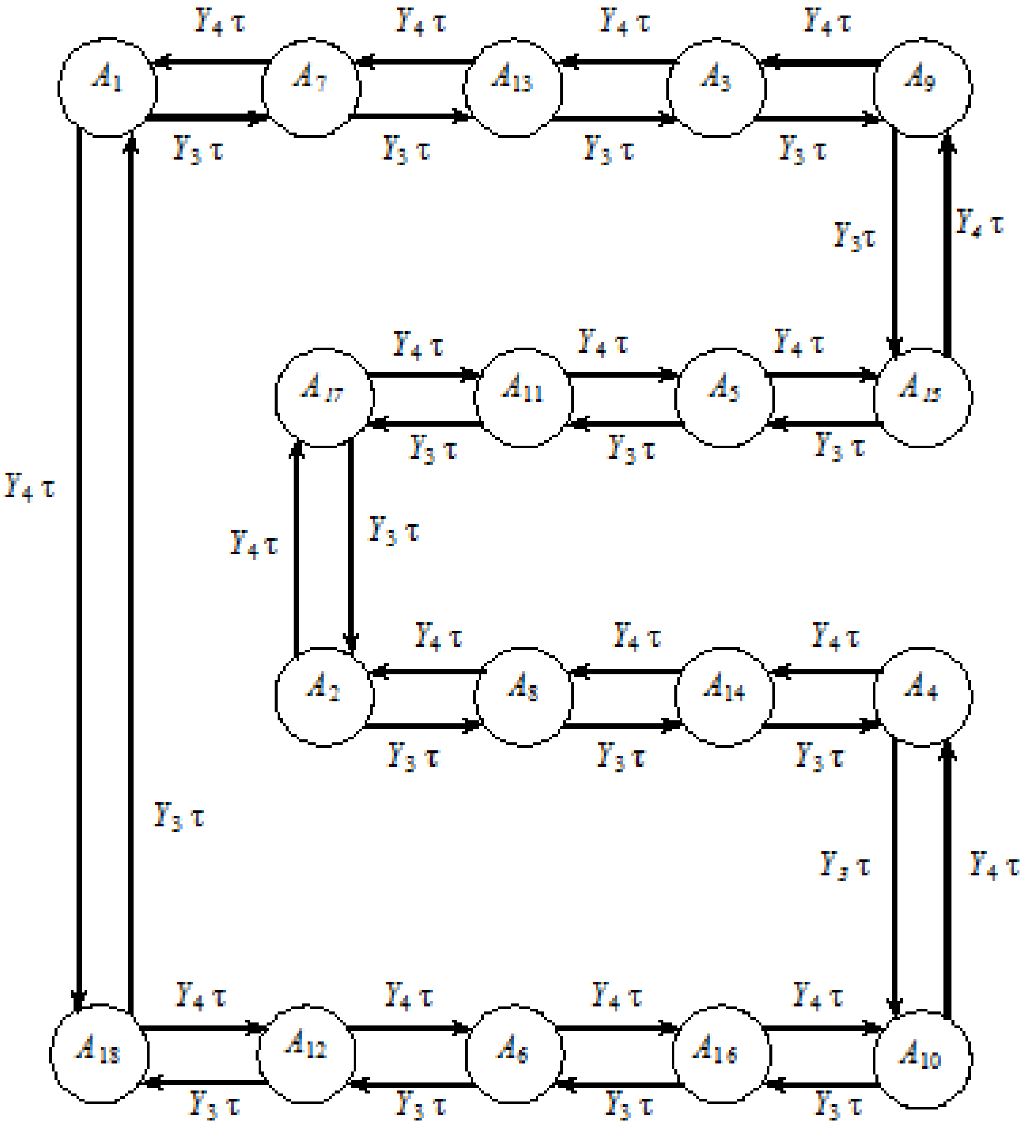
## METHOD FOR BUILDING A COUNTER ON THE MUSP [4].

As the memory of the counter, we choose the class $L_N^B$ MUSP for 18 states. This multilevel scheme contains the MFIS and two triggers for three (see two-level diagram of memory of the class $L_N^B$). We build a two-stage scheme on a class MUSP, in which the second stage is used as a delay in the output signal necessary for reliable operation of the counter. Each stage MUSP is synchronized by the signal $\tau_i$ ($i$ = 1, 2). The algorithm of the reversible counter, which is considered as a multifunctional machine of the 2nd kind, is that the transitions in the MFIS (upper level) MUSP are carried out in one $\pi_j$ state block with constant stable states of strategy automata (triggers on three lower level states), as in the low-order counter.



The algorithm of the reversible counter, which is considered as a multifunctional machine of the 3rd kind, is that transitions in low-level strategy machines due to the internal multi-functional memory organization system of a multilevel device perform enlarged transitions in the upper level MFIS.

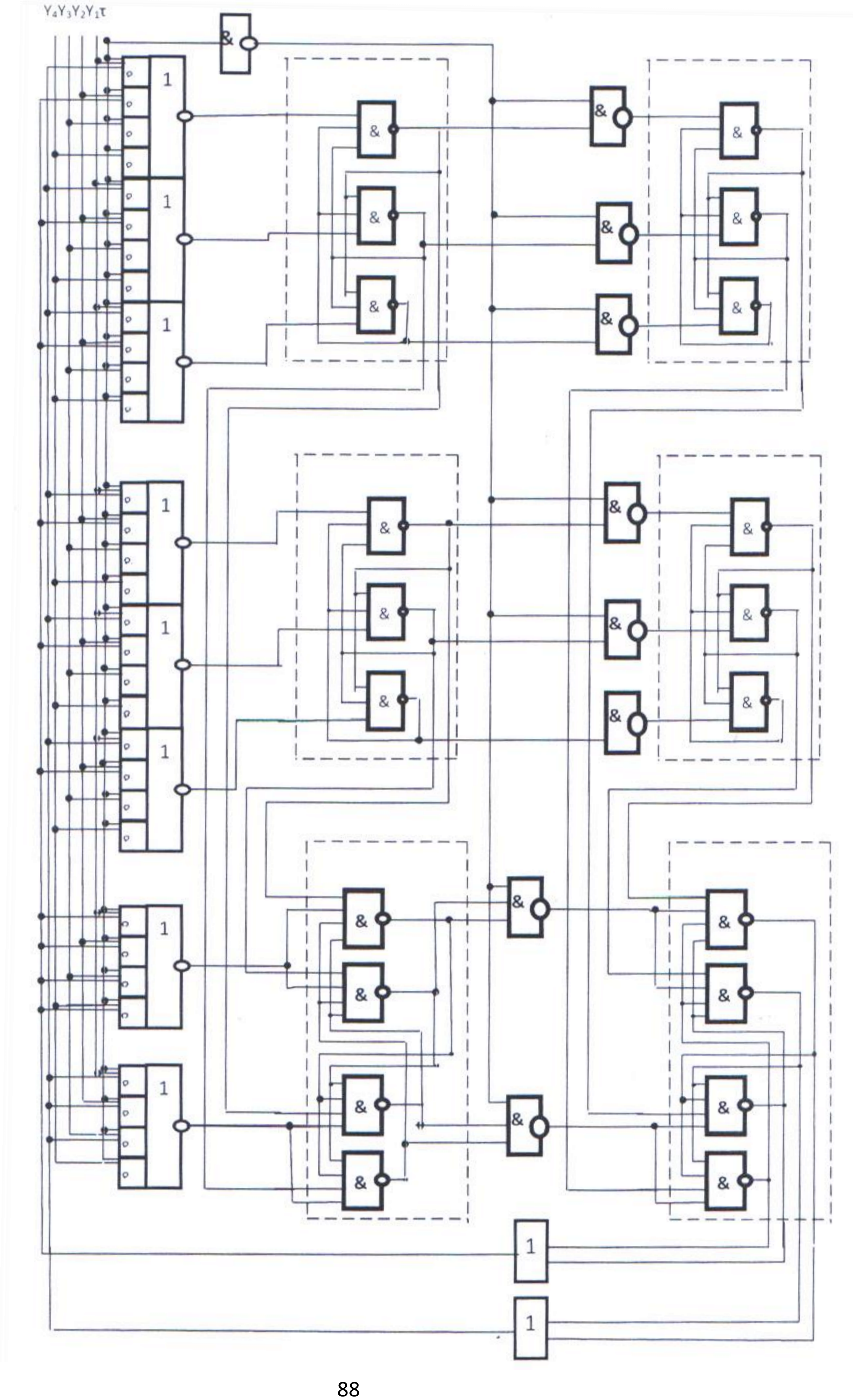
The scheme of a reversible counter implemented on a two-stage memory device that has the ability to perform various modes $Y_i$ ($Y_1$ - $Y_4$) of work modulo 18. The circuit of the counter is constructed traditionally.



The two-level MFIS of the upper level is constructed as a T-type trigger, which counts modulo 2, and the automata of a strategy as counters modulo 3.

When the clock signal $\tau_1$ appears, the value of the first stage of the counter changes in accordance with the operation mode $Y_i$ ($Y_1$ - $Y_4$) and the corresponding operation algorithm. When stable output signals appear at the output nodes of the first stage of the counter and the synchronization signal $\tau_2$, the values of the first stage of the counter are rewritten into the second stage of the counter.

A reversible counter on the MFIS can have four modes of operation, exceeding the functionality of reversible counters on triggers, which have only two modes.



Y4Y3Y2Y1τ
&
1

## CONTROL DEVICE WITH MEMORY ON TRIGGERS [12; 24].

The control unit (CU) in modern computers is part of the central information-processing device (processor), designed for automatic control of the computational process, which coordinates the work of all computer devices using synchronization and control signals that are developed during program execution . The most widely known for performing CUs in integrated components of processors is a CU with a Wilks-Stringer matrix structure and using registers on triggers, a two-matrix microprogram control circuit. The disadvantage of these block diagrams of a CU with a matrix structure is that due to the implementation of memory on triggers in register structures, only sequential processing of information is implemented.

## CONTROL DEVICE WITH MEMORY ON MUSP [4; 25].

Structurally, the CU is built on the MUSP differs from the CU built on the triggers, because the CU on the MUSP stores the general information of the algorithm in the strategy registers and the private information of the algorithm in registers i, has a decoder of the strategy register with matrix *M*, which receives the sync signal $\tau_2$ intended for organization transition functions in the general part of the algorithm, and a series of $DC_i$ decoders ($i$ = 1, 2, ..., n), on the output buses of which the matrices of microinstructions and the transitions $S_i$ are organized, which are intended for the organization of the transition functions s in a separate part of the algorithm.

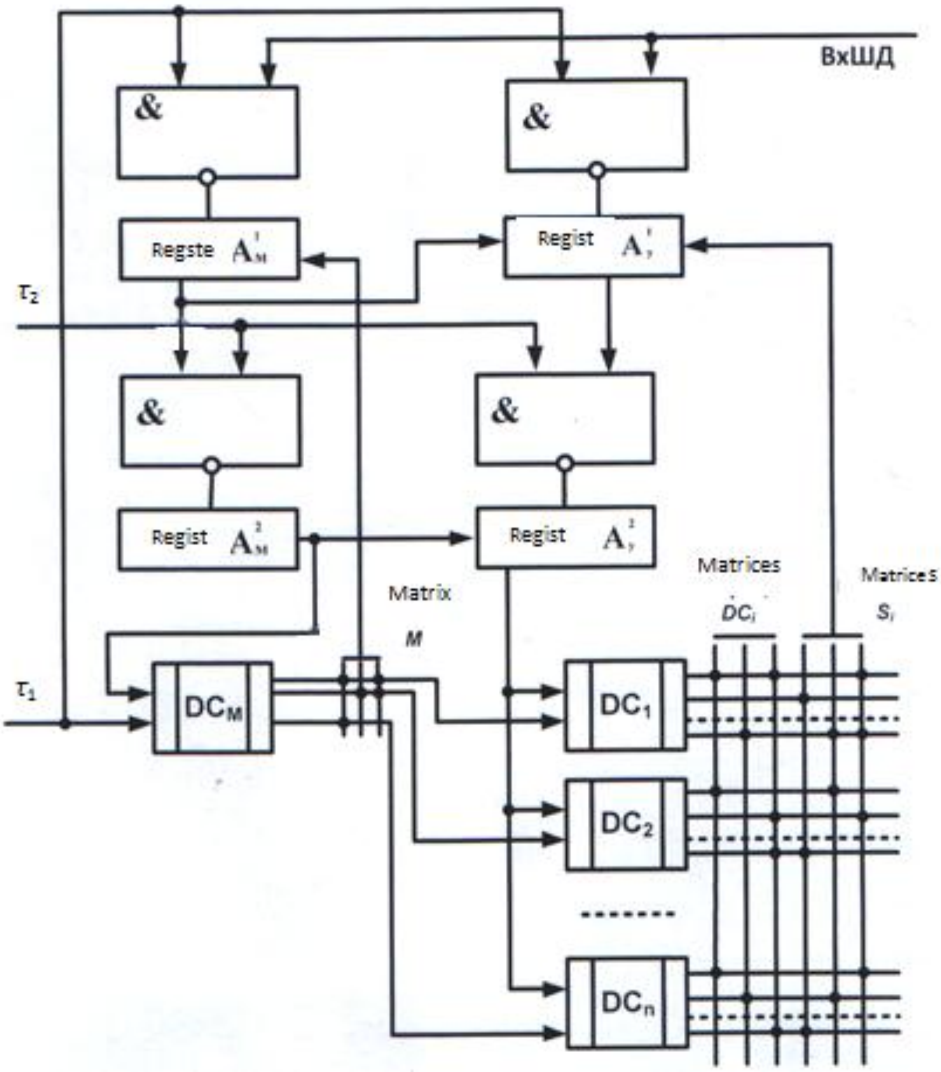
Functionally, in the control unit of the MUSP, an algorithm is used that changes over time (depending on the general input information) on its own reaction to those or other states of the registers, simultaneously processing general and private information.

## PROCESSOR [12; 26–27].

A processor is a device for automatically performing a sequence of operations that are caused by a problem solving program. It consists of two devices: operational and manager. The fourth generation processors also include internal (processor) memory devices and information input / output control devices.

## STRUCTURE OF THE PROCESSOR [12].

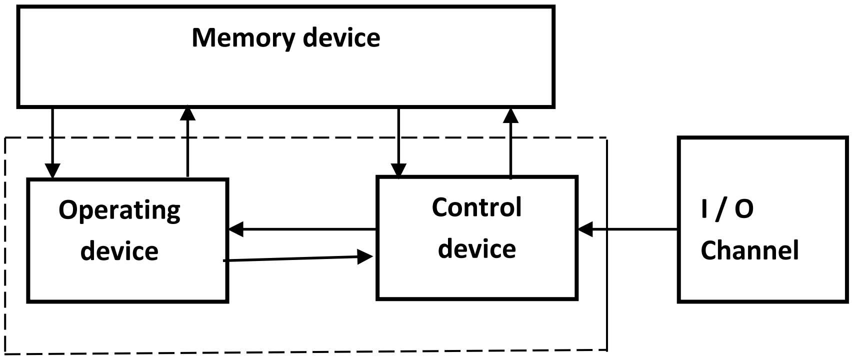
This is a combination of its functional blocks and the connection between them



.

## GENERALIZED STRUCTURAL SCHEME OF THE PROCESSOR WITH MEMORY ON TRIGGERS [12].

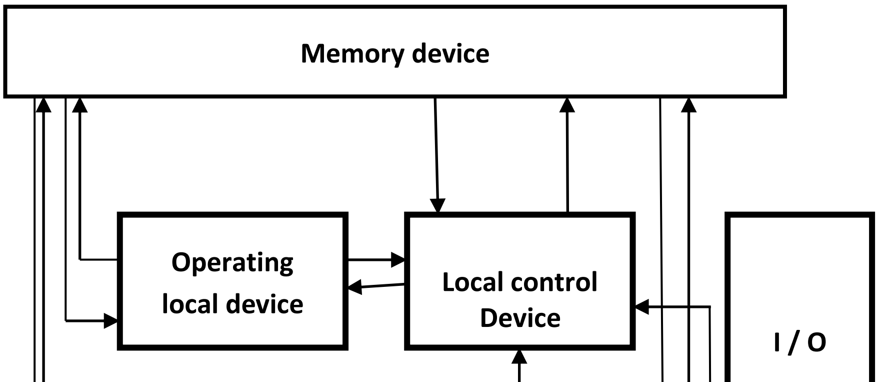
The classic generalized microprocessor block diagram with the principle of programmed control was proposed by Charles Babbage in the 19-th century. A generalized block diagram of a modern processor has this form.



## GENERALIZED STRUCTURAL SCHEME OF THE PROCESSOR WITH MEMORY ON MUSP [28].

A generalized block diagram of a microprocessor with the principle of hierarchical program control is proposed by L.F. Marakhovsky in 1996. The generalized block diagram of a processor with memory on MISP has the following form.
This structure becomes reconfigured.

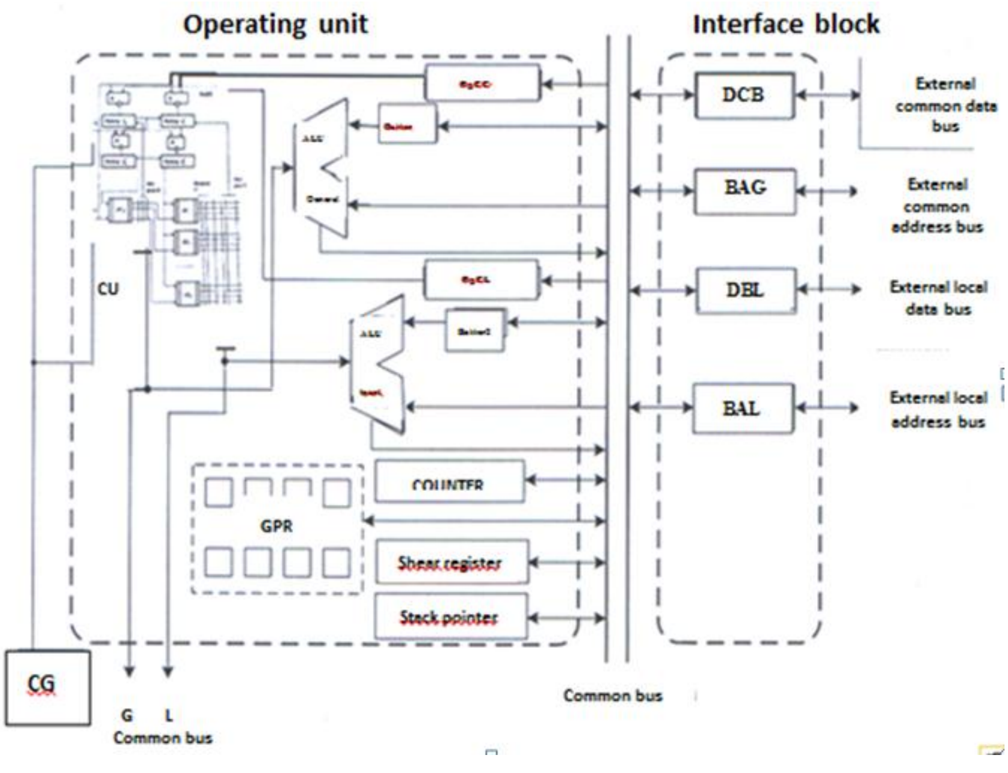
## COMPUTER WITH MEMORY ON TRIGGERS [26].



A computer consists of two components: a processor and RAM, which are interconnected by a system bus consisting of a data bus, a control bus and an address bus.

## COMPUTER WITH MEMORY ON MUSP [29–30].

The generalized block diagram of a computer at MFSP and MUSP has the form:



The computer becomes reconfigured. This is due to the fact that to change the command system, a reconfigured firmware control device with memory on the MISP is used. The basis of the architectural and structural organization of the reconfigured computer is the use of the principle of hierarchical program management.

## PRINCIPLES OF CONSTRUCTION OF RECONFIGURED PROCESSORS AND COMPUTERS [28–30].

The development of new principles for constructing reconfigured processors and computers on new elementary multi-level memory schemes allows you to memorize both common and private information simultaneously and change the structure of storing private information under the influence of common information, which makes it possible to increase the speed of processing hierarchical information.

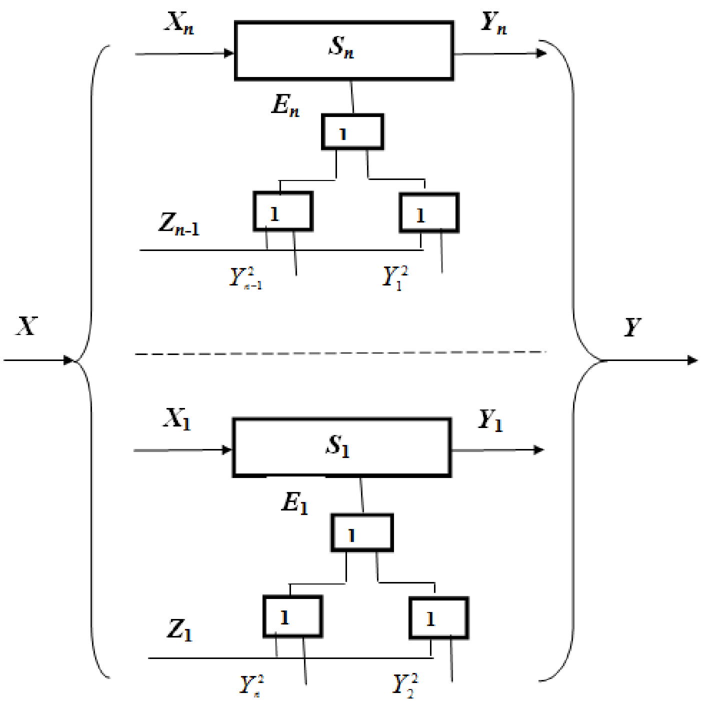
The new direction of building computers on MFIS and MUSP contributes to the progress of computing technology. It can be implemented on modern logic elements used in VLSI, FPGA, RAM, and can also affect the development of reconfigured devices, computers, computer systems and networks.

*MICROPROGRAMMING AND PROGRAMMING THEORY*

## COMPLEX OF CHARACTERISTICS OF SOFTWARE AND HARDWARE MEANS OF PROCESSORS AND COMPUTERS [4].

The set of characteristics of software and hardware make up the concept of the architecture of computers and processors.

## HIERARCHICAL ABSTRACT $F_I$ – AUTOMATON [4].

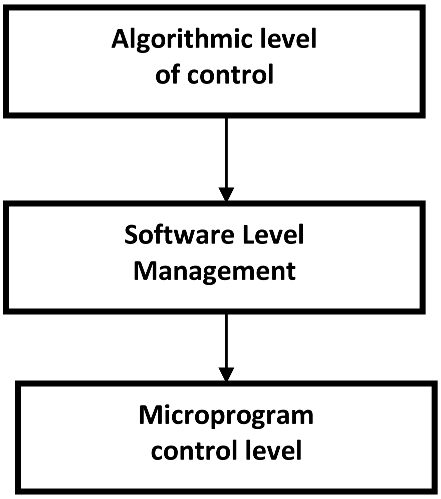


Each multifunctional $S_i$ sub-machine ($i$ = 1, 2, ..., $n$) of a hierarchical abstract $F_i$-automaton can be transferred from one state to another in parallel with other IA sub-machines. The operation of the $S_i$ sub-machine in a certain subset of states can be changed by the effects of the input letters of the information alphabet $X$ at the clock time t and the effects of the results of the work of other $S_i$ sub-machine with letters of the saving input alphabet $E$ at the moments of the internal clock Δ of automatic continuous time $T$.

A characteristic feature of IA $A$ is the possibility of interaction between $S_i$ sub-machines not only during the $t_i$ cycle, but also during the internal tact $\Delta_i$ of the automatic continuous time $T_i$. The memory of $S_i$ sub-machines represents a matrix structure, in which, during a clock $t_i$, under the influence of information $x_i(t)$ input signals, the $S_i$ sub-machine is able to change from one state to another in one state block (the row of the matrix structure of the automatic memory circuit). And during the internal clock $\Delta_i$, under the influence of the input signals saving $e_j(\Delta)$, the $S_i$ sub-machine can move from one state to another in one block $\mu_i$ (column of the matrix structure of the automatic memory circuit), that is, from a certain state of one block to a certain state of another block states.

## THREE LEVELS OF THE INTERNAL LANGUAGE OF THE COMPUTER [12].

In modern computers, there are three levels of internal language, which correspond to three levels of management: algorithmic, software and microprogram. Each of the levels can perform two main functions: serve as a universal means of displaying the input language (ie, the language in which the problem algorithm is formulated) and as means of interpreting some operators through others.

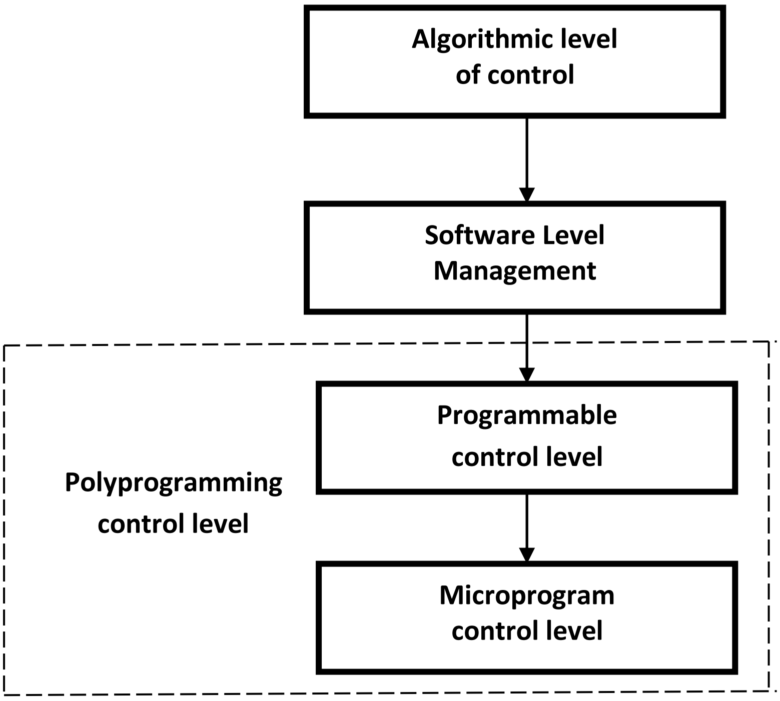

Algorithmic level of control
Software Level Management
Microprogram control level

At the same time, all control levels are in a certain hierarchical relationship, which allows the expression of a higher level operator through lower level operators.

The nature of the connections between the management levels, as well as the functions of each of them, most significantly reflects the features of the architecture and structure of the processors.

## FOURTH GOVERNANCE LEVEL [4–5; 29].

The principle of hierarchical program management proposed by Professor L.F. Marakhovsky, who breaks the control information into *n* levels, allows you to enter the fourth level of control of the microprogram, which is common to the microprogram level. Miliprogram and microprogram levels are combined into a polyprogram level, which provides processing of general and separate information simultaneously.


Algorithmic level of control
Software Level Management
Programmable control level
Polyprogramming control level
Microprogram control level

This improves the speed of information processing in the class of learning algorithms (and other reconfigured) that change over time under the influence of General information, your reactions to certain input words.

## THE PRINCIPLE OF MICRO-PROGRAMMABLE [24; 31].

The principle of microprogramming is implemented through inclusion in the structure of the processor special memory block for storing firmware. Firmware processors provide the programmer in addition to the command language effective language of micro-ops, which is basically a passive storage device (ROM). Along with this, the principle of microprogramming simplifies the process of development, modification and system change teams, and is also a tool of functional flexibility in the orientation of computers and processors to solve entire classes of problems.

## THE PRINCIPLE OF BUILDING POLYPROGRAM PROCESSORS [4-5; 29].

The principle of construction preprogramming processors is realized through the inclusion in the structure of the processor special memory block to MUSP to save General information of miniprogram. This unit provides additional functionality in the firmware of the processors in the direction of increasing modifications and changes to the system commands and in the process leads to the possibility of simultaneous processing of common and private information.

## THE PROGRAM [12].

Software for the solution of any task is a formal description of the algorithm in the computer, which is made in the form of a sequence of commands that control the process of solving the problem.

In the computer's memory is stored the processing program and data (set of numbers) that are processed by the micro-operations these commands in the processor. This approach is based on the principle of program control and the principle of conservation programs in the computer memory

## THE PROGRAMME TEAM [32].

Each team determines the machine action required to implement any one of the operation. That the processor is executing a program implementing the algorithm for solving the problem, is a sequence of operations carried out in the order given by the program.

## THE COMMAND SYSTEM [32].

The command system is the internal language of computers. It combines system operations and addressing system. In line with this, every word of command consists of two parts: operating (specifies the type of operation that is performed in one of the functional devices of the machine) and address (specifies the address of the memory cells of the memory block where the stored data codes and commands). The teams often introduce additional features that determine the specifics of the operation, the indication modification the command address, signs of the type of the storage device, and sometimes the pointers of the addresses of the following commands. The use of the combination of these features allows you to increase the functionality of the team.

**MICROOPERATION [32].**

Every operation consists of smaller micro-operations corresponding to a single elementary transformation under the influence of a certain functional of the signal (the macro).

**MICROPROGRAM [31-33].**

Firmware algorithmically describes a machine with memory triggers, seeing it functioning consistently every time *T* move him from one state to another during one external clock *T* the automaton discrete time. The firmware in a more General way is presented as autogram.

## JUSTIFICATION OF THE TERM "POLYGRAM" [5].

For the algorithmic description of memory automata on the MUSP, the term “polygram” is introduced.

1. The terms "microprogram", "autogram" and other methods of describing with memory on triggers are oriented, quite justified, on the description of transition functions in automata only when the input signal $x(t)$ is applied;

2. the terms “microprogram”, “autogram” and other methods of describing with memory on triggers are oriented towards the realization of storing sequential information in registers on triggers;

3. the terms “microprogram”, “autogram” and other description methods are not focused on the description of transition functions in memory machines on MUSP under the influence of the input word $p(T) = x(t)$, $e(\Delta)$ of automatic continuous time;

4. The polygram is focused not only on the implementation of storing hierarchical information in registers on the MUSP and converting the input information into output, but also by changing the region of the stored information by the input signal $e(\Delta)$.

## TERM "POLYGRAM" [5].

The term "polygram" combines the term "milligram", which remembers, processes and issues general information, with the term "autogram" or "microprogram", which processes and issues separate (private) information simultaneously (in parallel). The term “polygram” is associated with a circuit implementation of control on multi-level registers with a multi-functional memory organization system. The polygram is focused not only on converting the input information into the output, but also on changing the subsets of the $a_i$ states in which the automaton can function with a certain preserving e(Δ) input signal. This feature allows you to use the description of a polygram of hierarchical automata (HA) with memory on the MFIS and MUSP, which allow you to simultaneously process public and private information in one machine cycle $T$.

## DESCRIPTION OF THE STATE $a_k$ HA [5].

A polygram describes each state of $a_k$ HA as a combination of ai states of $S_i$ sub-devices (built on the MFIS).

$$a_k = \bigcup_i a_i .$$

## POINT OF THE POLYGRAM [5].

At each point of the polygram, the mode of operation of the $S_i$ sub-machine for one external clock cycle $T_0$ (machine clock) of automatic continuous time is described, which is simply used in the polygram as $T$ elementary $p_i(t)$ of the input words of the $S_i$ sub-machines.

$$R_k = \bigcup_i p_i .$$

## TRANSITION TO NEW STATE HA *A* [5].

Under the influence of the input word $R_k(T)$, HA A can move to the new state $a_s$ and produce one of the output signals $\mathrm{Y}_\mathrm{k}$, consisting of a set of output signals $\mathrm{Y}_\mathrm{i}$ of certain types of $S_i$ 1st, 2nd, and 3rd type substomats.

$$Y_k^1(t) = \bigcup_i Y_i^1(t);$$

$$Y_k^2(T) = \bigcup_i Y_i^2(T);$$

$$Y_k^3(\Delta) = \bigcup_i Y_i^3(\Delta),$$

where $Y_i^1(t), Y_i^2(T), Y_i^3(\Delta)$ the output signals are determined according to the functions of the outputs.

## OPERATION OF UNDER-AUTOMATS AT CONSTANTABLE *e<sub>j</sub>* INPUT SIGNALS [5].

With unchanged sets of $e_j$-preserving input signals, the $S_i$ substates operate in certain $\pi_i$ blocks of their states and, therefore, HA A also functions in the same time interval in certain blocks of their $\pi_k$ states.

$$\pi_K = \bigcup_i \pi_i$$

## UNAMBIGUOUS DEFINITION OF THE BLOCK $\pi_k$ STATES IN HA A [5].

The sets of $e_j$-saving input signals determine the $\pi_i$ blocks of states in which $S_i$ sub-machines operate, and the set of $E_k$-saving $e_j$ input signals uniquely determine the $\pi_k$ blocks of states in which the entire HA A operates.

$$E_K = \bigcup_i e_j .$$

## POLYGRAM POINT AT DETERMINED ELEMENTARY $R_I$ INPUT WORD [5].

The item of the polygram with the deterministic elementary $R_i$ input word is written in the following form:

$$\begin{aligned} KE.\, R_1 &\rightarrow Y_1^i - K_1 E_1, \\ R_2 &\rightarrow Y_2^i - K_2 E_2, \\ &\text{........................} \;, \\ R_m &\rightarrow Y_m^i - K_m E_m \;, \end{aligned}$$

where $K$ is the state of HA;

$E$- preserving input signal HA;

$R_j\,(j = \overline{1,m})$- elementary input word HA;

$K_j\,(j = \overline{1,m})$- the state of the HA, to which the transition from the state $K$ is performed when executing line $j$;

$E_j(j=\overline{1,m})$- preserving the input signal, in which the $K_j$ state is memorized;

$Y_j^i(j=\overline{1,m})$- output signal HA.

## DESCRIPTION OF THE LINE ITEM *KE* [5].

Each line ($R_j \rightarrow Y_j - K_jE_j$) of a *KE* polygram is described as $S_k$ lines:

$$\begin{gathered}
a_1e_1.\,p_{j_1} \rightarrow Y_{j_1}^i - a_{j_1}e_{j_1}, \\
a_2e_2.\,p_{j_2} \rightarrow Y_{j_2}^i - a_{j_2}e_{j_2}, \\
\dots\dots\dots\dots\dots\dots\dots\dots, \\
a_qe_q.\,p_{j_q} \rightarrow Y_{j_q}^i - a_{j_q}e_{j_q},
\end{gathered}$$

where $q$ is the number of lines $S_k(k=\overline{1,q})$, that are in the description of the line of the polygram;

$$j(j=\overline{1,m});$$

$p_{j_i}$ - elementary input string word;

$$S_k(k=\overline{1,q});$$

$Y_{j_i}^l$ - output line vector $S_k(k=\overline{1,q})$;

$a_{j_i}$ - state of the submachine $S_s$, to which the transition from the state $a_{s_i}$ when the line $S_k$ is executed;

$e_i$ - saving input signal.

Such a description of the behavior of the hierarchical automaton *A* in the *KE* state is the point of the polygram of the hierarchical algorithm of the automaton *A*.

The paragraph of the *KE* polygram, as we see from its description, has a hierarchical structure: it is first described in paragraph *KE* of the line of the polygram, and then the lines *ae* of each line of the paragraph *KE* of the polygram.

## THE PROCESS OF TRANSFORMING INFORMATION IN HA *A* [5].

The change in the internal structure of memorizing states in $S_i$ sub-automata and in general in HA *A*, as well as the transformation of incoming external information in HA *A*, are just two interrelated parts of the same information conversion process in HA *A*.

## GENERAL POLYGRAM LINE [5].

A polygram row of the form

$$KE.\ R_j\ E_i \rightarrow Y_j - K_jE_j,$$

is called a common line, emphasizing by this term that this line represents a transition in two variables $R_j$ (x(t)), $E_i$ (e(Δ)) and all three vectors of the above command.

## TRANSITION LINE THIRD KIND [5].

In addition to common lines in a polygram, there may be lines in which only one of the vectors is explicitly represented, while other command vectors are omitted. The vector skip is depicted with *Ø*. If the components of a vector are missing, instead of a vector, its present component is represented.

View string

$KE.\ E_i \rightarrow Y_j - K_jE_j$ with $E_i \neq E_j$

called the third order transition line, since it describes the transition performed only by the components of the vector $E_i$ of the stored input signals.

## LINE OF PROBABLE TRANSITION OF THE FIRST TYPE [5].

View string

$KE.\ R_P\ E_i \rightarrow Y_j - K_jE_j$ with $E_i = E_j$

called the probabilistic transition line type 1, since it describes the transition carried out under the influence of the components of the vector $R_P$ of the input signals, which set the output state, which is not remembered.

**LINE OF PROBABLE TRANSITION OF THE SECOND TYPE [5].**

View string

$KE.\ R_j\ E_B \rightarrow Y_j - K_jE_j$

called the probabilistic transition line type 2, because it describes the transition, carried out under the influence of the vector component $E_B$ of the probabilistic input signals.

**LINE OF FUZZY TRANSITION [5].**

View string

$KE.\ R_P\ E_B \rightarrow Y_j - K_jE_j$

called the string of fuzzy transition, because it describes the transition, carried out under the influence of two components of the vector $R_P$ and $E_B$ input signals.

**TRANSITION LINE [5].**

View string

$KE.\ R_j \rightarrow \varnothing - K_jE_j$,

named simply as a transition line, characteristic for automata of the 1st and 2nd kind, since it represents only the transition to the next state $K_jE_j$ without the output signal vector $Y_j$, and also without the transition component $E_i$.

## OUTPUT LINE [5].

View string

$$KE.\ R_j \rightarrow Y_j - \varnothing,$$

named by the output line, since it does not represent the transition to the state vector $K_j$ and the vector $E_j$ which does not preserve the new state is not represented.

## LINE OF *C*-VIEW [5; 30].

If the vectors $E_i$ and $E_j$ coincide in the line, they can be omitted. The string then (for example, common) takes the form

$$K.\ R_j \rightarrow Y_j - K_j,$$

and the vectors $E_i$ and $E_j$ ($E_i = E_j$) in this case are implied and define the well-defined block $\pi_j$ ($K$, $K_j \in \pi_j$) of the states to be remembered. Such a line is called C-type and is used when describing the microprogram or autograph [30] of automata the Mealy, Moore and C-automata that use triggers in memory as registers whose state is saved with one $e(\Delta)$ input signal. Such a description of the firmware once again emphasizes that the Mealy, Moore, and C-automata machines are a subset of the Marakhovsky multifunctional automata, on the basis of which the polygram describing the HA $A$, based on the automaton memory circuits is considered.

## FAILURE STABILITY SYSTEMS HA *A* [5].

The above-mentioned components of the vectors at the same time in the external clock cycle T can initiate various operations in controlled objects, as well as the functional shutdown of one of the sub-machine $S_i$ HA, while determining its uselessness or erroneousness in operation. This property is important when creating fault-tolerant digital devices [8].

An HA system is considered to be fault-tolerant or insensitive to faults if its organization provides for the elimination of defective $S_i$ sub-machines from the operating region of an HA through the use of hardware, information and algorithmic redundancy from the operating area.

The part of HA A intended for processing general information is called an automaton of the strategy HA or abbreviated ASHA $A_M$. The structure of HA and is hierarchical and can be fault tolerant.

## OPPORTUNITY OF THE POLYGRAM [5].

The polygram allows describing the HA *A* as a whole from the general standpoint of the functioning of hierarchical systems with parallel execution of the branches of the algorithm with sub-machine $S_i$ with parallel execution of the algorithm by the strategy automaton processing the general information.

Thus, the polygram describes not only the tasks of the $S_i$ sub-machines, but their inter-level interaction in HA *A*, due to the saving input signals from the strategy's sub-machines.

## CORRECT POLYGRAM [5].

A polygram is called correct, which, when implemented in an automaton, synthesized by formal methods, leads to the functioning of a reliable operation of the device, corresponding to its intended purpose.

## SPEED V SOLUTIONS OF PROBLEMS IN THE SEQUENTIAL PROCESSING OF GENERAL AND PRIVATE INFORMATION [22].

The speed V of solving problems in the case of sequential processing of general and private information during program control largely depends on the number of calls to the memory device for rebuilding a reconfigured system, which in the first approximation is determined by the formula:

$$V = \frac{1}{k_1 t_1 + k_2 t_0},$$

where $t_i$ is the time to select a word from memory when performing an operation in a separate device;

$k_1$ is the average number of accesses to a memory device when performing an operation in a separate device;

$k_2$ is the average number of calls to the memory device for restructuring the processing algorithm;

*$t_0$* is the time to select a word from memory when accessing a memory device to restructure the processing algorithm.

## THE TYPE OF COMMANDS IN A COMPUTER WITH MEMORY ON TRIGGERS [12].

In accordance with the specifics of the numerical methods of the program are recorded in a storage device (memory) as a sequence of four, three, two or unicast commands.

To write the code of each command, a word of standard length (bit depth) is given. In memory, commands are located in consecutive memory cells (one after another).

Type of three-address command

| | | | |
|---|---|---|---|
| Operation code | Cell address code of the 1st operand | Cell address code of the 2nd operand | Cell address code of the 3rd operand |

Unicast command type

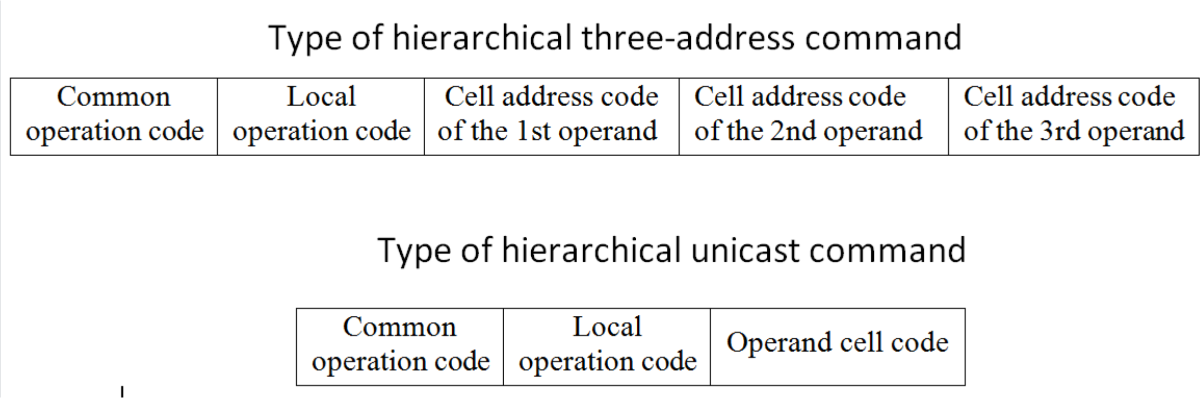
| | |
|---|---|
| Operation code | Operator Cell Address Code |

Although the address (the number of the memory cell where it is stored) is not indicated in the command (with the exception of the four-address address, in which the fourth address is the address of the command), it is determined by the fact that program instructions are entered into memory cells in sequence. The address of each next command is calculated from the address of the previous command, by adding 1 to it.

## TYPE OF COMMANDS IN A COMPUTER WITH MEMORY ON MUSP [33].

For devices that use MUSP as memory, such commands are fundamentally inappropriate. This is due to the fact that they are designed for sequential processing of information. In parallel processing of information, where the general and private information is processed in parallel, an additional cell is required in the address of the command that would define the code of the operand of the general information.

In this regard, it is proposed to enter into the command codes an additional code for the common operand.



Type of hierarchical three-address command

| Common operation code | Local operation code | Cell address code of the 1st operand | Cell address code of the 2nd operand | Cell address code of the 3rd operand |
|---|---|---|---|---|

Type of hierarchical unicast command

| Common operation code | Local operation code | Operand cell code |
|---|---|---|

It should be noted that the private operation code processes the operand information, and the common operation code changes the private operation code and therefore it does not need to use operands for processing.

## METHODS OF DESCRIPTION OF HIERARCHICAL PROGRAM MANAGEMENT [33].

Methods for describing hierarchical program management in the form of a polygram with two levels of control, although in principle there may be more.

The first level is determined by the firmware that processes the incoming information in the form of operands. The second level is defined by the milprogram, which changes the structure of the firmware. The second level changes the firmware commands in one clock cycle of machine time *T*. Such an approach allows the processor to switch from one program to another for one clock cycle of computer time, which is impossible in modern programming languages

## SPEED V SOLUTIONS FOR PROBLEMS WITH SIMULTANEOUS TREATMENT OF GENERAL AND PRIVATE INFORMATION [4].

The speed V of solving problems while simultaneously processing general and private information at the polygram level of control can be calculated in the first approximation by the formula:

$$V = \frac{1}{k_1 t_1},$$

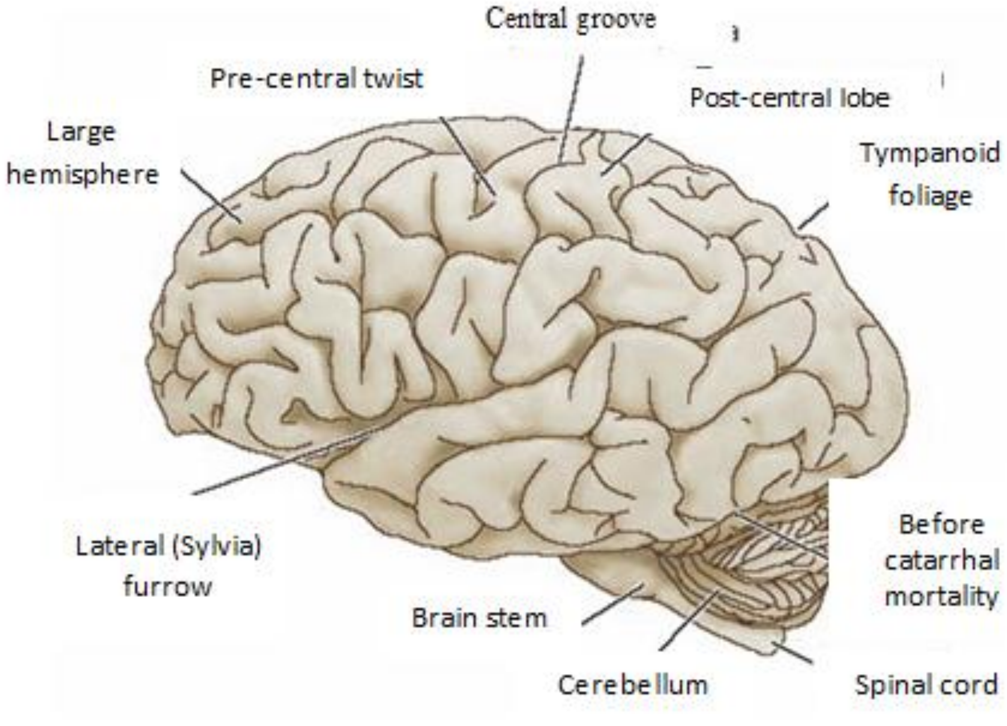
where $t_i$ is the time to select a word from memory when performing an operation in a separate device and simultaneously retrieving a word to rearrange the processing algorithm;

$k_1$ - the average number of accesses to the memory device during the operation.

## *MODEL OF DIGITAL ARTIFICIAL NEURON*

### HUMAN BRAIN [33]

This is the most complex machine in the known universe.



The human brain

The generalized structural diagram of the human brain consists of symmetrical subsystems that are associated with external receptors of the human body, such as: eyes, ears, mouth, body skin, as well as aura receptors around the human body, consisting of 8 levels. Subsystems of the human brain consist of the cerebral hemispheres, the occipital part of the brain, the visual cusps, the diencephalon, the medulla oblongata, the midbrain, the pons, the cerebellum, and the spinal cord, as well as the descending and ascending pyramidal nerve trunks (main flow).

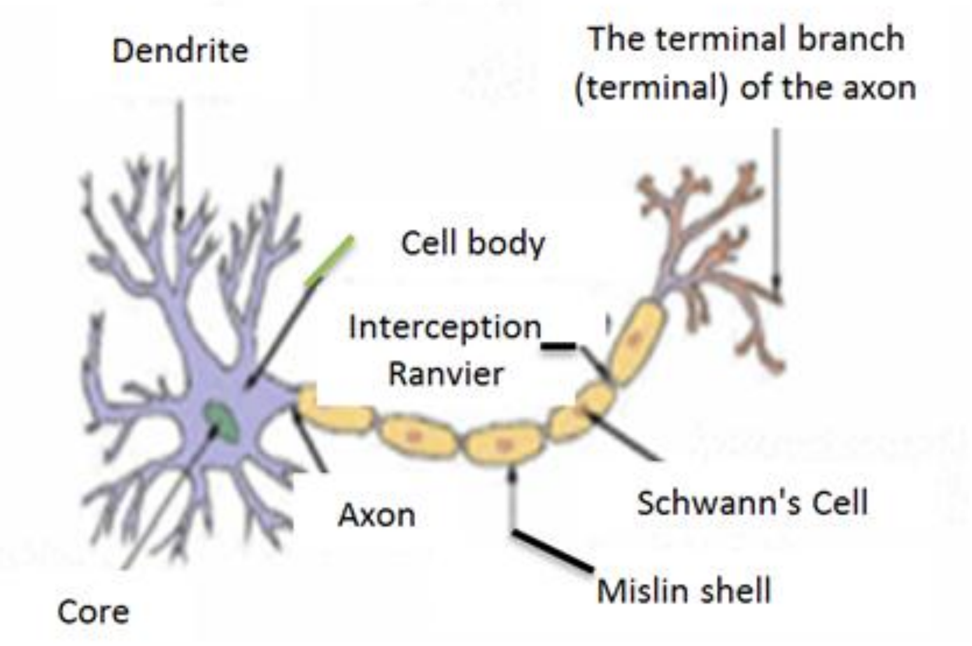
## NEURAL NETWORK (biological neural network) [39]

This is a collection of neurons in the brain and spinal cord of the central nervous system (CNS), and the ganglion of the peripheral nervous system (PNS), which are connected or functionally integrated in the nervous system, perform specific physiological functions.

A neural network consists of a group or groups of chemically or functionally connected neurons. One neuron can be connected with many other neurons, and the total number of neurons and connections in the network can be quite large. The place of contact of neurons is called a synapse; a typical synapse is an axodendritic chemical. The transmission of pulses is carried out chemically by means of mediators or electrically by passing ions from one cell to another.

## BIOLOGICAL NEURON [38].

The central nervous system has a cellular structure. The unit is a nerve cell, a neuron. The neuron has the following basic properties:



The typical structure of a neuron

1. Participates in the metabolism and dissipates energy. Changes the internal state over time, reacts to input signals and generates output effects and therefore is an active dynamic system.
2. Has many synapses - contacts for information transfer.
3. A neuron interacts by exchanging electrochemical signals of two types: electrotonic (with attenuation) and nerve impulses (spikes) that propagate without attenuation.

Biological neuron contains the following structural units:

*The body of a neuron* [49] has a nucleus containing chromosomes encoding cell genetic information using DNA. The nucleus is surrounded by a cytoplasm of complex structure. The bodies of the neuron and the capillaries of the blood vessels constitute the so-called gray matter, which is really gray.

*Cell body (t)* - soma: contains the nucleus (s), mitochondria (provide the cell with energy), other organelles that support the vital activity of the cell.

*Dendrites (d)* - input fibers, collect information from other neurons. Activity in the dendrites changes smoothly. Their length is usually no more than 1 mm.

*Dendrites [49]* are in fact an extension of the area of the neuron, which makes it possible to increase the amount of information received by the neuron. The number of dendrites is different for different types of neurons - up to 20. Each dendrite can be very branched. In length, they can also vary - from a few microns to millimeters.

*Membrane* - maintains a constant composition of the cytoplasm inside the cell, provides for the conduction of nerve impulses.

*Cytoplasm* is the internal environment of the cell. It differs in the concentration of K +, Na +, Ca ++ ions and other substances in comparison with the extracellular medium.

*The axon (a),* one or none of each cell, is a long, sometimes more than a meter, output cell nerve fiber. The impulse is generated in the axon knoll (ah). Axon provides impulse conduction and transmission of effects to other neurons or muscle fibers (mV). Toward the end, the axon often branches.

*Each neuron [49]* has only one axon, whose length can be from several microns to a meter, depending on the type of cell. Most axons have branches (collaterals). With the help of endings on them (end feet), the axon is connected to other cells.

*Synapse (c)* - the place of contact of nerve fibers - transmits excitation from cell to cell. Synapse transmission is almost always unidirectional. Presynaptic and postsynaptic cells are distinguished in the direction of impulse transmission.

*Synapses [49]* (the term of Sherrington - from the Greek synapsis - connection) represent some kind of connecting formations between axon endings and other neurons. Synapses can communicate with any parts of another neuron — the body of a neuron, dendrites, axons, or synapses of other cells. They contain chemicals that, when involved, can affect the activity of other cells.

*Schwann cells (Swiss).* Specific cells consisting almost entirely of myelin, an organic insulating substance. The nerve fiber is wrapped tightly with 250 layers of myelin. Uninsulated nerve fiber sites between Schwann cells are called Ranvier intercepts (RH). Due to myelin isolation, the speed of propagation of nerve impulses increases 5 * 10 times and reduces the energy costs of conducting impulses. Myelinated fibers are found only in higher animals.

In the human central nervous system there are from 100 to 1000 types of nerve cells, depending on the degree of detail chosen. They differ in the dendrite pattern, the presence and length of the axon and the distribution of synapses around the cell. Cells are strongly interconnected. A neuron may have more than 1000 synapses. Similar in function cells form clusters, spherical or parallel layered. There are hundreds of clusters in the brain. The cerebral cortex is also a cluster. The bark thickness is 2 mm, and the area is about a square foot.

*Neurons are surrounded by glial cells [49]* - glia (from the Greek glya - glue). Glia is composed of fatty lipids, which give - like milk - white color. The ratio between the mass of neurons and the mass of the glial cells surrounding them turned out to be a very important indicator for the implementation of complex nervous processes. Interestingly, with evolution, the ratio of glial tissue of the cortex to the mass of its nerve cells increases more and more, and in humans it turns out to be many times greater than in other mammals.

## FUNCTIONAL PROPERTIES OF THE BRAIN AND NEURON [34–36].

We are not yet fully aware of what a neuron is.

1. a neuron is a complex automaton that has a hierarchical, approximately eight-level structure, the maximum area of perception of which is limited to approximately the number of elements equal to 1121, i.e. the number of states;

2. on the one hand, the neuron cell has an internal core that stores information and is able to perceive information through two channels: stimulating and inhibiting (tuning), and on the other hand it has the ability to rebuild and establish connections with other neurons, which creates corresponding patterns and images with which the person operates;

3. The brain has a high functional reliability of work (with damage to the brain by about 50%, it continues to function normally and the person is able to perform skilled work!). In addition, the neuron cells partially break down and are replaced by new ones, which indicates a strict control over the working capacity of a living cell, and when it is found to be partially inoperable, it is replaced;

4. A neuron has a large number of exits, reaching approximately 10,000, which allow it to communicate with other neurons, sometimes located at a sufficiently large distance. At the same time, it is suggested that the active signal from a neuron when it is excited goes targeted to a specific neuron, and not immediately to all the neurons with which it has a connection. Address communication, again, it is assumed, may inhibit input signals (other input signals that determine the direction of the active signal are not yet known!);

5. All living matter - from a virus to a person, is equipped with a reliable transmission conveyor, a hereditary information thesaurus. It consists of, extremely rational, communication systems, with incredible accuracy and reliability of information

transfer, fantastic protection from interference. And all this, in unbelievable, small volumes. Universal and communication language. It is the same for all living organisms;

6. while studying the processes of signaling activity, the most important phenomenon, which is of paramount importance for the processing of information, is revealed. The senses are arranged in such a way that the reaction to a long-acting, repetitive signal is extinguished. At the same time, very sharply, a new signal is perceived, different from others. Thanks to this feature, the system eliminates new information from the stream of messages, signals. New information appears, above all, with changes in the environment. The ability to form temporary connections, under the action of even insignificant stimuli, is of crucial importance for the cognitive activity of the brain, in the study of the laws of the surrounding world;

7. According to the first signal system, any living organism receives signals from the outside world, about the state of the environment, from Space, through wave radiation. The second signal system (speech) is the invention of the human brain. It depends entirely on the first, receiving, interpreting and abstracting signals. A person can transmit and receive processed information only in his own environment, where the code of communication is generally accepted: words, language, speech. Communicate with the system, the same level of development, the human system can only learn the language of this system.

Thus, the second signaling system is a superstructure above the first.

8. It seems that, nevertheless, all the experience accumulated by a person, all information obtained during life, does not pass without a trace, but is recorded and stored at a subconscious level and is transmitted, genetically, to descendants. Nature bothers about reliable duplication of brain functions, about protecting the information thesaurus of man. Nature bothers about reliable duplication of brain functions, about protecting the person’s information thesaurus. Nature reliably controls and protects the brain.

9. The transmission of electrical impulses [49] through synaptic connections is the transmission of information to and from the brain. It occurs chemically using two categories of substances: peptides and neurotransmitters. Neurotransmitters act locally, affecting only their immediate environment (serotonin, dopamine, acetylcholine). Peptides, by contrast, can act over long distances.

Summing up, V. Gagin writes [21]: “When considering the development of living and inanimate Nature, it is clear that the Golden Section law, the adaptation of living organisms, the hierarchy of information processing in the human brain, the high functional reliability of the brain (in the case of brain damage) 50% of it continues to function!) And the genetic memory of many generations of people. "

## ARTIFICIAL NEURON [4; 35; 37].

The building material for the brain is a neuron. An artificial neuron is used to build neural networks or a neurocomputer, which has two input beams of signals: excitatory and inhibitory neurons, the ability to rearrange its information storage structure and allow neurons to communicate with other neurons in a certain way, having thousands of output signals. The function of the output signals of the neuron is called the activation function or the trigger function, the transfer function. The result is sent only to one output.

## RELATED INFORMATION [35].

Information mediated by human labor is called related information, due to the fact that it is accumulated in the human brain through its training.

The primary flow of related information provides the livelihoods and survival of each person as an element of the social system, and the secondary flow of related information increases the efficiency of the primary information flow and ensures the survival of the system as a whole.

## THE CONDITION OF INFORMATION BALANCE SYSTEM [35].

The state of the system, in which the information flow created by it contains equal amounts of functional and structural information, is called the state of information balance of the system.

## FREE INFORMATION [35].

Free information is called information due to the fact that it is synthesized by the human brain when it perceives the reaction of the external environment to human-made actions.

## INFORMATION CAPACITY ELEMENTAL SYSTEMS [35].

The system of N elements in the state of information equilibrium has a structure consisting of n hierarchical levels. The amount of information that elements of each hierarchical level of the system create is a constant value. The amount of information that each element of the system contributes to the information flow is called the information capacity of the system element. In the hierarchical structure of the system, all its elements are ordered from the highest (private) to the lowest (general) level in descending order of their information capacity.

## NUMBER OF HIERARCHICAL LEVELS OF THE OPERATIONAL MEMORY OF THE BRAIN [35].

The number of hierarchical levels of the information structure of the operational memory of the human brain lies in the range from 7 to 8.

## ENTROPY [41].

Entropy is a measure of the system's freedom: the more entropy, the more states available to the system, the more degrees of freedom it has.

## STATIC ENTROPY [35].

The entropy represented by the amount of information generated on average by one element of the system is called static entropy, which has the dimension nit / element.

## ENTROPY DYNAMIC [35].

The entropy represented by the amount of information generated on average per unit of time is called the dynamic entropy, which has the dimension of nit / time if the information is measured in natural units

## INFORMATION CHARACTERISTICS OF THE HUMAN BRAIN [35].

This characteristic determines the upper limit of the number of levels of complexity of images that a person can perceive and with which he can operate in the process of thinking. The upper limit of the amount of information carried by images of each level of complexity is approximately 735.1 nits / element. This constant sets the limits between which a person is still able to establish connections. This restriction, in our opinion, determines the assertion of the theorem of Gödel on the incompleteness of deductive systems [40], and also characterizes the properties of the short-term memory of the human brain.

This consequence implies that the information flow circulating in the RAM cannot exceed 5586.76 nits / element.

## NUMBER OF ATTRIBUTES THAT CAN BE OPERATED BY MAN [35].

The number of signs that a person can operate in the process of perception and thinking is 54 nit / element. This number is divided into an equal number of structural and functional features, the numbers of which are 27 nit / element.These results were confirmed in the study of the social structure of the working class: “Of the 54 signs in the questionnaire, 27 signs (properties) were selected that seemed most important for the classification of individuals ...”.

## “MENTAL ABILITIES” OF A MAN [35].

A person cannot develop his mental abilities in the way he can develop muscles. The only thing he can do is to improve his art in the application of mental abilities. This is an important distinction that a layman does not distinguish. The essence of this improvement is the acquisition of information of certain levels of complexity and the development of methods for its logical processing.

## LIMITATIONS OF HUMAN MEMORY [35].

The total number of images that make up the information flow, capable of circulating in the operational (short-term) memory of the human brain is approximately defined up to 1121, and the information capacity of the threshold for both functional and structural information is approximately 27. In the alphabets of most developed modern languages (including the section between letters) is the number 27.

The knowledge of the world by man is limited within arbitrarily intersecting areas not only by the volume of short-term memory, but also by the volume of long-term memory.

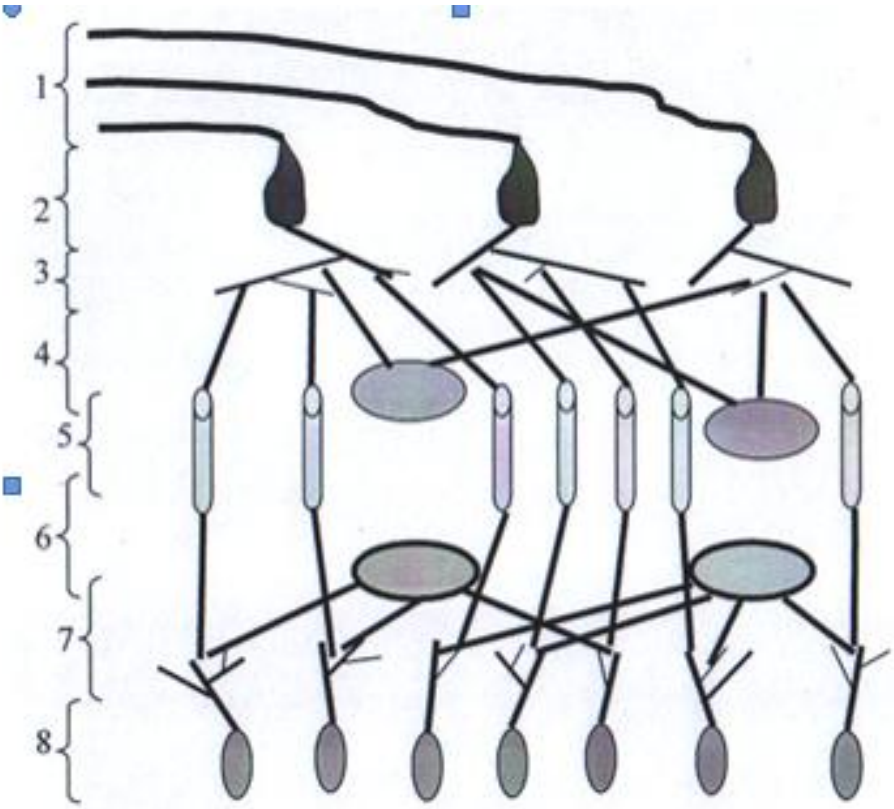
## PROBLEMS OF CREATING A MODEL OF THE HUMAN BRAIN [4; 35: 42–44].

The human brain has several advantages over all technical-cybernetic devices for such important properties:

1. The input signals from the external environment, affect the eyes, ears, body, on the taste of food, etc. In this regard, to create a model of the human brain, appropriate sensors should be developed and preferably their output signals should be best represented in digital form for better interface with digital devices.
2. Information that comes from the external environment is summarized. This is seen in the example of the eye. There are

about 18–20 million receptors in the eye, and about 72 thousand cones that summarize the visible information through the eye receptors. That is, information is compressed approximately 256 times already at the second level. This problem of data compression is important to understand and solve technically.

In fig. depicts the biological scheme of the horizontal organization of information compression and, importantly, 8 levels are shown, which corresponds to the levels of the human brain.

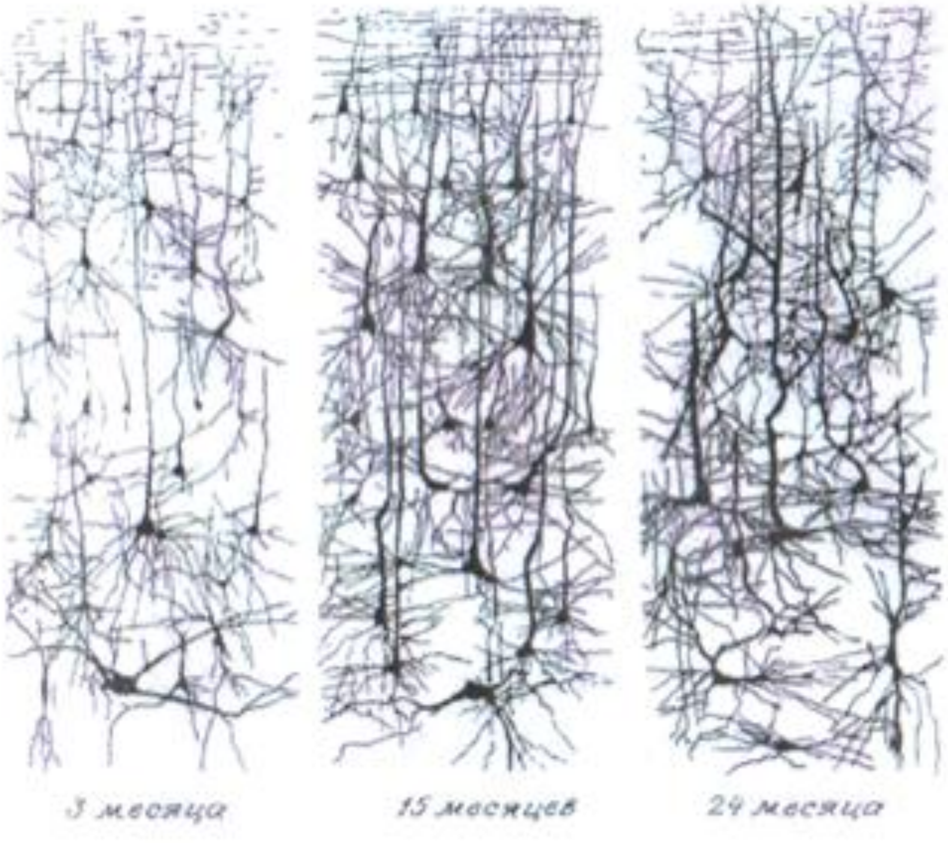


3. It is necessary to consider the constantly natural expanding connection between the neurons of the human brain.

4. This image shows how connections between a child from three months to two years gradually begin to be established, which are

necessary for generalizing the obtained (expanding) information, building appropriate patterns and models reflecting the real world of an individual person. Thus, the problem is to create opportunities to expand the connections between artificial neurons while increasing information or generalizing it.

5. The human brain has between 14 and 20 billion neurons. This is a fairly large structure in terms of the number of neurons, which is difficult to physically create at the present stage of technological development, and even more so to manage it. The talented mathematician Frank Plampton Ramsay proved that complete disorder is impossible in such large structures as the human brain, the



Universe, etc. Each sufficiently large set of numbers, points, or objects necessarily contains an ordered structure. Work in this direction confirmed this important result. However, the problem of creating ordered structures in the human brain model remains.

5. In the brain there is no computer, logical theories, positional number systems, but only its own logic for obtaining information, compressing information, choosing the path of communication with other cells, to summarize this information. Calculations, reasoning, number systems and any other algorithms are derived from those models that have already been generalized and are of interest to a person, as N. Nikitin writes. in an interesting work "The control logic of the cell" [43].

6. In many ways, the work of the brain, as it is customizable, affects the weather, mood, and music, and programs on TV and other unusual situations, and often dreams. The whole range of these external influences injected into an artificial model of the brain is now considered unreal.

7. If it were possible to create a machine that would exceed the human brain, and the person was unable to understand it, then he would try to "fix" it or destroy it.

## NEURONS [45].

Neurons (from the Greek. Nėuron) - nerve cells, structural and functional units of the nervous system. The neuron consists of the body and the processes departing from them - relatively short dendrites and a long axon. Neurons conduct nerve impulses from receptors to the central nervous system and from the central nervous system to the executive organs. Neurons interact with each other and with the cells of the executive organs through synapses. Some synapses cause depolarization of the neuron, others - hyperpolarization; the first ones are exciting, the second ones are braking. Usually, stimulation of a neuron requires irritation from several excitatory synapses.

Dendrites - as a rule, short and highly branched processes, which serve as the main site of formation of excitatory and inhibitory synapses affecting a neuron (different neurons have a different ratio of the length of the axon and dendrites). A neuron can have several dendrites and usually only one axon. One neuron can have connections with 20 thousand other neurons.

The place of excitation generation in most neurons is the axonal mound - the formation of an axon from the body at the site of axon detachment. For all neurons, this zone is called a trigger.

## DIFFERENCES BETWEEN BIOLOGICAL AND ARTIFICIAL NEURON [45].

Neural networks built on artificial neurons reveal some signs that make it possible to make an assumption about the similarity of their structure with the brain structure of living organisms. However, even at the lowest level of artificial neurons, there are significant differences. For example, an artificial neuron is an inertialess system, that is, a signal at the output appears simultaneously with the appearance of signals at the input, which is quite uncharacteristic of a biological neuron that has a memory. From the informational characteristics of the neuron of the brain of living organisms we select:

1. input signals to a neuron consist of two types: stimulating and inhibiting;
2. the neuron remembers the information;
3. output signals of a neuron, depending on the intensity of the functional load, form one or another type of fiber, i.e. has a certain functional direction.

## PRINCIPLES OF BUILDING THE SYSTEM OF THE ARTIFICIAL BRAIN [4].

1. The system is heterogeneous - the principle of diversity is used, but these inhomogeneities are connected by a consistent interaction.
2. The system is multifunctional - and each part of it is capable of performing not only its own functions, but also duplicating the functions of other parts of the system.
3. System management functions - duplication of control commands is required, the presence of a separate “feedback”, independent control over the execution of control commands.
4. The presence of one System Management Center is fraught with the consequences of its failure - another reserve Center is needed.
5. The system is obliged to ensure its sustainable development, self-learning, resilience, to have the ability to adequately respond

to external influences, to possess the necessary "safety margin", or "unsinkability" of the System.

6. In the event of the destruction of a part of the System, the possibility of its restoration (regeneration) is necessarily provided for, and, as an extreme case, if the entire System is destroyed, its disaster recovery (Phoenix effect) requires a separate Disaster Recovery Center (Phantom).

7. The Main Condition - all this should be described by a Single Language, which does not allow double interpretation. ”

8. All these principles and conditions are feasible subject to availability:

- Primary and Backup Mathematical Systems;
- Primary and Backup programming and management languages;
- Complete and duplicated interconnection between the System Blocks with the Svoy-Alien recognition system;
- Independent Analysis and Prediction Centers.

## MODELS OF ARTIFICIAL NEURON ON CIRCUITS OF AUTOMATIC MEMORY [4].

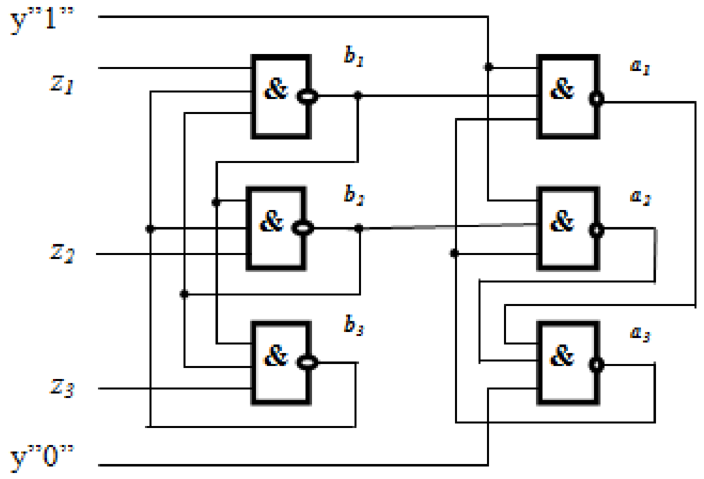
The most acceptable device at the present time for building a digital model of an artificial neuron is the open structure of multifunctional memory circuits with a matrix structure of state memory. It, like a neuron, has two sets of input signals: establishing x(t) (exciting) and saving e(Δ) (inhibiting). The excitatory input signals of a biological neuron from several other neurons, which allow accumulating (summing up) information and overcoming its threshold for outputting an active signal, can be completely replaced by a digital coincidence circuit from several outputs of other neurons. Such coincidence schemes, of which there can be more than one, could be implemented on conventional AND-OR schemes. The MFIS is described by a structural digital language, which allows the developer to determine the main characteristics of the memory circuit and the required characteristics for constructing an automaton strategy for the construction of MUSPs that have a semi-closed structure using known formulas.

MUSP is a digital artificial neuron circuit (CIN), which allows the use of deterministic, probabilistic and fuzzy transitions, inaccessible to triggers. The CIN self-monitors its performance when used in conjunction with machines of the 4th kind to detect catastrophic failures in memory circuits.

The proposed prototype of the CIN neuron has the possibility of restructuring the structure of the memorized states when the general information changes in the $A_M$ strategy automaton.

## FUNCTIONAL SCHEME OF THE MINIMAL CIN MODEL WITHOUT CONTROL SCHEME [4; 46].

The choice of the CIN, where the MFIS is described by the symbol number 21, and the $A_M$ strategy automaton by the symbol number 111, is not accidental for the realization of the CIN.



The ratio of the number of elements of the MFIS group, which is associated with the three elements of the $A_M$ strategy automaton, has a value of 0.6 (6), which is close to the number φ ≈ 0.618, characteristic of the "golden" section. However, this does not limit the developer of the CIN to design an arbitrary structure on the MISP, consisting of the MFIS and the $A_M$ strategy automaton.

## REGISTER FOR CIN [4].

The register consisting of eight bits of the CIN with control schemes for detecting catastrophic failures can be represented by analogy with an axon in a biological neuron. Such a register can have 68 (more than 46,600) states that can be transmitted by a code for exciting the corresponding CIN.

## MEMBRANE QIN [4].

Traditionally built on an adder, the inputs of which receive signals from artificial neurons with corresponding weights and when the threshold is reached, the neuron itself is excited. In our case, instead of an adder, it is proposed to use a reversible counter, which can sum the exciting input signals with specific weights and subtract the decelerating input signals with the corresponding weights. When the counter reaches its maximum value, the neuron is energized and after issuing the excitation signal (spike) is reset.

## GOLDEN PROPORTION [41].

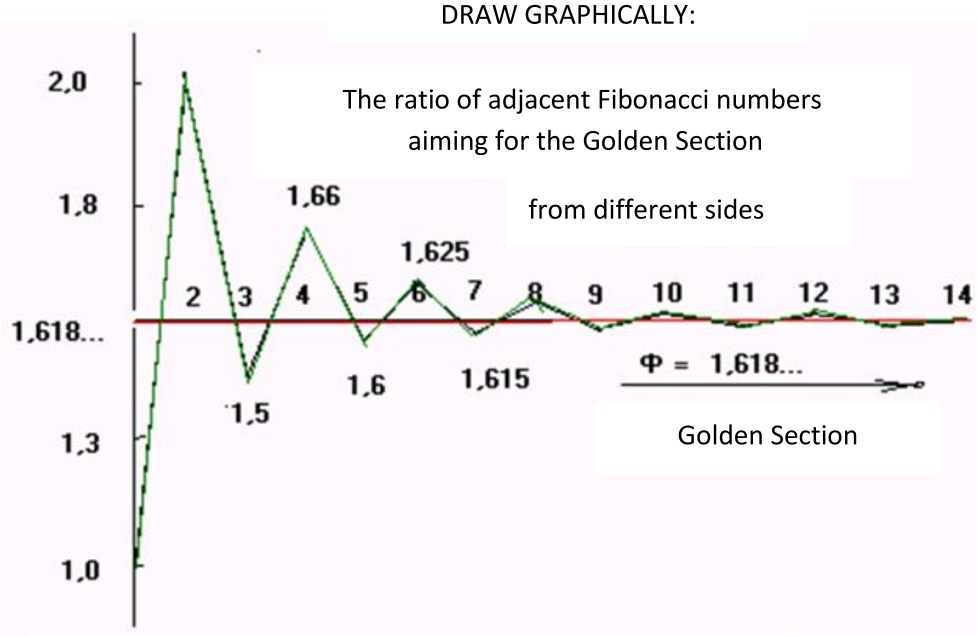
The golden proportion is surprisingly invariant.

$\Phi = (\sqrt{5} + 1) / 2 = 1{,}618\ldots;$

$\varphi = 1/\Phi = (\sqrt{5}-1)/2 = 0{,}618\ldots$

The Golden Proportion reflects the irrationality in the proportions of Nature, and the Fibonacci numbers reflect integerness in the organization of Nature. The combination of both laws is a dialectical unity of two principles: continuous and discrete, mobile and inert, rational and irrational. It is the perfection, the ideal, towards which the creative principle should strive.

DRAW GRAPHICALLY:



The ratio of adjacent Fibonacci numbers aiming for the Golden Section

from different sides

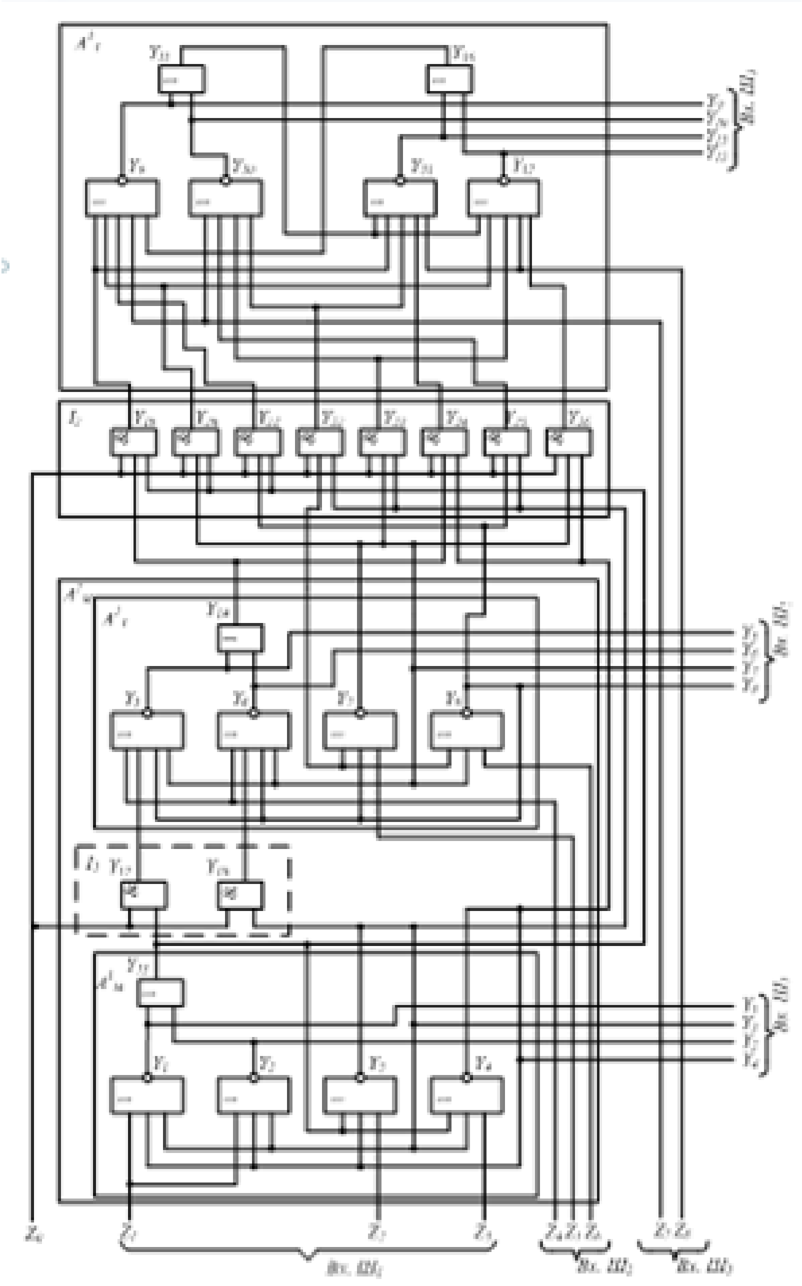
Golden Section

Periodic damped oscillation relative to Φ

## THREE-LEVEL SCHEME OF MEMORY [4].

The three-level memory circuit, as the basis of a neuron, resembles an artificial neuron with advanced transition and state memory capabilities.

• With different sets $e_j(\Delta)$ of input signals from the strategy automaton of the three-level memory scheme, in the πj state blocks, the MFIS can function as 9 *RS* flip-flops with different sets of their own states.\



• Three-level memory circuit, in blocks of $l_i$ ($i$ = 1,2,3) states can function as different three elementary six stable memory circuits.

• In general, the three-level memory scheme functions as an elementary 18-stable memory scheme.

In operation, a three-level memory circuit can operate in three modes:

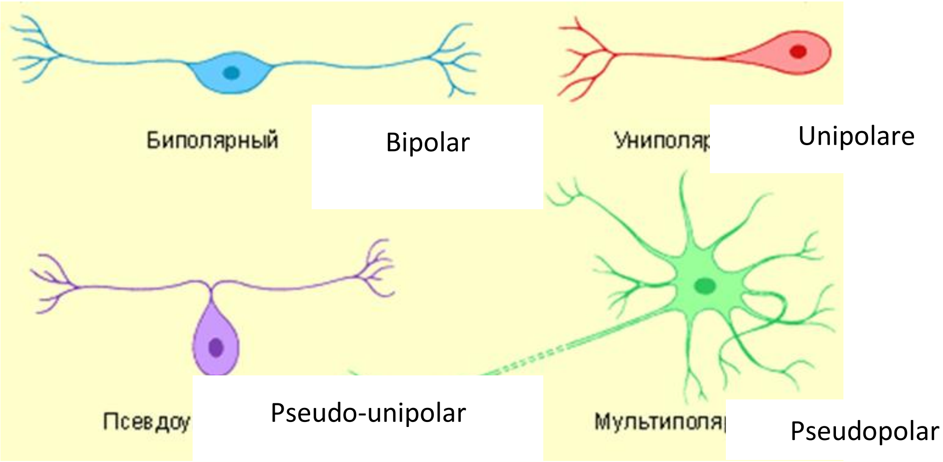
- In deterministic mode, realizing unambiguous and enlarged transitions;
- In probabilistic mode, realizing probabilistic transitions of the first and second types;
- In fuzzy mode, realizing fuzzy transitions.

An elementary neuron on a 3-level memory circuit has 5 transitions. These transitions provide functional advantages over known artificial neurons.

## IMPORTANT FUNCTIONAL POSSIBILITIES OF THE BRAIN [47–48].

- the brain has a short-term and long-term memory, with different properties for storing information; In order for long-term memory to be more reliable, new information should be acquired gradually and repeated at regular intervals.
- neurons and neural connections in their brain change; The smallest detail can cause a change in connections and structures in the brain;
- neurons have a different shape and size, form processes of two types: axons and dendrites;
- Neurons can be highly specific (primary areas of the cerebral cortex), for example, to respond only to a high or low tone in a sound, or only to an acute angle or a rounded line in an image or

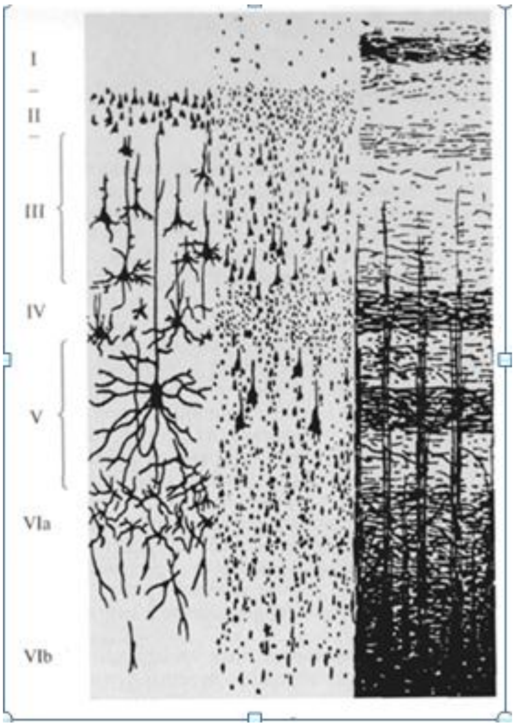
visible object. The neurons of the secondary cortex are, on the contrary, multimodal and have very complex characteristics. For example, they can respond to a certain number of stimuli, regardless of whether they are visual or auditory. There are neurons that do not respond to a single sensory stimulus, which means that their role is different. For example, there are attention units neurons that react to the change of old and new stimuli [49].



• Neuroglia. Neuroglia cells are concentrated in the central nervous system, where their number is ten times the number of neurons. They fill the space between the neurons, providing them with nutrients. Neuroglial cells may be involved in the preservation of information in the form of RNA codes. When the cells are dama Pseudo-unipolar tively divide, f Pseudopolar the injury site; neuroglia cells of another type turn into phagocytes and protect the body from viruses and bacteria.

• Synapses. The transfer of information from one neuron to another occurs in the synapses. Usually, the axon of one neuron and the dendrites or the body of another are connected via synapses. The synapses are also connected to the neurons of the end of the muscle fibers. The number of synapses is very large: some brain cells can have up to 10,000 synapses.

• In the brain in the I - IV layers, there is a perception and processing of incoming signals to the cortex in the form of nerve impulses. Efferent paths leaving the crust form mainly in the V – VI layers. It can be seen from the figure that the VI layer is divided into two layers: Via and Vib, which indicates a 7-level structure of the brain.



• the same activity structures the brain of different people differently;

• the brain is designed so that we often remember only the essence of the phenomena, missing details; It is important to concentrate on the meaning, try to find common patterns, link all the details into a logical system.

- A man’s brain concentrates more on substance, and a woman’s brain on detail.

The listed functionality is far from the only, though important.

## STRUCTURE OF THE NERVOUS SYSTEM [49].

In the structure of the nervous system there are:

- spinal cord,
- brain stem,
- the forebrain, the main part of which is the new cortex - the neocortex (neocortex).

In the brain produce:

- bark,
- amygdala (generating emotions),
- hippocampus (processing information just received and translating it from short-term memory into long-term memory),
- thalamus (receiving sensory information and transmitting it to the cortex),
- brain stem (controlling autonomic functions of the body, such as breathing; this is, as it were, the link between the brain and the spinal cord).

The brainstem includes such departments as:

- diencephalon (diencephalon),
- midbrain,
- posterior brain.

The structure of the cerebral cortex [49]. The cerebral cortex consists of two hemispheres - the right and left, each of which, in turn, consists of their lobes. The hemispheres are connected by nerve fibers, which form a bridge, and are called corpus callosum (corpus callosum).

The outer surface of the neocortex is a gray matter consisting of neurons with their bodies, dendrites, and blood capillaries. The inner layer is a white substance (subcortex) with axons emerging from the cortex and glia. Gray matter consists of 6 layers. They can be divided into two groups according to their functional role:

- 4 outer layers accept axons from other areas of the brain;
- two inner layers send axons to other areas of the brain.

The structure of the cerebral cortex is hierarchical, i.e. each level of the hierarchy does its work. In this regard, there are areas of the cortex:

- primary (projection),
- secondary,
- tertiary (integrative, associative).

## QUESTIONS OF LIFE CELLS (NEURON MEMBRANE) [47]

Debatable is the question of the presence or absence of a membrane through which signals from the synapses to the neuron.

Ling's physical theory is an alternative to the generally accepted membrane theory, which explains the four fundamental properties of a cell by the properties of its plasma membrane:

(1) semipermeability;

2) the ability to selectively accumulate substances;

(3) the ability to maintain osmotic stability;

(4) the ability to generate electrical potentials.

These properties are fundamental because our understanding of their mechanism determines our ideas on almost any matter of cell activity.

Ling's theory is not based on the properties of the membrane, but on the sorption properties of proteins, wherever they are in the cell.First of all, we are talking about the binding of the most mass cell components - water and K + ions. The adsorption properties of proteins do not remain unchanged, but depend on the conditions of the microenvironment.

Like any new theory, Ling's theory makes it possible to take a fresh look at the well-established ideas in the physiology and biophysics of the cell and outline new research prospects. So, Na, K-ATPase is not a pump, but an ionic receptor; It explains why this enzyme can selectively bind to Na + and K + ions and why the selectivity to these cations can change.

## LITERATURE

1. Explanatory dictionary of the Russian language: a Textbook for students of NATs. schools / [V. G. Biryukov, V. G. Vetvitsky, L. M. Gaydarova, al]. Ed. by M. I. Makhmutov, A. V. Tekucheva, H.M. Shan – L.: Prosveschenie, 1982. -384.. 2. V. M. Glushkov, Synthesis of digital automata. – M.: Fizmatgiz, 1962. – 476 p. 3. V. M. Glushkov, Theory of algorithms. –K.: SVPS, 1961. – 167 p.

4. Marakhovskii, L. F. fundamentals of the theory of design of discrete devices. Logical design of discrete devices on the schemes automatic memory: monograph. – K.: CHU, 1996. -128 p.

5. Marakhovsky L. F. Handbook of digital computing: (information technology processing hierarchical information in digital computers) - K .: // Interdisciplinary research in science and education. – 2018. No. 7B; URL: mino.esrae.ru/197-1616 (date of the application: 31.05.2018). in English. - 481 p.

6. Dictionary A-I Dictionary of the literary terms.

7. Wikipedia, the free encyclopedia. Paradigm (philosophy)

8. Promonenkov V. K., information Paradigm // "Academy of Trinitarizm", M., El № 77-6567, publ.20589, 16.05.2015.– 17 c.

9. Marakhovskii L. F. Comp utern schemotechnika: navch. Oleshko. – K.: KNEU, 2008. -360 p.

10. Zadeh L. A. the Concept of linguistic variable and its application to making approximate solutions: TRANS. s angl.. – M.: Mir, 1976. – 165 p.

11. Handbook of digital computing: (CPU and memory) / B. N. Malinowski, E. I., Bryukhovychi, E. L. Denisenko et al. / Under the editorship of B. N. Malinowski. – K.: "Techno", 1979. – 366 p.

12. Mishchenko V. A., Kosminski V. D., Semashko, A. N. Multifunction machines and circuitry of a computer / ed. by V. A. Mishchenko. – M.: Radio and communication, 1982. – 249 p.

13. Marachovsky L. F. Basic Concepts to Build the Next Generation of Reconfigurable Computing Systems. International Journal Of Applied And Fundamental Research. – 2013. – No. 2 URL: www.science-sd.com/455-24170 (20.11.2013). 6 p.

14. Marakhovskii L. F., Mikhno N. L. Structural machine. – The patent.–Seastrunk in Sovereign restr patent Ukraine on Corin model No. 25816 from 27 August 2007 – (51) IPC (2006) G06F 1/00 – bull. 13.– 12 S.

15. Marakhovskii L. F., Gavrilenko V. V., Mikhno N. L. Mathematical machines and principles of digital automata third kind. // Visnyk National transport University. –CH1 K.: NTU.– The issue 16 and Part 2. – K.: NTU.– The issue 17, 2008. With 329-335.

16. Marakhovskii L. F., Gavrilenko V. V., Zaitsev O. V., Mikhno N. L. Structural synthesis automats for odnochasno obrobki zagalna okremo nformac // national University bioresurs I prirodokoristuvannya Ukraine. – K., 2009 R. – VIP. 139. – S. 114-120.

17. Marakhovskii, L. F. a Multifunctional memory circuits. – Kiev: USIM, No. 6.-1996.– S. 59-69

18. Marakhovskii L. F. a Multilevel device automatic memory. – Kiev: the USIM. – No. 1.– 1998.– P. 66-72

19. Marakhovskii L. F. a Multilevel device automatic memory. II h. – Kiev: the USIM. – No. 2. – 1998. – S. 63-69

20. The Gagin V. System analysis "the Blade of life." – Odessa: El. ed., 2001. – 315 p.

21. Palagin A.V. Reconfigurable computing systems: Fundamentals and applications / A. V. Palagin, V. N. Opanasenko. – Kyiv: Prosvita, 2006. – 280 p. 22. S. Husson Microprogramming control. – M.: Mir, 1973. – Vol. 1 – 240 s.; 1974. – Vol. 2. – 477 p.

23. Marakhovskii L. F., Mikhno N. L. Scriptography divice control. – The patent. Seastrunk in Sovereign restr patents of Ukraine for invention No. 87871 from 28. 08 2009 R. – (51) IPC (2009) G06F 9/00 – bull. 16. – 6 p.

24. Murakhovski VI Iron PC. New opportunities. – SPb.: Peter 2005. 592 p.

25. Kalyaev I. A., Levin I. I., E. A. Semernikov Rekonfiguriruemyh multiconference computational structures M.: SSC RAS, 2008. – 395 p.

26. Mikhno N. L. How to build a reconfigurable processor "element" level / Zbirnik of naukovih] DETOUT, a Series of "Transport system I technology", 2011, VIP. 18. – S. 84-94.

27. Mikhno N. L. Methods of constructing reconfigurable computer systems on the elements of automatic memory/ Zbirnik of naukovih] DETOUT, a Series of "Transport system I technology", 2011, VIP. 19. – S. 146-152.

28. Marakhovskii L. F., Mikhno N. L. E-obchislyuval machine. – The patent. Seastrunk in Sovereign restr patent Ukraine on Corin model No. 34167 from R. on July 25 2008 – (51) IPC (2006) G06F 17/00 – bull. 14. – 10 p.

29. V. M. Glushkov, Y. V. Kapitonova, A. T. Mishchenko, Logic design of discrete devices. – K.: Of Sciences. Dumka, 1987. – 264 p.

30. Handbook of digital computing (software) / B. N. Malinovsky, V. V. Lipaev, T. F. Slobodyanyuk, etc. Under the editorship of B. N. Malinowski. – K.: Techno, 1991. -207 p.

31. S. I. Baranov, Synthesis of microprogramming automata (graph-schemes and machines). –L.: Energy, 1979. – 232 p.

32. Marakhovskii L. F., Romanky Yu. a. Methods of development programmnogo security. Zbirnik of naukovih] DEUT Mn in osut I science of Ukraine. A series of "Transport system I technology". – VIP. 26. – K.: DETOT, 2015. – 10 p.

33. Marakhovskii, L. F. Artificial intelligence: a monograph– K.: DETOT, 2014. – 241 p.

34. Hoursin L. A. the Beginning of the theory of systems of public type. – K.: "Svit", 2001. – 279 p.

35. Hoursin L. A. On the essence of the informational flows as reflection of dynamic structure of substantial base of short-term

memory of the human brain. – M.: "STI", ser. 2, No. 9, 1970. – S. 10-19.

36. Nikitin A.V., Artificial neuron // "Academy of Trinitarizm", M., El № 77-6567, publ.20230, 20.02.2015.

37. Seancev I. V. Neural networks: main models. – Voronezh, publishing house. "Voronezh State University", 1999. – 76 p.

38. Neural network. Wikipedia, the free encyclopedia

39. Nagel E., Newman J. R. Godel's Theorem. Abridged translation from English. Yu. A. Tatev. – Moscow: Znanie, 1970. – 60 p.

40. The Gagin V. System analysis "the Blade of life." – Odessa: El. ed., 2001. – 315 p.

41. R. L. Graham and V. Rödl. Numbers in Ramsey Theory, in Surveys in Combinatorics and. London Mathematical Society Lecture Notes Series, 1987, No. 123, pp.111-153.

42. Nikitin A.V., the control Logic of the cell // "Academy of Trinitarizm", M., El № 77-6567, publ.17037, 29.11.2011

43. Giordano Bruno. Wikipedia, the free encyclopedia

44. Nero. Wikipedia, the free encyclopedia

45. Stakhov A. P. The Mathematics of Harmony. From Euclid to Contemporary Mathematics and Computer Science.– Canada, World Scientic, 2010. -600 p.

46. Medina, John. The rules of pulp. 12 intellectual survival and prosperity. Washington, University of Washington, 2015.– 25 C.

47. Karpova I. S., candidate of medical Sciences. Central nervous system (CNS). Basics of structure and function. Lecture 1

48. Ling G. a Physical theory of the living cell: the Unnoticed revolution. – Saint Petersburg: Nauka, 2008. – 376 p.

49. Chernigovskaya T. V. , Gavrilova T. A. . Multimedia e-learning course on neurolinguistics "Minerva" (issued on CD) with the support of the Soros Foundation http://www.genlingnw.ru/Staff/Chernigo/Minerva/index.html

50. Chernigovskaya T. V. , M. I. Grineva . Actual problems and achievements of neuroscience in the context of education. Electronic textbook. http://islam-dl.spbu.ru/library/katalog/GEN_E956_F10_1

51. Videos for courses "Psycholinguistics", "Neurolinguistics", "Cognitive processes and the brain." http://coglab-spb.ru/index.php/ru/lectures.

## SUBJECT INDEX