

Minimization of the automatic machine structure process of accounting and control of railway automation and telemechanics devices

Dilshod Baratov^{1,*} and *Elmurod Astanaliev*¹

¹Tashkent state transport university, Tashkent, Uzbekistan

Abstract. The article explores the features of electronic document management of technical documentation of railway automation and telemechanics. A block diagram of an automaton model of technical documentation is developed. The proposed block diagram of a formalized model of technical documentation consists of matrices of external microoperations, internal microoperations, code creation for the following microcomponents. The number of internal states of the firmware is completely determined by the number of logic elements of the algorithms. The size of the matrices depends on the number of operators and logical conditions in the logic circuits of the algorithms. In this paper, a non-maximal group of compatible elements of logic circuits of algorithms is used as a microcommand, which led to the simplification of the matrix of the circuit of the firmware automaton. In the considered method of forming the maximum groups of compatible elements of logic schemes of algorithms, micro operations included in each micro command are performed simultaneously in one microtact. The structure of the firmware of the machine is simplified due to the transition from simple to complex micro-ops.

1 Introduction

The traditional way to automate electronic document management of technical documentation (EDMTD) is that the development of software gradually changes to the background mode, since even minor changes in the transport scheme and technological process mean the need to reprogram the application and consume more amount of time and money. As a result, there will be no time to update appeals, based on changing conditions and requirements of railway transport [1-3]. Actively developing systems associated with enterprise automation require a large number of trained personnel due to the high complexity. In addition, the number of qualified specialists in the field of automation is growing insufficiently fast [6]. Thus, the task is to create an easy-to-use, reliable automation tool, in the arsenal of which there are not only tools for describing a business process, but also its implementation. The possibility of implementation is important, since a pure descriptive tool is interesting only from the point of view of the analysis of technological processes and can

* Corresponding author: baratovdx@yandex.ru

be used only as part of a technical task when performing a specific model of the work process [7, 8]. EDMTD modeling is presented in the works of both Russian [9-14] and foreign authors [15-20].

Based on the logic schemes of algorithms (LSA) developed in the papers [20-22] for the process of accounting and control of railway automation and telemechanics devices (CARCD), this article presents a block diagram of an automatic model of technical documentation (TD), microcommands are formed according to the logic diagrams of algorithms TD, the problems of simplifying the structure of the automaton of the CARCD process are considered.

The purpose of this work is to describe and define an automated workflow model. EDMTD is considered as an object of modeling. To implement the model, an approved machine of the theory of automata will be used, adapted to modern programming technologies. The task of creating and implementing EDMTD is very urgent today. Significant financial, material and time resources are spent at enterprises and organizations of the railway to solve this problem.

2 Development of a block scheme of an automaton model of technical documentation

On the basis of a microprogram machine (MA), a block diagram of a formalized TD model has been developed.

Based on the MA, the Wilks-Stinger scheme was implemented, which is used when there are no strict requirements for the speed of machine control in the synthesis of microprograms with the least control [23]. The proposed structural diagram of the formalized TD model consists of matrices of external micro-operations M1, internal micro-operations M2, code creation for the following micro-components M3. The presented model has become a new system paradigm for the representation of TDs [4, 5].

Each external micro-operation $Z_{V_{qg}}$ is a control operator, and the internal one $Z_{\alpha_{qg}}$ is a control logical α_{qg} condition that checks a condition whose values are fulfilled ($\alpha_{qg} = 1$; marked with a + sign) or non-fulfilled condition ($\alpha_{qg} = 0$; marked with a - sign). On the structural diagram, a circle denotes a circuit that serves to query the value $Z_{\alpha_{qg}}$ of the condition it is checking (Fig. 1).

Firmware, i.e. the sequence of microoperations is conveniently described in the language of LSA, and external microoperation V_{qg} is associated with the operator $Z_{V_{qg}}$, and internal microoperation α_{qg} is associated with the logical condition $Z_{\alpha_{qg}}$.

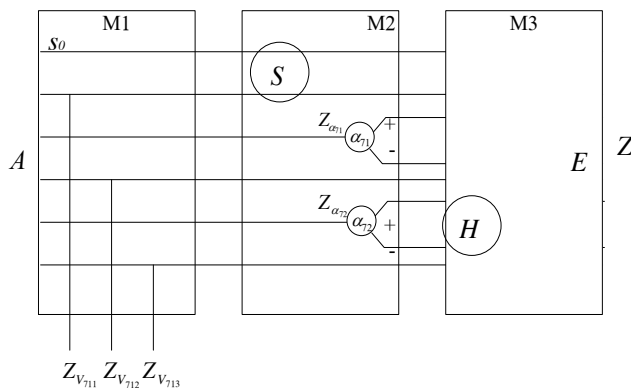


Fig. 1. Block scheme of a formalized TD model.

When performing this operation, each micro-command includes only one micro-operation (external or internal) and includes only one operator or logical condition during each microtact. The number of internal states of MA is completely determined by the number of LSA elements. The size of the matrices M1 and M2 depends on the number of operators and logical conditions in the LSA. In the specific case, when only operators enter the LSA, the matrix M2 is absent. In this case, successive microcommands are generated each time in the M3 matrix.

If there are logical conditions in the LSA, the matrix M2 is needed, in which they are formed $Z_{\alpha_{qg}}$. If the value checked in $Z_{\alpha_{qg}}$ the logical condition (if it enters the LSA without inversion), the order of execution of the LSA elements is violated. Then, in M3, the necessary number of micro-commands must be formed in order to correctly perform LSA.

If combine a separate micro-command with each element of the LSA, even a very simple algorithm requires that MA fulfill a large number of internal states. At the same time, there is no need to distinguish the internal position for each LSA element. Some external microoperations may not be performed sequentially, but simultaneously in a single microtact. This occurs when the operator performing operations on the corresponding microtacts can work in parallel. LSA statements executed simultaneously are not related to the internal state of a single MA statement. This minimizes the number of internal states of MA. The combination of individual microoperations over time leads to an increase in speed, since the number of microtacts required to execute the algorithm decreases.

Simplification of the MA scheme can be obtained by simultaneously performing external and internal microoperations. Then the number of microcommands of MA will be determined not by the number of LSA elements, but by the number of groups of simultaneously performed microoperations. Consider the LSA technological process of accounting and control of railway automation and telemechanics devices AT, which includes planning and repair of equipment, control of reception and storage, as well as the movement of devices in the repair and maintenance department:

$$\begin{aligned}
 A_T = & V_0 V_{711} V_{712} V_{713} V_{714} \downarrow^{717} V_{715} V_{716} V_{717} \downarrow^{715} \\
 & V_{718} \alpha_{711} \uparrow^{711} V_{7110} V_{7111} \overline{\alpha_{712}} \uparrow^{712} \downarrow^{711} V_{719} \omega \uparrow^{715} \\
 & \times \downarrow^{712} \downarrow^{714} V_{7115} \alpha_{713} \uparrow^{713} V_{7116} \alpha_{714} \uparrow^{714} V_{7117} \omega \\
 & \uparrow^{716} \downarrow^{713} V_{7118} \omega \uparrow^{717} \downarrow^{712} V_{7112} V_{7113} V_{7114} V_k
 \end{aligned} \tag{1}$$

The CARCD process is presented in the LSA language [24,25]. The main elements are the operators $V_{qg}, q = \overline{1, Q}, g = \overline{1, G}$ that correspond to the elementary actions of the process, logical conditions $\alpha_k, k = \overline{1, K}$ - the probability of their fulfillment depends on the current state of the process CARCD, represented by arrows $\alpha_k \uparrow^p, p = \overline{1, P}$, where p is the index of the arrow. Here will be distinguish 6 groups of simultaneously performed microoperations:

$$\begin{aligned}
 A_T = & \underbrace{V_0 V_{711} V_{712} V_{713} V_{714} \downarrow^{717} V_{715} V_{716} V_{717} \downarrow^{715} V_{718} \alpha_{711}}_1 \\
 & \underbrace{\uparrow^{711} V_{7110} V_{7111} \overline{\alpha_{712}} \uparrow^{712} \downarrow^{711} V_{719} \omega \uparrow^{715}}_2 \\
 & \times \underbrace{\downarrow^{712} \downarrow^{714} V_{7115} \alpha_{713} \uparrow^{713} V_{7116} \alpha_{714}}_3 \underbrace{\uparrow^{714} V_{7117} \omega}_4 \\
 & \underbrace{\uparrow^{716} \downarrow^{713} V_{7118} \omega}_5 \underbrace{\uparrow^{717} \downarrow^{712} V_{7112} V_{7113} V_{7114} V_k}_6
 \end{aligned} \tag{2}$$

It follows from (1) that MA will have six internal states. The $M3$ matrix in Fig. 2 is constructed when compared with microcommands 1, 2, 3, 4, 5, 6 of the code combinations 001, 010, 011, 100, 101, 110, respectively.

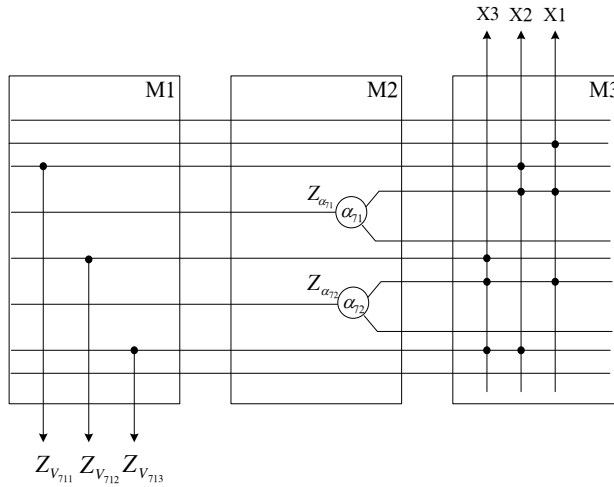


Fig. 1. The structure of the matrix $M3$.

Thus, will be examined the method of phasing the micro-commands in stages, and as a result will be switched from the micro-command, which includes only one external or internal micro-operation, to the micro-command, which includes the entire group of external and internal micro-operations. To form such microcommands, a firmware should be implemented in addition to the LSA, providing information on the possibility of simultaneous operation of various operators and the distribution of offsets for each operator. Obviously, the operator V_{qg} and the logical condition α_{qg} cannot be included in one micro-command if the value can be changed by the operator V_{qg} . The task of forming the smallest possible number of LSA microcommands is complex.

When considering various possible ways of constructing an MA circuit, it is considered that only one algorithm is implemented. However, a software control method is used precisely when several different algorithms need to be implemented in the machine.

The structure of the MA transition graph will largely depend on the choice of microcommands when comparing internal states with microcommands [26-28].

A special requirement is the development of methods that allow to minimize and encode the machine, taking into account various requirements.

3 Method for simplifying the structure of the automatic machine the process of accounting and control of railway automation and telemechanics devices

The subsequent simplification of the MA structure can be achieved by switching from simple to complex micro commands. In complex micro-commands, it is characteristic that the micro-operations included in it may not be performed simultaneously, but sequentially over time, and the set of these micro-operations may change depending on the values of the conditions checked when performing internal micro-operations included in this micro-command.

When forming simple micro commands, will be assumed that each functional block (FB) has one input, the impact of which should be removed after this FB has finished its work (i.e., the corresponding micro operation is performed).

In the technological process of CRCAD, representing LSA A71 in practice, two algorithmic States V_{qg}^* (completed) and V_{qg} (not completed) can be used for one operator. The operators V_{qg}^* and V_{qg} are never executed simultaneously and can be called inverses.

The presence of inverse operators in the LSA that describe the operation of the firmware machine allows to build a machine with complex micro-commands.

In (2), a sequence of executing operators $V_{q_1g_1}, V_{q_2g_2}, \dots, V_{q_n g_n}$ is required, and the FB corresponding to the operator V_{qg} can be enabled after the FB corresponding to the operator $V_{q(g-1)}$ has completed its work, i.e. when forming simple micro-commands, only one operator can be included in each of them, and the number of internal States of the MA will be equal to the number of operators. Can allow the operator V_{qg} to be executed simultaneously with the operators $V_{q(g+1)}, \dots, V_{q(g+m)} (g \leq n)$, if there is no operator V_{qg}^* among the latter, and map not one, but several groups of operators that make up a complex microcommand to each internal state of the MA. But now operators $V_{qg}, \dots, V_{q(g+m)}$ that are part of the same micro-command must be executed sequentially, i.e. the value $Z_{V_{q(g+1)}}$ cannot appear before the operator $V_{q(g+1)}$ ends. In order to ensure the specified sequence of function blocks, each time before executing an operator $V_{q(g+1)}$, must check whether the operator V_{qg} is executed. Therefore, in addition to the usual logical conditions in the LSA, must enter special waiting logical conditions α that check the execution of the corresponding operators. Will be denote such logical conditions by χ_i , assuming that $\chi_i = 1$, if the operator V_{qg} is executed and $\overline{\chi_i} = 1$, if the operator V_{qg}^* is executed.

LSA (4.4) will have the form:

$$\begin{aligned}
 A71 &= \downarrow^1 V_{711} \chi_{711} \uparrow^1 \downarrow^2 V_{712} \chi_{712} \uparrow^2 \downarrow^3 V_{713} \chi_{713} \\
 &\uparrow^3 \downarrow^4 V_{714} \chi_{714} \uparrow^4 \downarrow^{717} \downarrow^5 V_{715} \chi_{715} \uparrow^5 \downarrow^6 V_{716} \chi_{716} \uparrow^6 \\
 &\downarrow^7 V_{717} \chi_{717} \uparrow^7 \downarrow^{715} \downarrow^8 V_{718} \chi_{717} \uparrow^7 \alpha_{711} \uparrow^{711} \downarrow^{10} V_{7110} \\
 &\chi_{7110} \uparrow^{10} \downarrow^{11} V_{7111} \chi_{7111} \uparrow^{11} \downarrow^{716} \overline{\alpha_{712}} \uparrow^{712} \downarrow^{711} \downarrow^{19} \\
 &\downarrow^9 V_{719} \chi_{719} \uparrow^9 \omega \uparrow^{715} \downarrow^{714} \downarrow^{15} V_{7115} \chi_{7115} \\
 &\uparrow^{15} \alpha_{713} \uparrow^{713} \downarrow^{16} V_{7116} \chi_{7116} \uparrow^{16} \alpha_{714} \uparrow^{714} \downarrow^{17} \\
 &V_{7117} \chi_{7117} \uparrow^{17} \omega \uparrow^{716} \downarrow^{713} \downarrow^{18} V_{7118} \chi_{7118} \uparrow^{18} \\
 &\omega \uparrow^{717} \downarrow^{712} \downarrow^{12} V_{7112} \chi_{7112} \uparrow^{12} \downarrow^{13} V_{7113} \chi_{7113} \uparrow^{13} \downarrow^{14} \\
 &V_{7114} \chi_{7114} \uparrow^{14} \downarrow^1 \equiv \tag{3} \\
 &\equiv V_{711}^* \overline{\chi_{711}} \uparrow^1 \downarrow^2 V_{712}^* \\
 &\overline{\chi_{712}} \uparrow^2 \downarrow^3 V_{713}^* \overline{\chi_{713}} \uparrow^3 \downarrow^4 V_{714}^* \overline{\chi_{714}} \uparrow^4 \downarrow^{717} \downarrow^5 \\
 &V_{715}^* \overline{\chi_{715}} \uparrow^5 \downarrow^6 V_{716}^* \overline{\chi_{716}} \uparrow^6 \downarrow^7 V_{717}^* \overline{\chi_{717}} \\
 &\uparrow^7 \downarrow^{715} \downarrow^8 V_{718}^* \overline{\chi_{717}} \uparrow^7 \overline{\alpha_{711}} \uparrow^{711} \downarrow^{10} V_{7110}^*
 \end{aligned}$$

$$\begin{aligned}
 & \overline{\chi_{7110}} \uparrow^{10} \downarrow^{11} V_{*7111} \overline{\chi_{7111}} \uparrow^{11} \downarrow^{716} \alpha_{712} \\
 & \uparrow^{712} \downarrow^{711} \downarrow^{19} \downarrow^9 V_{*719} \overline{\chi_{719}} \uparrow^9 \omega \uparrow^{715} \downarrow^{714} \downarrow^{15} \\
 & V_{*7115} \overline{\chi_{7115}} \uparrow^{15} \overline{\alpha_{713}} \uparrow^{713} \downarrow^{16} V_{*7116} \overline{\chi_{7116}} \\
 & \uparrow^{16} \overline{\alpha_{714}} \uparrow^{714} \downarrow^{17} V_{*7117} \overline{\chi_{7117}} \uparrow^{17} \omega \uparrow^{716} \\
 & \downarrow^{713} \downarrow^{18} V_{*7118} \overline{\chi_{7118}} \uparrow^{18} \omega \uparrow^{717} \downarrow^{712} \downarrow^{12} V_{*7112} \\
 & \overline{\chi_{7112}} \uparrow^{12} \downarrow^{13} V_{*7113} \overline{\chi_{7113}} \uparrow^{13} \downarrow^{14} V_{*7114} \overline{\chi_{7114}} \uparrow^{14} \tag{3'}
 \end{aligned}$$

Will be still assume that the values of all logical conditions included in the LSA are changed only by the operators of this LSA. After the algorithm is executed, the values of the conditions α can change from the outside, while the conditions χ do not change. Another feature of logical conditions χ is the absence of ramifications in the LSA, the introduction of a time delay, fixing the moment when the corresponding operators are executed. Therefore, for ease of recording, such LU's can not be included in the LSA, although the corresponding logical blocks will be included in the machine.

Then write the LSA in the form:

$$\begin{aligned}
 A71 &= V_{711} V_{712} V_{713} V_{714} \downarrow^{717} V_{715} V_{716} V_{717} \\
 & \downarrow^{715} V_{718} \alpha_{711} \uparrow^{711} V_{7110} V_{7111} \downarrow^{716} \overline{\alpha_{712}} \uparrow^{712} \\
 & \downarrow^{711} V_{719} \omega \uparrow^{715} \times \downarrow^{714} V_{7115} \alpha_{713} \uparrow^{713} \\
 & V_{7116} \alpha_{714} \uparrow^{714} V_{7117} \omega \uparrow^{716} \downarrow^{713} V_{7118} \omega \uparrow^{717} \\
 & \downarrow^{712} V_{7112} V_{7113} V_{7114} \equiv V_{*711} V_{*712} V_{*713} \\
 & V_{*714} \downarrow^{717} V_{*715} V_{*716} V_{*717} \downarrow^{715} V_{*718} \\
 & \overline{\alpha_{711}} \uparrow^{711} V_{*7110} V_{*7111} \downarrow^{716} \alpha_{712} \\
 & \uparrow^{712} \downarrow^{711} V_{*719} \omega \uparrow^{715} \times \downarrow^{714} V_{*7115} \\
 & \overline{\alpha_{713}} \uparrow^{713} V_{*7116} \alpha_{714} \uparrow^{714} V_{*7117} \omega \uparrow^{716} \\
 & \downarrow^{713} V_{*7118} \omega \uparrow^{717} \downarrow^{712} V_{*7112} V_{*7113} V_{*7114} \tag{4}
 \end{aligned}$$

Fig. 2 shows the structure of the MA with complex micro-commands based on the LSA (4). The automaton has twelve internal States. Therefore, the expansion of simple micro-commands and the formation of complex micro-commands make it possible to simplify the structure of the MA.

The method of forming complex micro-commands is similar to forming simple micro-commands, but now all operators can be included in one micro-command, except inverses. Therefore, when creating complex micro-commands, the definition of conflicting LSA elements will change somewhat.

Definition. Two elements of the LSA Y_i and Y_j are contradictory if:

- 1) they are inverse operators;
- 2) Y_i is a logical condition, a Y_j – an operator that can be executed after verification Y_i , and the logical condition is included in the shift distribution of the operator Y_j .

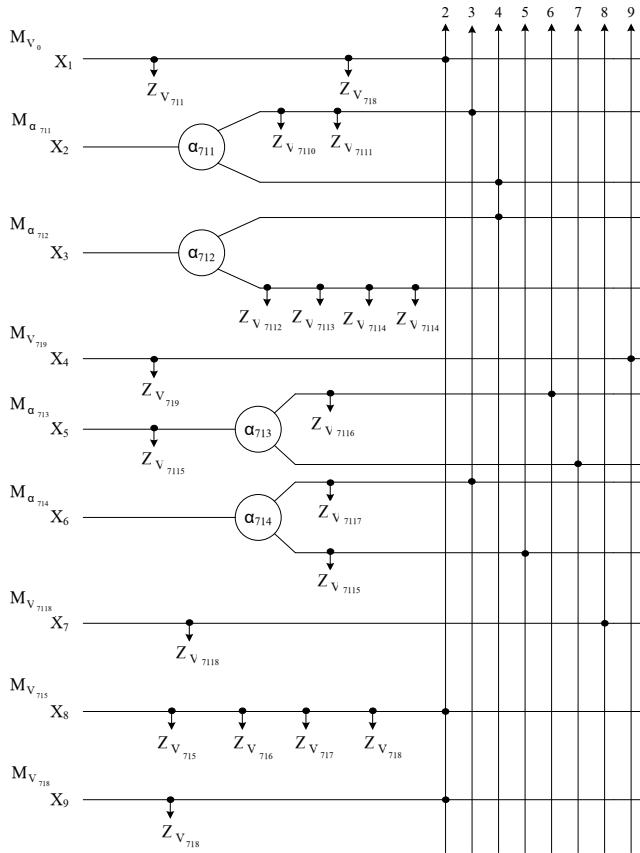


Fig. 3. Structure of MA LSA A71 with complex micro commands.

Thus, in contrast to a simple one, a complex microcommand can include a logical condition α_q and an operator V_g that changes the value of this logical condition ($V_g - \{\alpha_q\}$), if only the check α_q must be performed after the operator V_g is executed.

Using the previously described method of forming micro-commands, taking into account this definition of inconsistency, will be made complex micro-commands for LSA (4):

$$\begin{aligned}
 U^{A71} &= V_0 V_{711} V_{712} V_{713} V_{714} V_{715} V_{716} V_{717} \\
 &V_{718} \downarrow^{715} \downarrow^{717} \alpha_{711} \uparrow^{711} V_{7110} V_{7111} \downarrow^{716} \\
 &\overline{\alpha_{712}} \uparrow^{712} \downarrow^{711} V_{719} V_{718} \omega \uparrow^{715} V_{7115} \\
 &\downarrow^{714} \alpha_{713} \uparrow^{713} V_{7116} \alpha_{714} \uparrow^{714} V_{7117} \omega \uparrow^{716} \\
 &\downarrow^{714} V_{7115}^* \omega \uparrow^{714} \downarrow^{713} V_{7118} V_{715}^* V_{716}^* \\
 &V_{717}^* V_{718}^* \omega \uparrow^{717} \downarrow^{712} V_{7112} V_{7113} V_{7114} V_k
 \end{aligned} \tag{5}$$

In the presented logical scheme, no pair of operators is executed simultaneously. The following order of shifts is set for this LSA:

$$\begin{aligned}
 &V_{711} - \{\alpha_{711}, \alpha_{712}\}; V^*_{711} - \{\alpha_{711}, \alpha_{712}\}; \\
 &V_{712} - \{\alpha_{712}\}; V^*_{712} - \{\alpha_{712}\}; \\
 &V_{713} - \{-\}; V^*_{713} - \{-\}; V_{714} - \{-\}; \\
 &\dots, V_{7118} - \{-\}; V^*_{7118} - \{-\} \tag{6}
 \end{aligned}$$

Creating complex micro-commands from a group F_{V_1} where the first one starts with the operator V_1 .

$$\begin{aligned}
 M_{V_0} &= \left\{ \begin{array}{l} V_{711} \rightarrow V_{712} \rightarrow V_{713} \rightarrow V_{714} \rightarrow V_{715} \\ \rightarrow V_{716} \rightarrow V_{717} \rightarrow V_{718} \rightarrow [\alpha_{711}] \end{array} \right\} M_{\alpha_{711}} \\
 &= \left\{ \begin{array}{l} \alpha_{711} \left| \begin{array}{l} \&\rightarrow V_{7110} \rightarrow V_{7111} \rightarrow [\alpha_{712}] \\ \&\rightarrow [V_{719}] \end{array} \right. \end{array} \right\} \\
 M_{\alpha_{712}} &= \alpha_{712} - \left| \begin{array}{l} \&\rightarrow V_{7112} \rightarrow V_{7113} \rightarrow V_{7114} \rightarrow V_k \\ \&\rightarrow [V_{719}] \end{array} \right. \\
 M_{V_{719}} &= \{V_{719} \rightarrow \omega^{715} \rightarrow V_{718} \rightarrow [\alpha_{711}]\} \\
 M_{\alpha_{713}} &= \left\{ \begin{array}{l} V_{7115} \rightarrow \alpha_{713} \left| \begin{array}{l} \&\rightarrow V_{7116} \rightarrow [\alpha_{714}] \\ \rightarrow [V_{7118}] \end{array} \right. \end{array} \right\} \\
 M_{\alpha_{714}} &= \left\{ \begin{array}{l} \alpha_{714} \left| \begin{array}{l} \&\rightarrow V_{7117} \rightarrow \omega^{716} \rightarrow [\alpha_{712}] \\ \rightarrow V_{7115} \rightarrow [\alpha_{713}] \end{array} \right. \end{array} \right\} \\
 M_{V_{7118}} &= \left\{ \begin{array}{l} V_{7118} \rightarrow \omega^{717} \rightarrow V^*_{715} \rightarrow V^*_{716} \\ \rightarrow V^*_{717} \rightarrow V^*_{718} \rightarrow [\alpha_{711}] \end{array} \right\} \tag{7}
 \end{aligned}$$

In (7), two complex micro-commands are reflected; the mA scheme, whose internal States are compared with these micro-commands, is shown in Fig.4.

Compatible groups of elements of this logic scheme of algorithms are mapped to the corresponding simple micro-commands in order to make the transition to complex micro-commands and reduce the number of internal MA States.

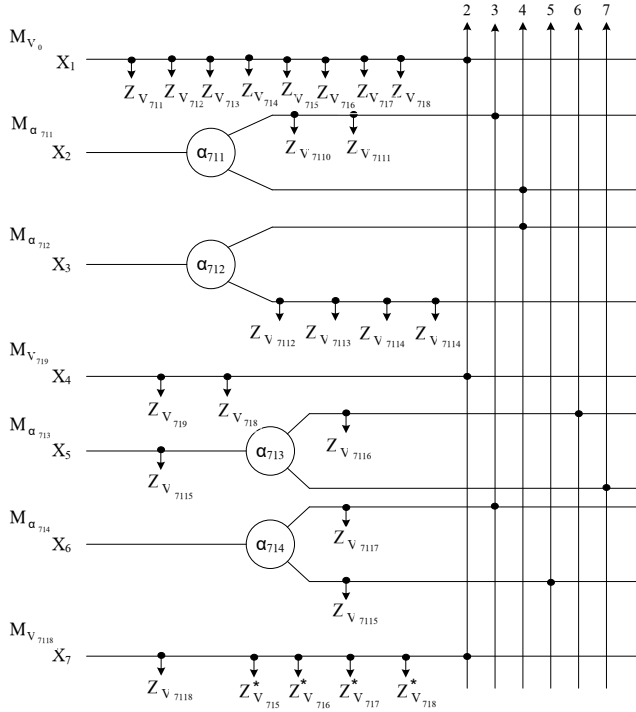


Fig. 4. Structure of MA LSA A71 with complex micro commands.

For comparison, will be presented compatible groups of elements of this LSA corresponding to simple micro commands:

$$\begin{aligned}
 M_{V_0} &= \{V_0 \rightarrow V_{711} \rightarrow V_{712} \rightarrow V_{713} \rightarrow V_{714} \rightarrow \\
 &\quad \{V_{715} \rightarrow V_{716} \rightarrow V_{717} \rightarrow V_{718} \rightarrow [\alpha_{711}] \} \\
 M_{\alpha_{711}} &= \left\{ \begin{array}{l} \&^{\rightarrow} V_{7110} \rightarrow V_{7111} \rightarrow [\alpha_{712}] \\ \alpha_{711} \quad \quad \quad \&_{\rightarrow} [V_{719}] \end{array} \right\} \\
 M_{\alpha_{712}} &= \left\{ \begin{array}{l} \&^{\rightarrow} [V_{719}] \\ \alpha_{712} - \left| \&_{\rightarrow} V_{7112} \rightarrow V_{7113} \rightarrow V_{7114} \rightarrow V_k \right. \end{array} \right\} \\
 M_{V_{719}} &= \{V_{719} \rightarrow [V_{718}] \} \\
 M_{\alpha_{713}} &= \left\{ \begin{array}{l} \&^{\rightarrow} V_{7116} \rightarrow [\alpha_{714}] \\ V_{7115} \rightarrow \alpha_{713} \quad \quad \quad \rightarrow [V_{7118}] \end{array} \right\} \\
 M_{\alpha_{714}} &= \left\{ \begin{array}{l} \&^{\rightarrow} V_{7117} \rightarrow [\alpha_{712}] \\ \alpha_{714} \quad \quad \quad \rightarrow V_{7115} \rightarrow [\alpha_{713}] \end{array} \right\} \\
 M_{V_{7118}} &= \{V_{7118} \rightarrow [V_{715}] \} \\
 M_{V_{715}} &= \{V_{715} \rightarrow V_{716} \rightarrow V_{717} \rightarrow V_{718} \rightarrow \} \\
 M_{V_{718}} &= \{V_{718} \rightarrow [\alpha_{711}] \}
 \end{aligned} \tag{8}$$

An MA constructed using simple micro commands has eight internal States (Fig. 5).

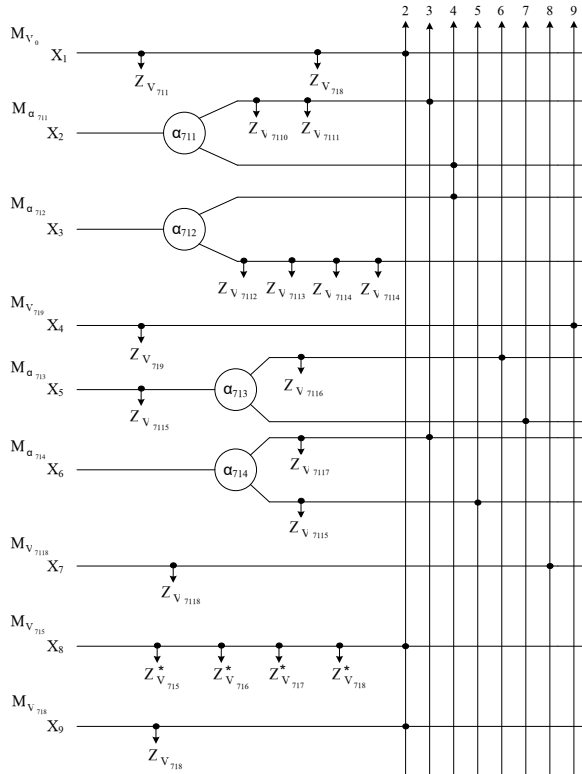


Fig. 5. Structure of MA LSA A71 with simple micro commands.

Thus, the transition to complex micro-commands complicates the $M1$ and $M2$ matrices, but simplifies and reduces the number of internal MA States.

Based on the results of the research presented in this paper, will be formulate the following conclusions:

The proposed structural diagram of a formalized model of the technical documentation takes into account matrix external micro-operations, matrix of internal micro-operations, as well as matrix code generation the following macro. Simplification of the MA scheme is achieved by simultaneously performing external and internal micro-operations. The number of MA micro-commands is determined not by the number of LSA elements, but by the number of groups of simultaneously performed micro-operations.

A method for simplifying the MA scheme has been developed that allows simultaneous minimization and encoding of the machine, taking into account various requirements related to the document flow of automation and telemechanics farms. When comparing micro-commands, maximum performance is achieved, i.e. the LSA is performed in the minimum number of clock cycles of the MA, and the MA itself has a minimum or close to it number of internal States.

The use of non-maximal groups of compatible LSA elements as micro-commands has led to simplification of mA scheme matrices. The use of a micro-command system simplifies the combination part of the MA.

With the considered method of forming maximum groups of compatible LSA elements, all micro operations included in each microcomand (which is typical for them) are performed simultaneously in one microtact.

Simplification of the MA structure was achieved by switching from simple to complex micro commands. A complex micro-command is characterized by the fact that the micro-

operations included in it may not be performed simultaneously, but sequentially over time, and the set of these micro-operations may change depending on the values of the conditions checked when performing internal micro-operations included in this micro-command.

References

1. D.Kh.Baratov, A.Kh. Boltaev, E.T.Astanaliev, *Functional Support of the Automated Accounting System and Control of Devices in Railway Automation Telemechanics*, International Journal of Advanced Research in Science, Engineering and Technology **6**, 3, 8572-8580 (2019)
2. V. Shamanov, *Generalized mathematical model of the process of operating automation systems and telemechanics*, Automation on Transport **2**, 2, 163-179 (2016)
3. N.M. Aripov, D. X.Baratov, *Questions intellectualization of management of transportation processes for railways*, International scientific journal **9**, 53-55 (2016)
4. P.Y. Bulavsky, D.S. Markov, *Hierarchical multi-matrix formalization of a simulation model of electronic document management of technical documentation*, Actual issues of the development of railway automation systems of telemechanics: Sat. scientific proceedings / ed. Vl. V. Sapozhnikov, SPb., 52-59 (2013)
5. N. Aripov, R. Aliyev, D. Baratov, E. Ametova, *Features of Construction of Systems of Railway Automatics and Telemechanics at the Organization of High-Speed Traffic in the Republic of Uzbekistan*, Procedia Engineering **134**, 175-180 (2016)
6. P.Y. Bulavsky, *Conceptual model of electronic document management of technical documentation*, Transport of the Russian Federation **1 (32)**, 60-63 (2011)
7. A.G. Mamikonov, *Development methods for automated control systems* (M.: Energiya, 1973)
8. N.M. Aripov, D.Kh. Baratov, E.K. Ametova, *Formalization of electronic document management of railway automation and telemechanics based on simulation*, Bulletin Of The Tashkent Railway Transport Engineering Institute **2-3**, 90-97 (2017)
9. D. Baratov, *The issues of creating a formalized model of the technical documentation*, International scientific journal **4 (1)**, 22-23 (2017)
10. D.S. Markov, A.A. Likov, V.B. Sokolov, T.Yu. konstantinova, *Concept and tools for dynamic batch modeling of complex Queuing systems*, Collection: intelligent Transport systems 49-58 (2017)
11. Vasilenko, M.N., Bulavsky, P.E., & Baratov, D. (2009). Document flow technologies for order optimization. World of Transport, 7 (4), 110-115.
12. M.F. Stepanov, A.M. Stepanov, *Mathematical modeling of intelligent self-organizing systems: implementation of the action planning mechanism*, Information technologies and nanotechnologies, 1681-1688 (2018)
13. F. Guo, C.T. Jahren, Y. Turkan, *Electronic Document Management Systems for the Transportation Construction Industry*. International Journal of Construction Education and Research **1-16** (2019)
14. M.T. Sun, J.L. Hou, *The architecture and models for document security reasoning in an electronic document management system*, Journal of the Chinese Institute of Industrial Engineers **20(4)**, 305-316 (2003)
15. S.S. Sokolov, A.S. Karpina, V.D. Gaskarov, *Methods and models of designing the secure system of electronic document management in transport logistic cluster*. Bulletin of

- Astrakhan State Technical University. Series: Management, Computer Sciences and Informatics **(3)**, 40-52 (2016)
16. C.M. Afonso, J.L. Roldán Salgueiro, M.J. Sánchez Franco, M.D.L.O. González, *The moderator role of Gender in the Unified Theory of Acceptance and Use of Technology (UTAUT): A study on users of Electronic Document Management Systems* (2012)
 17. S.F.M. Yatin, A.A.M. Ramli, H. Shuhaimi, H. Hashim, et.al., *Electronic Document Management System: Malaysian Experience*, Australian Journal of Basic and Applied Sciences **9(3)**, 82-89 (2015)
 18. P. Aurelia, T. Ana, *A document management system modeling*, ANALELE UNIVERSITĂȚII DIN ORADEA, 1479 (2018)
 19. N.M.Aripov, D.Kh. Baratov, *Methods of building the mathematical mode of electronic document management of technical documentation for railway automation*, Automation on Transport **3**, 1, 98-111 (2017)
 20. D.Kh. Baratov, N.M. Aripov, *Formalization of electronic technical document management of railway automatics and telemechanics*, Europäische Fachhochschule, **8**, 33-35 (2016)
 21. D.Kh. Baratov, N.M. Aripov, D.Kh. Ruziev, *Formalized Methods of Analysis and Synthesis of Electronic Document Management of Technical Documentation*, 2019 IEEE East-West Design & Test Symposium (EWDTS), 1-9 (2019)
 22. S. Furth, J. Baumeister, *Semantification of large corpora of technical documentation. In Enterprise Big Data Engineering, Analytics, and Management (pp. 171-200)*, IGI Global (2016)
 23. A. Aguilar, C. Lozoya, L.M. Orona, *A hamming distance and fuzzy logic-based algorithm for P2P content distribution in enterprise networks*, Peer-to-Peer Networking and Applications **12(5)**, 1323-1335 (2019)
 24. M.V. Wilkes, J.B. Stringer, *Micro-programming and the design of the control circuits in an electronic digital computer*, Mathematical Proceedings of the Cambridge Philosophical Society. Cambridge University Press **49**, 2, 230-238
 25. D.X. Baratov, *The issues of creating a formalized model of the technical documentation*, International scientific journal **4 (1)**, 22-23 (2017)
 26. V.V. Sapozhnikov, D.V. Efanov, *Fundamentals of reliability theory and technical diagnostics* (Moscow, Lan, 2019)
 27. D. Baratov, N. Aripov, O. Muxiddinov, X. Jumanov, *Block diagram of the automatic model of technical*, IOP Conference Series: Materials Science and Engineering **918(1)**, 012084 (2020)
 28. N.M. Aripov, D.K. Baratov, *Methods of building the mathematical mode of electronic document management of technical documentation for railway automation*, Automation on Transport **3(1)**, 98-111 (2017)