ISSN 2226-3780

10. SINEMA Server – Siemens Support. Configuration Manual [Text]: C79000-G8976-C241-01. – Nürnberg, Germany: Siemens AG, 2011. – 215 p. – Available at: \www/URL: https://cache.industry.siemens.com/dl/files/033/52750033/att_105666/v1/PH_SINEMA-SERVER_76.PDF
11. Shea, N. Siemens vs Allen-Bradley: Function Blocks [Electronic resource] / N. Shea // DMC, Inc. – April 01, 2010. – Available at: \www/URL: https://www.dmcinfo.com/latest-thinking/blog/articletype/articleview/articleid/171

РАЗРАБОТКА ВЕРХНЕГО УРОВНЯ АВТОМАТИЗИРОВАННОЙ SCADA-СИСТЕМЫ ПОДАЧИ ВЛАГИ В ПОЧВЕННО-РАСТИТЕЛЬНЫЙ ПОКРОВ

Разработан верхний уровень системы подачи воды в почвенно-растительный покров с использованием программного продукта фирмы Siemens Simatic WinCC. Исходя из водно-физических свойств почв, разработаны три режима подачи воды в почвенно-растительный покров (ручной, первого и второго типа). Структура разработанной системы – это количество режимов управления, структура сети и передачи данных.

Ключевые слова: почвенно-растительный покров, режимы подачи, структура системы, верхний уровень системы.

Belei Oksana, *PhD, Associate Professor, Department of Information and Telecommunication Technologies and Systems, Ivano-Frankivsk National Technical University of Oil and Gas, Ukraine, e-mail: oksana_kl@meta.ua, ORCID: http://orcid.org/0000-0002-2386-4106*

Zamikhovska Olena, *PhD, Associate Professor, Department of Information and Telecommunication Technologies and Systems, Ivano-Frankivsk National Technical University of Oil and Gas, Ukraine, ORCID: http://orcid.org/0000-0001-7123-4080*

Stasyuk Roman, *PhD, Associate Professor, Department of Information and Telecommunication Technologies and Systems, Ivano-Frankivsk National Technical University of Oil and Gas, Ukraine, ORCID: http://orcid.org/0000-0002-4724-6118*

UDC 004.2
DOI: 10.15587/2312-8372.2017.108845

Babakov R.

USING OF METHOD OF REPLACEMENT OF INPUT VARIABLES IN MICROPROGRAM FINITE-STATE MACHINE WITH DATAPATH OF TRANSITIONS

Запропоновано використовувати відомий метод заміни вхідних змінних для оптимізації апаратурних витрат у мікропрограмному автоматі з операційним автоматом переходів. Метод дозволяє використовувати для синтезу схеми пристрою гетерогенний елементний базис, що сприяє зменшенню апаратурних витрат в схемі автомата. В результаті застосування даного методу розроблено нову структурну модель автомата, для якої визначені критерії ефективності в порівнянні зі структурою-прототипом.

Ключові слова: *мікропрограмний автомат, операційний автомат переходів, заміна вхідних змінних, оптимізація апаратурних витрат.*

1. Introduction

One of the central parts of modern computing systems is a control unit whose characteristics largely determine the characteristics of the system as a whole [1, 2]. The increase in the complexity of computing tasks observed today increases the requirements for the speed of computing systems. One way to improve the speed of control units is to use highly effective structural solutions, one of which is a microprogram finite-state machine (FSM) [2]. FSM with a canonical structure, along with high speed, is characterized by high hardware amount for implementing the logical circuit of the device [2, 3]. This makes it urgent to develop new structures and methods for synthesizing MPA, aimed at hardware amount optimization while maintaining an acceptable speed.

At present, many methods of FSM optimization leading to various structural realizations are known [3–6]. One of these methods is the method of replacement of input variables, which makes it possible to reduce the number of different Boolean terms in the system of equations for the FSM transition functions [5, 6]. Another structural solution is a microprogram finite-state machine with datapath of transitions (FSM with DT), in which the transition function is represented as a set of partial functions and has a schematic interpretation in the form of a datapath [7, 8].

In works [2, 5, 6] the possibility of joint use of various optimization methods, allowing under certain conditions to increase the efficiency of the resulting FSM structures in terms of hardware amount, is shown. With regard to the microprogram finite-state machine with datapath of transitions, the application of known methods of hardware amount optimization is unexplained for today.

2. The object of research and its technological audit

The object of this research is a microprogram finite-state machine with datapath of transitions [7, 8], the structural diagram of which is shown in Fig. 1 and in this paper is denoted by symbol U_1.

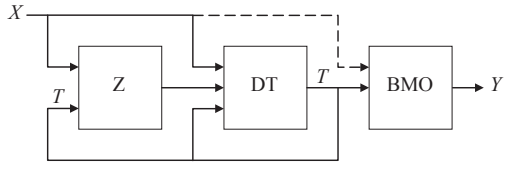

Fig. 1. Structural diagram of the microprogram finite-state machine with datapath of transitions (structure U_1)

The structure includes the following blocks (Fig. 1):
– DT: datapath of transitions that performs transformations of the state code of an FSM using a number of transition operations (TO) under the control of an TO code represented by Z signals;
– Z: forms the code of the transition operation Z on the basis of the code of the current state T and the signals of the logical conditions (LC) X;
– BMO: block of microoperations, which forms a set of microoperations Y.

The presence of connection shown in dashed lines in Fig. 1, allows to consider this structure as a Mealy FSM, the absence – as Moore FSM.

In the FSM with DT, the abstract transition function is divided into a set of partial functions, each of which implements a proper subset of FSM transitions [9]. The partitioning is performed in such manner that for each partial function, a uniform law of arithmetic-logical transformation of state codes (transition operation) can be specified, which makes it possible to realize all transitions from the corresponding subset. This law is implemented as a separate combination circuit, the hardware amount in which does not depend on the number of FSM transitions that it realizes. A set of combinational circuits corresponding to a set of partial transition functions form an operating part (OP) in which the outputs of the combinational circuits are multiplexed under control of signals Z.

OP together with the memory register (MR) form the datapath of transitions in which the MR is the single register circuit and in each operation it acts simultaneously as a register of input data and a result register (Fig. 2).

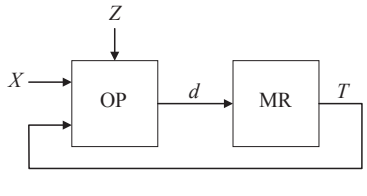

Fig. 2. Structural diagram of datapath of transitions

Let FSM be given by flow-chart of the algorithm containing a set of states $A=\{a_1,...,a_M\}$ and a set of logical conditions $X=\{x_1,...,x_L\}$ [2]. With the increasing complexity of the control algorithms implemented by FSM with DT, an increase in the number L of logic conditions appearing as input signals of the microprogram FSM is often associated. In the structure of U_1 (Fig. 1), an increase in L leads to an increase in the input signals in the block Z along the X line. In many cases, this makes it impossible to synthesize this block in the basis of memory devices (in particular, in the FPGA block memory), since it requires the use of memory devices with a large number of address inputs [2, 5].

The only acceptable basis for synthesis of the block Z is the basis of the LUT elements, which is the main resource of the FPGA crystal [10]. At the same time, the increase in hardware amount in the logical circuit of FSM with DT, associated with the implementation of the block Z in the basis of LUT elements, worsens such characteristics of the circuit as cost, size, power consumption, reliability and others.

Thus, one of the ways to reduce the hardware amount in the logical circuit of the microprogram finite-state machine with datapath of transitions is implementation of the block Z in the basis of FPGA block memory. This is possible if the number of input signals of the block Z is reduced. The latter can be achieved by increasing the number of levels of transformation of logical signals in the circuit of the FSM due to the application of the known method of replacement of input variables [5, 6].

3. The aim and objectives of research

The aim of research is to apply the method of replacement of input variables to a microprogram finite-state machine with datapath of transitions. This would allow under certain conditions to reduce the hardware amount in the logical circuit of the FSM with DT. The resulting structure may be more preferable than other known structures of FSM.

To achieve this aim it is necessary:

1. To develop a structural model of FSM with DT and replacement of input variables.

2. To determine the conditions of effectiveness of this structure in comparison with structure U_1.

3. To demonstrate the possibility of applying this approach.

4. Research of existing solutions of the problem

The development of methods for hardware optimization in the logical circuits of control units is a classic problem of the theory of digital automata. In this direction, in the literature, it is, first of all, necessary to select works [2–6] containing a number of original techniques and structural solutions oriented toward the use of a heterogeneous elemental basis.

Also within the area of this problem the following directions can be indicated:
– structural decomposition of the logical circuit of FSM [11–13];
– special coding of FSM states [14–16];
– synthesis taking into account the features of the elemental basis [17–19].

Thus, in [11], a method for converting of object codes is proposed, according to which the number of logical signal conversion levels in the FSM structure increases. At the expense of simplification of each of the levels, and also by providing the possibility of implementing the circuit in a heterogeneous elemental basis, the total hardware amount in the circuit of the automaton is reduced. Similar methods, oriented to a specific elemental basis (PAL, FPGA), are described in [12, 13].

Another approach to reducing hardware amount is the use of special methods for FSM states coding [14–16]. Moreover, in addition to reducing the cost of the circuit, in a number of cases, an increase in performance, a reduction in power consumption, an increase in noise immunity, and other positive effects are also achieved.

ISSN 2226-3780

Also, an alternative way to achieve savings in hardware amount is use of the features of the elemental basis. In modern FPGAs there are such typical functional blocks as multiplexers, registers, block and distributed memory, and others. According to the authors of works [17–19], the use of these blocks in the FSM circuit allows to reduce the number of LUT elements used, using them for the implementation of other blocks of the computing system (for example, the operating device).

Thus, the results of the analysis allow to conclude that there are many different ways to optimize the hardware amount in the circuits of digital control units. Nevertheless, the possibility and expediency of using these approaches in a microprogram finite-state machine with datapath of transitions remains unexplored today.

5. Methods of research

A specific feature of FSM is that almost always for any state $a_i \in A$ the next condition is fulfilled:

$$|X(a_i)| \ll L, \tag{1}$$

where $X(a_i)$ is a subset of X which elements significantly affect the transitions from the state a_i [6]. Execution for the given flow-chart inequality (1) allows to apply to the FSM with DT a known method of replacement of input variables corresponding to the signals of logical conditions [5, 6].

Let FSM with DT be given a flow-chart in which any transition depends on no more than G logical conditions. In this case, a lot of signals will be fed to the input of block Z and the datapath of transitions, and a special circuit M will be added to the structure of the FSM, which realizes the function:

$$P = P(X, T). \tag{2}$$

Adding a block M to the structure U_1 leads to the structure of FSM with DT and replacement of input variables (Fig. 3). Let's denote this structure by the symbol U_2.

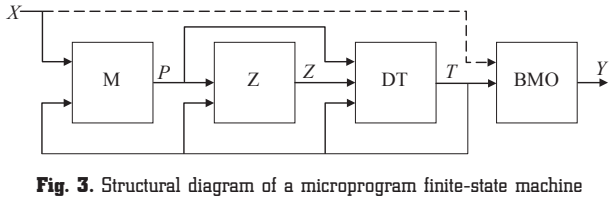

Fig. 3. Structural diagram of a microprogram finite-state machine with datapath of transitions and replacement of input variables (structure U_2)

Here block Z realizes function:

$$Z = Z(P, T), \tag{3}$$

and is constructed by the system of equations:

$$\begin{cases} z_1 = z_1(T_1, ..., T_R, p_1, ..., p_G); \\ z_2 = z_2(T_1, ..., T_R, p_1, ..., p_G); \\ ... \\ z_{R_Z} = z_{R_Z}(T_1, ..., T_R, p_1, ..., p_G), \end{cases} \tag{4}$$

where R is the digit capacity of the state code of the FSM. Block M is synthesized in accordance with the following system of equations:

$$\begin{cases} p_1 = p_1(T_1, ..., T_R, x_1, ..., x_L); \\ p_2 = p_2(T_1, ..., T_R, x_1, ..., x_L); \\ ... \\ p_G = p_G(T_1, ..., T_R, x_1, ..., x_L). \end{cases} \tag{5}$$

Let's build an inequality, in the performance of which the hardware amount in the logical circuit of FSM with U_2 structure will be less than in the circuit with the structure U_1:

$$H^{U_2} + H_Z^{U_2} + H_{OP}^{U_2} + H_{MR}^{U_2} + H_{BMO}^{U_2} < H_Z^{U_1} + H_{OP}^{U_1} + H_{MR}^{U_1} + H_{BMO}^{U_1}. \tag{6}$$

Here, each term is the numerically expressed hardware amount in the block determined by the subscript entering the FSM structure, defined by the superscript. When using the FPGA basis, LUT elements, which are regular FPGA function blocks, can be used as the hardware amount unit [10].

Assuming that $H_{MR}^{U_1} = H_{MR}^{U_2}$ and $H_{BMO}^{U_1} = H_{BMO}^{U_2}$, let's reduce inequality (6) to the following:

$$H^{U_2} + H_Z^{U_2} + H_{OP}^{U_2} < H_Z^{U_1} + H_{OP}^{U_1}. \tag{7}$$

In the left-hand side of the inequality, the additional term $H_M^{U_2}$ is the quantity corresponding to the hardware amount in the block M. When synthesizing an FSM circuit in a heterogeneous element basis, the block M can be realized with standard multiplexers or PLA with a large number of inputs and a small number of outputs [5, 6].

The method of coding of input variables assumes that the growth of hardware amount due to the addition of the block M is accompanied by their simultaneous decrease in the block Z. Thus, the coding of input variables allows to reduce the number of inputs of the block Z to the value $(G+R)$. This not only reduces expenses in the block Z, but also simplifies its implementation in the basis of memory devices. Thus, for $G \ll L$, in the general case, the inequality $H_Z^{U_2} < H_Z^{U_1}$ will be satisfied.

As for the datapath of transitions, the coding of the input variables does not fundamentally affect its internal structure. In the U_2 structure, the DT input will receive G structured (binary) signals p_i instead of L signals of logical conditions in the case of the U_1 structure. This can lead to the appearance of common circuit fragments in various combinational circuits of operational part, as a consequence, to reduction in the hardware amount in the DT circuit of the U_1 structure in comparison to the U_2 structure.

6. Research results

As the research results, we will illustrate the proposed approach using the example of synthesis of FSM with U_2 structure. Let FSM be given by flow-chart Γ marked by states of Moore FSM (Fig. 4). The flow-chart contains $M = 10$ states $a_0 - a_9$, for encoding which requires $R = 4$ bits. Let's apply to flow-chart a method of replacement of input variables. According to Fig. 4, $G = 2$, that is $P = \{p_1, p_2\}$.

Let's build a table for replacement of input variables $x_1 - x_5$ with variables p_1, p_2, for which the method

described in [5] is applicable. Result of coding of logical conditions is presented in Table 1.

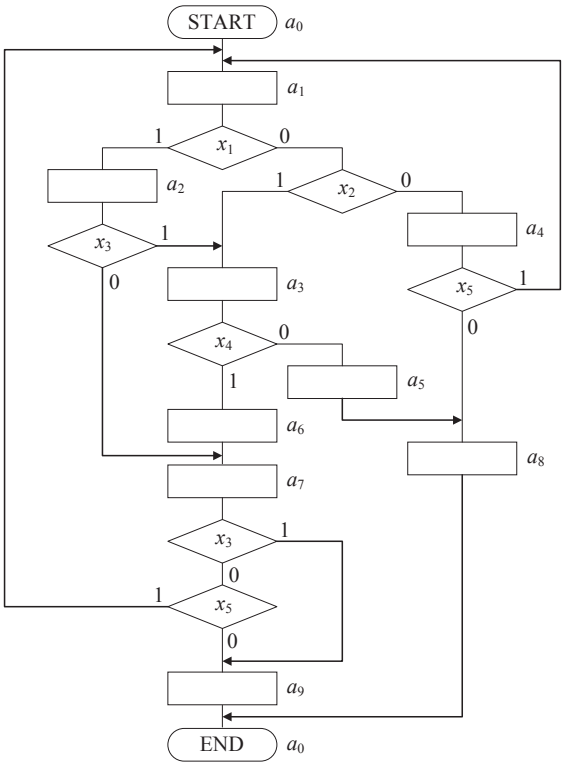

Fig. 4. Flow-chart Γ

Table 1

Table of input variables replacement (flow-chart Γ)

a_i	p_1	p_2	a_i	p_1	p_2
a_0	–	–	a_5	–	–
a_1	x_1	x_2	a_6	–	–
a_2	x_3	–	a_7	x_3	x_5
a_3	x_4	–	a_8	–	–
a_4	–	x_5	a_9	–	–

Let's perform the synthesis of the FSM according to the flow-chart, shown in Fig. 4, in which instead of variables x_1, x_3, x_4 a variable p_1 is used, instead of variables x_2, x_5 is a variable p_2.

By analogy with [20], let's code the states $a_0 - a_9$ with intermediate codes K_I from the set of integers in the range [0; 5] as follows:

$$K_I(a_0)=3,\quad K_I(a_1)=8,\quad K_I(a_2)=4,\quad K_I(a_3)=0,$$
$$K_I(a_4)=13,\quad K_I(a_5)=11,\quad K_I(a_6)=5,\quad K_I(a_7)=2,$$
$$K_I(a_8)=6,\quad K_I(a_9)=7.$$

Structural (binary) state codes represented by the binary vectors $<T_1, T_2, T_3, T_4>$, will be chosen equal to the corresponding intermediate codes in the format of four-bit binary unsigned numbers:

$$K_S(a_0)=0011,\quad K_S(a_1)=1000,\quad K_S(a_2)=0100,$$
$$K_S(a_3)=0000,\quad K_S(a_4)=1101,\quad K_S(a_5)=1011,$$
$$K_S(a_6)=0101,\quad K_S(a_7)=0010,\quad K_S(a_8)=0110,$$
$$K_S(a_9)=0111.$$

Let's choose the following transition operations defined on the set of intermediate state codes:

$$O_1:\ K_I(a^{t+1})=K_I(a^t)+5, \tag{8}$$
$$O_2:\ K_I(a^{t+1})=K_I(a^t)+11, \tag{9}$$
$$O_3:\ K_I(a^{t+1})=K_I(a^t)\times 4, \tag{10}$$
$$O_4:\ K_I(a^{t+1})=K_I(a^t)/2. \tag{11}$$

When using this TO, let's assume that the result of each operation is always reduced to the range [0; 15], which is expressed by the mathematical operation «mod 16» (taking the remainder of the integer division by 16).

Let's code operations $O_1 - O_4$ with unique two-digit codes $< z_1, z_2 >$:

$$K_Z(O_1)=00,\quad K_Z(O_2)=01,\quad K_Z(O_2)=10,\quad K_Z(O_3)=11.$$

Let's associate with each FSM transition one of the TO $O_1 - O_4$ as shown in the table of transitions (Table 2) [2, 5].

In Table 2, the O_i column contains the transition operation performed during transition from state a_m by condition P_h.

In graphical form, the synthesis result is shown in Fig. 5.

Table 2

Table of transitions of microprogram finite-state machine with structure U_2

a_m	$K_S(a_m)$	a_s	$K_S(a_s)$	P_h	O_h	Z_h	h
a_0	0011	a_1	1000	1	O_1	–	1
a_1	1000	a_2	0100	p_1	O_4	z_1z_2	2
		a_3	0000	$\bar{p}_1p_2$	O_3	z_1	3
		a_4	1101	$\bar{p}_1\bar{p}_2$	O_1	–	4
a_2	0100	a_3	0000	p_1	O_3	z_1	5
		a_7	0010	$\bar{p}_1$	O_4	z_1z_2	6
a_3	0000	a_5	1011	p_1	O_2	z_2	7
		a_6	0101	$\bar{p}_1$	O_1	–	8
a_4	1101	a_1	1000	p_2	O_2	z_2	9
		a_8	0110	$\bar{p}_2$	O_4	z_1z_2	10
a_5	1011	a_8	0110	1	O_2	z_2	11
a_6	0101	a_7	0010	1	O_4	z_1z_2	12
a_7	0010	a_9	0111	p_1	O_1	–	13
		a_1	1000	$\bar{p}_1p_2$	O_3	z_1	14
		a_9	0111	$\bar{p}_1\bar{p}_2$	O_1	–	15
a_8	0110	a_0	0011	1	O_4	z_1z_2	16
a_9	0111	a_0	0011	1	O_4	z_1z_2	17

According to Table 2, a system of equations (4) is constructed, each equation of which is defined by the following expression:

$$z_i=\bigvee_{h=1}^{H} C_{ih}A_m^hP_h. \tag{12}$$

In expression (12) the following notation is used:

– C_{ih}: a boolean variable that takes true value if and only if the variable z_i is written in row h and column Z_h of the table;

ISSN 2226-3780

– A_m^h: the value of the column $K_S(a_m)$ in the row h;
– P_h: conjunctive term, formed by variables from the set P, indicated in the column P_h of the row h of the table.

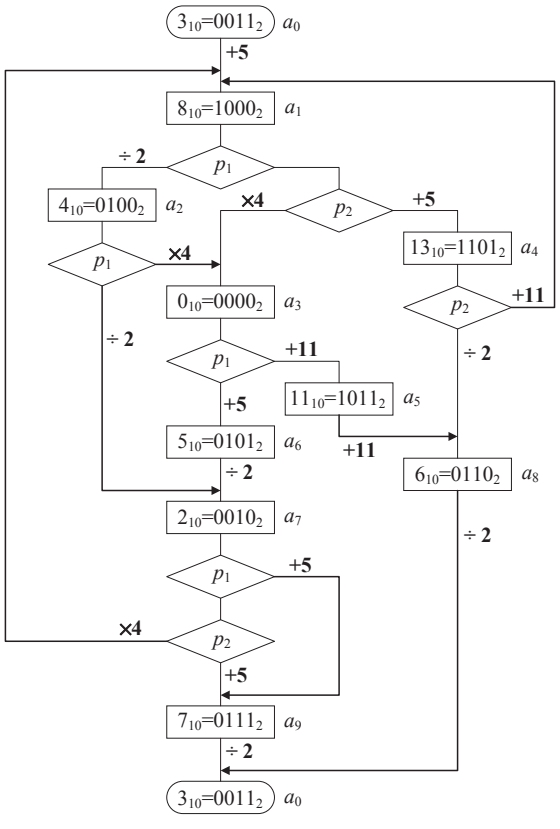

Fig. 5. Result of synthesis of finite-state machine with structure U_2

To synthesize the block M, it is necessary to construct a system of equations (5) according to Table 1. The procedure for synthesizing the block M, taking into account the features of the used elemental basis, is described in detail in [2].

The considered example of synthesis of FSM with DT with U_2 structure allows to make a conclusion about the possibility of using the method of replacement of input variables in the microprogram finite-state machine with datapath of transitions. The condition for the expediency of applying this approach is determined by the expression (7).

7. SWOT analysis of research results

Strength. The advantage of the proposed structure of the microprogram finite-state machine with datapath of transitions and replacement of input variables is the smaller number of input signals of the block Z in comparison with the prototype structure. This makes it possible to use for the synthesis of the block Z a basis of memory devices, which can be represented by FPGA block memory modules.

At the same time, the block M added to the structure can be realized in the basis of multiplexers, which are also standard functional nodes of modern FPGAs. The saving of LUT elements due to this allows them to be used for the synthesis of other FSM or a computing system units. Ultimately, this approach makes it possible to synthesize, on the basis of the same FPGA series, FSM of a higher complexity, or to use microchips of lower complexity for a given FSM.

Weakness. A decrease in the number of block Z inputs in the proposed structure of FSM with DT is possible due to the introduction of an additional block M. If the used FPGA does not contain a sufficient number of multiplexers, a basis of LUT elements can be used to implement the block M, which can level the hardware decrease in the block Z. In any case, the effectiveness of the proposed structure by the criterion of hardware amount in comparison with the prototype structure is not absolute and is different in each specific case. If for the given FSM the structure proposed in this paper has more hardware amount than the prototype structure, then a prototype structure (or other, more optimal FSM structure that meets the specified design criteria) should be used to design the FSM.

Opportunities. Practical use of control units on the basis of a microprogram finite-state machine with datapath of transitions and replacement of input variables requires the development of formalized methods for synthesizing this structure. Also, the area of its effective application, expressed by a set of values or ranges of values of the parameters of the FSM, must be determined. The modern process of designing of digital devices is impossible without the use of specialized CAD systems, which leads to the problem of developing a synthesizable VHDL model oriented to a certain elemental basis (for example, FPGA) [9]. Also, using a similar model for a given FSM, the hardware amount, expressed, for example, in the number of LUT elements and used FPGA internal memory units, can be experimentally determined.

Threats. At present, many FSM structures are known that are oriented to optimization of various parameters of the FSM circuit. At the same time, there are still no unambiguous ways of selecting one or another FSM structure in each particular case. Attempts to develop specialized CADs, allowing in automatic or automated mode to select the optimal FSM structure from a variety of known structures in accordance with specified design criteria, remain unrealized. The choice of the optimal structure «manually» requires the designer to know a large number of different structures and methods for their synthesis. At the same time, there is always a risk that the structure chosen by the designer is not optimal in a particular case. This negatively affects the cost of finished products based on FSM with chosen structure.

Thus, SWOT analysis of research results allows to identify the main directions for further research. Among them: development of a formal method for synthesizing a microprogram finite-state machine with datapath of transitions and replacement of input variables, constructing a synthesizable VHDL model of this structure and determining the area of its effective use.

8. Conclusions

1. In this article, a new structure of a microprogram finite-state machine with datapath of transitions that uses the principle of changing input variables is proposed. In this structure, the number of input variables of the block

Z decreases, which contributes to a decrease in the number of components of Boolean terms in the system of equations realized by the block Z. As a result, there is a decrease in hardware amount in the block Z, while increasing amount due to the introduction of an additional block M in the structure of the FSM with DT, which replaces the input variables.

2. The condition of efficiency of the proposed structure is formulated according to the criterion of hardware amount in comparison with the known structure of FSM with DT without replacement of input variables. The condition is determined by the inequality (7), each part of which is the sum of hardware amount in individual blocks of corresponding structure.

3. An example of synthesis of FSM with DT and replacement of input variables by flow-chart is considered. This example reflects the main stages in the synthesis of the FSM and can later be used to develop a formalized method for synthesizing this structure.

References

1. Glushkov, V. M. Sintez tsifrovyh avtomatov [Text] / V. M. Glushkov. – Moscow: Fizmatgiz, 1962. – 476 p.
2. Baranov, S. I. Sintez mikroprogrammnyh avtomatov [Text] / S. I. Baranov. – Leningrad: Energiia, 1979. – 232 p.
3. Drozdov, E. A. Optimizatsiia struktur tsifrovyh avtomatov [Text] / E. A. Drozdov. – Moscow: Soetskoe radio, 1975. – 352 p.
4. Barkalov, A. A. Sintez mikroprogrammnyh ustroistv upravleniia [Text] / A. A. Barkalov, A. V. Palagin. – Kyiv: Institute of Cybernetics of NAS of Ukraine, 1997. – 135 p.
5. Barkalov, A. A. Sintez ustroistv upravleniia na programmiruemyh logicheskih ustroistvah [Text] / A. A. Barkalov. – Donetsk: Donetsk National Technical University, 2002. – 262 p.
6. Baranov, S. I. Tsifrovye ustroistva na programmiruemyh BIS s matrichnoi strukturoi [Text] / S. I. Baranov, V. A. Skliarov. – Moscow: Radio i sviaz', 1986. – 272 p.
7. Barkalov, A. A. Operational formation of state codes in microprogram automata [Text] / A. A. Barkalov, R. M. Babakov // Cybernetics and Systems Analysis. – 2011. – Vol. 47, No. 2. – P. 193–197. doi:10.1007/s10559-011-9301-y
8. Babakov, R. Research of efficiency of microprogram final-state machine with datapath of transitions [Text] / R. Babakov, A. Barkalov, L. Titarenko // 2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). – IEEE, 2017. – P. 203–206. doi:10.1109/cadsm.2017.7916115
9. Babakov, R. M. Algebraic Interpretation of a Microprogram Finite-State Machine with Datapath of Transitions [Text] / R. M. Babakov, A. A. Barkalov. – Cybernetics and Systems Analysis. – 2016. – Vol. 52, No. 2. – P. 191–198. doi:10.1007/s10559-016-9814-5
10. Grushvitskii, R. I. Proektirovanie sistem na mikroshemah programmiruemoi logiki [Text] / R. I. Grushvitskii, A. H. Mursaev, E. P. Ugriumov. – St. Pererburg: BHV-Peterburg, 2002. – 608 p.
11. Barkalov, A. A. Structural decomposition as a tool for the optimization of an FPGA-based implementation of a mealy FSM [Text] / A. A. Barkalov, L. A. Titarenko, A. A. Barkalov // Cybernetics and Systems Analysis. – 2012. – Vol. 48, No. 2. – P. 313–322. doi:10.1007/s10559-012-9410-2
12. Kania, D. Decomposition-based synthesis and its application in PAL-oriented technology mapping [Text] / D. Kania // Proceedings of the 26th Euromicro Conference. EUROMICRO 2000. Informatics: Inventing the Future. – Maastricht: IEEE Compuetr Society Press, 2000. – P. 138–145. doi:10.1109/eurmic.2000.874626
13. Rawski, M. The Influence of Functional Decomposition on Modern Digital Design Process [Text] / M. Rawski, T. Luba, Z. Jachna, P. Tomaszewicz // Design of Embedded Control Systems. – Boston: Springer US, 2005. – P. 193–204. doi:10.1007/0-387-28327-7_17
14. Amann, R. Optimal state chains and state codes in finite state machines [Text] / R. Amann, U. G. Baitinger // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 1989. – Vol. 8, No. 2. – P. 153–170. doi:10.1109/43.21834
15. Bacchetta, P. Low-power state assignment techniques for finite state machines [Text] / P. Bacchetta, L. Daldos, D. Sciuto, C. Silvano // 2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No.00CH36353). – 2000. – Vol. 2. – P. 641–644. doi:10.1109/iscas.2000.856410
16. Czerwinski, R. B09: State assignment method for high speed FSMS [Text] / R. Czerwinski, D. Kania // IFAC Proceedings Volumes. – 2004. – Vol. 37, No. 20. – P. 216–221. doi:10.1016/s1474-6670(17)30599-2
17. Barkalov, A. A. Partial reconfiguration of compositional microprogram control units implemented on FPGAs [Text] / A. A. Barkalov, M. Wegrzyn, R. Wiszniewski // IFAC Proceedings Volumes. – Vol. 39, No. 21. – P. 116–119. doi:10.1016/s1474-6670(17)30169-6
18. Bomar, B. W. Implementation of microprogrammed control in FPGAs [Text] / B. W. Bomar // IEEE Transactions on Industrial Electronics. – 2002. – Vol. 49, No. 2. – P. 415–422. doi:10.1109/41.993275
19. Solovjev, V. Synthesis of sequential circuits on programmable logic devices based on new models of finite state machines [Text] / V. Solovjev // Proceedings Euromicro Symposium on Digital Systems Design. – IEEE Compuetr Society Press, 2001. – P. 170–173. doi:10.1109/dsd.2001.952274
20. Babakov, R. M. Intermediate algebra of transitions in microprogram final-state machine [Text] / R. M. Babakov // Radio Electronics, Computer Science, Control. – 2016. – No. 1. – P. 64–73. doi:10.15588/1607-3274-2016-1-8

ПРИМЕНЕНИЕ МЕТОДА ЗАМЕНЫ ВХОДНЫХ ПЕРЕМЕННЫХ В МИКРОПРОГРАММНОМ АВТОМАТЕ С ОПЕРАЦИОННЫМ АВТОМАТОМ ПЕРЕХОДОВ

Предложено использовать известный метод замены входных переменных для оптимизации аппаратурных затрат в микропрограммном автомате с операционным автоматом переходов. Метод позволяет использовать для синтеза схемы устройства гетерогенный элементный базис, что способствует уменьшению аппаратурных затрат в схеме автомата. В результате применения данного метода разработана новая структурная модель автомата, для которой определены критерии эффективности в сравнении со структурой-прототипом.

Ключевые слова: микропрограммный автомат, операционный автомат переходов, замена входных переменных, оптимизация аппаратурных затрат.

Babakov Roman, *PhD, Associate Professor, Department of Applied Mechanics and Computer Technologies, Vasyl' Stus Donetsk National University, Vinnytsia, Ukraine, e-mail: r.babakov@donnu.edu.ua, ORCID: http://orcid.org/0000-0001-7196-0912*