Розробка засобів обчислювальної техніки та дослідження комп'ютерних мереж

УДК 004.274

A.A. Barkalov[1,2], professor,
L.A. Tytarenko[2], professor,
K.N. Yefimenko[1], PhD.,
I.J. Zeleneva[1], PhD.
[1]Donetsk National Technical University, Ukraine
[2]University of Zielona Gora,Poland,
Abar54@mail.ru, Irina@cs.dgtu.donetsk.ua

Optimization of the Addressing Scheme in Compositional Microprogram Control Unit with Code Sharing

A method for reducing the hardware amount in the scheme of compositional microprogram control units (CMCU) with code sharing is proposed oriented to FPGA technology. The method is based on the use of three sources for encoding of pseudoequivalent OLC classes, and a multiplexer, which allows choosing one of these sources. Such an approach would reduce the number of LUT elements in the addressing circuit of CMCU. Application of the specified method to the finite state machine of addressing CMCU, under its implementation with FPGA, leads to reduction of the number of LUT-elements in the circuit of the control unit. Results of the research of the developed structures are represented, allowing defining their efficiency and the area of optimum application. An example of the proposed method application is given.

Key words: compositional microprogram control units, GSA, operational linear chains, FPGA, LUT, logic circuit.

Introduction

The compositional microprogram control units (CMCU) are an effective tool for implementation of the linear control algorithms [1,2]. One of the CMCU models is the model with sharing of the codes [3], allowing under certain conditions to reduce hardware expenses in the scheme of microcommands addressing. Now chips like FPGA (field–programmable gate arrays) are widely used for implementation of digital devices schemes [4,5]. The basis of these VLSI's (Very Large Scale Integration circuits) is represented by the macrocells of the plate type, called LUT (look–up table). As a rule, LUT-elements have limited number of inputs (4-6) [6,7]. For the reduction of the number of LUT in the scheme CMCU it is necessary to reduce the number of arguments and terms in system of functions of addressing of microcommands [1,8]. In the real operation one of the approaches to the solution of this task, based on multiplexing of three sources of codes of classes of the pseudo-equivalent operator linear circuits (OLC) is offered. The offered method is the development of the results obtained in [9,10].

The aim of this research is the reduction of the number of LUT-elements in the scheme CMCU with division of codes at the expense of multiplexing of sources of codes of classes of the pseudo-equivalent OLC's.

Objective of the research is to develop a method of synthesis of CMCU with division of codes, allowing optimizing the addressing scheme of microcommands.

The control algorithm is provided in the form of the algorithm graphs scheme (GSA) [8]. This choice is defined by visualization of similar representation and wide use of the device GSA in practice of engineering design.

Compositional MCU (microprogram control unit) with division of codes

Let GSA G = G(B, E) be provided by sets of nodes B and arcs E connecting them. Let $B = b_0 \cup b_E \cup E_1 \cup E_2$, where b_0 is the initial node, b_E is the finite node, E_1 –is a set of operator nodes and E2 is a set of the conditional nodes of GSA G. In the nodes of operator $b_q \in E_1$ the sets of microoperations $Y(b_q) \subseteq Y$ are recorded, where $Y = \{y_1,\ldots,y_N\}$ is a set of microoperations. Let us introduce some definitions [2].

Definition 1. An operational linear chain of GSA G is the final sequence of operational nodes $\alpha_g = \langle b_{g1},\ldots,b_{gF_g}\rangle$, such that for any pair of neighboring components b_{gi}, b_{gi+1}, where i is a component of train α_g, exists an arch $\langle b_{gi}, b_{gi+1}\rangle \in E$.

Definition 2. Node $b_q \in D^g$, where D^g is a set of the nodes entering OLC α_g, is called input OLC α_g, if there is an arch $\langle b_t, b_q \rangle \in E$, where $b_t \notin D^g$.

Definition 3. Node $b_q \in D^g$ is called output OLC α_g, if there is an arch $\langle b_q, b_t \rangle \in E$, where $b_t \notin D^g$.

Definition 4. OLC α_i, α_j are called pseudo-equivalent OLC, if their outputs are connected with an input of the same node $b_q \in B$

Let for some GSA G a set of OLC $C = \{\alpha_1, \ldots, \alpha_G\}$ be created defining partition in a set E_1 [3], and let $|E_1| = M$. Let us match each node $b_q \in E_1$ with the microcommand MI_q with the address $A(b_q)$ having digit capacity

$$R = \lceil \log_2 M \rceil \quad (1)$$

Let $F_{max} = \max(F_1, \ldots, F_G)$ be the maximum number of components in OLC.

Let us encode each OLC $\alpha_g \in C$ with the binary code $K(\alpha_g)$, having R_1 of discharges, where

$$R_1 = \lceil \log_2 G \rceil \quad (2)$$

For determination of any node $b_q \in D^g$ R_2 of the discharges representing a code $K(b_q)$ will be enough. Thus

$$R_2 = \lceil \log_2 F_{max} \rceil \quad (3)$$

Let the following condition be satisfied for GSA G:

$$R_1 + R_2 = R \quad (4)$$

In this case for the implementation of algorithm G it is expedient to use the CMCU model with division of codes (fig. 1).

For encoding OLC this model uses variables $\tau_r \in \tau$, where $|\tau| = R_1$. Codes of components are selected so that natural addressing of microcommands [1] could be executed. For this purpose the code of the first component of any OLC is equal to 0, the second is 1 and so on. It is natural that these decimal numbers are provided by their binary R_2-digit equivalents.

Let us designate CMCU (fig. 1) with U_1.

In CMCU U_1 the addressing scheme of microcommands (ASM) implements a system of functions of excitation of C counter and Tr trigger

$$\Phi = \Phi(\tau, X),$$
$$\Psi = \Psi(\tau, X). \quad (5)$$

Thus, the address of the microcommand is as follows

$$A(b_q) = K(\alpha_g) * K(b_q), \quad (6)$$

where the node b_q is a part of OLC $\alpha_g \in C$, * – a concatenation sign.

Composition MCU U_1 functions as follows. On Start signal in Tr and C the initial address of the microprogram is skidded, and the trigger of selection of TB is set in a single status. In this case Fetch = 1 and it allows selecting the commands of the control memory (CM). If the read microcommand does not correspond to OLC output, the signal y_0 is created at the same time with microoperations $Y(b_q)$. If $y_0 = 1$, a unit is added to the contents of C and the next component of the current OLC is addressed. If the output of OLC is reached, $y_0 = 0$. Thus, the input address of the next OLC is created by the scheme ASM. In case of achieving the termination of the microprogram y_E signal is created, TB trigger is nullified, and the selection of microcommands stops.

The number of LUT elements in the scheme ASM depends on the number of arguments and terms in the system (5). The paper offers a method, which allows reducing the complexity of functions in the system (5) and reducing hardware expenses in the scheme ASM.

The main idea of the offered method

Let OLC $\alpha_g \in C_1$, if Og is not connected to the finite top of GSA G. Let us find the partition $\Pi_C = \{B_1, \ldots, B_I\}$ of the set C1 into classes of pseudo-equivalent OLC (POLC). Let us perform coding $\alpha_g \in C$ so that the greatest possible number of classes $B_i \in \Pi_C$, where $|\Pi_C| = I$, would be represented by one generalized interval of R_1-dimensional Boolean space. Let n_i be the number of generalized intervals representing a class. Let us provide a set Π_C as $\Pi_C = \Pi_A \cup \Pi_B$. Thus, the sets Π_A and Π_B are built as follows:

$$(n_i = 1) \rightarrow B_i \in \Pi_A,$$
$$(n_i > 1) \rightarrow B_i \in \Pi_B. \quad (7)$$

The source of codes of classes $B_i \in \Pi_A$ is the register Tr. Thus, class code $B_i \in \Pi_A$ is defined by an appropriate interval of R_1-dimensional Boolean space.

Let us encode classes $B_i \in \Pi_B$ with binary codes with digit capacity $C(B_i)$

$$R_3 = \lceil \log_2(|\Pi_B| + 1) \rceil. \quad (8)$$

We use for coding of classes $B_i \in \Pi_B$ variables from the set $Z = \{z_1, \ldots, z_{R_3}\}$. The block of the converter of codes (BCC) is necessary for the formation of codes with $C(B_i)$.

This block realises the system of functions

$$Z = Z(\tau) \quad (9)$$

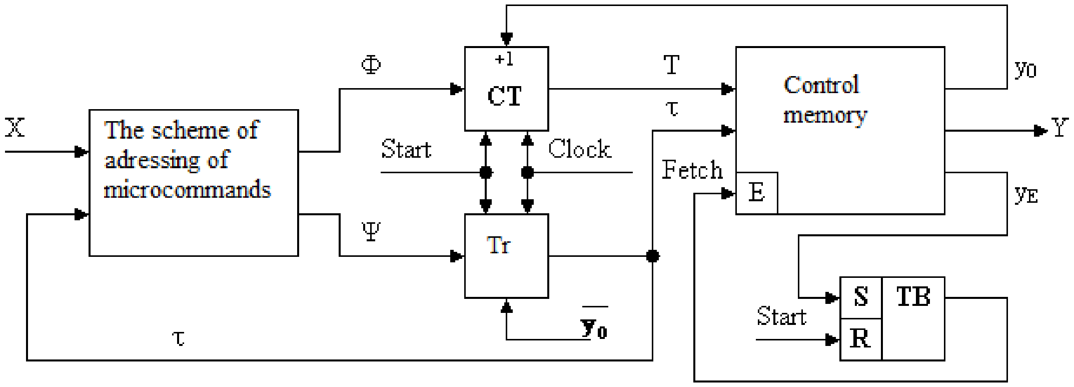


Figure 1 – The block scheme CMCU with division of codes

Modern microcircuits FPGA incorporate blocks of built-in memory EMB (embedded memory block) [6,7]. These blocks have possibility of reconfiguration, which is reduced to the change in the number of inputs and outputs at fixed capacity V_0.

$$V_0 = 2^S \cdot t_F \tag{10}$$

In formula (10) variable S is the number of inputs, and t_F is the number of exits EMB. As a rule, the following configurations of modern EMB [6, 7] are possible: 16K×1, 8K×2, 4K×4, 2K×8, 1K×16, 512×32, 256×64 (bits).

This model has a number of differences from the model U_1. First, the unit ASM is divided into two units. The unit ASM_1 implements the transitions determined by a set of ΠB, and the unit ASM_2 – by a set of Π_A. The MSC multiplexer serves for the choice of a source of codes, using a variable Ex. Ex determined by the value of the variable state of the trigger TM, controlled by the additional variable y_M. The unit BCC is a source of codes of classes for the circuit ASM_1, creating a part of this code. The second source of code $K(B_i)$ for $B_i \in \Pi_B$ is the control memory CM. Source of codes for the circuit ASM_2 is the register of Tr.

It means that parameters S and t_F belong to the following sets: $S \in \{14,13,\ldots 8\}$ and $t_F \in \{1,2,4,8,16,32,64\}$. In case of the fixed value of t_F the number of cells in EMB is defined by the following formula:

$$V = \lceil V_0 / t_F \rceil \tag{11}$$

For the implementation of CM, M of cells of EMB are enough, thus, the unit has t_M of outputs:

$$t_M = \lceil V_0 / M \rceil \tag{12}$$

Let the following inequality take place for some GSA G and a microcircuit of FPGA:

$$N + 3 + R_3 > t_M > N + 3. \tag{13}$$

In this case $\Delta R = t_M - (N+3)$ discharges of a code $K(B_i)$ it is expedient to create on free outputs of EMB ($B_i \in \Pi_B$).

Remained $R_3 - \Delta R$ of discharges of the code are created by the circuit BCC. It is equivalent to the representation of a set of Z as follows:

$$Z = Z^1 \cup Z^2 . \tag{14}$$

Variables $Z_r \in Z^1$ are created by BCC, $Z_r \in Z^2$ and variables – by CM. It is obvious that the intersection of these sets is empty.

In this case for the implementation of the scheme of a control unit the CMCU U_2 model (fig. 2) is offered.

On Start signal in Tr and ST zero codes (the address of the first microcommand) are skidded, and TF and TM trigger are set respectively in “1” (Fetch=1) and “0” (Ex = 0). Until the address of the input is reached, U_2 functions as U_1. When the address of the output of OLC $\alpha_g \in B_i$ is reached, the variable $y_M = 1$ can be created. In this case Ex = 1 and the jump address is created by the circuit ASM_1.

For this purpose a system of functions is created:

$$\begin{aligned} \Phi^1 &= \Phi^1(Z, X^1); \\ \Psi^1 &= \Psi^1(Z, X^1). \end{aligned} \tag{15}$$

If $B_i \in \Pi_A$, the variable y_M is not created. Thus Ex = 0 and a jump address is defined by the circuit ASM_2.

For this purpose a system of functions is created:

$$\begin{aligned} \Phi^2 &= \Phi^2(\tau, X^2); \\ \Psi^2 &= \Psi^2(\tau, X^2). \end{aligned} \tag{16}$$

Appropriate functions are transferred to the output of MSC and required codes $K(\alpha_g)$ and $K(b_q)$ are loaded in Tr and CT respectively. Functioning proceeds normally up to y_E variable formation (achievement of control algorithm termination).

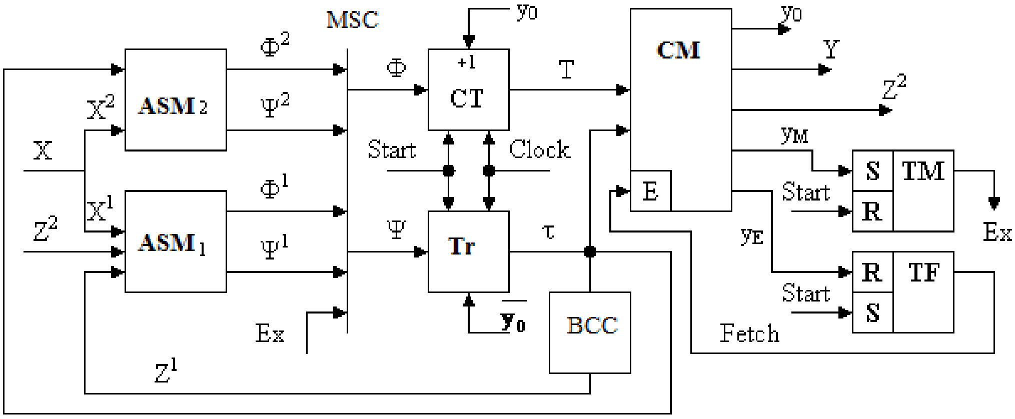


Figure 2 – The block scheme CMCU U_2

Such approach allows reducing the number of terms in system (5) to absolute minimum. If

$$R_3 < R_1 \tag{17}$$

the number of arguments in system (15) decreases in comparison with respective functions of system (5). The drawbacks of this approach are the existence of units MSC and BCC, consuming some resources of the crystal, and the increase in digit capacity of words of CM. However, the scheme MSC is implemented using three-stable outputs of macrocells, therefore additional LUT elements are not required.

Obviously, the application of the offered method makes sense if the number of LUT elements in the unit BCC is much less than the parameter Δ_{LUT}. Parameter Δ_{LUT} is defined by the difference of the number of LUT elements in the unit and the units ASM_1 and ASM_2.

Implementation of the circuit CMCU U_2

In real operation the method of synthesis of CMCU U_2, including the following stages, is offered:

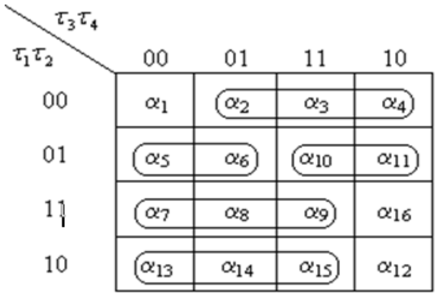
1. Formation for GSA G of sets of C, C_1, and $\Pi_{C.}$
2. Optimum coding of OLC $\alpha_g \in C_1$ and coding OLC components.
3. Formation of sets Π_A and Π_B.
4. Coding of classes $B_i \in \Pi_B$ by codes C(Bi).
5. Formation of the transition table for classes $B_i \in \Pi_A$.
6. Formation of the transition table for classes $B_i \in \Pi_B$.
7. Formation of contents of the control memory.
8. Formation of the truth table for unit BCC.
9. Scheme CMCU synthesis in the given base.

Stages 1-4 are executed by known techniques [1-3]. The stage 9 is connected with the development of the CMCU VHDL models and the use of standard industrial packets [6, 7]. These stages are not of special interest in the illustration of the diagram CMCU U_2 synthesis. So we do not consider these stages in our article.

Let for some GSA G_1 the set of OLC $C = \{\alpha_1,\ldots,\alpha_{16}\}$ and partition $\Pi_C = \{B_1,\ldots,B_7\}$ be obtained. Thus, $\alpha_{16} \notin C_1$, and classes of $B_i \in \Pi_C$ are defined as follows: $B_1 = \{\alpha_1\}$, $B_2 = \{\alpha_2,\alpha_3,\alpha_4\}$, $B_3 = \{\alpha_5,\alpha_6\}$, $B_4 = \{\alpha_7,\alpha_8,\alpha_9\}$, $B_5 = \{\alpha_{10},\alpha_{11}\}$, $B_6 = \{\alpha_{12}\}$, $B_7 = \{\alpha_{13},\alpha_{14},\alpha_{15}\}$. So, G = 16, R_1 = 4, $\tau = \{\tau_1,\ldots,\tau_4\}$.

Optimum coding of OLC is executed so that the greatest possible number of classes $B_i \in \Pi_C$ could be represented by one interval of R_1-dimensional boolean space [1]. One of options of coding is provided in fig. 3.

From Karnaugh map (fig. 3) it is possible to receive the following intervals defining classes $B_i \in \Pi_C$. The class B_1 is defined by interval 0000; class B_2 – by intervals 00*1 and 001*; class B_3 – by interval 010*; class B_4 – by intervals 110* and 11*1; class B_5 – by interval 011*; class B_6 – by interval 1010; class B_7 – by intervals 100* and 10*1.



$\tau_1\tau_2$ \ $\tau_3\tau_4$	00	01	11	10
00	α_1	α_2	α_3	α_4
01	α_5	α_6	α_{10}	α_{11}
11	α_7	α_8	α_9	α_{16}
10	α_{13}	α_{14}	α_{15}	α_{12}

Figure 3 – OLC codes $\alpha_g \in C$

Thus, $\Pi_A = \{B_1, B_3, B_5, B_6\}$, where $K(B_1)$ = 0000, $K(B_3)$ = 010*, $K(B_5)$ = 011* and $K(B_6)$ = 1010. Obviously, $\Pi_B = \{B_2, B_4, B_7\}$ and their coding requires $R_3 = 2$ discharges. So, Z = $\{z_1,z_2\}$. Let's encode classes $B_i \in \Pi_B$ as follows: $C(B_2)$ = 00, $C(B_4)$ = 01, $C(B_7)$ = 10. The transition table is created on the basis of the generalized formulas of transitions [1]. For example, from GSA G1 it is possible to obtain the following formulas:

$$B_1 \to x_1 b_3 \vee \overline{x}_1 x_2 b_8 \vee \overline{x}_1 \overline{x}_2 b_{12}; \qquad (18)$$
$$B_2 \to x_4 b_{12} \vee \overline{x}_4 b_{17}.$$

Columns of the transition table include the following information: initial class (Bi column); class code (for Π_A it is code K (Bi), and for Π_B it is C (Bi));

transition address ($A(b_q)$); logical conditions determining transition (X_h column); functions of excitation of triggers of the Tr register (column Ψ_h^1 for Π_B and Ψ_h^2 for Π_A); functions of excitation of triggers of the CT counter (column Φ_h^2 for Π_A and Φ_h^1 for Π_B); transition number (column h).

Let $A(b_3) = 000100$, $A(b_8) = 001000$, $A(b_{12}) = 010101$, $A(b_{17}) = 110010$. Then the fragments of transition tables for formulas (13) are given in tab. 1 and tab. 2.

Table 1. Transition table for class $B_1 \in \Pi_A$

B_i	$K(B_i)$	$A(b_q)$	X_h	Ψ_h^2	Φ_h^2	h
B_1	0000	000100	x_1	D_4	–	1
		001000	$\overline{x}_1 x_2$	D_3	–	2
		010101	$\overline{x}_1 \overline{x}_2$	$D_2 D_4$	D_6	3

Table 2. The transition table for class $B_2 \in \Pi_B$

B_i	$C(B_i)$	$A(b_q)$	X_h	Ψ_h^1	Φ_h^1	h
B_2	00	010101	x_4	D_2D_4	D_6	1
		110010	$\overline{x}_4$	D_1D_2	D_5	2

The system (15) can be obtained from the transition table for classes $B_i \in \Pi_B$. So, from tab. 2 we have, for example,

$$D_1 = \overline{z}_1 \overline{z}_2 \overline{x}_4 ; \; D_6 = \overline{z}_1 \overline{z}_2 x_4 .$$

The system (16) can be obtained from the transition table for classes $B_i \in \Pi_A$. So, from tab. 1 we have, for example,

$$D_2 = \overline{\tau}_1 \overline{\tau}_2 \overline{\tau}_3 \overline{\tau}_4 \overline{x}_1 \overline{x}_2 ; \; D_6 = \overline{\tau}_1 \overline{\tau}_2 \overline{\tau}_3 \overline{\tau}_4 \overline{x}_1 \overline{x}_2 .$$

From the obtained example follow such sets: $Z^1 = \{Z_1\}$ and $Z^2 = \{Z_2\}$. In this case BCC implements only a part of the code $K(B_i)$. In this case Z_1 discharge is implemented only.

The table BCC has columns: Bi, K $(B_i)_j$, C (B_i), Z_h, h. Here K $(B_i)_j$ is the code of the class Bi corresponding to one of the generalized intervals; Z_h is the discharges of the code C (B_i) accepting single value in h^{th} row of the table. For the considered example, the number of lines $H_{ПК} = 6$. This parameter is defined by the sum number of the intervals representing the codes of classes $B_i \in \Pi_B$. For our example the unit BCC is provided in tab. 3.

Table 3. Table of address transformer unit

B_i	$K(B_i)_j$	$C(B_i)$	Z_h	h
B_2	00*1	00	–	1
	001*		–	2
B_4	110*	01	–	3
	11*1		–	4
B_7	100*	10	z_1	5
	10*1		z_1	6

From tab. 3 we have system (9): $z_1 = \tau_1 \overline{\tau}_2 \overline{\tau}_3 \vee \tau_1 \overline{\tau}_2 \tau_4$.

It should be mentioned that for the approach offered in [10], BCC should also implement the equation $z_2 = \tau_1 \tau_2 \overline{\tau}_3 \vee \tau_1 \tau_2 \tau_4$.

Implementation of CM unit has the following features. The part of the code $K(B_i)$, determined by a set of Z_2, is located in cells of CM, the addresses of which correspond to OLC outputs. This stage is executed in a trivial way and is not considered in this article.

Conclusion

The method of optimization of CMCU offered in the article is based on multiplexing three sources of codes of classes of the pseudo-equivalent OLC. Such approach allows reducing the number of terms in the system of functions of excitation of triggers of the register and the counter of addresses of microcommands to the greatest possible value. If the CMCU with division of codes is considered as a Moore machine, the offered approach allows reducing the number of terms to the value of this parameter at the equivalent Mealy machine. Besides, the number of LUT elements decreases in the circuit of the transformer of codes, as not all the addresses of outputs of OLC are subject to conversion and not all discharges of codes of classes are created.

The drawback of the offered approach is introduction of the multiplexer, which enters an additional time delay in a cycle of operation of CMCU. However, reduction of number of terms leads to the reduction of the number of levels in the circuit and the time delay from introduction of MSC is compensated. The researches conducted by the authors showed that the offered method allows reducing to 40% the number of LUT elements in relation to the initial CMCU. Thus, the time of the cycle of the CMCU U_2 time was always less, than at the CMCU U_1.

Scientific novelty of the offered method consists in the use of the features of CMCU (existence of classes of the pseudo-equivalent OLC) for reduction of the number of LUT elements in the circuit CMCU with division of codes.

The practical significance of the method is in the reduction of the space of a crystal of FPGA, occupied by the circuit CMCU that allows obtaining the circuits of smaller cost, than the analogs known from literature.

References

1. Barkalov A. Logic synthesis for compositional microprogram control units / A.Barkalov, L.Titarenko. – Berlin: Springer, 2008. – 272 p.

2. Баркалов А.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС / А.А. Баркалов, Л.А. Титаренко. – Донецк: УНИТЕХ, 2009. –336 с.

3. Barkalov A. Logic synthesis for FSM-based control units / A. Barkalov., L. Titarenko. – Berlin: Springer, 2009. – 233 p.

4. Maxfield S. The Design Warrior's Guide to FPGAs / S. Maxfield. – Amsterdam: Elsevier, 2004. – 541 p.

5. Грушвицкий Р.И. Проектирование систем на микросхемах с программируемой структурой / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов. – С-Пб: БХВ – Петербург, 2006. – 736 с.

6. Електронний ресурс. – Режим доступу: xilinx.com.

7. Електронний ресурс. – Режим доступу: altera.com.

8. Baranov S. Logic and System Design of Digital Systems / S.Baranov. - Tallinn: TTU, 2008. – 266 p.

9. Оптимизация схемы КМУУ с преобразователем адреса микрокоманд / А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко, Я.М. Липински // Наукові праці ДонНТУ. Серія: «Проблеми моделювання та автоматизації проектування» (МАП-2011). – 2011. – Випуск 9 (179). – С. 26-35.

10. Баркалов А.А. Оптимизация схемы КМУУ с разделением кодов / А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко // Наукові праці ДонНТУ. Серія: «Інформатика, кібернетика та обчислювальна техніка» (ІКОТ-2011). – 2011. – Випуск 14 (188). – С. 68-73.

Надійшла до редакції 28.06.2013.

О.О. БАРКАЛОВ[1,2], Л.О. ТИТАРЕНКО[2], К.М. ЄФІМЕНКО[1], І.Я. ЗЕЛЕНЬОВА[1]

[1]ДВНЗ «Донецький національний технічний університет» (Україна)

[2]Університет Зеленогурьський (Польща)

ОПТИМІЗАЦІЯ СХЕМИ АДРЕСАЦІЇ КОМПОЗИЦІЙНОГО МІКРОПРОГРАМНОГО ПРИСТРОЮ КЕРУВАННЯ ІЗ РОЗПОДІЛОМ КОДІВ

В роботі запропоновано метод зменшення апаратурних витрат у схемі КМПК із розподілом кодів, який орієнтовано на технологію FPGA. Метод засновано на використанні трьох джерел кодів класів псевдоеквівалентних ОЛЛ та мультіплексору, який дозволяє вибрати одне з цих джерел. Такій підхід дозволить зменшити число LUT елементів у схемі адресації КМПК. Наведено приклад використання запропонованого методу.

Ключові слова: композиційний мікропрограмний пристрій керування, ГСА, операторний лінійний ланцюг, FPGA, LUT, логічна схема.

А.А. БАРКАЛОВ[1,2], Л.А. ТИТАРЕНКО[2], К.Н. ЕФИМЕНКО [1], И.Я. ЗЕЛЕНЕВА [1]

[1] ГВУЗ «Донецкий национальный Технический Университет» (Украина)

[2]Университет Зеленогурский (Польша)

ОПТИМИЗАЦИЯ СХЕМЫ АДРЕСАЦИИ КОМПОЗИЦИОННОГО МИКРОПРОГРАММНОГО УСТРОЙСТВА УПРАВЛЕНИЯ С РАЗДЕЛЕНИЕМ КОДОВ

В работе предложен метод уменьшения аппаратурных затрат в схеме КМУУ с разделением кодов, ориентированный на технологию FPGA. Метод основан на использовании трех источников кодов классов псевдоэквивалентных ОЛЦ и мультиплексора, который позволяет выбирать один из этих источников. Такой подход позволяет уменьшить число LUT элементов в схеме адресации КМУУ. Приведен пример использования предложенного метода.

Ключевые слова: композиционное микропрограммное устройство управления, ГСА, операторная линейная цепь, FPGA, LUT, логическая схема.