Addressing scheme CMCU (composition microprogram control unit) optimization with division of codes

The method of reduction of hardware expenses in the scheme CMCU with division of the codes, oriented on the FPGA technology is offered. The method is based on use of three sources of codes of classes of the pseudo-equivalent OLC (operator-linear circuit) and the multiplexer, allowing to select one of these sources. Such approach will allow to reduce number of LUT elements in the addressing scheme CMCU. The example of application of the offered method is given.

Keywords: CMCU, GSA, OLC, FPGA, logic scheme(circuit)

Introduction

The composition microprogram control units (CMCU) are an effective remedy of implementation of the linear control algorithms [1,2]. One of the CMCU models is the model with division of the codes [3], allowing under certain conditions to reduce hardware (instrumental) expenses in the addressing scheme of microcommands. Now chips (microcircuits) like FPGA (field–programmable gate arrays) are widely used in case of implementation of schemes of digital devices [4,5]. Basis of these VLSI's (Very Large Scale Integration circuits) represent 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 reduction of number of LUT in the scheme CMCU it is necessary to reduce number of arguments and terms in system of functions of addressing of microcommands [1,8]. In the real operation one of 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 development of the results received in operations [9,10].

The aim of this research is reduction of 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 the codes, allowing to optimize 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 broad use of the device GSA in practice of engineering design.

Composition MCU (microprogram control unit) with division of codes

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

Definition 1. As an operational linear chain of GSA G is called the final sequence of operational nodes $\alpha_g = \langle b_{g1},\ldots,b_{gF_g} \rangle$ such that for any pair next component b_{gi}, b_{gi+1}, where i - of 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 - set of the nodes, entering in OLC α_g, is called as 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 as exit 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 as pseudo-equivalent OLC, if their exits are connected with an input of the same node $b_q \in B$

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

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

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

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

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

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

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

Let for GSA G the following condition be satisfied:

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

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

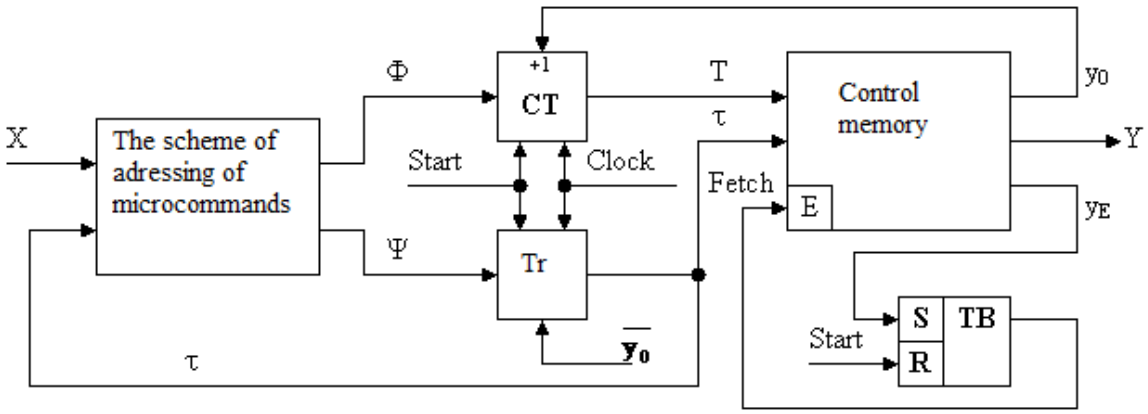

Figure 1 – The block scheme CMCU with division of codes

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

Let's agree further to designate CMCU (fig. 1) with U_1 character.

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

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

Thus the address of the microcommand is represented in a look

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

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 case of this Fetch = 1 that allows selection of commands of the control memory (CM). If the read microcommand doesn't correspond to OLC output, at the same time with microoperations $Y(b_q)$ the signal y_0 is created. If y_0 = 1, to contents of C is added unit and the following component of the current OLC is addressed. If an output of OLC reach, y_0 = 0. Thus the address of an input of the following OLC is created by the scheme ASM. In case of achievement of the termination of the microprogram y_E signal is created, the TB trigger is nullified, and selection of microcommands stops.

The number of LUT elements in the scheme ASM depends from number of arguments and terms in system (5). In the present work the method, allowing to reduce complexity of functions in system (5) is offered and, hence, to reduce hardware expenses in the scheme ASM.

The main idea of an offered method

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

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

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's encode classes $B_i \in \Pi_B$ with binary codes with digit capacity $C(B_i)$

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

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

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

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 change the number of inputs and exits at fixed capacity V_0.

$$V_0 = 2^S \cdot t_F \qquad (10)$$

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

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:

$$V = \lceil V_0 / t_F \rceil \qquad (11)$$

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

$$t_M = \lceil V_0 / M \rceil \qquad (12)$$

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

$$N + 3 + R_3 > t_M > N + 3 \qquad (13)$$

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

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

$$Z = Z^1 \cup Z^2 \qquad (14)$$

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

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

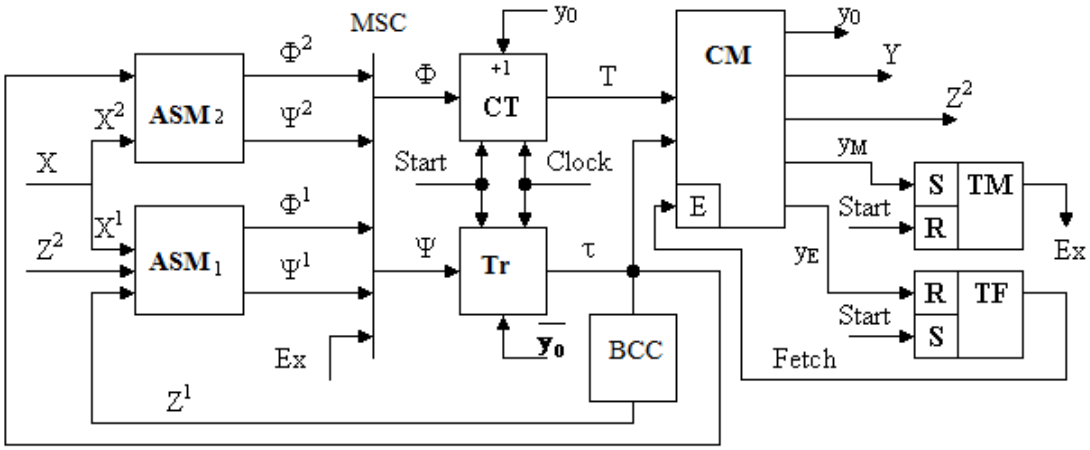

Figure 2 – The block scheme CMCU U_2

This model has a row of differences from model U_1. First, the unit ASM is partitioned 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 a 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 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. Offered CMCU functions as follows.

On Start signal in Tr and ST zero codes (the address of the first microcommand) are skidded, and the TF and TM trigger are set respectively in "1" (Fetch=1) and "0" (Ex = 0). While the address of an input won't reach, U_2 functions as U_1. In case of achievement of the address of an output of OLC $\alpha_g \in B_i$, the variable $y_M = 1$ can be created. In this case Ex = 1 and a jump address is created by the circuit ASM_1. The system of functions is for this purpose created:

$$\Phi^1 = \Phi^1(Z, X^1); \quad \Psi^1 = \Psi^1(Z, X^1). \qquad (15)$$

If the $B_i \in \Pi_A$, that the variable y_M isn't created. Thus Ex = 0 and a jump address is defined by the circuit ASM_2. The system of functions is for this purpose created:

$$\Phi^2 = \Phi^2(\tau, X^2); \quad \Psi^2 = \Psi^2(\tau, X^2). \tag{16}$$

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

Such approach allows to reduce number of terms in system (5) to an absolute minimum. In case of condition execution:

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

the number of arguments in system (15) in comparison with appropriate functions from system (5) decreases. Shortcomings of this approach is existence of the units MSC and BCC, consuming some resources of a crystal, and increase in digit capacity of words of CM. However, the scheme MSC is implemented at the expense of use of three-stable outputs of macrocells therefore additional LUT elements it isn't required.

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

Features of implementation of the circuit CMCU U_2

In the 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 Π_C.
2. Optimum coding of OLC $\alpha_g \in C_1$ and coding OLC component.
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 a 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 to development of the CMCU VHDL models and use of standard industrial packets [6,7]. These stages don't represent a particular interest for an illustration of synthesis of the diagram CMCU U_2. In this regard we don't 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 received. 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 $B_i \in \Pi_C$ was represented by one interval of R_1-dimensional boolean space [1]. One of options of coding is provided in fig. 3.

$\tau_1\tau_2$ \ $\tau_3\tau_4$	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 $\alpha_g \in C$

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 an interval 0000; class B_2 – by an intervals 00*1 and 001*; class B_3 – by an interval 010*; class B_4 – by an intervals 110* and 11*1; class B_5 – by an interval 011*; class B_6 – by an interval 1010; class B_7 – by an intervals 100* and 10*1. 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]. Let, for example, from GSA G1 it is possible to receive the following formulas:

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

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

transition address ($A(b_q)$); logical conditions defining transition (X_h column); functions of excitation of triggers of the register of Tr (column Ψ_h^1 for Π_B and Ψ_h^2 for Π_A); functions of excitation of triggers of the CT counter (a 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 fragments of transition tables for formulas (13) are given in tab. 1 and tab. 2.

Table 1. The transition table for a 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_2D_4	D_6	3

Table 2. The transition table for a 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 received 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 received from the transition table for classes $B_i \in \Pi_A$. So, from tab. 1 we have, for example,

$$D_2 = \overline{\tau_1}\tau_2\overline{\tau_3}\tau_4 x_1 x_2;\ D_6 = \overline{\tau_1}\tau_2\tau_3\tau_4 x_1 x_2.$$

Let from a reviewed example the following sets follow: $Z^1 = \{Z_1\}$ and $Z^2 = \{Z_2\}$. In case of this BCC implements only part of a 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 $K(B_i)_j$ – the code of the class Bi corresponding to one of the generalized intervals; Z_h – the discharges of a code $C(B_i)$ accepting single value in 'h'-th row of the table. For a reviewed example, number of lines $H_{ПК} = 6$. This parameter is defined by summary number of the intervals representing codes of classes $B_i \in \Pi_B$. For a reviewed example the unit BCC is provided to tab. 3.

Table 3. Table of the unit of the transformer of the address

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): $z_1 = \tau_1\overline{\tau_2}\,\overline{\tau_3} \vee \tau_1\overline{\tau_2}\tau_4$.

Let's mark that for the approach offered in [10], BCC shall implement and the equation $z_2 = \tau_1\tau_2\overline{\tau_3} \vee \tau_1\tau_2\tau_4$.

In case of implementation of the unit CM there are following features. The part of a code $K(B_i)$, determined by a set of Z_2, is located in cells of CM which addresses corresponds to OLC outputs. This stage is executed in a trivial way and in this article isn't considered.

Conclusion

The method of optimization of CMCU offered in article is based on multiplexing of three sources of codes of classes of the pseudo-equivalent OLC. Such approach allows to reduce with guarantee number of terms in 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 considered as a Moore machine, offered approach allows to reduce number of terms to 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 addresses of outputs of OLC are subject to conversion and not all discharges of codes of classes are created.

Lack 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 carries to reduction of number of levels in the circuit and the time delay from introduction of MSC is compensated. The researches conducted by authors showed that the offered method allows to reduce to 40% number of LUT elements in relation to the initial CMCU. Thus cycle of the CMCU U_2 time always was less, than at the CMCU U_1.

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

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

List of references

1. Barkalov A. Logic synthesis for compositional microprogram control units./ A.Barkalov., L.Titarenko. – Berlin: Springer, 2008. – 272 pp.
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 pp.
4. Maxfield S. The Design Warrior's Guide to FPGAs./ S.Maxfield. – Amsterdam: Elsevier, 2004. – 541 pp.
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. xilinx.com.
7. altera.com.
8. Baranov S. Logic and System Design of Digital Systems./ S.Baranov. - Tallinn: TTU, 2008. – 266 pp.
9. Баркалов А.А. 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). Випуск 9 (179). – Донецьк: ДонНТУ, 2011. – 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). Випуск 14 (188) / – Донецьк: ДВНЗ «ДонНТУ», 2011. – С.68-73.