

Оптимизация схемы композиционного микропрограммного устройства управления с общей памятью

А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко, Я.М. Липински

Предложен метод уменьшения аппаратных затрат в схеме композиционного микропрограммного устройства управления с общей памятью, ориентированный на технологию *Field-programmable gate arrays*. Метод основан на использовании двух источников кодов классов псевдоэквивалентных операторных линейных цепей и мультиплексора, позволяющего выбрать один из этих источников. Такой подход позволит уменьшить число *look-up table* элементов в схеме адресации композиционного микропрограммного устройства управления. Приведен пример применения предложенного метода.

A method for reducing the hardware amount in the circuit of composition microprogrammed control units with common memory is suggested oriented to the field-programmable gate arrays technology. The method is based on the use of two sources of codes classes of pseudo-equivalent OLC and a multiplexer to choose one of these sources. Such an approach would reduce the number of look-up table elements in the addressing circuit of composition microprogrammed control units. An example of the proposed method application is given.

Запропоновано метод зменшення апаратних витрат у схемі композиційного мікропрограмного пристрою керування із загальною пам'яттю, який орієнтовано на технологію *Field-programmable gate arrays*. Метод засновано на використанні двох джерел кодів класів псевдоеквівалентних операторних лінійних ланцюгів та мультиплексора, який дозволяє вибрати одне з цих джерел. Такий підхід дозволить зменшити число *Look-up table* елементів у схемі адресації композиційного мікропрограмного пристрою керування. Наведено приклад використання запропонованого методу.

Введение. Если алгоритм управления некоторой системы представлен линейной граф-схемой алгоритма (ГСА) [1], то для реализации схемы устройства управления (УУ) может быть использована модель композиционного микропрограммного устройства управления (КМУУ) с общей памятью [2, 3]. В настоящее время для реализации схем УУ широко используются программируемые логические интегральные схемы (ПЛИС) вида *FPGA (field-programmable gate arrays)* [4, 5]. Основу *FPGA* представляют макроэлементами *LUT (look-up table)*, имеющие ограниченное число входов (4–6) [6, 7]. Для оптимизации схемы УУ на *FPGA* необходимо уменьшить количество аргументов и термов в реализуемых системах булевых функций [8]. В статье предлагается метод решения этой задачи для КМУУ с общей памятью. Метод основан на использовании двух источников кодов классов псевдоэквивалентных операторных линейных цепей (ОЛЦ) и мультиплексора источников кодов (МИК).

Цель исследования – оптимизация схемы КМУУ с общей памятью путем мультиплекси-

рования источников кодов псевдоэквивалентных ОЛЦ.

Задача исследования – разработка метода синтеза КМУУ с общей памятью, позволяющего уменьшить число *LUT*-элементов в его комбинационной части.

Композиционное МУУ с общей памятью

Пусть ГСА $\Gamma = \Gamma(B, E)$ представлена множествами вершин B и соединяющих их дуг E . Пусть $B = b_0 \cup b_E \cup E_1 \cup E_2$, где b_0 – начальная вершина, b_E – конечная вершина, E_1 – множество операторных вершин и E_2 – множество условных вершин ГСА Γ . В операторных вершинах $b_q \in E_1$ записываются наборы микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ – множество микроопераций. В условных вершинах $b_q \in E_2$ записываются элементы множества логических условий $X = \{x_1, \dots, x_L\}$. Введем ряд определений, взятых из [2].

Определение 1. Операторной линейной цепью ГСА Γ называется конечная последовательность операторных вершин $\alpha_g = \langle b_{g_1}, \dots, b_{g_{F_g}} \rangle$ такая, что для любой пары соседних компонент $b_{g_i}, b_{g_{i+1}}$, где i – номер компоненты кортежа α_g , существует дуга $\langle b_{g_i}, b_{g_{i+1}} \rangle \in E$.

Ключевые слова: композиционное микропрограммное устройство управления (КМУУ), граф-схема алгоритма (ГСА), операторные линейные цепи (ОЛЦ), *field-programmable gate arrays (FPGA)*, логическая схема.

Определение 2. Вершина $b_q \in D^g$, где D^g – множество вершин, входящих в ОЛЦ α_g , называется входом ОЛЦ α_g , если существует дуга $\langle b_i, b_q \rangle \in E$, где $b_i \in D^g$.

Определение 3. Вершина $b_q \in D^g$ называется выходом ОЛЦ α_g , если существует дуга $\langle b_q, b_i \rangle \in E$, где $b_i \notin D^g$.

Определение 4. ОЛЦ α_i, α_j называются псевдоэквивалентными ОЛЦ, если их выходы связаны со входом одной и той же вершины $b_q \in B$.

Пусть для некоторой ГСА Γ сформировано множество ОЛЦ $C = \{\alpha_1, \dots, \alpha_G\}$, определяющее разбиение на множестве E_1 [3], и пусть $|E_1| = M$. Поставим в соответствие каждой вершине $b_q \in E_1$ микрокоманду MI_q с адресом $A(b_q)$, имеющим разрядность

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

Используем для адресации микрокоманд переменные $T_r \in T$, где $|T| = R$. Адресация выполняется таким образом, чтобы выполнялось условие

$$A(b_{g_{i+1}}) = A(b_{g_i}) + 1, \quad (2)$$

где $b_{g_i}, b_{g_{i+1}} \in D^g$ и $\langle b_{g_i}, b_{g_{i+1}} \rangle \in E$.

В этом случае УУ может быть реализовано в виде КМУУ U_1 (рис. 1), называемом КМУУ с общей памятью [2, 3]. Принципы функционирования КМУУ U_1 достаточно полно представлены в литературе [1–3].

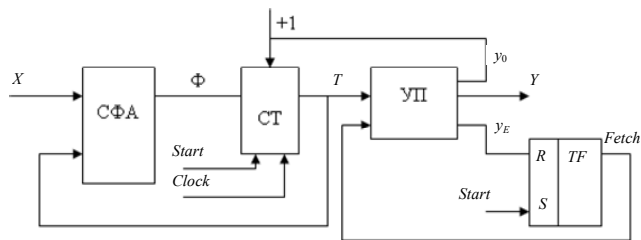


Рис. 1. Структурная схема КМУУ U_1

В КМУУ U_1 схема формирования адреса (СФА) реализует систему функций возбуждения триггеров счетчика СТ, определяемую как

$$\Phi = \Phi(T, X). \quad (3)$$

При этом, как правило, счетчик имеет информационные входы типа D [6, 7]. По сигналу *Start* счетчик СТ устанавливается в ноль, что соответствует адресу первой микрокоманды реализуемого алгоритма. По сигналу $y_0 = 1$ содержимое СТ увеличивается на единицу, что соответствует режиму (2). Управляющая память (УП) хранит наборы микроопераций $Y(b_q) \subseteq Y$ и переменные y_0 (управление СТ) и y_E (признак окончания алгоритма). Триггер считывания *TF* формирует сигнал *Fetch*, разрешающий выборку микрокоманд из УП. При достижении окончания алгоритма формируется переменная $y_E = 1$, что приводит к $Fetch = 0$ и прекращению выборки из УП.

При использовании *FPGA* схемы СФА, СТ и *TF* реализуются на *LUT*, а схема УП – на встроенных блоках памяти *EMB* (*embedded memory block*). Основным недостатком КМУУ U_1 – значительное число термов в системе функций (3). Это приводит к увеличению числа *LUT* элементов и их уровней в схеме СФА. Для устранения этого недостатка используют оптимальную адресацию микрокоманд, что приводит к КМУУ U_2 [9]. Однако такая адресация не всегда возможна. Уменьшение числа термов в системе (3) гарантируется в КМУУ U_3 , где осуществляется преобразование адресов микрокоманд в коды классов псевдоэквивалентных ОЛЦ [2, 3]. Однако это связано с введением в схему КМУУ дополнительного блока преобразователя адресов (БПА), потребляющего некоторые ресурсы кристалла. В статье предлагается метод синтеза КМУУ, позволяющий уменьшить как число термов в системе (3), так и число *LUT* элементов в схеме БПА.

Основная идея предлагаемого метода

Найдем разбиение $\Pi_C = \{B_1, \dots, B_I\}$ множества ОЛЦ C на классы псевдоэквивалентных ОЛЦ. Выполним адресацию микрокоманд так, чтобы при выполнении условия (2) максимально возможное число классов $B_i \in \Pi_C$ выражалось одним интервалом R -разрядного булева пространства. Представим разбиение Π_C в виде $\Pi_C = \Pi_1 \cup \Pi_2$, где $\Pi_1 \cap \Pi_2 = \emptyset$. Пусть $B_i \in \Pi_1$, если этот класс представляется более

чем одним интервалом пространства кодирования. Остальные классы принадлежат множеству Π_2 .

Закодируем классы $B_i \in \Pi_1$ двоичными кодами $K(B_i)$ разрядности

$$R_1 = \lceil \log_2 I_1 \rceil, \quad (4)$$

где $I_1 = |\Pi_1|$. Используем для кодирования переменные $\tau_r \in \tau$, где $|\tau| = R_1$. Если $\Pi_1 \neq \emptyset$ и $\Pi_2 \neq \emptyset$, то для реализации схемы УУ предлагается модель U_4 (рис. 2).

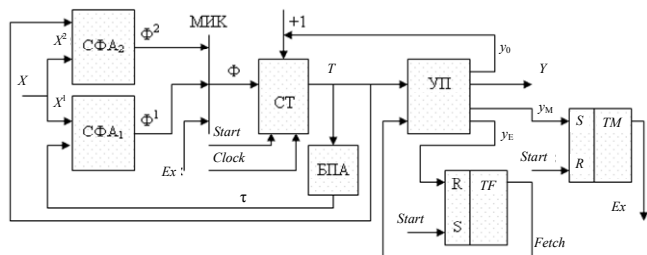


Рис. 2. Структурная схема КМУУ U_4

КМУУ U_4 функционирует следующим образом. По сигналу *Start* в *CT* заносится нулевой адрес, триггеры *TF* и *TM* устанавливаются соответственно в 1 (*Fetch* = 1) и 0 (*Ex* = 0). Схема *СФА₁* служит для задания переходов из классов $B_i \in \Pi_1$. При этом $y_M = 1$, $Ex = 1$ и

$$\Phi^1 = \Phi^1(\tau, X^1). \quad (5)$$

Схема *СФА₂* служит для задания переходов из классов $B_i \in \Pi_2$. При этом $y_M = Ex = 0$ и

$$\Phi^2 = \Phi^2(T, X^2). \quad (6)$$

Мультиплексор *МИК* служит для выбора источника функций возбуждения *CT*:

$$\Phi = Ex \Phi^1 \vee \overline{Ex} \Phi^2. \quad (7)$$

При этом $\Phi^1 \cup \Phi^2 = \Phi$. Очевидно, через *МИК* передаются только функции $\varphi_r \in \Phi^1 \cap \Phi^2$.

Блок *БПА* служит для преобразования адресов выходов ОЛЦ в коды классов $B_i \in \Pi_1$. Для такого преобразования служат функции

$$\tau = \tau(T). \quad (8)$$

Сравнение блоков *УП* КМУУ U_1 и U_4 показывает, они отличаются наличием выхода y_M . Этот выход используется для управления триггером *TM* и, следовательно, мультиплексором *МИК*.

Такой подход позволяет уменьшить число термов в системе (3) до абсолютно возможного минимума. Кроме того, уменьшается сложность блока *БПА* в сравнении с КМУУ U_3 . Отметим, что при $\Pi_1 = \emptyset$ КМУУ U_4 вырождается в U_2 . При $\Pi_2 = \emptyset$ КМУУ U_4 превращается в КМУУ U_3 . Недостаток – увеличение числа выходов блока *УП*. Однако этот блок строится из реконфигурируемых блоков *ЕМВ*, имеющих строго определенное число выходов [6, 7]. При этом очевидна высокая вероятность наличия неиспользованных выходов.

В статье предлагается метод синтеза КМУУ U_4 , включающий следующие этапы:

- формирование множества ОЛЦ C для ГСА Γ ;
- формирование разбиения Π_C множества C ;
- оптимальная адресация микрокоманд;
- кодирование классов $B_i \in \Pi_1$;
- формирование таблицы переходов для классов $B_i \in \Pi_1$;
- формирование таблицы переходов для классов $B_i \in \Pi_2$;
- формирование таблицы блока преобразователя адреса;
- формирование таблицы содержимого управляющей памяти;
- реализация схемы КМУУ в заданном элементном базисе.

Пример применения предложенного метода

Пусть для некоторой ГСА Γ получено множество ОЛЦ $C = \{\alpha_1, \dots, \alpha_{12}\}$, где $\alpha_1 = \langle b_1, \dots, b_4 \rangle$, $\alpha_2 = \langle b_5 \rangle$, $\alpha_3 = \langle b_6, b_7, b_8 \rangle$, $\alpha_4 = \langle b_9, b_{10} \rangle$, $\alpha_5 = \langle b_{11}, b_{12}, b_{13}, b_{14} \rangle$, $\alpha_6 = \langle b_{15}, b_{16} \rangle$, $\alpha_7 = \langle b_{17}, \dots, b_{20} \rangle$, $\alpha_8 = \langle b_{21}, \dots, b_{24} \rangle$, $\alpha_9 = \langle b_{25}, b_{26} \rangle$, $\alpha_{10} = \langle b_{27}, b_{28} \rangle$, $\alpha_{11} = \langle b_{29}, b_{30}, b_{31} \rangle$, $\alpha_{12} = \langle b_{32} \rangle$. Пусть эти ОЛЦ могут быть разбиты на пять классов, где $B_1 = \{\alpha_1, \alpha_2, \alpha_3\}$, $B_2 = \{\alpha_4, \alpha_5, \alpha_6\}$, $B_3 = \{\alpha_7, \alpha_8, \alpha_9\}$, $B_4 = \{\alpha_{10}, \alpha_{11}\}$, $B_5 = \{\alpha_{12}\}$. Пусть выход ОЛЦ $\alpha_{12} \in C$ связан с входом вершины b_E . Как известно, переходы из таких ОЛЦ не рассматриваются, так как их последняя вершина должна включать переменную y_E [2].

Сделаем адресацию микрокоманд так, чтобы выполнялось условие (2) и максимально возможное число классов представлялось одним обобщенным интервалом R -мерного булева пространства. В рассматриваемом примере $M = 32$, то есть $R = 5$ и $T = \{T_1, \dots, T_5\}$. Один из возможных вариантов оптимальной адресации микрокоманд показан на рис. 3. Этот рисунок содержит видоизмененную карту Карно, достаточную для получения обобщенных интервалов. Судя по рис. 3, класс B_1 соответствует интервалу 00^{***} , класс $B_2 - 01^{***}$, класс B_3 соответствует двум интервалам: 10^{***} и 1100^* , класс B_4 соответствует двум интервалам: 11011 и 11110 . Интервал для класса B_5 не имеет значения. Символ $U_4(\Gamma)$ означает, что КМУУ U_4 реализуется по ГСА Γ .

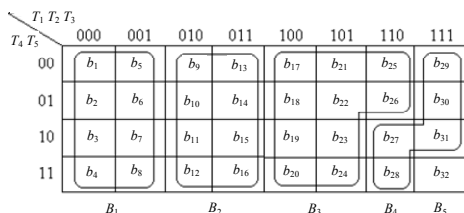


Рис. 3. Адреса микрокоманд КМУУ $U_4(\Gamma)$

Итак, $\Pi_1 = \{B_3, B_4\}$, $I_1 = 2$, $\Pi_2 = \{B_1, B_2\}$. Для кодирования классов $B_i \in \Pi_1$ достаточно $R_1 = 1$ переменной, т.е. $\tau_r \in \{\tau_1\}$. Коды классов $B_i \in \Pi_2$ уже были найдены, так как они совпадают с соответствующими интервалами. Таким образом, $K(B_1) = 00^{***}$, $K(B_2) = 01^{***}$. Так как для обоих кодов $T_1 = 0$, то классы B_1 и B_2 можно идентифицировать переменной T_2 . Пусть $K(B_3) = 0$, $K(B_4) = 1$.

Таблицы переходов для классов $B_i \in \Pi_C$ строятся по системе обобщенных формул перехода [2, 3]. Пусть в рассматриваемом примере эта система имеет вид:

$$\begin{aligned}
 B_1 &\rightarrow x_1 b_9 \vee \overline{x_1} x_2 b_{11} \vee \overline{x_1} x_2 x_3 b_{13} \vee \overline{x_1} x_2 x_3 b_{15}; \\
 B_2 &\rightarrow x_2 b_5 \vee x_2 x_5 b_6 \vee x_2 x_5 x_6 b_{17} \vee x_2 x_5 x_6 b_{22}; \\
 B_3 &\rightarrow x_4 x_6 b_{21} \vee x_4 x_6 b_{25} \vee x_4 x_3 b_{27} \vee x_4 x_3 b_{29}; \\
 B_4 &\rightarrow x_1 b_{32} \vee \overline{x_1} b_{23}.
 \end{aligned} \tag{9}$$

Таблицы переходов имеют следующие столбцы B_i , $K(B_i)$, b_q , $A(b_q)$, X_h , Φ_h , h . Здесь X_h – набор ЛУ, определяющий связь между классом

B_i и вершиной b_q в формулах перехода вида (9); Φ_h – набор функций возбуждения СТ, равных единице для формирования адреса $A(b_q)$. Разница между таблицами заключается в том, что для классов $B_i \in \Pi_1$ код представляется переменными $\tau_r \in \tau$, а для $B_i \in \Pi_2$ – переменными $T_r \in T$. Для КМУУ $U_4(\Gamma)$ переходы для классов $B_i \in \Pi_1$ представлены в табл. 1, а для классов $B_i \in \Pi_2$ – в табл. 2. Первая таблица имеет $H_1 =$ шесть строк, а вторая – $H_2 =$ восемь строк.

Таблица 1. Переходы для классов $B_i \in \Pi_1$

B_i	$K(B_i)$	b_q	$A(b_q)$	X_h	Φ_h	h
B_3	0	b_{21}	10100	$x_4 x_6$	$D_1 D_3$	1
		b_{25}	11000	$x_4 \overline{x_6}$	$D_1 D_2$	2
		b_{27}	11010	$\overline{x_4} x_3$	$D_1 D_2 D_4$	3
B_4	1	b_{32}	11111	x_1	$D_1 D_2 D_3 D_4 D_5$	4
		b_{23}	10110	$\overline{x_1}$	$D_1 D_3 D_4$	5

Таблица 2. Переходы для классов $B_i \in \Pi_2$

B_i	$K(B_i)$	b_q	$A(b_q)$	X_h	Φ_h	h
B_1	0	b_9	01000	x_1	D_2	1
		b_{11}	01010	$\overline{x_1} x_2$	$D_2 D_4$	2
		b_{13}	01100	$\overline{x_1} \overline{x_2} x_3$	$D_2 D_3$	3
		b_{15}	01110	$\overline{x_1} \overline{x_2} \overline{x_3}$	$D_2 D_3 D_4$	4
B_2	1	b_5	00100	x_2	D_3	5
		b_6	00101	$\overline{x_2} x_5$	$D_3 D_5$	6
		b_{17}	10000	$\overline{x_2} \overline{x_5} x_6$	D_1	7
		b_{22}	10101	$\overline{x_2} \overline{x_5} \overline{x_6}$	$D_1 D_3 D_5$	8

Для табл. 1 и 2 адреса микрокоманд используются, указанные на рис. 3. В табл. 1 показана система функций (5). Например, $D_1 = 1$, $D_2 = \overline{\tau_1} x_4 \overline{x_6} \vee \overline{\tau_1} \overline{x_4} \vee \overline{\tau_1} x_1$ (после минимизации). В табл. 2 – система (7). Например, $D_1 = T_1 \overline{x_2} \overline{x_5}$, $D_2 = \overline{T_1}$ (после минимизации). Из этих таблиц имеем также множества $X^1 = \{x_1, x_3, x_4, x_6\}$, $X^2 = \{x_1, x_2, x_3, x_5, x_6\}$, $\Phi^1 = \Phi^2 = \Phi$.

Для формирования таблицы БПА необходимо найти соответствие между кодами классов $B_i \in \Pi_2$ и адресами выходов ОЛЦ $\alpha_g \in B_i$. Эти адреса определяются обобщенными интервалами. Таблица включает столбцы α_g , $A(\alpha_g)$, B_i , $K(B_i)$, τ_g , g , где $A(\alpha_g)$ – адрес выхода ОЛЦ $\alpha_g \in C$. Для КМУУ $U_4(\Gamma)$ эта таблица имеет $G_4 = 5$ строк (табл. 3).

Таблица 3. Блок преобразователя адреса КМУУ $U_4(\Gamma)$

α_g	$A(\alpha_g)$	B_i	$K(B_i)$	τ_g	g
α_7	10***	B_3	0	–	1
α_8	10***	B_3	0	–	2
α_9	11000	B_3	0	–	3
α_{10}	11011	B_4	1	τ_1	4
α_{11}	111**	B_4	1	τ_1	5

Из табл. 3 имеем систему (8), представляющую $\tau_1 = T_1 T_2 T_3 T_4 T_5 \vee T_1 T_2 T_3$.

Синтез схемы МУК сводится к реализации системы (7). Эта задача – тривиальна, а схема включает R LUT элементов, каждый из которых имеет по три входа.

Для формирования содержимого УП необходимо выполнить преобразование исходной ГСА Γ , которое заключается в следующем:

- Если вершина $b_q \in E_1$ не является выходом ОЛЦ $\alpha_g \in C$, то в нее включается переменная y_0 .
- Если вершина $b_q \in E_1$ связана с вершиной b_E , то в нее вводится переменная y_E .
- Если вершина $b_q \in E_1$ является выходом ОЛЦ $\alpha_g \in B_i$, где $B_i \in \Pi_1$, то в нее вводится переменная y_M .

Например, в вершину $b_2 \in D^1$ необходимо ввести y_0 , в вершину $b_{20} - y_M$, а в вершину $b_{32} - y_E$. Далее в ячейки УП с адресом $A(b_q)$ записывается набор $Y(b_q)$ и необходимые элементы множества $\{y_0, y_M, y_E\}$. Этот этап выполняется обычным способом и в статье не рассматривается.

Реализация схемы КМУУ U_4 сводится к реализации систем функций (5)–(8) на LUT элементах и УП на EMB. Для решения этой задачи используются стандартные промышленные пакеты [6, 7]. Этот этап выходит за пределы данной статьи.

Заключение. Предлагаемый в статье метод оптимизации КМУУ основан на мультиплексировании двух источников кодов классов псевдоэквивалентных ОЛЦ. Такой подход позволяет гарантированно уменьшить число термов в системе функций возбуждения триггеров счетчика адресов микрокоманд до максимально возможной величины. Если КМУУ с общей памятью рассматривать как автомат Мура, то предлагаемый подход позволяет уменьшить число термов до величины этого параметра как у эквивалентного автомата Мили. Кроме того, уменьшается число LUT элементов в схеме преобразователя

адреса, так как не все адреса выходов ОЛЦ подлежат преобразованию.

Недостаток предложенного подхода – введение мультиплексора, который вносит дополнительную задержку в цикл работы КМУУ. Однако уменьшение числа термов ведет к уменьшению числа уровней в схеме и задержка от введения МИК компенсируется. Проведенные авторами исследования показали, что предложенный метод позволяет до 48% уменьшить число LUT элементов по отношению к исходному КМУУ с общей памятью. При этом время цикла КМУУ U_4 всегда было меньше, чем у КМУУ U_1 . Результаты сравнения КМУУ $U_2 - U_4$ показали, что каждое из них может иметь минимальные аппаратные затраты. Это определяется параметрами исходной ГСА. Таким образом, при синтезе КМУУ с общей памятью на FPGA выбор должен проводиться между КМУУ U_2, U_3 и U_4 .

Научная новизна предложенного метода заключается в использовании особенностей КМУУ (наличие классов псевдоэквивалентных ОЛЦ) для уменьшения числа LUT элементов в схеме КМУУ.

Практическая значимость метода заключается в уменьшении площади кристалла FPGA, занимаемой схемой КМУУ с общей памятью, что позволяет получить схемы, обладающие меньшей стоимостью, чем известные из литературы аналоги.

Дальнейшие направления работы связаны с разработкой метода выбора оптимальной структуры КМУУ с общей памятью при реализации схемы в базисе FPGA.

1. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units. – Berlin: Springer, 2008. – 272 p.
2. Баркалов А.А., Титаренко Л.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС. – Донецк: УНИТЕХ, 2009. – 336 с.
3. Barkalov A., Titarenko L. Logic synthesis for FSM-based control units. – Berlin: Springer, 2009. – 233 p.
4. Maxfield S. The Design Warrior's Guide to FPGAs. – Amsterdam: Elsevier, 2004. – 541 p.
5. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах с программируемой структурой. – СПб.: БХВ – Петербург, 2006. – 736 с.
6. xilinx.com
7. altera.com
8. Baranov S. Logic and System Design of Digital Systems. – Tallinn: TTU, 2008. – 266 p.
9. Баркалов А.А., Ковалев С.А., Ефименко К.Н. Оптимизация числа LUT-элементов в композиционном микропрограммном устройстве управления с общей памятью // УСиМ. – 2006. – № 4. – С. 50–55.

Поступила 20.11.2010

Тел. для справок: (0622) 301-0735 (Донецк)

E-mail: a.barkalov@jie.uz.zgora.pl

© А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко, Я.М. Липински, 2011