

ОПТИМИЗАЦИЯ СХЕМЫ КМУУ С ОБЩЕЙ ПАМЯТЬЮ

Баркалов А.А.¹, Титаренко Л.А.¹, Ефименко К.Н.², Липински Я.М.¹

Университет Зеленогурский (Польша)¹,

ГВУЗ «Донецкий национальный технический университет» (Украина)²

1. 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 макроячейки 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 настоящей работе 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).

Целью исследования является оптимизация схемы КМУУ с общей памятью за счет мультиплексирования источников кодов псевдоэквивалентных ОЛЦ.

Задачей исследования является разработка метода синтеза КМУУ с общей памятью, позволяющего уменьшить число LUT-элементов в комбинационной части КМУУ.

2. Композиционное МУУ с общей памятью

Пусть ГСА $\Gamma = \Gamma(V, E)$ представлена множествами вершин V и соединяющих их дуг E . Пусть $V = b_0 \cup b_E \cup E_1 \cup E_2$, 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 $\langle b_{g_i}, b_{g_{i+1}} \rangle \in E$.

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 $\langle b_t, b_q \rangle \in E$, где $b_t \notin D^g$.

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

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

Пусть для некоторой ГСА Γ сформировано множество ОЛЦ $S = \{\alpha_1, \dots, \alpha_G\}$, 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].

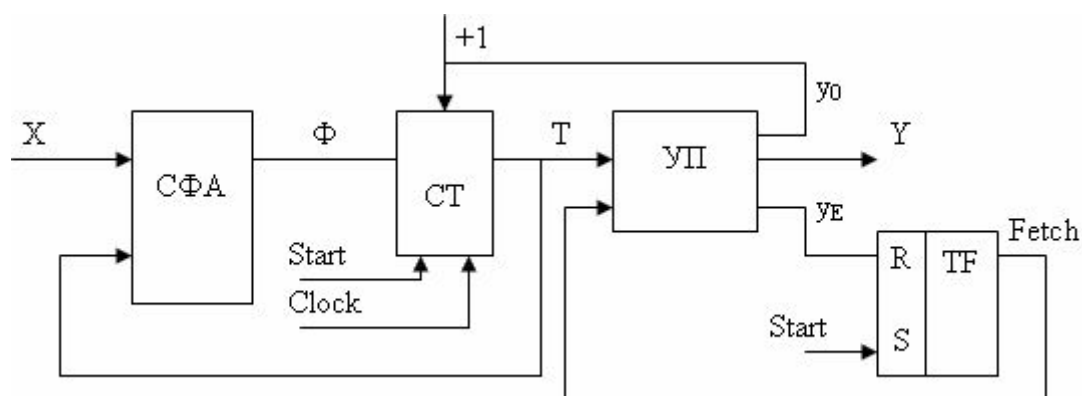


Рис. 1. Структурная схема КМУУ U_1

В КМУУ U_1 схема формирования адреса (СФА) реализует систему функций возбуждения триггера 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 является 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), который потребляет некоторые ресурсы кристалла. В настоящей работе предлагается метод синтеза КМУУ, позволяющий уменьшить как число термов в системе (3), так и число LUT элементов в схеме БПА.

3. Основная идея предлагаемого метода

Найдем разбиение $\Pi_C = \{B_1, \dots, B_1\}$ множества ОЛЦ C на классы псевдоэквивалентных ОЛЦ. Выполним адресацию микрокоманд так, чтобы при выполнении условия (2) макси-

мально возможное число классов $V_i \in \Pi_C$ 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 $V_i \in \Pi_1$, если этот класс представляется более, чем одним интервалом пространства кодирования. Остальные классы принадлежат множеству Π_2 .

Закодируем классы $V_i \in \Pi_1$ двоичными кодами $K(V_i)$ 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).

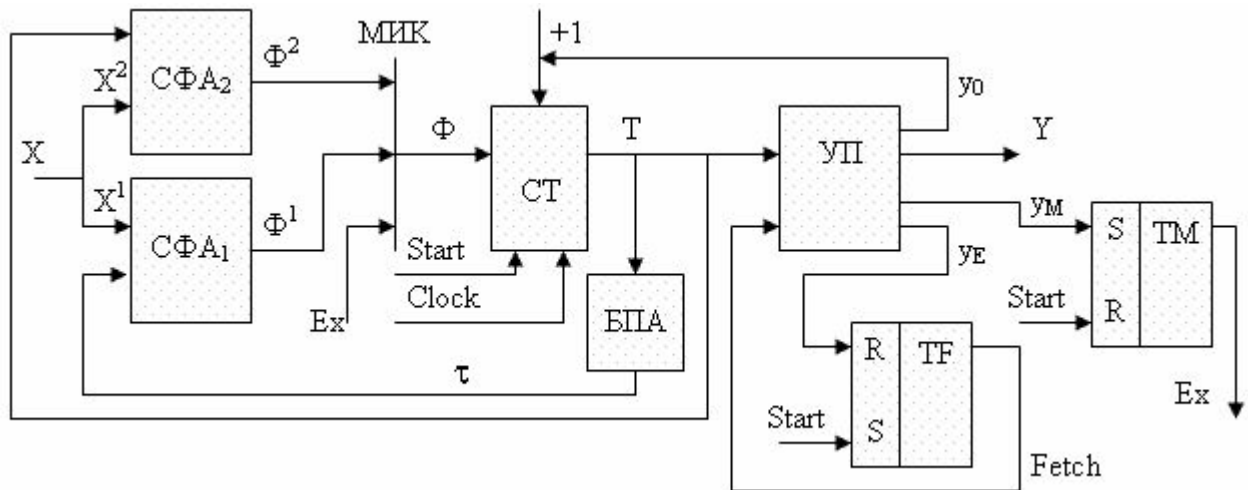


Рис. 2. Структурная схема КМУУ U_4

КМУУ U_4 функционирует следующим образом. По сигналу Start в СТ заносится нулевой адрес, триггера TF и TM устанавливаются соответственно в 1 (Fetch = 1) и 0 ($Ex = 0$). Схема СФА1 служит для задания переходов из классов $V_i \in \Pi_1$. 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 $V_i \in \Pi_2$. При этом $y_M = Ex = 0$ и

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

Мультиплексор МИК служит для выбора источника функций возбуждения СТ:

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

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 $V_i \in \Pi_1$. 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 . Как уже ясно, 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 по сравнению 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 . Недостатком является увеличение числа выходов блока УП. Однако этот блок строится из реконфигурируемых блоков ЕМВ, которые имеют строго определенное число выходов [6,7]. При этом имеется высокая вероятность наличия неиспользованных выходов.

В настоящей работе предлагается метод синтеза КМУУ U_4 , включающий следующие

этапы:

1. Формирование множества ОЛЦ S для ГСА Γ .
2. Формирование разбиения Π_C множества S .
3. Оптимальная адресация микрокоманд.
4. Кодирование классов $V_i \in \Pi_1$.
5. Формирование таблицы переходов для классов $V_i \in \Pi_1$.
6. Формирование таблицы переходов для классов $V_i \in \Pi_2$.
7. Формирование таблицы блока преобразователя адреса.
8. Формирование таблицы содержимого управляющей памяти.
9. Реализация схемы КМУУ в заданном элементном базисе.

4. Пример применения предложенного метода

Пусть для некоторой ГСА Γ получено множество ОЛЦ $S = \{\alpha_1, \dots, \alpha_{12}\}$, 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 $I=5$ классов, где $V_1 = \{\alpha_1, \alpha_2, \alpha_3\}$, $V_2 = \{\alpha_4, \alpha_5, \alpha_6\}$, $V_3 = \{\alpha_7, \alpha_8, \alpha_9\}$, $V_4 = \{\alpha_{10}, \alpha_{11}\}$, $V_5 = \{\alpha_{12}\}$. Пусть выход ОЛЦ $\alpha_{12} \in S$ связан с входом вершины b_E . Как известно, переходы из таких ОЛЦ не рассматриваются, так как их последняя вершина должна включать переменную u_E [2].

Выполним 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 приведен на рис. 3. Этот рисунок содержит видоизмененную карту Карно, которая достаточна для получения обобщенных интервалов. Из рис. 3 имеем: класс V_1 соответствует интервалу 00^{***} , класс $V_2 - 01^{***}$, класс V_3 соответствует двум интервалам: 10^{***} и 1100^* , класс V_4 соответствует двум интервалам: 11011 и 11110 . Интервал для класса V_5 не имеет значения. Символ $U_4(\Gamma)$ означает, что КМУУ U_4 реализуется по ГСА Γ .

		$T_1 T_2 T_3$							
		000	001	010	011	100	101	110	111
$T_4 T_5$	00	b_1	b_5	b_9	b_{13}	b_{17}	b_{21}	b_{25}	b_{29}
	01	b_2	b_6	b_{10}	b_{14}	b_{18}	b_{22}	b_{26}	b_{30}
	10	b_3	b_7	b_{11}	b_{15}	b_{19}	b_{23}	b_{27}	b_{31}
	11	b_4	b_8	b_{12}	b_{16}	b_{20}	b_{24}	b_{28}	b_{32}
		V_1		V_2		V_3		V_4	V_5

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

Итак, $\Pi_1 = \{V_3, V_4\}$, $I_1 = 2$, $\Pi_2 = \{V_1, V_2\}$. Для кодирования классов $V_i \in \Pi_1$ достаточно $R_1 = 1$ переменной, то есть $\tau_r \in \{\tau_1\}$. Коды классов $V_i \in \Pi_2$ уже были найдены, так как они совпадают с соответствующими интервалами. Таким образом, $K(V_1) = 00^{***}$, $K(V_2) = 01^{***}$. Так как для обоих кодов $T_1 = 0$, то классы V_1 и V_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:

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

Таблицы переходов имеют следующие столбцы $B_i, K(B_i), b_q, A(b_q), X_h, \Phi_h, h$. Здесь X_h – набор ЛУ, который определяет 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 $H_1 = 6$ строк, а вторая – $H_2 = 8$ строк.

Таблица 1

Таблица переходов 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	$\overline{x_4 x_6}$	$D_1 D_2$	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

Таблица переходов 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	$\overline{x_1 x_2 x_3}$	$D_2 D_3$	3
		b_{15}	01110	$\overline{x_1 x_2 x_3}$	$D_2 D_3 D_4$	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	$\overline{x_2 x_5 x_6}$	D_1	7
		b_{22}	10101	$\overline{x_2 x_5 x_6}$	$D_1 D_3 D_5$	8

Для табл.1 и табл.2 адреса микрокоманд берутся из рис. 3. Из табл.1 имеем систему функций (5). Например, $D_1 = 1, D_2 = \overline{\tau_1 x_4 x_6} \vee \tau_1 x_4 \vee \tau_1 x_1$ (после минимизации). Из табл. 2 имеем систему (7). Например, $D_1 = \overline{T_1 x_2 x_5}, D_2 = \overline{T_1}$ (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 клас-

сов $V_i \in \Pi_2$ и адресами выходов ОЛЦ $\alpha_g \in V_i$. Эти адреса определяются обобщенными интервалами. Таблица включает столбцы α_g , $A(\alpha_g)$, V_i , $K(V_i)$, τ_g , g . Здесь $A(\alpha_g)$ – адрес выхода ОЛЦ $\alpha_g \in C$. Для КМУУ $U_4(\Gamma)$ эта таблица имеет $G_4 = 5$ строк (табл. 3).

Таблица 3

Таблица блока преобразователя адреса КМУУ $U_4(\Gamma)$

α_g	$A(\alpha_g)$	V_i	$K(V_i)$	τ_g	g
α_7	10***	V_3	0	–	1
α_8	10***	V_3	0	–	2
α_9	11000	V_3	0	–	3
α_{10}	11011	V_4	1	τ_1	4
α_{11}	111**	V_4	1	τ_1	5

Из табл. 3 имеем систему (8), которая в нашем случае имеет вид $\tau_1 = T_1 T_2 \bar{T}_3 T_4 T_5 \vee T_1 T_2 T_3$.

Синтез схемы МИК сводится к реализации системы (7). Эта задача является тривиальной, а схема включает R LUT элементов, каждый из которых имеет по 3 входа.

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

1. Если вершина $b_q \in E_1$ не является выходом ОЛЦ $\alpha_g \in C$, то в неё включается переменная y_0 .
2. Если вершина $b_q \in E_1$ связана с вершиной b_E , то в неё вводится переменная y_E .
3. Если вершина $b_q \in E_1$ является выходом ОЛЦ $\alpha_g \in V_i$, где $V_i \in \Pi_1$, то в неё 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 тривиальным образом и в нашей статье не рассматривается.

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

5. Заключение

Предлагаемый в работе 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 у эквивалентного автомата Мили. Кроме того, уменьшается число LUT элементов в схеме преобразователя адреса, так как не все адреса выходов ОЛЦ подлежат преобразованию.

Недостатком предложенного подхода является введение мультиплексора, который вносит дополнительную задержку в цикл работы КМУУ. Однако уменьшение числа термов ведёт 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 производиться 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.

Литература

1. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units. – Berlin: Springer, 2008. 272 pp.
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 pp.
4. Maxfield S. The Design Warrior's Guide to FPGAs. – Amsterdam: Elsevier, 2004. – 541 pp.
5. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах с программируемой структурой – С-Пб: 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 pp.
9. Баркалов А.А., Ковалев С.А., Ефименко К.Н. Оптимизация числа LUT-элементов в композиционном микропрограммном устройстве управления с общей памятью// Управляющие системы и машины. –2006. – №4. – С. 50-55.