

УДК 004.274

А.А. Баркалов¹, д-р. техн. наук, проф.,
Л.А. Титаренко¹, д-р. техн. наук, проф.,
К.Н. Ефименко², канд. техн. наук, доц.,
И.Я. Зеленева², канд. техн. наук, доц.

¹Університет Зеленогурский, г. Зеленая Гура, Польша,

²ГВУЗ «Донецкий национальный технический университет», г. Донецк, Украина

E-mail: A.Barkalov@iie.uz.zgora.pl

Оптимизация схемы адресации КМУУ с разделением кодов

Предлагается метод уменьшения аппаратных затрат в схеме КМУУ с разделением кодов, ориентированный на технологию FPGA. Метод основан на использовании трех источников кодов классов псевдоэквивалентных ОЛЦ и мультиплексора, позволяющего выбрать один из этих источников. Такой подход позволит уменьшить число LUT элементов в схеме адресации КМУУ. Приведен пример применения предложенного метода.

Ключевые слова: КМУУ, ГСА, ОЛЦ, FPGA, логическая схема

Введение

Композиционные микропрограммные устройства управления (КМУУ) являются эффективным средством реализации линейных алгоритмов управления [1,2]. Одной из моделей КМУУ является модель с разделением кодов [3], позволяющая при определенных условиях уменьшить аппаратные затраты в схеме адресации микрокоманд. В настоящее время микросхемы типа FPGA (field-programmable gate arrays) широко используются при реализации схем цифровых устройств [4,5]. Основу этих СБИС представляют макроячейки табличного типа, называемые LUT (look-up table). Как правило, LUT-элементы имеют ограниченное число входов (4-6) [6,7]. Для уменьшения числа LUT в схеме КМУУ необходимо уменьшить число аргументов и термов в системе функций адресации микрокоманд [1,8]. В настоящей работе предлагается один из подходов к решению этой задачи, основанный на мультиплексировании трех источников кодов классов псевдоэквивалентных операторных линейных цепей (ОЛЦ). Предлагаемый метод является развитием результатов, полученных в работах [9,10].

Целью исследования является уменьшение числа LUT-элементов в схеме КМУУ с разделением кодов за счет мультиплексирования источников кодов классов псевдоэквивалентных ОЛЦ.

Задачей исследования является разработка метода синтеза КМУУ с разделением кодов, позволяющего оптимизировать схему адресации микрокоманд.

Алгоритм управления представлен в виде граф-схемы алгоритма (ГСА) [8]. Этот выбор определен наглядностью подобного представления и широким применением аппарата ГСА в практике инженерного проектирования.

Композиционное МУУ с разделением кодов

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 [2].

Определение 1. Операторной линейной цепью ГСА Γ называется конечная последовательность операторных вершин $\bar{b}_g = \langle b_{g_1}, \dots, b_{g_g} \rangle$ такая, что для любой пары соседних компонент $b_{g_i}, b_{g_{i+1}}$, где i – номер компоненты кортежа \bar{b}_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 \bar{b}_g , называется входом ОЛЦ \bar{b}_g , если существует дуга $\langle b_i, b_q \rangle \in E$, где $b_i \notin D^g$.

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

Определение 4. ОЛЦ \bar{b}_i, \bar{b}_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 $C = \{b_1, \dots, b_G\}$, 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

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

Пусть $F_{\max} = \max(F_1, \dots, F_G)$ – максимальное число компонент в ОЛЦ. Закодируем каждую ОЛЦ $b_g \in C$ двоичным кодом $K(b_g)$, имеющим R_1 разрядов, где

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

Для определения любой вершины $b_q \in D^g$ достаточно R_2 разрядов, представляющих код $K(b_q)$. При этом

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

Пусть для ГСА Γ выполняется следующее условие:

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

В этом случае для реализации алгоритма Γ целесообразно использовать модель КМУУ с разделением кодов (рис. 1).

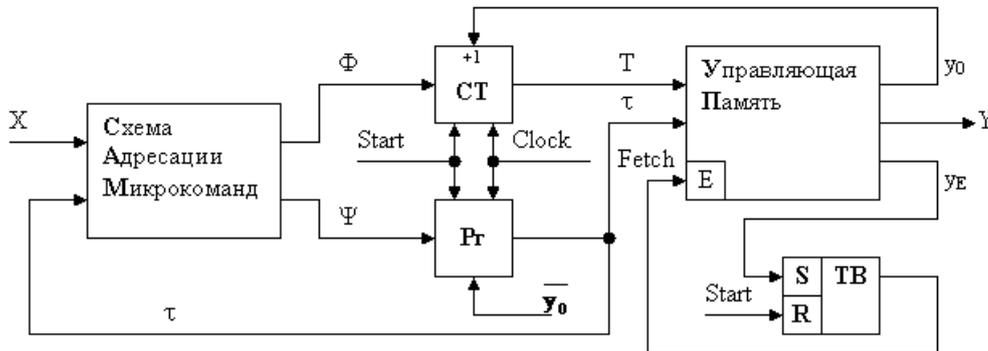


Рисунок 1 – Структурная схема КМУУ с разделением кодов

В этой модели для кодирования ОЛЦ используются переменные $\tau_r \in \tau$, где $|\tau| = R_1$. Для кодирования компонент ОЛЦ используются переменные $T_r \in T$, где $|T| = R_2$. Коды компонент выбраны так, чтобы выполнялась естественная адресация микрокоманд [1]. Для этого код первой компоненты любой ОЛЦ равен 0, второй – 1 и так далее. Естественно, что эти десятичные числа представлены их двоичными R_2 -разрядными эквивалентами.

Условимся в дальнейшем обозначать КМУУ (рис. 1) символом U_1 .

В КМУУ U_1 схема адресации микрокоманд (САМ) реализует систему функций возбуждения счетчика СТ и триггера Pr

$$\begin{aligned} \Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X). \end{aligned} \quad (5)$$

При этом адрес микрокоманды MI_q представляется в виде

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

где вершина b_q входит в состав ОЛЦ $b_g \in C$, * – знак операции конкатенации.

Композиционное МУУ U_1 функционирует следующим образом. По сигналу Start в Pr и СТ заносится начальный адрес микропрограммы, а триггер выборки ТВ устанавливается в единичное состояние. При этом Fetch = 1, что разрешает вы-

борку команд из управляющей памяти (УП). Если считанная микрокоманда не соответствует выходу ОЛЦ, то одновременно с микрооперациями $Y(b_q)$ формируется сигнал y_0 . Если $y_0 = 1$, то к содержанию СТ прибавляется единица и адресуется следующая компонента текущей ОЛЦ. Если выход ОЛЦ достигнут, то $y_0 = 0$. При этом адрес входа следующей ОЛЦ формируется схемой САМ. При достижении окончания микропрограммы формируется сигнал y_E , триггер ТВ обнуляется, и выборка микрокоманд прекращается.

Число LUT-элементов в схеме САМ зависит от числа аргументов и термов в системе (5). В настоящей работе предлагается метод, позволяющий уменьшить сложность функций в системе (5) и, следовательно, уменьшить аппаратные затраты в схеме САМ.

Основная идея предлагаемого метода

Пусть ОЛЦ $b_g \in C_1$, если O_g не связан с конечной вершиной ГСА Γ . Найдем разбиение $\Pi_C = \{V_1, \dots, V_I\}$ множества C_1 на классы псевдоэквивалентных ОЛЦ (ПОЛЦ). Выполним кодирование $b_g \in C$ так, чтобы максимально возможное число классов $V_i \in \Pi_C$, где $|\Pi_C| = I$, представлялось одним обобщенным интервалом R_1 -мерного булева пространства. Пусть p_i – число обобщенных интервалов, представляющих класс. Предста-

вим множество Π_C в виде $\Pi_C = \Pi_A \cup \Pi_B$. При этом множества Π_A и Π_B строятся следующим образом:

$$\begin{aligned} (n_i = 1) &\rightarrow V_i \in \Pi_A, \\ (n_i > 1) &\rightarrow V_i \in \Pi_B. \end{aligned} \tag{7}$$

Источником кодов классов $V_i \in \Pi_A$ является регистр Pr. При этом код класса $V_i \in \Pi_A$ определяется соответствующим интервалом R_1 -мерного булева пространства.

Закодируем классы $V_i \in \Pi_B$ двоичными кодами $C(V_i)$ разрядности

$$R_3 = \lceil \log_2(|\Pi_B| + 1) \rceil. \tag{8}$$

Используем для кодирования классов $V_i \in \Pi_B$ переменные из множества $Z = \{z_1, \dots, z_{R_3}\}$. Для формирования кодов $C(V_i)$ необходим блок преобразователя кодов (БПК). Этот блок реализует систему функций

$$Z = Z(\tau). \tag{9}$$

Современные микросхемы FPGA имеют в своем составе блоки встроенной памяти EMB (embedded memory block) [6,7]. Эти блоки имеют возможность реконфигурации, которая сводится к изменению числа входов и выходов при фиксированной емкости V_0

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

В формуле (10), переменная S означает число входов, а t_F число выходов EMB. Как правило, возможны следующие конфигурации совре-

менных EMB [6,7]: $16K \times 1$, $8K \times 2$, $4K \times 4$, $2K \times 8$, $1K \times 16$, 512×32 , 256×64 (бит). Это значит, что параметры S и t_F принадлежат следующим множествам: $S \in \{14, 13, \dots, 8\}$ и $t_F \in \{1, 2, 4, 8, 16, 32, 64\}$. При фиксированной величине t_F число ячеек в EMB определяется следующей формулой:

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

Для реализации УП достаточно M ячеек EMB, При этом блок имеет t_M выходов:

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

Пусть для некоторой ГСА Γ и микросхемы FPGA имеет место следующее неравенство:

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

В этом случае $\Delta R = t_M - (N + 3)$ разрядов кода $K(V_i)$ целесообразно формировать на свободных выходах EMB ($V_i \in \Pi_B$). Оставшиеся $R_3 - \Delta R$ разрядов кода формируются схемой БПК. Это равносильно представлению множества Z в следующем виде:

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

Переменные $Z_r \in Z^1$ формируются БПК, а переменные $Z_r \in Z^2$ – УП. Очевидно, что пересечение этих множеств пусто.

В этом случае для реализации схемы устройства управления предлагается модель КМУУ U_2 (рис. 2).

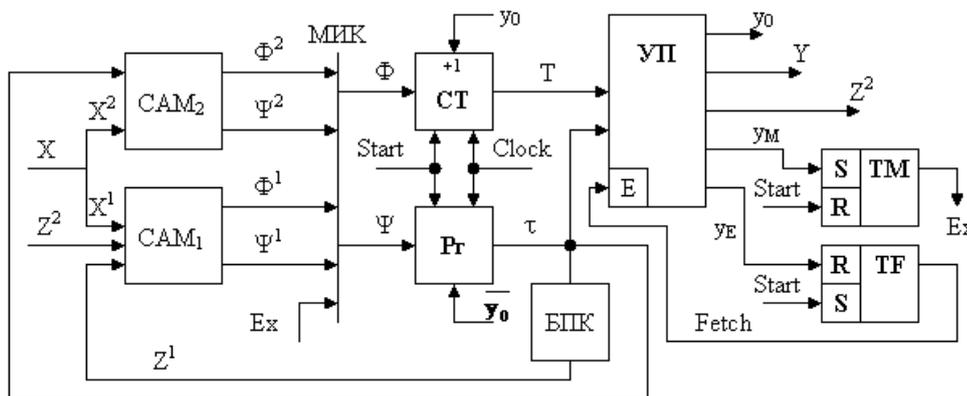


Рисунок 2 – Структурная схема КМУУ U_2

Эта модель имеет ряд отличий от модели U_1 . Во-первых, блок САМ разделен на два блока. Блок САМ₁ реализует переходы, определяемые множеством Π_B , а блок САМ₂ – множеством Π_A . Мультиплексор МИК служит для выбора источника кодов, используя переменную E_x . Значение переменной E_x определяется состоянием триггера ТМ, управляемого дополнительной переменной u_M . Блок БПК является источником кодов классов для схемы САМ₁, формируя часть этого кода. Вторым источником кода $K(V_i)$ для $V_i \in \Pi_B$ яв-

ляется управляющая память УП. Источниками кодов для схемы САМ₂ является регистр Pr. Предлагаемое КМУУ функционирует следующим образом.

По сигналу Start в Pr и СТ заносятся нулевые коды (адрес первой микрокоманды), а триггера TF и ТМ устанавливаются соответственно в "1" (Fetch = 1) и "0" ($E_x = 0$). Пока адрес входа не достигнут, то U_2 функционирует как U_1 . При достижении адреса выхода ОЛЦ $b_s \in V_i$ может формироваться переменная $u_M = 1$. В этом случае $E_x = 1$

и адрес перехода формируется схемой САМ₁. Для этого формируется система функций:

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

Если $V_i \in \Pi_A$, то переменная u_M не формируется. При этом $E_x = 0$ и адрес перехода определяется схемой САМ₂. Для этого формируется система функций:

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

Соответствующие функции передаются на выход МИК и требуемые коды $K(b_g)$ и $K(b_q)$ загружаются соответственно в Рг и СТ. Функционирование продолжается обычным образом до формирования переменной u_E (достижение окончания алгоритма управления).

Такой подход позволяет уменьшить число термов в системе (5) до абсолютного минимума. При выполнении условия

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

уменьшается число аргументов в системе (15) по сравнению с соответствующими функциями из системы (5). Недостатками данного подхода является наличие блоков МИК и БПК, потребляющими некоторые ресурсы кристалла, и увеличение разрядности слов УП. Однако, схема МИК реализуется за счет использования тристабильных выходов макроячеек, поэтому дополнительных LUT-элементов не требуется.

Очевидно, применение предложенного метода имеет смысл, если число LUT-элементов в блоке БПК будет значительно меньше параметра Δ_{LUT} . Параметр Δ_{LUT} определяется разностью числа LUT-элементов в блоке САМ и блоках САМ₁ и САМ₂.

Особенности реализации схемы КМУУ U₂

В настоящей работе предлагается метод синтеза КМУУ U₂, включающий следующие этапы:

1. Формирование для ГСА Г множеств C , C_1 , и Π_C .
2. Оптимальное кодирование ОЛЦ $b_g \in C_1$ и кодирование компонент ОЛЦ.
3. Формирование множеств Π_A и Π_B .
4. Кодирование классов $V_i \in \Pi_B$ кодами $C(V_i)$.
5. Формирование таблицы переходов для классов $V_i \in \Pi_A$.
6. Формирование таблицы переходов для классов $V_i \in \Pi_B$.
7. Формирование содержимого управляющей памяти.
8. Формирование таблицы истинности бло-

ка БПК.

9. Синтез схемы КМУУ в заданном базисе.

Этапы 1-4 выполняются по известным методикам [1-3]. Этап 9 связан с разработкой VHDL-моделей КМУУ и использованием стандартных промышленных пакетов [6,7]. Эти этапы не представляют особого интереса для иллюстрации синтеза схемы КМУУ U₂. В этой связи мы не рассматриваем данные этапы в нашей статье.

Пусть для некоторой ГСА Г₁ получено множество ОЛЦ $C = \{b_1, \dots, b_{16}\}$ и разбиение $\Pi_C = \{V_1, \dots, V_7\}$. При этом $b_{16} \notin C_1$, а классы $V_i \in \Pi_C$ определяются следующим образом: $V_1 = \{b_1\}$, $V_2 = \{b_2, b_3, b_4\}$, $V_3 = \{b_5, b_6\}$, $V_4 = \{b_7, b_8, b_9\}$, $V_5 = \{b_{10}, b_{11}\}$, $V_6 = \{b_{12}\}$, $V_7 = \{b_{13}, b_{14}, b_{15}\}$. Итак, $G = 16$, $R_1 = 4$, $\tau = \{\tau_1, \dots, \tau_4\}$.

Оптимальное кодирование ОЛЦ $b_g \in C_1$ выполняется так, чтобы максимально возможное число классов $V_i \in \Pi_C$ представлялось одним интервалом R₁-мерного булева пространства [1]. Один из вариантов кодирования представлен на рис. 3.

		$\tau_3 \tau_4$			
		00	01	11	10
$\tau_1 \tau_2$	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}

Рисунок 3 – Коды ОЛЦ $b_g \in C$

Из карты Карно (рис. 3) можно получить следующие интервалы, определяющие классы $V_i \in \Pi_C$. Класс V_1 определяется интервалом 0000; класс V_2 – интервалами 00*1 и 001*; класс V_3 – интервалом 010*; класс V_4 – интервалами 110* и 11*1; класс V_5 – интервалом 011*; класс V_6 – интервалом 1010; класс V_7 – интервалами 100* и 10*1. Таким образом, $\Pi_A = \{V_1, V_3, V_5, V_6\}$, где $K(V_1) = 0000$, $K(V_3) = 010^*$, $K(V_5) = 011^*$ и $K(V_6) = 1010$. Очевидно, $\Pi_B = \{V_2, V_4, V_7\}$ и для их кодирования необходимо R₃ = 2 разряда. Итак, $Z = \{z_1, z_2\}$. Закодируем классы $V_i \in \Pi_B$ следующим образом: $C(V_2) = 00$, $C(V_4) = 01$, $C(V_7) = 10$.

Таблица переходов формируется на основе обобщённых формул переходов [1]. Пусть, например, из ГСА Г₁ можно получить следующие

формулы:

$$B_1 \rightarrow x_1 b_3 \vee \overline{x_1 x_2} b_8 \vee \overline{x_1 x_2} b_{12}; \tag{18}$$

$$B_2 \rightarrow x_4 b_{12} \vee \overline{x_4} b_{17}.$$

Столбцы таблицы переходов включают следующую информацию: исходный класс (столбец B_i); код класса (для Π_A это код $K(B_i)$, а для Π_B – $C(B_i)$); адрес перехода ($A(b_q)$); логические условия определяющие переход (столбец X_h); функции возбуждения триггеров регистра Pr (столбец Ψ_h^1 для Π_B и Ψ_h^2 для Π_A); функции возбуждения триггеров счетчика CT (столбец Φ_h^2 для Π_A и Φ_h^1 для Π_B); номер перехода (столбец h).

Пусть $A(b_3) = 000100$, $A(b_8) = 001000$, $A(b_{12}) = 010101$, $A(b_{17}) = 110010$. Тогда фрагменты таблиц переходов для формул (13) приведены в табл. 1 и табл. 2.

Таблица 1. Таблица переходов для класса $B_1 \in \Pi_A$

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	$\overline{x_1 x_2}$	$D_2 D_4$	D_6	3

Таблица 2. Таблица переходов для класса $B_2 \in \Pi_B$

B_i	$C(B_i)$	$A(b_q)$	X_h	Ψ_h^1	Φ_h^1	h
B_2	00	010101	x_4	$D_2 D_4$	D_6	1
		110010	$\overline{x_4}$	$D_1 D_2$	D_5	2

Система (15) может быть получена из таблицы переходов для классов $B_i \in \Pi_B$. Так, из табл. 2 имеем, например $D_1 = \overline{z_1 z_2} x_4$; $D_6 = z_1 z_2 x_4$. Система (16) может быть получена из таблицы переходов для классов $B_i \in \Pi_A$. Так, из табл. 1, имеем, например, $D_2 = \tau_1 \tau_2 \tau_3 \tau_4 x_1 x_2$; $D_6 = \tau_1 \tau_2 \tau_3 \tau_4 x_1 x_2$.

Пусть из рассматриваемого примера следуют следующие множества: $Z^1 = \{Z_1\}$ и $Z^2 = \{Z_2\}$. При этом БПК реализует только часть кода $K(B_i)$. В данном случае реализуется только разряд Z_1 .

Таблица БПК имеет столбцы: B_i , $K(B_i)_j$, $C(B_i)$, Z_h , h . Здесь $K(B_i)_j$ – код класса B_i , соответствующий одному из обобщенных интервалов; Z_h – разряды кода $C(B_i)$, принимающие единичное значение в h -й строке таблицы. Для рассматриваемого примера, число строк $N_{ПК} = 6$. Этот параметр определяется суммарным числом интервалов, представляющих коды классов $B_i \in \Pi_B$. Для рассматриваемого примера блок БПК представлен в табл. 3.

Таблица 3. Таблица блока преобразователя адреса

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

Из табл. 3 имеем систему (9): $z_1 = \tau_1 \tau_2 \tau_3 \vee \tau_1 \tau_2 \tau_4$. Отметим, что для подхода, предложенного в [10], БПК должен реализовать и уравнение $z_2 = \tau_1 \tau_2 \tau_3 \vee \tau_1 \tau_2 \tau_4$.

При реализации блока УП имеются следующие особенности. Часть кода $K(B_i)$, определяемая множеством Z^2 , помещается в ячейки УП, адреса которых соответствуют выходам ОЛЦ. Этот этап выполняется тривиальным образом и в данной статье не рассматривается.

Заключение

Предлагаемый в работе метод оптимизации КМУУ основан на мультиплексировании трех источников кодов классов псевдоэквивалентных ОЛЦ. Такой подход позволяет гарантированно уменьшить число термов в системе функций возбуждения триггеров регистра и счетчика адресов микрокоманд до максимально возможной величины. Если КМУУ с разделением кодов 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 кодов, так как не все адреса выходов ОЛЦ подлежат преобразованию и не все разряды кодов классов формируются.

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 40% уменьшить число LUT-элементов по отношению к исходному КМУУ. При этом время цикла КМУУ U_2 всегда было меньше, чем у КМУУ U_1 .

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

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

Список использованной литературы

1. Barkalov A. Logic synthesis for compositional microprogram control units./ A.Barkalov., L.Titareno. – 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.Titareno. – 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” (КОТ-2011). Випуск 14 (188) / – Донецьк: ДВНЗ «ДонНТУ», 2011. – С.68-73.

Надійшла до редколегії 10.04.2013

О.О. БАРКАЛОВ¹, Л.О. ТИТАРЕНКО¹, К.М. ЄФІМЕНКО², І.Я. ЗЕЛЕНЬОВА²

¹Університет Зеленогурський (Польща)

²ДВНЗ «Донецький національний технічний університет» (Україна)

ОПТИМІЗАЦІЯ СХЕМИ АДРЕСАЦІЇ КМПК ІЗ РОЗПОДІЛОМ КОДІВ

В роботі запропоновано метод зменшення апаратних витрат у схемі КМПК із розподілом кодів, який орієнтовано на технологію FPGA. Метод засновано на використанні трьох джерел кодів класів псевдоеквівалентних ОЛЛ та мультиплексу, 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: КМПК, ГСА, ОЛЛ, FPGA, логічна схема

A.A. BARKALOV¹, L.A. TITARENKO¹, K.N. EFIMENKO², I.J. ZELENJOVA²

¹University of Zielona góra (Poland)

²Donetsk National Technical University (Ukraine)

Optimization of the Addressing Scheme in Compositional Microprogram Control Unit with Code Sharing. The article is devoted to development of methods of synthesis and optimization of compositional microprogram control units (CMCU) of FPGA (field-programmable logic arrays). For optimization of resources of a system-on-a-chip used at realization of the compositional microprogram control unit the method of a structural reduction. A method for reducing the hardware amount in the circuit of CMCU with code sharing is proposed oriented on FPGA technology. The method is based on the use of three sources of codes classes of pseudoequivalent OLC and a multiplexer to choose 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 research of the developed structures are resulted, allowing defining their efficiency and an area of optimum application. An example of the proposed method application is given.

Keywords: compositional microprogram control units, GSA, operational linear chains, FPGA, logic circuit.