УДК 004.274

А.А. Баркалов (д-р техн. наук, проф.)[1,2], К.Н. Ефименко (канд. техн. наук, доц.)[2], И.Я. Зеленева (канд. техн. наук, доц.)[2]
1) Университет Зеленогурский, Польша,
2) ГВУЗ «Донецкий национальный технический университет», г. Донецк
кафедра компьютерной инженерии
E-mail: irina@cs.dgtu.donetsk.ua; A.Barkalov@iie.uz.zgora.pl

ОПТИМИЗАЦИЯ СХЕМЫ АДРЕСАЦИИ КМУУ С ЭЛЕМЕНТАРНЫМИ ЦЕПЯМИ

Предлагается метод уменьшения аппаратурных затрат в схеме КМУУ с элементарными цепями, ориентированный на технологию FPGA. Метод основан на использовании трех источников кодов классов псевдоэквивалентных ЭОЛЦ 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, ЭОЛЦ, FPGA, логическая схема.*

Общая постановка проблемы

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 аппаратурные 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]. Для уменьшения числа LUT в схеме КМУУ необходимо уменьшить число аргументов и термов в системе функций адресации микрокоманд [1,6]. В настоящей работе предлагается один из подходов к решению этой задачи, основанный на мультиплексировании трех источников кодов классов псевдоэквивалентных элементарных операторных линейных цепей (ЭОЛЦ). Предлагаемый метод является развитием результатов, полученных в работах [7,8].

Цель исследования заключается в уменьшении числа LUT-элементов в схеме КМУУ с элементарными цепями за счет мультиплексирования трех источников кодов классов псевдоэквивалентных ЭОЛЦ.

Задачей исследования является разработка метода синтеза КМУУ с элементарными цепями, позволяющего оптимизировать по схему адресации микрокоманд.

Алгоритм управления представлен в виде граф-схемы алгоритма (ГСА) [6]. Этот выбор определен наглядностью подобного представления и широким применением аппарата ГСА в практике инженерного проектирования.

Композиционное МУУ с элементарными цепями

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 Г.

© Баркалов А.А., Ефименко К.Н., Зеленева И.Я., 2013

В операторных вершинах $b_q \in E_1$ записываются наборы микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1,...,y_N\}$ – множество микроопераций. В условных вершинах $b_q \in E_2$ записываются элементы множества логических условий $X = \{x_1,...,x_L\}$. Введем ряд определений, взятых из [2].

Определение 1. Операторной линейной цепью (ОЛЦ) ГСА Г называется конечная последовательность операторных вершин $\alpha_g = \langle b_{g1},...,b_{gF_g} \rangle$ такая, что для любой пары соседних компонент b_{gi}, b_{gi+1}, где i – номер компоненты кортежа α_g, существует дуга $\langle b_{gi}, b_{gi+1} \rangle \in E$.

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 $b_t \notin D^g$. Элементарная ОЛЦ (ЭОЛЦ) имеет только один вход.

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

Пусть для некоторой ГСА Г сформировано множество ЭОЛЦ $C = \{\alpha_1,...,\alpha_G\}$, 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

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

Пусть $F_{max} = \max(F_1,...,F_G)$ – максимальное число компонент в ЭОЛЦ. Закодируем каждую ЭОЛЦ $\alpha_g \in C$ двоичным кодом $K(\alpha_g)$, 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 Г выполняется следующее условие:

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

В этом случае для реализации алгоритма Г целесообразно использовать модель КМУУ с элементарными цепями (рис. 1).

В этой модели для кодирования ЭОЛЦ используются переменные $\tau_r \in \tau$, где $|\tau| = R_1$. Для кодирования компонент ЭОЛЦ 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 ЭОЛЦ 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 триггера Рг

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

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 ЭОЛЦ $\alpha_g \in C$, * – знак операции конкатенации.

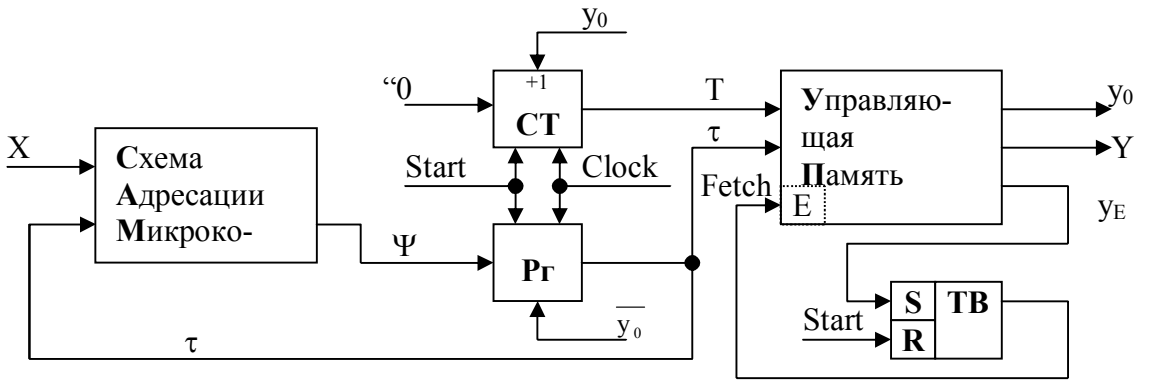


Рисунок 1 – Структурная схема КМУУ с элементарными цепями

Композиционное МУУ U_1 функционирует следующим образом. По сигналу Start в Рг и СТ заносится начальный адрес микропрограммы, а триггер выборки ТВ устанавливается в единичное состояние. При этом Fetch = 1, что разрешает выборку команд из управляющей памяти (УП). Если считанная микрокоманда не соответствует выходу ЭОЛЦ, то одновременно с микрооперациями $Y(b_q)$ формируется сигнал y_0. Если $y_0 = 1$, то к содержимому СТ прибавляется единица и адресуется следующая компонента текущей ЭОЛЦ. Если выход ОЛЦ достигнут, то $y_0 = 0$. При этом адрес входа следующей ЭОЛЦ 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 аппаратурные затраты в схеме САМ.

Основная идея предлагаемого метода

Пусть ЭОЛЦ $\alpha_g \in C_1$, если O_g не связан с конечной вершиной ГСА Г. Найдем разбиение $\Pi_C = \{B_1,...,B_I\}$ множества C_1 на классы псевдоэквивалентных ЭОЛЦ (ПЭОЛЦ). Выполним кодирование $\alpha_g \in C$ так, чтобы максимально возможное число классов $B_i \in \Pi_C$, где $|\Pi_C| = I$, представлялось одним обобщённым интервалом R_1-мерного булева пространства. Пусть n_i – число обобщённых интервалов, представляющих класс. Представим множество Π_C в виде $\Pi_C = \Pi_A \cup \Pi_B$. При этом множества Π_A и Π_B строятся следующим образом:

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

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

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

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

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

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

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 емкостиV_0

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

В формуле (10), переменная S означает число входов, а t_F число выходов ЕМВ. Как правило, возможны следующие конфигурации современных ЕМВ [6,7]: 16К×1, 8К×2, 4К×4, 2К×8, 1К×16, 512×32, 256×64 (бит). Это значит, что параметры S и t_F принадлежат следующим множествам: $S \in \{14,13,...,8\}$ и $t_F \in \{1,2,4,8,16,32,64\}$. При фиксированной величине t_F число ячеек в ЕМВ определяется следующей формулой:

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

Для реализации УП достаточно М ячеек ЕМВ, При этом блок имеет t_M выходов:

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

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

$$N + 3 + R_3 > t_M > N + 3 . \quad (13)$$

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

$$Z = Z^1 \cup Z^2 . \quad (14)$$

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).

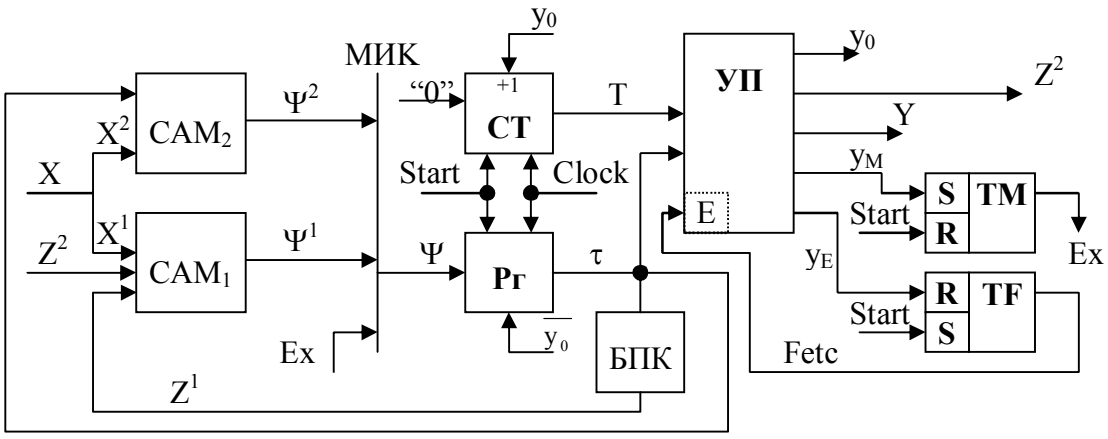


Рисунок 2 – Структурная схема КМУУ U_2

Эта модель имеет ряд отличий от модели U_1. Во-первых, блок САМ разделен на два блока. Блок $САМ_1$ реализует переходы, определяемые множеством Π_B, а блок $САМ_2$ – множеством Π_A. Мультиплексор МИК служит для выбора источника кодов, используя переменную Ex. Значение переменной Ex определяется состоянием триггера ТМ, управляемого дополнительной переменной y_M. Блок БПК является источником кодов классов для схемы $САМ_1$, формируя часть этого кода. Источниками кодов для схемы $САМ_2$ является регистр Рг.

Предлагаемое КМУУ функционирует следующим образом. По сигналу Start в Рг и СТ заносятся нулевые коды (адрес первой микрокоманды), а триггера TF и ТМ устанавливаются соответственно в "1" (Fetch = 1) и "0" (Ex = 0). Пока адрес входа не достигнут, то U_2 функционирует как U_1. При достижении адреса выхода ЭОЛЦ $\alpha_g \in B_i$ может формироваться переменная $y_M = 1$. В этом случае Ex = 1 и адрес перехода формируется схемой $САМ_1$. Для этого формируется система функций:

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

Если $B_i \in \Pi_A$, то переменная y_M не формируется. При этом Ex = 0 и адрес перехода определяется схемой CAM_2. Для этого формируется система функций:

$$\Psi^2 = \Psi^2(\tau, X^2). \quad (16)$$

Соответствующие функции передаются на выход МИК и требуемый код $K(\alpha_g)$ загружается в Рг. Функционирование продолжается обычным образом до формирования переменной y_E (достижение окончания алгоритма управления).

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

$$R_3 < R_1 \quad (17)$$

уменьшается число аргументов в системе (10) по сравнению с соответствующими функциями из системы (5). Недостатками данного подхода является наличие блоков МИК и БПК, потребляющими некоторые ресурсы кристалла, и увеличение на 1 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 CAM и блоках CAM_1 и CAM_2.

Особенности реализации схемы КМУУ U_2.

В настоящей работе предлагается метод синтеза КМУУ U_2, включающий следующие этапы:

1. Формирование для ГСА Г множеств C, C_1, и Π_C.
2. Оптимальное кодирование ЭОЛЦ $\alpha_g \in C_1$ и кодирование компонент ЭОЛЦ.
3. Формирование множеств Π_A и Π_B.
4. Кодирование классов $B_i \in \Pi_B$ кодами $C(B_i)$.
5. Формирование таблицы переходов для классов $B_i \in \Pi_A$.
6. Формирование таблицы переходов для классов $B_i \in \Pi_B$.
7. Формирование содержимого управляющей памяти.
8. Формирование таблицы истинности блока 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 [4]. Эти этапы не представляют особого интереса для иллюстрации синтеза схемы КМУУ U_2. В этой связи мы не рассматриваем данные этапы в нашей статье.

Пусть для некоторой ГСА Γ_1 получено множество ЭОЛЦ $C = \{\alpha_1,...,\alpha_{16}\}$ и разбиение $\Pi_C = \{B_1,...,B_7\}$. При этом $\alpha_{16} \notin C_1$, а классы $B_i \in \Pi_C$ определяются следующим образом: $B_1 = \{\alpha_1\}$, $B_2 = \{\alpha_2, \alpha_3, \alpha_4\}$, $B_3 = \{\alpha_5, \alpha_6\}$, $B_4 = \{\alpha_7, \alpha_8, \alpha_9\}$, $B_5 = \{\alpha_{10}, \alpha_{11}\}$, $B_6 = \{\alpha_{12}\}$, $B_7 = \{\alpha_{13}, \alpha_{14}, \alpha_{15}\}$. Итак, G = 16, $R_1 = 4$, $\tau = \{\tau_1,...,\tau_4\}$.

Оптимальное кодирование ЭОЛЦ $\alpha_g \in C_1$ выполняется так, чтобы максимально возможное число классов $B_i \in \Pi_C$ представлялось одним интервалом R_1-мерного булева пространства [1]. Один из вариантов кодирования представлен на рис. 3.

Из карты Карно (рис. 3) можно получить следующие интервалы, определяющие классы $B_i \in \Pi_C$. Класс B_1 определяется интервалом 0000; класс B_2 – интервалами 00*1 и 001*; класс B_3 – интервалом 010*; класс B_4 – интервалами 110* и 11*1; класс B_5 – интервалом 011*; класс B_6 – интервалом 1010; класс B_7 – интервалами 100* и 10*1. Таким образом, $\Pi_A = \{B_1, B_3, B_5, B_6\}$, где $K(B_1) = 0000$, $K(B_3) = 010*$, $K(B_5) = 011*$ и $K(B_6) = 1010$. Очевидно,

$\Pi_B = \{B_2, B_4, B_7\}$ и для их кодирования необходимо $R_3 = 2$ разряда. Итак, $Z = \{z_1, z_2\}$. Закодируем классы $B_i \in \Pi_B$ следующим образом: $C(B_2) = 00$, $C(B_4) = 01$, $C(B_7) = 10$.

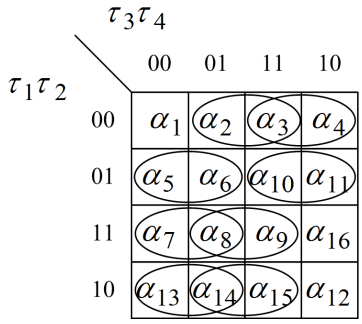


Рисунок 3 – Коды ЭОЛЦ $\alpha_g \in C$

Таблица переходов формируется на основе обобщённых формул переходов [1]. Пусть, например, из ГСА Γ_1 можно получить следующие формулы:

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

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 Рг (столбец Ψ_h^1 для Π_B и Ψ_h^2 для Π_A); 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	h
B_1	0000	000100	x_1	D_4	1
		001000	$\overline{x}_1 x_2$	D_3	2
		010101	$\overline{x}_1 \overline{x}_2$	$D_2 D_4$	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	h
B_2	00	010101	x_4	$D_2 D_4$	1
		110010	$\overline{x}_4$	$D_1 D_2$	2

Система (15) может быть получена из таблицы переходов для классов $B_i \in \Pi_B$. Так, из табл. 2 имеем, например $D_1 = \overline{z}_1 \overline{z}_2 \overline{x}_4$. Система (16) может быть получена из таблицы переходов для классов $B_i \in \Pi_A$. Так, из табл. 1, имеем, например, $D_2 = \overline{\tau}_1 \tau_2 \tau_3 \tau_4 x_1 \overline{x}_2$.

Пусть из рассматриваемого примера следуют 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

$H_{ПК} = 6$. 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): $z_1 = \tau_1\overline{\tau_2}\overline{\tau_3} \vee \tau_1\overline{\tau_2}\tau_4$. Отметим, что для подхода, предложенного в [8], БПК должен реализовать и уравнение $z_2 = \tau_1\tau_2\overline{\tau_3} \vee \tau_1\tau_2\tau_4$. 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.

Выводы

Предлагаемый в работе метод оптимизации КМУУ основан на мультиплексировании трех источников кодов классов псевдоэквивалентных ЭОЛЦ. Такой подход позволяет гарантированно уменьшить число термов в системе функций возбуждения триггеров регистра и счетчика адресов микрокоманд до минимальной возможной величины. Если КМУУ с элементарными цепями 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 ЭОЛЦ подлежат преобразованию и не все разряды кодов классов формируются. Проведенные авторами исследования показали, что предложенный метод позволяет до 42% 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 ЭОЛЦ) для уменьшения числа LUT-элементов в схеме КМУУ с элементарными цепями.

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. 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.– 736с.
6. Baranov S. Logic and System Design of Digital Systems / S Baranov. - Tallinn: TTU, 2008. – 266 pp.

7. 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). – 2011. – Випуск 9 (179). – С.26-35.
8. Оптимизация схемы КМУУ с элементарными цепями / [А.А. Баркалов, Л.А. Титаренко, К.Н. Ефименко, Я.М. Липински] // Наукові праці ДонНТУ. Серія „Проблеми моделювання та автоматизації проектування" (МАП-2011). – 2011. – Вип.10 (197). – С.21-30.

Надійшла до редакції:
28.03.2013

Рецензент:
д-р техн. наук, проф. Скобцов Ю.О.

О.О. Баркалов, К.М. Єфіменко, І.Я. Зеленьова
Університет Зеленогурський, ДВНЗ «Донецький національний технічний університет»
Оптимізація схеми адресації КМПК із елементарними ланцюгами. *В роботі запропоновано метод зменшення апаратурних витрат у схемі КМПК із елементарними ланцюгами, який орієнтовано на технологію FPGA. Метод засновано на використанні трьох джерел кодів класів псевдоеквівалентних ЕОЛЛ та мультиплексору, який дозволяє вибрати одне з цих джерел. Такий підхід дозволить зменшити число LUT елементів у схемі адресації КМПК. Наведено приклад використання запропонованого методу.*
Ключові слова: *композиційний мікропрограмний пристрій керування, ГСА, елементарний операторній лінійний ланцюг, FPGA, логічна схема.*

A.A. Barkalov, K.N. Yefimenko, I.J. Zelenjova
University of Zielona, Donetsk National Technical University
Optimization of Addressing Circuit for CMCU with Elementary Chains. *The article is devoted to the development of synthesis and optimization methods of compositional microprogram control units (CMCU) on FPGA (field-programmable logic arrays). Compositional microprogram control unit, particularly the structure of CMCU with code sharing, is effective tool for application of linear control algorithm. For optimization of a system-on-a-chip resources, applied in realization of the compositional microprogram control unit, the method of a structural reduction is used. A method for reducing the hardware amount in the circuit of CMCU with elementary chains oriented on FPGA technology is proposed. This method is based on the usage of three sources of codes classes of pseudoequivalent EOLC (elementary operational linear chains) and a multiplexer to choose one of these sources. Main steps of proposed method include: forming and optimal encoding of EOLC; definition of classes of pseudoequivalency; transforming of tne initial table of transitions and sharing it by two parts according to the classes; forming of control memory content; definition of functional expressions for the block of code transforming. Such an approach would reduce the number of LUT (look-up table)-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. Moreover, modern FPGA contains macrocells of embedded memory blocks (EMB). Such blocks can be reconfigured towards changing of input's and output's number with saving of fixed square of EMB macrocell. This property allows to apply the combinational circuit for multiplexer of code source using the tristable outputs of EMB. Proposed method uses this property for reducing the number of used macrocells in the logic circuit of CMCU. Results of research of the developed structures, allowing defining their efficiency and an area of optimum application, are discussed. And the example of the application of proposed method is given.*
Keywords: *compositional microprogram control units, GSA, elementary operational linear chains, FPGA, logic circuit.*