

Баркалов А. А.¹, Титаренко Л. А.², Зеленева И. Я.³, Цололо С. А.⁴¹Д-р техн. наук, профессор, ДонНТУ, профессор, Зеленогурский Университет (Польша)²Д-р техн. наук, профессор, ХНУРЭ, профессор, Зеленогурский Университет (Польша)³Канд. техн. наук, доцент, ДонНТУ, Украина, E-mail: irina@cs.dgtu.donetsk.ua⁴Канд. техн. наук, доцент, ДонНТУ, Украина

ОПТИМИЗАЦИЯ СХЕМЫ КОМПОЗИЦИОННОГО МИКРОПРОГРАММНОГО УСТРОЙСТВА УПРАВЛЕНИЯ С ОБЩЕЙ ПАМЯТЬЮ

Предлагается метод уменьшения аппаратных затрат в схеме композиционного микропрограммного устройства управления с общей памятью при реализации на микросхемах FPGA, с учетом ограничения на число входов LUT элементов. Метод основан на применении двух источников кодов классов псевдоэквивалентных операторных линейных цепей, а также на использовании избыточности встроенных блоков памяти FPGA. Такой подход позволяет уменьшить требуемое число LUT элементов в схеме адресации композиционного устройства, вследствие чего уменьшается площадь кристалла FPGA, занимаемая схемой КМУУ с общей памятью, что позволяет получить реализацию, обладающую меньшей стоимостью, чем известные из литературы аналоги. В статье описан пример применения предложенного метода.

Ключевые слова: композиционное микропрограммное устройство управления, операторная линейная цепь, граф-схема алгоритма, FPGA, мультиплексор, логическая схема.

ВВЕДЕНИЕ

Модель композиционного микропрограммного устройства управления (КМУУ) с общей памятью [2, 3] применяется для реализации схемы устройства управления (УУ) в том случае, когда граф-схема алгоритма (ГСА) управления некоторой системы относится к классу линейных [1]. Современный элементный базис предполагает широкое использование программируемых логических интегральных схем (ПЛИС) вида FPGA (field-programmable gate arrays) [4, 5] для реализации схем УУ. Основу FPGA представляют макроячейки LUT (look-up table), имеющие ограниченное число входов (4–6) [6, 7]. С учетом этого ограничения, 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. В данном решении учитывается такая особенность рассматриваемой структуры КМУУ, как совмещение функций регистра памяти и счетчика адреса микрокоманд на одном только счетчике. Основная идея метода заключается в 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 и избыточности встроенных блоков памяти.

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

Предлагаемый метод является развитием результатов, полученных в работе [9].

ПОСТАНОВКА ЗАДАЧИ

Обозначим, что исходными данными для задачи синтеза схемы композиционного микропрограммного устройства управления служит граф-схема управляющего алгоритма, т.е. некоторая ГСА $\Gamma = \Gamma(V, E)$, представленная множествами вершин V и соединяющих их дуг E [2]. 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 $\alpha_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$ 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 $b_t \notin D^g$.

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

Пусть для некоторой ГСА Γ сформировано множество ОЛЩ $C = \{\alpha_1, \dots, \alpha_G\}$, 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

$$R = \lceil \log_2 M \rceil. \tag{1}$$

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

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

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].

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

$$\Phi = \Phi(T, X). \tag{3}$$

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 ЕМВ (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 устране-

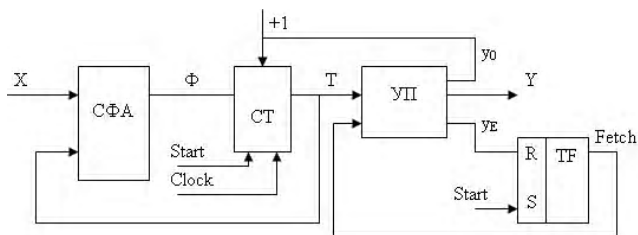


Рис. 1. Структурная схема КМУУ U_1

ния этого недостатка используют оптимальную адресацию микрокоманд, что приводит к КМУУ U_2 [9]. Однако такая адресация не всегда возможна. Уменьшение числа термов системе (3) гарантируется в КМУУ U_3 , где осуществляется преобразование адресов микрокоманд в коды классов псевдоэквивалентных ОЛЩ [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.

Как известно, LUT-элементы имеют ограниченное число входов $S (S = [4..6])$. Для уменьшения числа LUT-элементов целесообразно уменьшать число аргументов в реализуемых системах функций. Предлагаемый метод позволяет выполнить такое уменьшение числа аргументов.

ОСНОВНАЯ ИДЕЯ ПРЕДЛАГАЕМОГО МЕТОДА

Найдем разбиение Π_C множества ОЛЩ C на классы псевдоэквивалентных ОЛЩ. Пусть $\Pi_C = \{B_1, \dots, B_1\}$, то есть в ГСА имеется I псевдоэквивалентных ОЛЩ.

Пусть для реализации блока УП используются встроенные блоки памяти, входящие в состав FPGA [6, 7]. Эти блоки имеют фиксированную емкость V , однако число их входов t может меняться. Параметр t может принимать одно из фиксированных значений из множества $Q_F = \{1, 2, 4, 9, 18, 36\}$ [6, 7].

Для реализации системы функций $Y \cup \{y_0, y_E\}$ необходимо

$$n_f = \lceil (N + 2) / t_f \rceil \tag{4}$$

блоков ВБП. При этом параметр t_f (число выходов блока) определяется как

$$t_f = \lfloor V / 2^R \rfloor \tag{5}$$

Этот параметр должен быть выбран ближайшим к меньшему значению из множества Q_F . Общее число выходов блоков, входящих в УП, можно найти как

$$t_0 = n_f t_f. \tag{6}$$

Пусть выполняется условие

$$t_0 - (N + 3) = k > 0. \tag{7}$$

В этом случае множество Π_C можно разбить на два класса: $\Pi_C = \Pi_1 \cup \Pi_2$, где $\Pi_1 \cap \Pi_2 = \emptyset$. Класс Π_2 включает

$$I_2 = 2^k \tag{8}$$

элементов. Класс Π_1 будет включать

$$I_1 = I - I_2 \tag{9}$$

элементов. Рассмотрим случай, когда $I_1 \neq 0$ и $I_2 \neq 0$. При этом классы $V_i \in \Pi_1$ могут быть закодированы кодами $K(V_i)$, имеющими

$$R_1 = \lceil \log_2 I_1 \rceil \quad (10)$$

разрядов. Пусть для кодирования классов $V_i \in \Pi_1$ используются переменные $\tau_r \in \tau$, где $|\tau| = R_1$.

Закодируем классы $V_i \in \Pi_2$ кодами $C(V_i)$, имеющими

$$R_2 = k. \quad (11)$$

разрядов. Пусть для кодирования классов $V_i \in \Pi_2$ используются переменные $Z_r \in Z$, где $|Z| = R_2$. В этом случае для реализации УУ предлагается модель КМУУ U_4 (рис. 2).

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

$$\Phi^1 = \Phi^1(\tau, X^1). \quad (12)$$

Схема СФА₂ служит для задания переходов из классов $V_i \in \Pi_2$. При этом $u_m = E_x = 0$ и

$$\Phi^2 = \Phi^2(T, X^2). \quad (13)$$

Мультиплексор МИК служит для выбора источника функций возбуждения СТ:

$$\Phi = E_x \Phi^1 \vee \overline{E_x} \Phi^2, \quad (14)$$

при этом $\Phi^1 \cup \Phi^2 = \Phi$. Очевидно, через МИК передаются только функции $\phi_r \in \Phi^1 \cap \Phi^2$.

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

$$\tau = \tau(T). \quad (15)$$

Сравнение блоков УП КМУУ U_1 и U_4 показывает, что они отличаются наличием выхода u_m . Как уже стало ясно, этот выход используется для управления триггером ТМ и, следовательно, мультиплексором МИК.

Такой подход позволяет уменьшить число термов в системе (3) до абсолютного возможного минимума. Кроме того, уменьшается сложность блока БПА по сравнению с КМУУ U_3 . Как видно на рис. 2, коды классов $V_i \in \Pi_2$ генерируются управляющей памятью. Такой подход позволяет уменьшить число входов в LUT-элементах блоков СФА₁ и СФА₂. 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 статье предлагается метод синтеза КМУУ U_4 , включающий следующие этапы:

1. Формирование множества ОЛЦ С для ГСА Г.
2. Формирование разбиения Π_C множества С.
3. Оптимальная адресация микрокоманд.
4. Кодирование классов $V_i \in \Pi_1$ и $V_i \in \Pi_2$.
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.

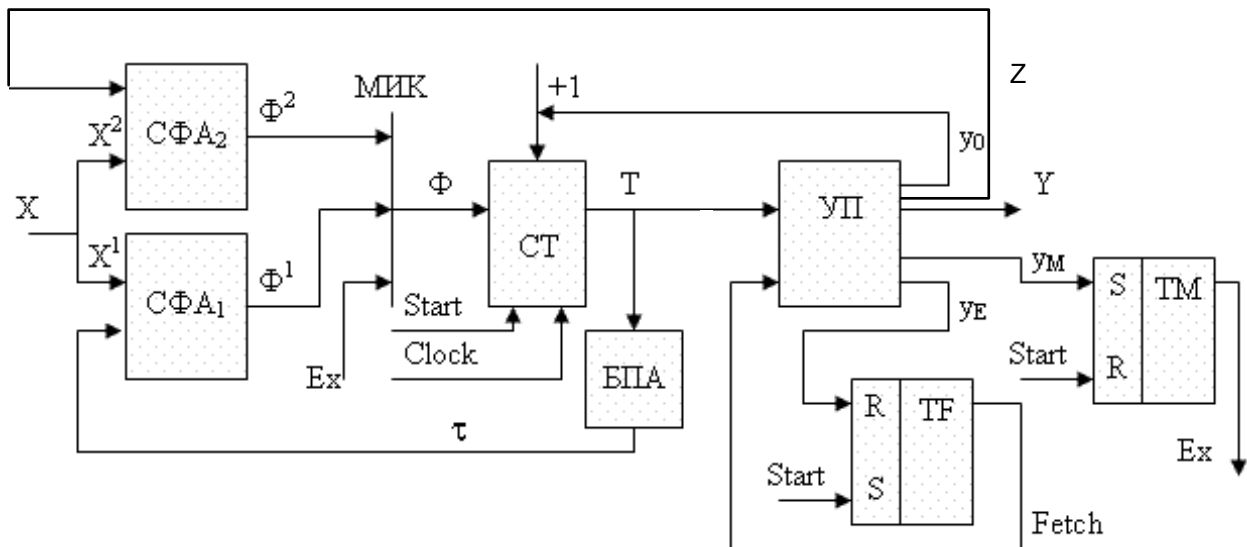


Рис. 2. Структурная схема КМУУ U_4

ПРИМЕР ПРИМЕНЕНИЯ ПРЕДЛОЖЕННОГО МЕТОДА

Пусть для некоторой ГСА Г 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 $I = 5$ 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 $\alpha_{12} \in C$ 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].

Выполним адресацию микрокоманд так, чтобы выполнялось условие (2) и максимально возможное число классов представлялось одним обобщенным интервалом R-мерного булевого 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. По 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 Г.

Пусть для реализации УП используются встроенные блоки памяти, причем их параметры и число микроопераций N таковы, что $R_2 = 1$. Следовательно, два класса $V_i \in P_C$ можно поместить в блок Π_2 . Пусть число входов LUT-элементов $S=4$. Тогда разбиение по классам целесообразно выполнить следующим образом: $\Pi_1 = \{V_1, V_2\}$ и $\Pi_2 = \{V_3, V_4\}$. Итак, имеем $I_1 = I_2 = 2$, $R_1 = R_2 = 1$, $\tau = \{\tau_1\}$, $Z = \{z_1\}$. Выполним кодирование классов $V_i \in P_C$. Результат кодирования не влияет на сложность схемы. Пусть $K(V_1) = 0$, $K(V_2) = 1$, $K(V_3) = 0$, $K(V_4) = 1$.

		$T_1 T_2 T_3$							
		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. Формирование адресов микрокоманд КМУУ $U_4(\Gamma)$

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

$$\begin{aligned}
 V_1 &\rightarrow x_1 b_9 \vee \overline{x_1} x_2 b_{11} \vee \overline{\overline{x_1} x_2} x_3 b_{13} \vee \overline{\overline{\overline{x_1} x_2} x_3} b_{15}; \\
 V_2 &\rightarrow x_2 b_5 \vee \overline{x_2} x_5 b_6 \vee \overline{x_2 x_5} x_6 b_{17} \vee \overline{\overline{x_2 x_5} x_6} b_{22}; \\
 V_3 &\rightarrow x_4 x_6 b_{21} \vee \overline{x_4} \overline{x_6} b_{25} \vee \overline{\overline{x_4} x_3} b_{27} \vee \overline{\overline{\overline{x_4} x_3}} b_{29}; \\
 V_4 &\rightarrow x_1 b_{32} \vee \overline{x_1} b_{23}.
 \end{aligned}
 \tag{16}$$

Таблицы переходов имеют следующие столбцы V_i , $K(V_i)$, $A(b_q)$, X_h , Φ_h , h . Здесь X_h – набор ЛУ, который определяет связь между классом V_i и вершиной b_q в формулах перехода вида (16); Φ_h – набор функций возбуждения СТ, равных единице для формирования адреса $A(b_q)$. Разница между таблицами заключается в том, что для классов $V_i \in \Pi_1$ код представляется переменными $\tau_r \in \tau$, а для $V_i \in \Pi_2$ – переменными $T_r \in T$. Для КМУУ $U_4(\Gamma)$ переходы для классов $V_i \in \Pi_1$ представлены в табл. 1, а для классов $V_i \in \Pi_2$ – в табл. 2. Первая таблица имеет $N_1 = 6$ строк, а вторая – $N_2 = 8$ строк.

Для табл. 1 и табл. 2 адреса микрокоманд определяются по рис. 3. Из табл. 1 получаем систему (12). Например, $D_1 = \tau_1 \overline{x_2} x_5$, $D_2 = \tau_1$ (после минимизации). Из табл. 2 получаем систему функций (13). Например, $D_1 = 1$, $D_2 = \overline{z_1} x_4 x_6 \vee \overline{z_1} x_4 \vee \overline{z_1} x_1$ (после минимизации). Из этих таблиц получаем также множества $X^1 = \{x_1, x_2, x_3, x_5, x_6\}$, $X^2 = \{x_1, x_3, x_4, x_6\}$, $\Phi^1 = \Phi^2 = \Phi$.

Для формирования таблицы БПА необходимо найти соответствие между кодами классов $V_i \in \Pi_1$ 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 $\alpha_g \in C$. Однако в рассматриваемом случае систему (15) можно получить тривиальным образом. Очевидно, что $\tau_1 = 1$ для $K(V_2)$. С другой стороны, $K(V_2) = 01***$. Следовательно, $\tau_1 = \overline{T_1} T_2$.

Синтез схемы МИК сводится к реализации системы (14). Эта задача является тривиальной, а схема включает R-число LUT элементов, каждый из которых имеет по 3 входа.

Для формирования содержимого УП выполняется преобразование исходной ГСА Г:

1. Если вершина $b_q \in E_1$ не является выходом ОЛЦ $\alpha_g \in C$, то в нее включается переменная y_q .
2. Если вершина $b_q \in E_1$ связана с вершиной b_E , то в нее 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 нее вводится переменная y_M .

Таблиця 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_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

Таблиця 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_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

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 данной 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].

ЗАКЛЮЧЕНИЕ

Предлагаемый в работе метод оптимизации КМУУ основан на мультиплексировании двух источников кодов классов псевдоэквивалентных ОЛЦ и использовании избыточности встроенных блоков памяти FPGA. Описанный выше способ адресации классов псевдоэквивалентных ОЛЦ ведет к уменьшению числа термов в системе функций возбуждения триггеров, составляющих счетчик адреса микрокоманд, до минимально возможной величины. 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 ведет к уменьшению числа уровней в схеме, и таким образом, 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 структурных особенностей КМУУ с общей памятью (совмещение функций счетчика адреса и регистра памяти, использование классов псевдоэквивалентных ОЛЦ) и особенностей FPGA (конфигурируемость встроенных блоков памяти) 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 базе FPGA.

СПИСОК ЛИТЕРАТУРЫ

1. *Barkalov, A. Logic synthesis for compositional microprogram control units / A. Barkalov, L. Titarenko. – Berlin : Springer, 2008. – 272 p.*

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 p.
4. Maxfield, S. The Design Warrior's Guide to FPGAs. – Amsterdam : Elsevier, 2004. – 541 p.
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. All Programmable Technologies from Xilinx Inc. [электронный ресурс]: ресурс содержит информацию о семействах ПЛИС производства Xilinx Inc – Режим доступа: <http://www.xilinx.com> – Загл. с экрана.
7. FPGA CPLD and ASIC from Altera [электронный ресурс]: технические характеристики ПЛИС производства Altera Corporation – Режим доступа: <http://www.altera.com>. – Загл. с экрана.
8. Baranov, S. Logic and System Design of Digital Systems / S. Baranov. – Tallinn : TTU, 2008. – 266 p.
9. Баркалов, А. А. Реализация композиционных микропрограммных устройств управления на FPGA-микросхемах / А. А. Баркалов, Л. А. Титаренко, А. Н. Мирошкин // Радиоэлектроника и информатика. – 2011. – № 1. – С. 52–55.

Стаття надійшла до редакції 03.02.2013.

Баркалов О. О.¹, Титаренко Л. О.², Зеленьова І. Я.³, Цололо С. О.⁴

¹Д-р техн. наук, професор, професор ДонНТУ, професор, Зеленогурський університет (Польща)

²Д-р техн. наук, професор, професор, Зеленогурський університет (Польща)

³Канд. техн. наук, доцент, ДонНТУ, Україна

⁴Канд. техн. наук, доцент, ДонНТУ, Україна

ОПТИМИЗАЦИЯ СХЕМИ КОМПОЗИЦИЙНОГО МИКРОПРОГРАМНОГО ПРИСТРОЮ КЕРУВАННЯ ІЗ ЗАГАЛЬНОЮ ПАМ'ЯТТЮ

Запропоновано метод зменшення апаратних витрат у схемі композиційного мікропрограмного пристрою керування із загальною пам'яттю при реалізації на микросхемах FPGA, із урахуванням обмеження на число входів LUT елементів. Метод засновано на застосуванні двох джерел кодів класів псевдоєквівалентних операторних лінійних ланцюгів, а також на використанні надмірності, що має місце у вбудованих блоках пам'яті FPGA. Такий підхід дозволяє зменшити потрібну кількість LUT елементів у схемі адресації композиційного пристрою, внаслідок чого зменшується площа кристалу FPGA, яку займає схема КМУУ із загальною пам'яттю, що дозволяє отримати реалізацію, якій притаманна менша вартість, ніж відомим з літератури аналогам. У статті наведено приклад використання методу.

Ключові слова: композиційний мікропрограмний пристрій керування, операторний лінійний ланцюг, граф-схема алгоритма, FPGA, мультиплексор, логічна схема.

Barkalov A. A.¹, Titarenko L. A.², Zelenyova I. Y.³, Tsololo S. A.⁴

¹Doctor of Science, Professor, DonNTU, professor of University of Zielona Góra

²Doctor of Science, Professor, KHTURE, professor of University of Zielona Góra

³Ph.D., associated professor, DonNTU, Ukraine

⁴Ph.D., associated professor, DonNTU, Ukraine

OPTIMIZATION OF COMPOSITIONAL MICROPROGRAMMING CONTROL UNIT WITH COMMON MEMORY

A method for reducing the hardware amount in the circuit of compositional microprogramming control unit with common memory oriented to FPGA technology is proposed. The restrictions of LUT's input number are taken into account. The method is based on the use of two sources of codes classes of pseudoequivalent operational linear chain and use a multiplexer to choose one of these sources. Also the surplus of FPGA's embedded memory blocks is used. Such an approach would reduce the number of LUT elements in the addressing subcircuit of compositional microprogramming control unit, that leads to reducing of common hardware amount and price of the unit. An example of the proposed method application is given.

Keywords: compositional microprogramming control unit, operational linear chain, flow-chart of algorithm, FPGA, multiplexer, logic circuit.

REFERENCES

1. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units, Berlin, Springer, 2008, 272 p.
2. Barkalov A. A., Titarenko L. A. Sintez mikroprogrammykh avtomatov na zakaznykh i programmiemykh SBIS, Doneck, UNITEK, 2009, 336 p.
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. Grushvickij R. I., Mursaev A. X., Ugryumov E. P. Proektirovanie sistem na mikrosxemax s programmiemoj strukturoj, S-Pb, BXV, Peterburg, 2006, 736 p.
6. All Programmable Technologies from Xilinx Inc (contains information about Xilinx Inc programmable devices), available at: <http://www.xilinx.com>.
7. FPGA CPLD and ASIC from Altera, technical features of Altera's devices, available at: <http://www.altera.com>.
8. Baranov S. Logic and System Design of Digital Systems, Tallinn, TTU, 2008, 266 p.
9. Barkalov A. A., Titarenko L. A., Miroshkin A. N. Realizaciya kompozicionnykh mikroprogrammykh ustrojstv upravleniya na FPGA-mikrosxemax, *Radioelektronika i informatika*, 2011, No.1, pp. 52–55.