

УДК 004.274

А.А. Баркалов¹, Л.А. Титаренко¹, К.Н. Ефименко²
Университет Зеленогурский, Польша¹,
Донецкий национальный технический университет, Украина²
A.Barkalov@iie.uz.zgora.pl¹; KN_Efimenko@mail.ru²

Оптимизация схемы КМУУ с разделением кодов

Предлагается метод уменьшения аппаратных затрат в схеме КМУУ с разделением кодов, ориентированный на технологию FPGA. Метод основан на использовании двух 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.

КМУУ, ГСА, ОЛЦ, FPGA, логическая схема

1. 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 двух источников кодов классов псевдоэквивалентных операторных линейных цепей (ОЛЦ). Предлагаемый метод является развитием результатов, полученных в работе [9].

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.

2. Базовая информация о КМУУ 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, 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 $\langle b_i, b_q \rangle \in E$, где $b_i \notin D^g$.

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 $b_q \in V$.

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

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

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

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

Для определения любой вершины $b_q \in D^s$ 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).

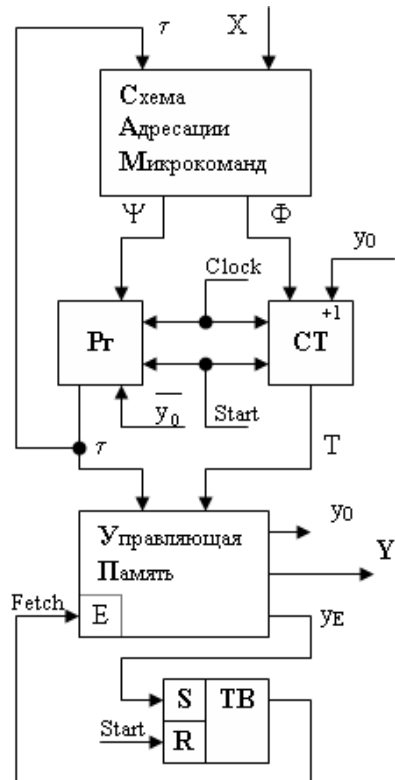


Рисунок 1 – Структурная схема КМУУ с разделением кодов

В этой модели для кодирования ОЛЦ используются переменные $\tau_i \in \tau$, где $|\tau| = R_1$.

Для кодирования компонент ОЛЦ используются переменные $T_i \in T$, 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 Рг

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

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

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 ОЛЦ, 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 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.

3. Основная идея предлагаемого метода

Пусть ОЛЦ $\alpha_g \in C_1$, если O_g не связан с конечной вершиной ГСА Γ . Найдем разбиение $\Pi_C = \{B_1, \dots, B_I\}$ множества C_1 на классы псевдоэквивалентных ОЛЦ (ПОЛЦ). Выполним кодирование $\alpha_g \in C$ так, чтобы максимально возможное число классов $B_i \in \Pi_C$, где $|\Pi_C| = I$, представлялось одним обобщенным интервалом R_1 -мерного булева пространства. Пусть n_i – число обобщенных 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:

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

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

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

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

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

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

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

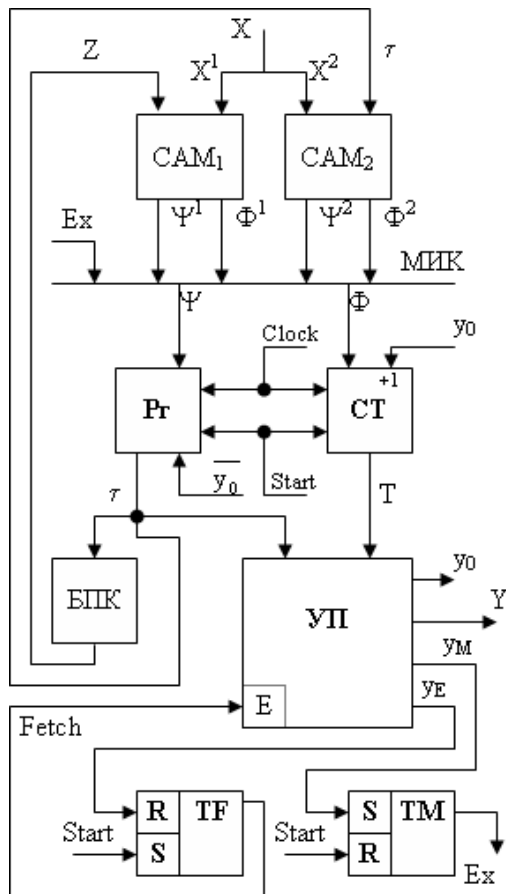


Рисунок 2 – Структурная схема КМУУ U_2

Эта модель имеет ряд отличий от модели U_1 . Во-первых, блок SAM разделен на два блока. Блок SAM_1 реализует переходы, определяемые

множеством Π_B , а блок SAM_2 – множеством Π_A . Мультиплексор МИК служит для выбора источника кодов, используя переменную Ex . Значение переменной Ex определяется состоянием триггера TM , управляемого дополнительной переменной y_M . Блок БПК является источником кодов классов для схемы SAM_1 . Источниками кодов для схемы SAM_2 является регистр R_r . Предлагаемое КМУУ функционирует следующим образом.

По сигналу $Start$ в R_r и CT заносятся нулевые коды (адрес первой микрокоманды), а триггера TF и TM 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 ОЛЦ $\alpha_g \in V_i$ может формироваться переменная $y_M = 1$. В этом случае $Ex = 1$ и адрес перехода формируется схемой SAM_1 . Для этого формируется система функций:

$$\begin{aligned} \Phi^1 &= \Phi^1(Z, X^1); \\ \Psi^1 &= \Psi^1(Z, X^1). \end{aligned} \quad (10)$$

Если $V_i \in \Pi_A$, то переменная y_M не формируется. При этом $Ex = 0$ и адрес перехода определяется схемой SAM_2 . Для этого формируется система функций:

$$\begin{aligned} \Phi^2 &= \Phi^2(\tau, X^2); \\ \Psi^2 &= \Psi^2(\tau, X^2). \end{aligned} \quad (11)$$

Соответствующие функции передаются на выход МИК и требуемые коды $K(\alpha_g)$ и $K(b_q)$ загружаются соответственно в R_r и CT . 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

$$R_3 < R_1 \quad (12)$$

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. Блок УП реализуется на реконфигурируемых блоках памяти EMB (embedded memory block) [5,6]. Эти блоки имеют строго определенное число выходов, входящих в множество $\{1, 2, 4, 8, 18, 36\}$. При этом имеется высокая вероятность наличие неиспользованных выходов блоков EMB , входящих в состав УП устройства U_1 .

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.

4. Особенности реализации схемы КМУУ U₂

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

1. Формирование для ГСА Г множеств C, C₁, и П_С.
2. Оптимальное кодирование ОЛЦ $\alpha_g \in C_1$ и кодирование компонент ОЛЦ.
3. Формирование множеств П_А и П_В.
4. Кодирование классов $V_i \in P_B$ кодами C(V_i).
5. Формирование таблицы переходов для классов $V_i \in P_A$.
6. Формирование таблицы переходов для классов $V_i \in P_B$.
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 [6,7]. Эти этапы 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 $C = \{\alpha_1, \dots, \alpha_{16}\}$ и разбиение $P_C = \{V_1, \dots, V_7\}$. При этом $\alpha_{16} \notin C_1$, а классы $V_i \in P_C$ определяются следующим образом: $V_1 = \{\alpha_1\}$, $V_2 = \{\alpha_2, \alpha_3, \alpha_4\}$, $V_3 = \{\alpha_5, \alpha_6\}$, $V_4 = \{\alpha_7, \alpha_8, \alpha_9\}$, $V_5 = \{\alpha_{10}, \alpha_{11}\}$, $V_6 = \{\alpha_{12}\}$, $V_7 = \{\alpha_{13}, \alpha_{14}, \alpha_{15}\}$. Итак, $G = 16$, $R_1 = 4$, $\tau = \{\tau_1, \dots, \tau_4\}$.

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

Из карты Карно (рис. 3) можно получить следующие интервалы, определяющие классы $V_i \in P_C$.

		$\tau_3 \tau_4$			
		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 $\alpha_g \in C$

Класс V₁ определяется интервалом 0000; класс V₂ – интервалами 00*1 и 001*; класс V₃ – интервалом 010*; класс V₄ – интервалами 110* и 11*1; класс V₅ – интервалом 011*; класс V₆ – интервалом 1010; класс V₇ – интервалами 100* и 10*1. Таким образом, $P_A = \{V_1, V_3, V_5, V_6\}$, где $K(V_1) = 0000$, $K(V_3) = 010^*$, $K(V_5) = 011^*$ и $K(V_6) = 1010$. Очевидно, $P_B = \{V_2, V_4, V_7\}$ и для их кодирования необходимо R₃ = 2 разряда. Итак, $Z = \{z_1, z_2\}$. Закодируем классы $V_i \in P_B$ следующим образом: $C(V_2) = 00$, $C(V_4) = 01$, $C(V_7) = 10$.

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

$$V_1 \rightarrow x_1 b_3 \vee \overline{x_1} x_2 b_8 \vee \overline{x_1} x_2 b_{12};$$

$$V_2 \rightarrow x_4 b_{12} \vee \overline{x_4} b_{17}.$$

(13)

Столбцы таблицы переходов включают следующую информацию: исходный класс (столбец V_i); код класса (для П_А это код K(V_i), а для П_В – C(V_i)); адрес перехода (A(b_q)); логические условия определяющие переход (столбец X_h); функции возбуждения триггеров регистра Рг (столбец Ψ_h^1 для П_В и Ψ_h^2 для П_А); функции возбуждения триггеров счетчика СТ (столбец Φ_h^2 для П_А и Φ_h^1 для П_В); 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.

Система (10) может быть получена из таблицы переходов для классов $V_i \in P_B$. Так, из табл. 2 имеем, например $D_1 = z_1 z_2 x_4$; $D_6 = \overline{z_1} z_2 x_4$. Система (11) может быть получена из таблицы переходов для классов $V_i \in P_A$. Так,

из табл. 1, имеем, например, $D_2 = \overline{\tau_1 \tau_2 \tau_3 \tau_4 x_1 x_2}$;
 $D_6 = \overline{\tau_1 \tau_2 \tau_3 \tau_4 x_1 x_2}$.

$z_1 = \overline{\tau_1 \tau_2 \tau_3} \vee \tau_1 \tau_2 \tau_3$; $z_2 = \overline{\tau_1 \tau_2 \tau_3} \vee \tau_1 \tau_2 \tau_3$. Остальные этапы синтеза реализуются тривиальным образом, и в нашей статье не рассматриваются.

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

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	z_2	3
	11*1		z_2	4
B_7	100*	10	z_1	5
	10*1		z_1	6

Из табл. 3 имеем систему (9):

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 / S. Maxfield. – 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. www.xilinx.com.
7. www.altera.com.
8. Baranov S. Logic and System Design of Digital Systems / S. Baranov. - Tallinn: TTU, 2008. - 266 p.
9. 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. – Вип. 9(179). – С.26-35.

Надійшла до редколегії 02.10.2011

**О.О. БАРКАЛОВ, Л.О. ТИТАРЕНКО,
К.М. ЄФІМЕНКО**

Зеленогурський університет, м. Зелена гора,
Польща,

Донецький національний технічний університет

**A.A. BARKALOV, L.A. TITARENKO,
K.N. EFIMENKO**

University of Zielona Gora, Poland,
Donetsk National Technical University

Оптимізація схеми 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 двох джерел кодів класів псевдоеквівалентних ОЛЛ 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.

КМПК, ГСА, ОЛЛ, FPGA, логічна схема

Optimization of Circuit for CMCU with Code Sharing

A method for reducing the hardware amount in the circuit of CMCU with code sharing is proposed, it is oriented on FPGA technology. The method is based on the use of two 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. An example of the proposed method application is given.

CMCU, GSA, OLC, FPGA, logic circuit