

УДК 004.3

А.А. БАРКАЛОВ¹, А.А. КРАСИЧКОВ², А.Н. МИРОШКИН²¹ Зеленогурский университет, Зеленая Гора, Польша.² Донецкий национальный технический университет, Украина.

РАСШИРЕНИЕ ФОРМАТА МИКРОКОМАНД В КОМПОЗИЦИОННОМ МИКРОПРОГРАММНОМ УСТРОЙСТВЕ УПРАВЛЕНИЯ С ЭЛЕМЕНТАРИЗАЦИЕЙ ОПЕРАТОРНЫХ ЛИНЕЙНЫХ ЦЕПЕЙ

Предлагается метод синтеза композиционных микропрограммных устройств управления с разделением кодов и элементаризацией граф-схемы алгоритма управления. Метод направлен на уменьшение аппаратных затрат (LUT-элементов) при реализации устройства управления в базисе FPGA. Уменьшение сложности схемы адресации микрокоманд достигается за счет введения в формат микрокоманды поля с кодом класса псевдоэквивалентной операторной линейной цепи. Приводятся условия, определяющие возможность применения предложенного метода. Приведен пример применения метода.

Ключевые слова: устройство управления, элементаризация, операторная линейная цепь, разделение кодов, FPGA, LUT-элемент.

Введение

При реализации алгоритмов управления сложность реализуемых систем обуславливает актуальность задачи уменьшения аппаратных затрат в схеме устройства [1]. При решении этой задачи необходимо учитывать особенности структурной схемы устройства управления и особенности элементного базиса. Среди особенностей, которые можно использовать, следует отметить наличие псевдоэквивалентных состояний, а также линейный характер ГСА управления [2].

Для реализации линейных алгоритмов управления, доля операторных вершин в которых больше 75%, целесообразно использовать УУ класса композиционных микропрограммных устройств управления (КМУУ) [2]. В качестве элементного базиса для реализации схем устройств управления настоящее время широко используется базис FPGA (Field-Programmable Gate Arrays) [3, 4]. Проблема минимизации аппаратных затрат решается путем уменьшения сложности основных структурных элементов устройства посредством уменьшения разрядности основных связей между ними. В настоящей работе предлагается один из путей решения этой проблемы: реализация управляющего устройства в виде КМУУ с разделением кодов и элементаризацией ОЛЦ.

Целью исследования является оптимизация комбинационной схемы КМУУ за счет введения в формат микрокоманд кодов классов псевдоэквивалентных операторных линейных цепей. Задачей исследования является разработка метода синтеза КМУУ, позволяющего уменьшить число LUT-эле-

ментов в схеме адресации микрокоманд. При этом алгоритм управления представляется в виде граф-схемы алгоритма (ГСА) [2].

1. Основные положения

Граф-схема алгоритма управления состоит из операторных и условных вершин, образующих множества E_1 и E_2 соответственно, а также множества дуг E . Начальную вершину ГСА обозначим b_0 , конечную – b_E . Операторная вершина b_q содержит набор микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ – множество микроопераций (выходных сигналов). Условная вершина b_g содержит один элемент $X(b_g)$ множества логических условий $X = \{x_1, \dots, x_L\}$. Если количество операторных вершин составляет 75% и более от общего количества вершин, речь идет о линейной ГСА.

Операторная линейная цепь (ОЛЦ) является последовательностью операторных вершин граф-схемы алгоритма. Каждая ОЛЦ α_g имеет произвольное число входов I_g^i и только один выход Q_g . Формальные определения ОЛЦ, их входов и выходов можно найти в [5]. ОЛЦ, которая имеет один вход и один выход, называется элементарной [2].

Операторные линейные цепи, выходы которых связаны с входом одной вершины, называются псевдоэквивалентными (ПОЛЦ). Множество таких ОЛЦ образуют класс B_i .

Пусть ГСА содержит G элементарных ОЛЦ α_g , которые составляют множество C . Для кодирования элементов этого множества достаточно

$$R_1 = \lceil \log_2 G \rceil \quad (1)$$

бит. Количество компонент в ОЛЦ α_g обозначим F_g . Максимальная длина $Q = \max(F_1, \dots, F_G)$ линейной цепи определяет разрядность кода компоненты ОЛЦ

$$R_2 = \lceil \log_2 Q \rceil. \quad (2)$$

Используем для кодирования элементарных ОЛЦ элементы $\tau_r \in \tau$, а для кодирования компонент – элементы $T_r \in T$, где $|\tau| = R_1$, $|T| = R_2$. Кодирование компонент выполняется в естественном порядке, то есть

$$K(b_{gi}) = K(b_{gi-1}) + 1, \quad (3)$$

где $g = 1, \dots, G$, $i = 1, \dots, F_g$.

Заметим, что каждая вершина $b_q \in E_1$ соответствует микрокоманде M_q , хранимой в управляющей памяти (УП) по адресу $A(b_q) = A_q$. Если адрес A_q получается вследствие конкатенации кода ОЛЦ и кода компоненты, то используется метод разделения кодов.

Для интерпретации ГСА может быть использована структура композиционного микропрограммного устройства управления с элементарными ОЛЦ и разделением кодов (рис. 1), обозначаемая в дальнейшем U_1 .

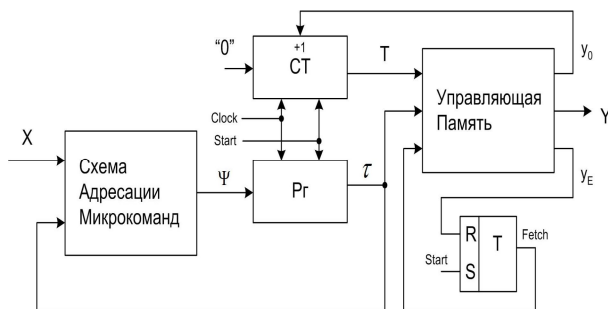


Рис. 1. Структурная схема КМУУ с элементарными ОЛЦ и разделением кодов

В КМУУ U_1 схема адресации микрокоманд (САМ) реализует функцию возбуждения регистра Pr $\Psi = \Psi(X, \tau)$. (4)

По сигналу $Start$ в Pr заносится начальный адрес микропрограммы, счетчик переводится в нулевое состояние, а триггер выборки TB устанавливается в состояние «1». При этом сигнал $Fetch = 1$, что разрешает выборку микрокоманд из 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 TB обнуляется и выборка микрокоманд прекращается.

На вход R счетчика СТ следует подавать сигнал $\overline{y_0} \vee Start$, что обеспечит перевод счетчика в нулевое состояние каждый раз при включении питания или при окончании очередной ОЛЦ.

Число термов в схеме САМ может быть уменьшено путем введения преобразователя кодов ОЛЦ в коды классов псевдоэквивалентных ОЛЦ (ПОЛЦ) [2]. Однако реализация такого преобразователя требует дополнительных ячеек ПЛИС.

В настоящей работе предлагается уменьшить сложность преобразователя кодов, для чего используются свободные ресурсы микросхем ППЗУ.

2. Основная идея предлагаемого метода

В исходной ГСА множество C_1 , состоящее из ОЛЦ α_g , не связанных с конечной вершиной, разбивается на классы $B_i \in \Pi_C$ псевдоэквивалентных ОЛЦ. Каждому классу B_i в соответствие ставится двоичный код $K(B_i)$ разрядности

$$R_3 = \lceil \log_2 I \rceil, \quad (5)$$

где I – количество классов ПОЛЦ. Пусть управляющая память КМУУ реализуется на ППЗУ, имеющих t выходов. При унитарном кодировании микроопераций [2] в ППЗУ хранятся микрокоманды разрядности

$$n_1 = N + 2, \quad (6)$$

где $N = |Y|$, а константа 2 учитывает переменные y_0 и y_E . Таким образом, R_4 бит слова управляющей памяти могут быть свободными, где

$$R_4 = \left\lceil \frac{n_1}{t} \right\rceil * t - n_1. \quad (7)$$

Если выполняется условие

$$R_4 \geq R_3, \quad (8)$$

то в формат микрокоманд можно включить поле FB с кодом $K(B_i)$, что ведет к КМУУ U_2 (Рис.2).

В КМУУ U_2 переменные $z_r \in Z$, где $|Z| = R_3$, представляют собой разряды кода $K(B_i)$. Схема САМ реализует функцию

$$\Psi = \Psi(Z, X). \quad (9)$$

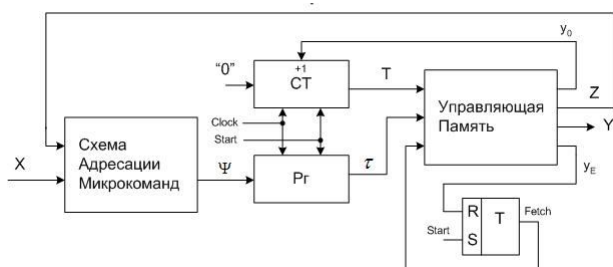


Рис. 2. Структурная схема КМУУ с элементарными ОЛЦ и разделением кодов после расширения формата микрокоманд

Остальные блоки КМУУ U_2 выполняют те же функции, что и одноименные блоки КМУУ U_1 . Отметим, что блоки САМ, СТ, РГ, ТВ реализуются в составе микросхемы ПЛИС. Для реализации блока УП используются внешние ППЗУ или блоки встроенной памяти (embedded ROM).

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

1. Формирование множеств C , C_1 и Π_C для ГСА Γ .
2. Кодирование ОЛЦ, их компонент и классов $V_i \in \Pi_C$.
3. Формирование содержимого управляющей памяти.
4. Формирование таблицы переходов КМУУ и функции $\Psi = \Psi(Z, X)$.
5. Синтез логической схемы КМУУ.

3. Пример применения метода

Рассмотрим применение метода на примере ГСА Γ_1 (рис. 3). Множество $C = \{\alpha_1, \dots, \alpha_5\}$ – элементарные ОЛЦ алгоритма, $C_1 = C \setminus \alpha_5$, $\Pi_C = \{V_1, \dots, V_3\}$, где $V_1 = \{\alpha_1\}$, $V_2 = \{\alpha_2, \alpha_3\}$, $V_3 = \{\alpha_4\}$. Следовательно, количество ОЛЦ $G = 5$, для кодирования которых используются $R_1 = 3$ переменных их множества $\tau = \{\tau_1, \tau_2, \tau_3\}$. Максимальная длина ОЛЦ $Q = 3$, для кодирования которых достаточно $R_2 = 2$ переменных из множества $T = \{T_1, T_2\}$. Общее количество операторных вершин $M = 9$, что требует разрядности $R = 4$ адреса слова в УП. Для кодирования $I = 3$ классов $V_i \in \Pi_C$ достаточно $R_3 = 2$ переменных.

Кодирование ОЛЦ $\alpha_g \in C$ и их компонент выполним с учетом ограничения (3). Адреса $A(b_q)$ микрокоманд КМУУ $U_2(\Gamma_1)$ приведены в табл. 1. Здесь и далее символ $U_i(\Gamma_j)$ означает, что КМУУ U_i интерпретирует ГСА Γ_j .

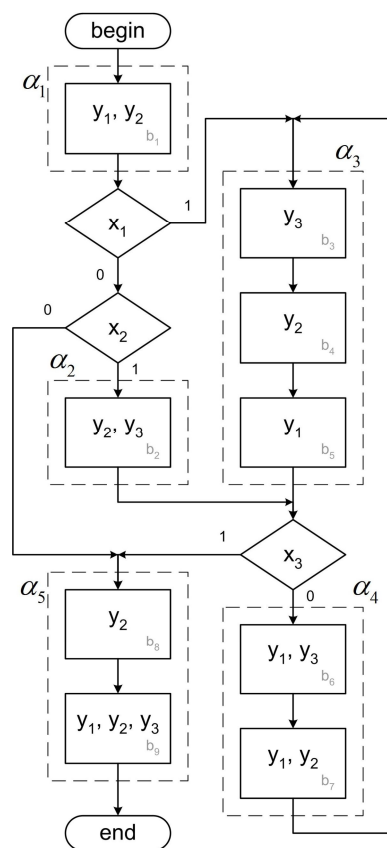


Рис. 3. Исходная ГСА Γ_1

Таблица 1
Адреса микрокоманд КМУУ $U_2(\Gamma_1)$

$T_1 T_2 \backslash \tau_1, \tau_2, \tau_3$	000	001	010	011	100
00	b_1	b_2	b_3	b_6	b_8
01	–	–	b_4	b_7	b_9
10	–	–	b_5	–	–
11	–	–	–	–	–

Из табл. 1 имеем, например, $A(b_6) = 01100$, $A(b_9) = 10001$ и так далее.

Коды классов $V_i \in \Pi_C$ зададим следующим образом: $K(V_1) = 00$, ..., $K(V_3) = 10$. Формат микрокоманд КМУУ U_2 включает поля y_0 , y_E , FY , FV , где поле FY содержит код набора микроопераций, а поле FV – код класса $V_i \in \Pi_C$. Если $y_0 = 1$, то содержимое поля FV игнорируется.

Пусть ГСА Γ_1 включает $N = 3$ микроопераций, а для реализации УП используются ППЗУ с

$t = 4$. В этом случае формула (7) дает $R_4 = 3$. Поскольку $R_4 > R_3$, применение данного метода возможно. Итак, в рассматриваемом примере $Z = \{z_1, z_2\}$. Содержимое УП КМУУ $U_2(\Gamma_1)$ показано в табл. 2.

Таблица 2

Содержимое управляющей памяти КМУУ $U_2(\Gamma_1)$

$A(b_q)$	y_0	FY	y_E	FB
$A(b_1)$	0	y_1, y_2	0	00
$A(b_2)$	0	y_2, y_3	0	01
$A(b_3)$	1	y_3	0	–
$A(b_4)$	1	y_2	0	–
$A(b_5)$	0	y_1	0	01
$A(b_6)$	1	y_1, y_3	0	–
$A(b_7)$	0	y_1, y_2	0	10
$A(b_8)$	1	y_2	0	–
$A(b_9)$	0	y_1, y_2, y_3	1	–

Как можно увидеть из табл. 2, Если вершина $b_q \in E_1$ не является выходом ОЛЦ $\alpha_g \in C_1$, то в ячейку с адресом $A(b_q)$ записывается микрооперация y_0 . В противном случае в эту ячейку записывается код $K(B_i)$, где $\alpha_g \in B_i$. Если вершина $b_q \in E_1$ связана с конечной вершиной ГСА, то в ячейку с адресом $A(b_q)$ заносится микрооперация y_E . Переходы из выходов ОЛЦ $\alpha_g \in C_1$ представлены следующей системой обобщенных формул переходов [2]:

$$\begin{aligned} B_1 &\rightarrow x_1 b_3 \vee \overline{x_1} x_2 b_2 \vee \overline{x_1} x_2 b_8; \\ B_2 &\rightarrow x_3 b_8 \vee \overline{x_3} b_6; \\ B_3 &\rightarrow b_3. \end{aligned} \tag{10}$$

Подобная система является основой для формирования таблицы переходов КМУУ U_2 со столбцами: $B_i, K(B_i), b_q, A(b_q), X_h, \Psi_h, h$. Назначение столбцов ясно из табл. 3, задающей переходы для класса $B_i \in \Pi_C$.

Таблица 3

Фрагмент таблицы переходов КМУУ $U_2(\Gamma_1)$

B_i	$K(B_i)$		b_q	$A(b_q)$			X_h	Ψ_h	h
	z_2	z_1		τ_1	τ_2	τ_3			
B_1	0	0	b_2	0	0	1	$\overline{x_1} x_2$	D_3	1
			b_3	0	1	0	x_1	D_2	2
			b_8	1	0	0	$\overline{x_1} x_2$	D_1	3

Адреса микрокоманд берутся из табл. 1. Отметим, что $\Psi = \{D_1, D_2, D_3\}$. Общее число строк $H_2(\Gamma_j)$ в таблице переходов КМУУ $U_2(\Gamma_j)$ совпадает с числом термов в системе обобщенных формул переходов. В нашем примере, $H_2(\Gamma_1) = 6$.

Система (9) формируется по таблице переходов. Так, из табл. 3 можно построить фрагменты ДНФ:

$$\begin{aligned} D_1 &= \overline{z_1} z_2 x_1 x_2; \\ D_2 &= z_1 z_2 x_1; \\ D_3 &= z_1 z_2 x_1 x_2. \end{aligned} \tag{11}$$

Для минимизации числа термов в системе (9) классы $B_i \in \Pi_C$ могут быть закодированы с использованием, например, алгоритма ESPRESSO [2].

Реализация логической схемы КМУУ U_2 сводится к реализации системы (9) на ПМЛ и к реализации УП на ППЗУ или встроенных блоках памяти. Для этой цели могут быть использованы известные методы [1, 2] или стандартные пакеты САПР.

Заключение

Предлагаемый метод расширения формата микрокоманд за счет введения поля с кодом класса ПОЛЦ для ГСА с элементарными операторными линейными цепями ориентирован на уменьшение числа LUT-элементов в схеме адресации микрокоманд. При этом число блоков встроенной памяти в управляющей памяти и число тактов выполнения алгоритма управления совпадают с соответствующими числами для базовой структуры U_1 КМУУ с разделением кодов.

Недостатком предлагаемого метода является ограничение (8) на возможность его применения.

Уменьшение числа термов в ДНФ функций возбуждения памяти может привести к уменьшению числа уровней в комбинационной части КМУУ. Это в свою очередь приводит к повышению быстродействия цифровой системы в целом.

Научная новизна предложенного метода состоит в использовании классов ПОЛЦ и свободных ресурсов блоков встроенной памяти для уменьшения числа LUT-элементов в схеме адресации микрокоманд устройства управления для ГСА с элементарными ОЛЦ. Практическая значимость метода заключается в уменьшении перематров необходимых для реализации устройства микросхем, что позволит реализовать схемы, обладающие меньшей стоимостью, чем известные аналоги.

Дальнейшие направления наших исследований связаны с разработкой САПР для синтеза схем КМУУ [5].

Литература

1. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем / В.В. Соловьев. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
2. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах / А.А. Баркалов. – Донецк: ДНТУ, 2002. – 262 с.
3. Грушвицкий Р.И. Проектирование систем с использованием микросхем программируемой логики / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов. –

СПб: БХВ. – Петербург, 2002. – 608 с.

4. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем / В.В. Соловьев. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.

5. Баркалов А.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС / А.А. Баркалов, Л.А. Титаренко. – Донецк: УНИТЕХ, 2009. – 336 с.

6. De Micheli G. Synthesis and Optimization of Digital Circuits / G De Micheli. – NY: McGraw-Hill, 1994. – 636 p.

Поступила в редакцию 18.03.2010

Рецензент: д-р техн. наук, проф., зав. каф. Автоматизированных систем управления, проф. Ю.А. Скобцов, Донецкий национальный технический университет, Донецк.

РОЗШИРЕННЯ ФОРМАТУ МІКРОКОМАНД У КОМПОЗИЦІЙНОМУ МІКРОПРОГРАМНОМУ ПРИСТРОЇ КЕРУВАННЯ З ЕЛЕМЕНТАРИЗАЦІЄЮ ОПЕРАТОРНИХ ЛІНІЙНИХ ЛАНЦЮГІВ

О.О. Баркалов, О.О. Красічков, О.М. Мірошкін

Пропонується метод синтезу композиційних мікропрограмних пристроїв керування з розділенням кодів та елементаризацією граф-схеми алгоритму керування. Метод спрямований на зменшення апаратних витрат (LUT-елементів) при реалізації пристрою керування в базисі FPGA. Зменшення складності схеми адресації мікрокоманд досягається за рахунок введення до формату мікрокоманд поля з кодом класу псевдоеквівалентного операторного лінійного ланцюгу. Приводяться умови, що визначають можливість застосування запропонованого методу. Приведений приклад застосування методу.

Ключові слова: пристрій керування, елементаризація, операторний лінійний ланцюг, розділення кодів, FPGA, LUT-елемент.

EXTENSION OF MICROINSTRUCTION FORMAT IN COMPOSITIONAL MICROPROGRAM CONTROL UNIT WITH CODE SHARING AND LINEAR CHAINS ELEMENTARIZATION

A.A. Barkalov, A.A. Krasichkov, A.N. Miroshkin

The new design method for compositional microprogram control units with code sharing and linear chains elementarization is proposed. The method targets on reduction in the number of FPGA LUT-elements in the combinational part of control unit. The additional field containing code of the class of pseudoequivalent operational linear chains are used for extension of microinstruction format. Conditions of proposed method using are given. Proposed method is illustrated by an example.

Key words: control unit, elementarization, operational linear chain, code sharing, FPGA, LUT-element.

Баркалов Александр Александрович – д-р техн. наук, проф. кафедри комп'ютерної інженерії, Донецький національний технічний університет, Донецьк, Україна; проф. Зеленогурського університета, Зелена Гора, Польща, e-mail: A.Barkalov@iie.uz.zgora.pl.

Красічков Алексей Александрович – канд. техн. наук, доцент кафедри комп'ютерної інженерії, Донецький національний технічний університет, Донецьк, Україна, e-mail: krasich@cs.dgtu.donetsk.ua.

Мірошкін Александр Николаевич – асистент кафедри комп'ютерної інженерії, Донецький національний технічний університет, Донецьк, Україна, e-mail: miroshkin@cs.dgtu.donetsk.ua.