

УДК 681.3

А.А. Баркалов<sup>1</sup>, д. н., проф.,  
В.Н. Струнилин<sup>2</sup>,  
Е.В. Струнилин<sup>2</sup>,<sup>1</sup> Зеленогурский университет, г. Зеленая гора, Польша  
A.Barkalov@iie.uz.zgora.pl<sup>2</sup> Донецкий национальный технический университет, г. Донецк, Украина  
vstrun@cs.dgtu.donetsk.ua, geKa@donntu.edu.ua

## Уменьшение аппаратурных затрат в устройстве управления с общей памятью

Предлагается метод уменьшения емкости управляющей памяти в схеме композиционного микропрограммного устройства управления с общей памятью. Метод основан на преобразовании адресов микрокоманд в коды наборов микроопераций и позволяет уменьшить аппаратурные затраты в схеме блока адресации микрокоманд.

**Ключевые слова:** композиционное микропрограммное устройство управления, операторная линейная цепь, логическая схема устройства управления, общая память, линейный алгоритм управления.

### Введение

Устройства управления (УУ) является важной частью практически каждой цифровой системы [1]. Одной из основных задач, возникающих при синтезе УУ, является уменьшение аппаратурных затрат в его схеме [2,3]. Методы решения этой задачи зависят как от характеристик реализуемого алгоритма управления, так и от используемого элементного базиса [4]. В настоящей работе рассматривается задача синтеза УУ по линейной граф - схеме алгоритма (ГСА) в базисе СБИС типа FPGA (field-programmable logic array) [5,6]. В этом случае целесообразно использовать модель композиционного микропрограммного устройства управления (КМУУ) [7]. Предлагаемый метод ориентирован на уменьшение емкости управляющей панели (УП) КМУУ с общей памятью. Для этого предлагается адаптация методов [8, 9] к условиям этого УУ.

Целью исследований является уменьшение емкости управляющей памяти КМУУ за счет введения преобразователя адресов микрокоманд в коды наборов микроопераций.

Задачей исследований является разработка метода синтеза КМУУ с общей памятью, позволяющего уменьшить число блоков встроенной памяти в схеме КМУУ по сравнению с известными методами. При этом алгоритм управления представляется в виде ГСА [1].

### Основные определения и общие положения

Пусть алгоритм управления цифровой системы представлен в виде ГСА Г, которая

характеризуется множеством вершин  $B = E_1 \cup E_2 \cup \{b_0, b_1\}$  и дуг Е, соединяющих эти вершин. Здесь  $E_1$  - множество операторных вершин, содержащих наборы микроопераций из множества микроопераций (МО)  $Y = \{y_1, \dots, y_n\}$ ;  $E_2$  - множество условных вершин, содержащих элементы множества логических условий (ЛУ)  $X = \{x_1, \dots, x_l\}$ ;  $b_0$  - начальная вершина;  $b_E$  - конечная вершина ГСА Г. Введем ряд определений [7].

**Определение 1.** Операторной линейной цепью ГСА Г называется конечная последовательность операторных вершин  $\alpha_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$ , такая, что для любой пары ее соседних вершин существует дуга  $\langle b_{gi}, b_{gi+1} \rangle \in E$ , где  $i = 1, \dots, F_g - 1$ .

**Определение 2.** Операторная вершина  $b_g \in D^g$ , где  $D^g$  - множество вершин, входящих в ОЛЦ  $\alpha_g$ , называется входом ОЛЦ  $\alpha_g$ , если существует дуга  $\langle b_t, b_q \rangle \in E$ , где  $b_t \notin D^g$ .

**Определение 3.** Операторная вершина  $b_q \in D^g$  называется выходом ОЛЦ  $\alpha_g$ , если существует дуга  $\langle b_q, b_t \rangle \in E$ ,  $b_t \notin D^g$ .

**Определение 4.** Операторные линейные цепи называются псевдоэквивалентными ОЛЦ (ПОЛЦ), если их выходы связаны с выходом одной и той же вершины ГСА Г.

Любая ОЛЦ  $\alpha_g$  имеет произвольное число входов, обозначаемых  $l_g^k$  ( $k = 1, \dots, F_g$ ) и образующих множество  $l_g$ , и только один выход, обозначаемый символом  $O_g$ .

**Определение 5.** Граф-схема алгоритма Г является линейной ГСА, если число ее операторных вершин не менее, чем в два раза превосходит число ОЛЦ.

Пусть для ГСА Г получено множество ОЛЦ  $C = \{\alpha_1, \dots, \alpha_G\}$ , где каждая ОЛЦ  $\alpha_g \in C$  включает максимально возможное число компонент [7]. Пусть  $I(\Gamma)$ ,  $O(\Gamma)$  соответственно множество входов и выходов ОЛЦ ГСА Г. Каждая вершина  $b_q \in E_1$  соответствует микрокоманде  $M_{q_i}$ , имеющей адрес  $A(b_q)$ . Выполним адресацию микрокоманд так, чтобы выполнялось условие

$$A(b_{qi+1}) = A(b_{qi}) + 1, \quad (1)$$

где  $g \in \{1, \dots, G\}$ ,  $i \in \{1, \dots, F_g - 1\}$ .

В этом случае для интерпретации линейной ГСА Г можно использовать модель КМУУ с общей памятью (рис. 1), обозначаемую в дальнейшем символом  $U_1$ . Эта модель включает блок адресации микрокоманд (БАМ), счетчик (СТ), блок управляющей панели (УП) и триггер выборки (ТВ). Устройство  $U_1$  функционирует следующим образом.

По сигналу Start в СТ записывается нулевой адрес, соответствующий началу микропрограммы, интерпретирующей ГСА Г, одновременно триггер ТВ устанавливается в единичное состояние (Fetch=1) и микрокоманды могут выбираться из блока УП. Если СТ содержит адрес  $A(b_q)$  и  $b_q \notin O(\Gamma)$ , то

одновременно с набором микроопераций  $Y(b_q)$ , записанных в вершине  $b_q \in E_1$ , УП формирует сигнал  $y_0$ . Если  $y_0=1$ , то содержимое СТ увеличивается на единицу по сигналу Clock. При этом выполняется безусловный переход, соответствующий равенству (1). В том случае, если  $b_q \in O(\Gamma)$ , то сигнал  $y_0$  не формируется, а блок БАМ вырабатывает функции возбуждения СТ

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

В этом случае по сигналу Clock в СТ формируется адрес перехода из выхода некоторой ОЛЦ  $\alpha_g \in C$ . Если  $\langle b_q, b_E \rangle \in E$ , то блок УП формирует сигнал  $y_E$ , вызывающий установку триггера ТВ в нулевое состояние. При этом Fetch = 0, выборка микрокоманд прекращается и КМУУ  $U_1$  прекращает функционирование.

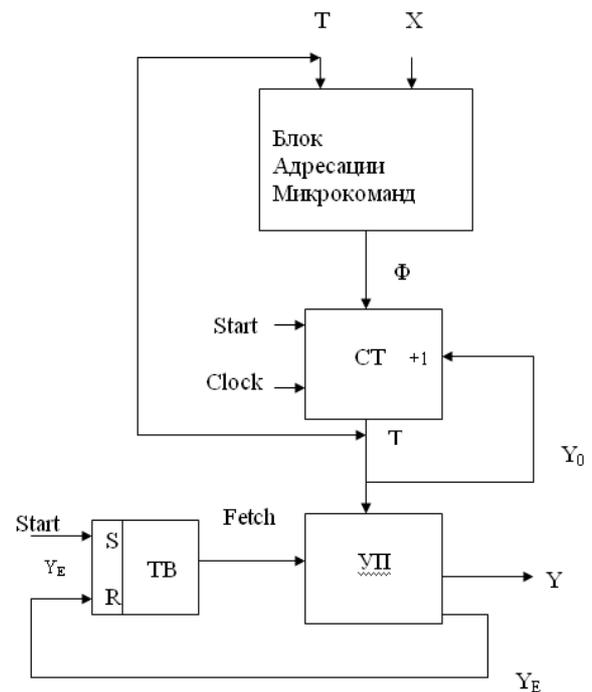


Рисунок 1 – Структурная схема КМУУ  $U_1$

При реализации схем КМУУ  $U_1$  на FPGA схемы БАМ, СТ и ТВ строятся на LUT элементах, а блоки ЕМВ используются для реализации управляющей памяти.

**Основные идеи предлагаемого метода**

В управляющей памяти КМУУ  $U_1$  хранится  $M = |E_1|$  микрокоманд, каждая из которых соответствует одной вершине  $b \in E_1$ .

Для адресации микрокоманд используется  $R_M$  адресных разрядов, где

$$R_M = \left\lceil \log_2 M \right\rceil. \tag{3}$$

Назовем множество  $Y(b_q) \subseteq Y$  микроопераций, записанных в вершине  $b \in E_1$ , набором микроопераций (НМО). При синтезе КМУУ в эти наборы могут добавляться дополнительные переменные  $y_0$  и  $y_E$ . Пусть в ГСА  $\Gamma$  имеется  $M_1$  попарно различных НМО. Тогда для их кодирования достаточно

$$R_I = \left\lceil \log_2 M_1 \right\rceil. \tag{4}$$

При выполнении условия

$$R_I < R_M \tag{5}$$

ёмкость УП может быть уменьшена за счет введения преобразователя адреса (ПА), выполняющего преобразование адреса  $A(b_q)$  в набор  $Y(b_q) \cup \{y_0, y_E\}$ . Это приводит к модели КМУУ  $U_2$  (рис. 2).

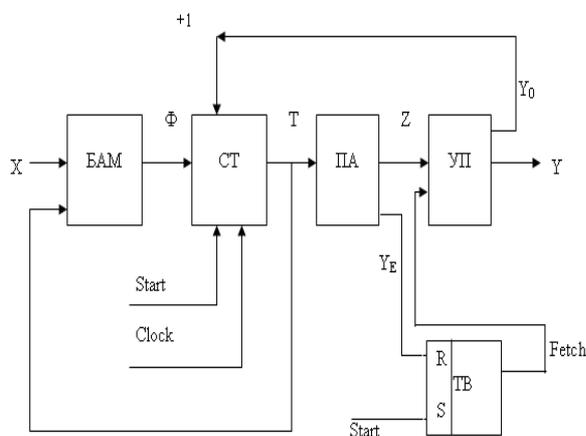


Рисунок 2 – Структурная схема КМУУ  $U_2$

Очевидно, принципы функционирования КМУУ  $U_1$  и  $U_2$  совпадают. Разница заключается в том, что в КМУУ  $U_2$  адреса микрокоманд преобразовываются в коды НМО. Для кодирования НМО используются переменные  $z_r \in Z$ , где  $|Z|=R_I$ . В настоящей работе предлагается метод синтеза КМУУ  $U_2$ .

**Метод синтеза КМУУ  $U_2$**

Предлагаемый метод синтеза включает следующие этапы:

1. Формирование множества ОЛЦ  $C = \{\alpha_1, \dots, \alpha_G\}$  по ГСА  $\Gamma$ .
2. Естественная адресация микрокоманд.
3. Построение множества расширенных наборов микроопераций  $V$ .
4. Если условие (5) выполняется, то синтез КМУУ  $U_2$  продолжается. В противном случае процесс завершается.
5. Кодирование НМО  $V_m \in V$  двоичными кодами  $K(V_m)$ .
6. Формирование содержимого управляющей памяти.
7. Формирование содержимого блока БПА.
8. Построение таблицы переходов КМУУ  $U_2$ .
9. Формирование системы функций (2).
10. Реализация схем блоков КМУУ в базе FPGA.

Рассмотрим более подробно некоторые из этих этапов.

Пусть для некоторой ГСА  $\Gamma_1$  с использованием методов [7] получено множество ОЛЦ  $C = \{d_1, \dots, d_{17}\}$ . Пусть для ГСА  $\Gamma_1$   $M=58$ .

Поскольку  $M:G = 3,41 > 2$ , то ГСА  $\Gamma_1$  является линейной и применение модели КМУУ имеет смысл. При этом  $R_M = 6$ ,  $\Phi = \{D_1, \dots, D_6\}$ ,  $T = \{T_1, \dots, T_6\}$ . Пусть множество  $V$  включает  $M_1 = 15$  наборов микроопераций, то есть  $R_I = 4$ . Очевидно,

условие (5) выполняется и применение модели  $U_2$  имеет смысл.

Закодируем НМО  $V_m \in V$  тривиальным образом:

$$K(V_1) = 0000, K(V_2) = 0001, \dots, K(V_{15}) = 1110.$$

Это позволяет сформировать содержимое УП. Пусть  $V_2 = \{y_1, y_3\}$ , тогда по адресу 0001 в УП записаны единицы в столбцах 1 и 3. Эта процедура (этап 7) выполняется для всех НМО.

На втором этапе в вершины ГСА  $\Gamma_1$  были введены дополнительные переменные  $y_0$  и  $y_E$ . После выполнения адресации, соответствующей (1), каждая вершина  $b_q \in E_1$  получает уникальный адрес  $A(b_q)$ .

Наибольший интерес представляет этап (7). Блок ПА задается таблицей, имеющей столбцы  $A(b_q), K(V_m), z_q, y_0, y_E, q$ . Каждая строка таблицы ПА соответствует одной микрокоманде исходной ГСА. В столбце  $K(V_m)$  расписывается код НМО для микрокоманды с адресом  $A(b_q)$ . В столбце  $Z_q$  расписывается переменные  $z_r \in Z$ , равные единице в коде  $K(V_m)$ . Этот столбец необходим, если блок ПА реализуется на LUT – элементах FPGA. Если  $b_q \in E_1$  не является выходом ОЛЦ, то в столбце  $y_0$  записывается 1. Если  $\langle b_q, b_E \rangle \in E$ , то в столбце  $y_E$  записывается 1.

Для формирования таблицы переходов КМУУ  $U_2$  необходимо построить систему формул перехода для вершин  $b_q \in O(\Gamma)$ . Этот этап также выполняется по известной методике [7]. Этап 9 выполняется тривиальным образом, а для решения задач этапа 10 используются промышленные САПР.

В заключение рассмотрим пример для этапа 7. Пусть ОЛЦ  $d_{17} \in C$  включает три компоненты:  $d_{17} = \langle b_{56}, b_{57}, b_{58} \rangle$ . Пусть  $Y(b_{56}) = V_{12}, Y(b_{57}) = V_2, Y(b_{58}) = V_{10}$  и пусть существует дуга  $\langle b_{58}, b_E \rangle \in E$ . Мы приняли,

что  $K(V_{12}) = 1011, K(V_2) = 1001, K(V_{10}) = 1001$ .

Очевидно, в вершины  $b_{56}$  и  $b_{57}$  добавляется переменная  $y_0$ , а в вершину  $b_E$  –  $y_E$ . Все это порождает фрагмент таблицы блока БА (таб. 1).

Таблица 1. Фрагмент таблицы ПА для КМУУ  $U_2$

$A(b_q)$	$K(V_m)$	$Z_q$	$y_0$	$y_E$	$q$
110111	1011	$Z_1 Z_3 Z_4$	1	0	56
111000	0001	$Z_4$	1	0	57
111001	1001	$Z_1 Z_4$	0	1	58

При построении табл. 1 было принято, что адрес  $A(b_q)$  равен двоичному эквиваленту числа  $(q - 1)_{10}$ . Естественно, что конкретные значения адресов микрокоманд зависят от результата этапа 2.

**Заключение**

Предлагаемый метод основан на преобразовании адресов микрокоманд в коды наборов микроопераций. При выполнении условия (5) метод позволяет уменьшить число встроенных блоков памяти в схеме КМУУ с общей памятью.

В КМУУ  $U_1$  требуется управляющая память, имеющая ёмкость

$$V(U_1) = 2^R \cdot M \cdot (N + 2). \tag{6}$$

В КМУУ  $U_2$  этот параметр уменьшается

до 
$$V(U_2) = 2^R \cdot I \cdot (N + 2). \tag{7}$$

Однако в КМУУ  $U_2$  вводится блок ПА, имеющий ёмкость

$$W(U_2) = 2^R \cdot M \cdot (R_I + 2). \tag{8}$$

Применение предлагаемого метода имеет смысл при выполнении условия

$$(V(U_1) / (V(U_2) + W(U_2))) > 1.$$

**Список использованной литературы**

1. Baranov S. Logic and System Design of Digital Systems / S. Baranov. – Tallin: TUT Press, 2008. – 266 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. Соловьев В.В. Логическое проектирование цифровых систем на ПЛИС / В.В. Соловьев, А. Климович. – М.: Горячая линия-Телеком, 2008. – 376 с.
4. DeMicheli G. Synthesis and Optimization of Digital Circuits / G. DeMicheli // McGraw-Hill. – 1994. – №4. – 636 p.
5. Altera Devices [Электронный ресурс]. – Режим доступа: [www.altera.com](http://www.altera.com)
6. Products & Services [Электронный ресурс]. – Режим доступа: [www.xilinx.com](http://www.xilinx.com)
7. Barkalov A.A. Logic Synthesis for Compositional Microprogram Control Units / A. Barkalov, L. Titarenko. – Berlin: Springer, 2008. – 272 p.
8. Оптимизация схемы композиционного микропрограммного устройства управления с разделением кодов / А.А. Баркалов, Р.В. Мальчева, А.А. Красичков и др. // Радиоэлектроника и информатика. – 2006. – №1. – С. 46–50.
9. Применение преобразования адресов в КМУУ с разделением кодов / А.А. Баркалов, Р.В. Мальчева, А.А. Красичков и др. // Труды седьмой МНПК «Современные информационные и электронные технологии», 22-26 мая 2006 г. – Одесса, 2006. – С. 182.

Надійшла до редколегії 26.03.2012

**О.О. БАРКАЛОВ<sup>1</sup>, В.М. СТРУНІЛІН<sup>2</sup>,  
Є.В. СТРУНІЛІН<sup>2</sup>**

<sup>1</sup> Зеленогурський університет, м. Зелена гора, Польща

<sup>2</sup> Донецький національний технічний університет

**A.A. BARKALOV<sup>1</sup>, V.N. STRUNILIN<sup>2</sup>,  
Y.V. STRUNILIN<sup>2</sup>**

<sup>1</sup> University of Zielona Gora, Poland,

<sup>2</sup> Donetsk National Technical University

**ЗМЕНШЕННЯ АПАРАТУРНИХ ВИТРАТ В ПРИБОРІ УПРАВЛІННЯ ІЗ ЗАГАЛЬНОЮ ПАМ'ЯТТЮ**

Пропонується метод зменшення ємності керуючої пам'яті у схемі композиційного мікропрограмного пристрою управління із загальною пам'яттю. Метод заснований на перетворенні адрес мікрокоманд на коди наборів мікрооперацій і дозволяє зменшити апаратні витрати у схемі блоку адресації мікрокоманд.

**Ключові слова:** композиційний мікропрограмний пристрій управління, операторний лінійний ланцюг, логічна схема пристрою управління, загальна пам'ять, лінійний алгоритм управління.

**REDUCTION OF HARDWARE EXPENSES IN CONTROL UNIT WITH THE GENERAL MEMORY**

The method for reducing size of control memory in the scheme of compositional microprogramming control unit with shared memory is proposed. The method is based on the transformation of microinstruction addresses to the set of micro-codes and hardware expenses in the block of addressing of microcommands are reduced.

**Keywords:** compositional microprogram control unit, operational linear object, logical circuit of control unit, common memory, linear control algorithm.