

## ПРЕОБРАЗОВАНИЕ КОДОВ В КОМПОЗИЦИОННЫХ МИКРОПРОГРАММНЫХ УСТРОЙСТВАХ УПРАВЛЕНИЯ

**Ключевые слова:** композиционное микропрограммное устройство управления, ПЛИС, структурная декомпозиция, оптимизация

### ВВЕДЕНИЕ

Методы оптимизации аппаратных затрат в схемах устройств управления (УУ) зависят как от характеристик реализуемого алгоритма управления, так и от параметров элементного базиса [1]. При реализации схем УУ на программируемых логических интегральных схемах (ПЛИС) типа FPGA (field programmable gate arrays) одним из методов оптимизации является функциональная декомпозиция [2, 3]. Это связано с тем, что основу таких ПЛИС составляют логические элементы (ЛЭ) табличного типа LUT (look-up table) [4, 5]. Если алгоритм управления носит линейный характер, то для его интерпретации целесообразно использовать модель композиционного микропрограммного устройства управления (КМУУ) [6]. В КМУУ система микроопераций реализуется на постоянных запоминающих устройствах (ПЗУ). Для этой цели могут быть эффективно использованы встроенные блоки памяти ЕМВ (embedded memory blocks) современных ПЛИС типа FPGA [7, 8]. Условимся использовать обозначение ПЛИС для FPGA, обозначение ЛЭ — для LUT и обозначение ВВП (встроенный блок памяти) — для ЕМВ. Для уменьшения числа ЛЭ в схеме УУ необходимо уменьшать как число аргументов в реализуемых функциях, так и число функций, зависящих от логических условий. Для решения первой задачи предлагается преобразование кодов входов операторных линейных цепей (ОЛЦ) алгоритма управления в коды классов псевдоэквивалентных ОЛЦ. Для решения второй задачи предлагается преобразование кодов входов ОЛЦ в адреса соответствующих им микрокоманд. При этом алгоритм управления представляется в виде граф-схемы алгоритма (ГСА) [1], а для задания УУ используется модель КМУУ с общей памятью [6]. Отметим, что предлагаемые методы могут быть адаптированы к другим формам представления алгоритма и другим моделям КМУУ.

### МОДЕЛЬ КМУУ С ОБЩЕЙ ПАМЯТЬЮ

Пусть алгоритм управления цифровой системы представлен в виде ГСА  $\Gamma$ , которая характеризуется множествами вершин  $B = E_1 \cup E_2 \cup \{b_0, b_E\}$  и дуг  $E$ , соединяющих эти вершины. Здесь  $E_1$  — множество операторных вершин, содержащих наборы микроопераций из множества микроопераций  $Y = \{y_1, \dots, y_N\}$ ;  $E_2$  — множество условных вершин, содержащих элементы множества логических условий (ЛУ)  $X = \{x_1, \dots, x_L\}$ ;  $b_0$  — начальная вершина;  $b_E$  — конечная вершина ГСА  $\Gamma$ . Введем ряд определений [6].

**Определение 1.** Операторной линейной цепью ГСА  $\Gamma$  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  $b_q \in D^g$ , где  $D^g \subseteq E_1$  — множество вершин, входящих в ГСА  $\Gamma$ , 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  $\langle b_q, b_t \rangle \in E$ , где  $b_t \notin D^g$ .

**Определение 4.** Цепи  $\alpha_i$  и  $\alpha_j$  называются псевдоэквивалентными ОЛЦ, если их выходы связаны с входом одной и той же вершины ГСА  $\Gamma$ .

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

Пусть для ГСА  $\Gamma$  получено множество ОЛЦ  $C = \{\alpha_1, \dots, \alpha_G\}$ , соответствующее разбиению минимальной мощности множества  $E_1$  на  $G$  классов, каждый из которых отвечает определению 1.

**Определение 5.** Граф-схема алгоритма  $\Gamma$  является линейной, если число ее операторных вершин не менее чем в два раза превосходит минимально возможное число  $G$  ее ОЛЦ.

Пусть  $I(\Gamma), O(\Gamma)$  — соответственно множество входов и выходов ОЛЦ  $\alpha_g \in C$ . Каждая вершина  $b_q \in E_1$  соответствует микрокоманде, имеющей адрес  $A(b_q)$ . Осуществим адресацию микрокоманд так, чтобы выполнялось условие

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

где  $g \in \{1, \dots, G\}$ ,  $i \in \{1, \dots, F_g - 1\}$ . В этом случае ГСА  $\Gamma$  может быть интерпретирована КМУУ  $U_1$  [7], структурная схема которого приведена на рис. 1.

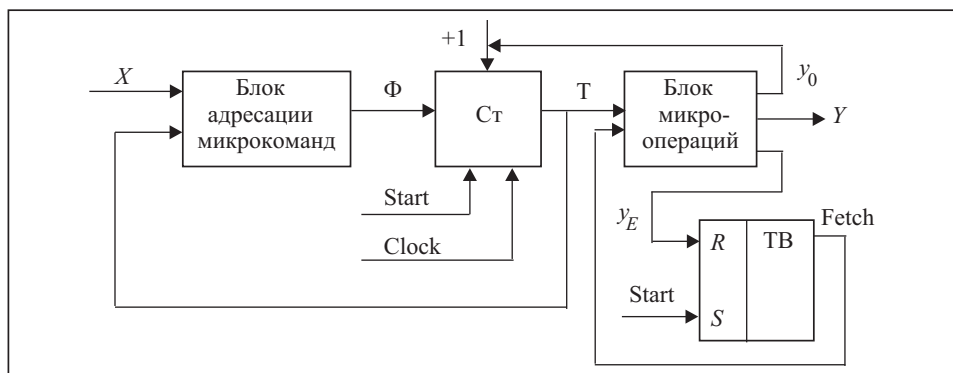


Рис. 1

Это устройство включает блок адресации микрокоманд (БАМ), счетчик (Сч), блок микроопераций (БМО) и триггер выборки (ТВ). КМУУ  $U_1$  функционирует следующим образом.

По сигналу Start в Сч записывается нулевой адрес, соответствующий началу микропрограммы, интерпретирующей ГСА  $\Gamma$ . Одновременно ТВ устанавливается в единичное состояние (Fetch = 1) и микрокоманды могут выбираться из БМО. Если Сч содержит адрес  $A(b_q)$  и  $b_q \notin O(\Gamma)$ , то одновременно с набором микроопераций  $Y(b_q)$ , записанных в вершине  $b_q \in E_1$ , БМО формирует сигнал  $y_0$ . Если  $y_0 = 1$ , то содержимое Сч увеличивается по сигналу Clock. При этом происходит безусловный переход, соответствующий (1). Если  $b_q \in O(\Gamma)$ , то сигнал  $y_0$  не

формируется, а 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 БМО формирует сигнал  $y_E$ , вызывающий установку в нулевое состояние. При этом Fetch = 0, выборка микрокоманд запрещается и КМУУ  $U_1$  прекращает функционирование.

При реализации схемы КМУУ  $U_1$  на ПЛИС схема БАМ, Ст и ТВ строится на логических элементах, а схема БМО реализуется на ВБП. Таким образом, модель позволяет сбалансированно использовать возможности современных ПЛИС. Ее недостатком является значительное число функций обратной связи  $T$ , совпадающее с разрядностью адреса микрокоманды

$$R_A = \lceil \log_2 M \rceil,$$

где  $M = |E_1|$ . В настоящей работе предлагается совместное использование методов структурной декомпозиции (представление блоков модели в виде композиции подблоков) и преобразования кодов блоков, основы которых изложены в [9].

#### ОСНОВНАЯ ИДЕЯ ПРЕДЛАГАЕМЫХ МЕТОДОВ

Функции (2) относятся к классу нерегулярных функций ( $i$ -функций), т.е. они определены на менее чем 50% возможных входных наборов [9]. При реализации  $i$ -функций на ЛЭ возникает проблема, связанная со значительным превышением числа аргументов функций (до 50) числа входов ЛЭ (до 6) современных ПЛИС [7, 8], что приводит к необходимости функциональной декомпозиции  $i$ -функций [2, 3]. Блок микроопераций КМУУ  $U_1$  реализует функции  $Y' = Y'(T)$ . Эти функции являются регулярными ( $r$ -функции), и для их реализации используются ВБП. Для уменьшения числа  $i$ -функций предлагается следующий подход.

Пусть  $I(\Gamma)$  — множество входов ОЛЦ  $\alpha_g \in C$ , т.е.

$$I(\Gamma) = \bigcup_{g=1}^G I_g,$$

и пусть  $|I(\Gamma)| = M_0$ . Поставим в соответствие каждому элементу  $I_g^k \in I(\Gamma)$  двоичный код  $K(I_g^k)$  разрядности

$$R_I = \lceil \log_2 M_0 \rceil.$$

Используем для кодирования входов ОЛЦ переменные  $z_r \in Z$ , где  $|Z| = R_I$ . Очевидно, что существует взаимно однозначное соответствие между кодами входов ОЛЦ и адресами микрокоманд, соответствующими этим входам. Таким образом, можно найти функцию  $\Phi = \Phi(Z)$ , позволяющую получить адреса микрокоманд, соответствующие входам, на основе кодов этих входов.

Пусть  $C' \subseteq C$  — множество ОЛЦ, выходы которых не связаны с входом конечной вершины, т.е.  $\alpha_g \in C'$ , если  $\alpha_g \in C$ , и  $\langle b_t, b_E \rangle \notin E$ , где  $b_t = O_g$ . Пусть  $\Pi_C = \{B_1, \dots, B_I\}$  — разбиение множества ОЛЦ  $C'$  на классы псевдоэквивалентных ОЛЦ. Поставим в соответствие каждому классу  $B_i \in \Pi_C$  двоичный код  $K(B_i)$  разрядности

$$R_C = \lceil \log_2 I \rceil \quad (3)$$

и используем для кодирования переменные  $\tau_r \in \tau$ , где  $|\tau| = R_C$ . Далее КМУУ  $U_1$  может быть преобразовано в КМУУ  $U_2$  (рис. 2).

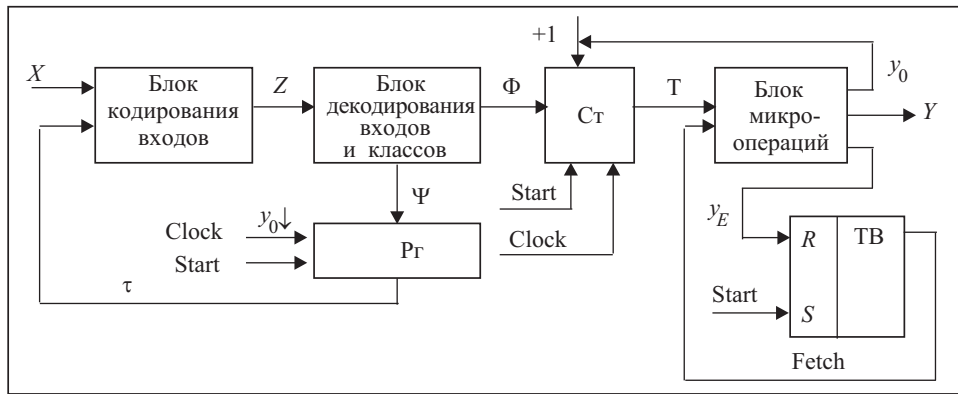


Рис. 2

В КМУУ  $U_2$  блок кодирования входов (БКВ) формирует переменные  $Z = Z(X, \tau)$ , кодирующие входы  $I_g^k \in I(\Gamma)$ . Блок декодирования входов и классов (БДВК) формирует функции

$$\Phi = \Phi(Z), \quad (4)$$

$$\Psi = \Psi(Z), \quad (5)$$

используемые для формирования в Ст адреса входа и в Рг кода класса псевдоэквивалентных ОЛЦ соответственно. Остальные блоки КМУУ  $U_2$  (Ст, БМО, ТВ) имеют то же назначение, что и в КМУУ  $U_1$ .

Устройство  $U_2$  функционирует следующим образом. По сигналу Start в Ст и Рг записываются нулевые коды, начинается выборка микрокоманд из БМО. Если Ст содержит адрес  $A(b_q)$  и  $b_q \notin O(\Gamma)$ , то формируется переменная  $y_0$ . При этом по сигналу Clock содержимое Ст увеличивается на единицу, а содержимое Рг не изменяется. Если  $b_q \in O(\Gamma)$ , то сигнал  $y_0$  не формируется. По сигналу Clock содержимое Ст модифицируется функциями (4), а содержимое Рг — функциями (5). Если  $\langle b_q, b_E \rangle \in E$ , то формируется переменная  $y_E$  и выборка микрокоманд прекращается.

Сравнение моделей  $U_1$  и  $U_2$  показывает, что КМУУ  $U_2$  основано на структурной декомпозиции БАМ и преобразовании кодов входов ОЛЦ в коды классов псевдоэквивалентных ОЛЦ и адреса входов. При этом БДВК реализуется на ВБП, что уменьшает число ЛЭ в схеме КМУУ (по сравнению с  $U_1$ ). При синтезе схемы КМУУ  $U_2$  используется тот факт, что число адресных входов  $S_F$  и выходов  $t_F$  блока памяти современных ПЛИС [7, 8] может настраиваться пользователем при сохранении постоянной емкости

$$V_F = 2^{S_F} t_F. \quad (6)$$

Блок БКВ имеет  $S_1 = R_I$  входов и  $t_1 = R_A + R_C$  выходов, при этом

$$t_0 = (V_F / 2^{R_I}) - (R_A + R_C) \quad (7)$$

выходов блока ВБП не используются для формирования функций  $\Phi$  и  $\Psi$ .

При реализации схемы БМО на блоках ВБП может быть использовано

$$t_1 = V_F / 2^{R_A}$$

выходов для формирования микроопераций  $y_n \in Y'$ . В общем случае для ре-

лизации БМО требуется

$$n_1 = \left\lceil \frac{N+2}{t_1} \right\rceil \quad (8)$$

БВП, имеющих емкость (6). Если  $n_1 > 1$ , то  $t_0$  микроопераций можно реализовать на БДВК. Это целесообразно только при выполнении условия

$$\left\lceil \frac{N+2-t_0}{t_1} \right\rceil < n_1. \quad (9)$$

Разобьем множество  $Y'$  на классы  $Y_I$  и  $Y_R$ , при этом  $y_n \in Y_I$ , если и только если эта микрооперация формируется лишь при переходе во входы ОЛЦ  $\alpha_g \in C$ . В противном случае микрооперация  $y_n \in Y_R$ . При выполнении условий (9) и

$$t_0 \geq n_I, \quad (10)$$

где  $n_I = |Y_I|$ , число БВП в схеме БМО уменьшается на единицу. Это приводит к КМУУ  $U_3$  (рис. 3), принцип функционирования которого не требует пояснения.

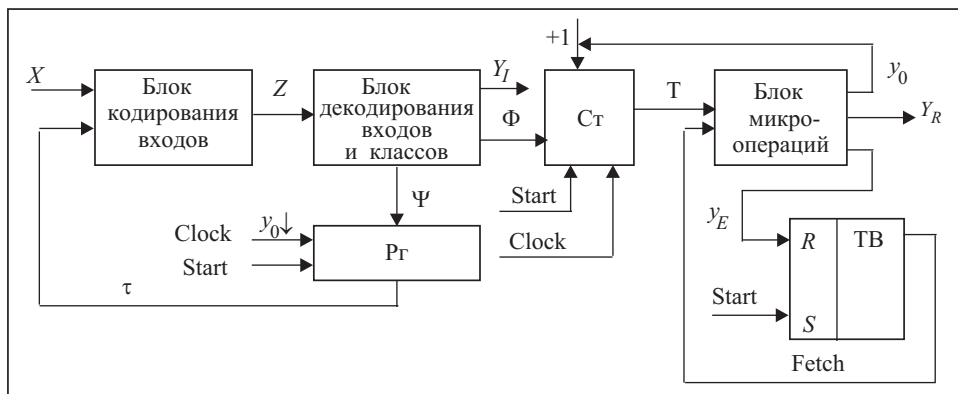


Рис. 3

В настоящее время БВП являются синхронными, что можно учесть при обеспечении устойчивости микроопераций  $y_n \in Y_I$ . Отметим, что КМУУ  $U_3$  — частный случай КМУУ  $U_2$  и становится эквивалентным, если  $Y_I = \emptyset$  или если не выполняется хотя бы одно из условий — (9) или (10). В настоящей работе предлагается метод синтеза КМУУ  $U_3$ , который иллюстрируется на примере ГСА  $\Gamma_1$  (рис. 4).

Для реализации  $r$ -функций условимся использовать БВП с конфигурацией  $128 \times 1, 64 \times 2, 32 \times 4, 16 \times 8, 8 \times 16$  бит, емкость которых  $V_F = 128$  бит при  $t_F = 1$ .

#### МЕТОД СИНТЕЗА И ПРИМЕР ЕГО ПРИМЕНЕНИЯ

Предлагаемый метод синтеза включает следующие этапы:

- 1) формирование множеств ОЛЦ  $C$ , входов  $I(\Gamma)$ , выходов  $O(\Gamma)$  и  $\Pi_C$  для ГСА  $\Gamma$ ;
- 2) линейная адресация микрокоманд;
- 3) разбиение множества  $Y'$ ;
- 4) кодирование входов ОЛЦ  $\alpha_g \in C$ ;
- 5) кодирование классов  $B_i \in \Pi_C$ ;
- 6) спецификация БКВ;
- 7) спецификация БДВК;
- 8) спецификация БМО;
- 9) реализация схемы КМУУ в заданном элементном базисе.

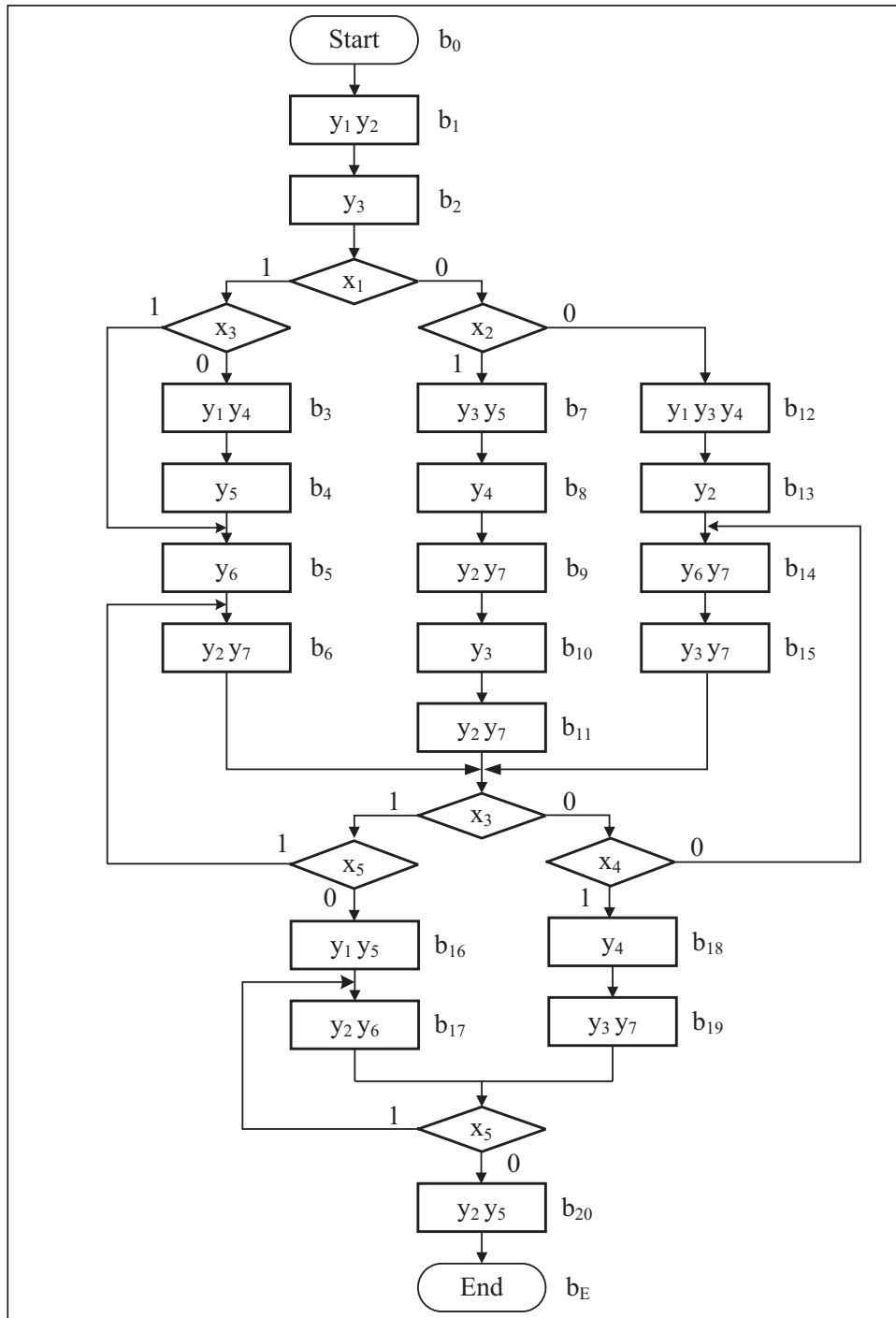


Рис. 4

Выполнение первого этапа не вызывает трудностей и выполняется по методике [6]. Для ГСА  $\Gamma_1$  формируется множество  $C = \{\alpha_1, \dots, \alpha_7\}$ , где  $\alpha_1 = \langle b_1, b_2 \rangle$ ,  $I_1^1 = b_1$ ,  $O_1 = b_2$ ;  $\alpha_2 = \langle b_3, \dots, b_6 \rangle$ ,  $I_2^1 = b_3$ ,  $I_2^2 = b_5$ ,  $I_2^3 = O_2 = b_6$ ;  $\alpha_3 = \langle b_7, \dots, b_{11} \rangle$ ,  $I_3^1 = b_7$ ,  $O_3 = b_{11}$ ;  $\alpha_4 = \langle b_{12}, \dots, b_{15} \rangle$ ,  $I_4^1 = b_{12}$ ,  $I_4^2 = b_{14}$ ,  $O_4 = b_{15}$ ;  $\alpha_5 = \langle b_{16}, b_{17} \rangle$ ,  $I_5^1 = b_{16}$ ,  $I_5^2 = O_5 = b_{17}$ ;  $\alpha_6 = \langle b_{18}, b_{19} \rangle$ ,  $I_6^1 = b_{18}$ ,

$O_6 = b_{19}$ ;  $\alpha_7 = \langle b_{20} \rangle$ ,  $I_7^1 = O_7 = b_{20}$ . Анализ множества  $C$  показывает, что для ГСА  $\Gamma_1$   $M = 20$ ,  $R_A = 5$ ,  $T = \{T_1, \dots, T_5\}$ ,  $\Phi = \{D_1, \dots, D_5\}$ ,  $I(\Gamma_1) = \{b_1, b_3, b_5, b_6, b_7, b_{12}, b_{14}, b_{16}, b_{17}, b_{18}, b_{20}\}$ , т.е.  $M_0 = 11$ ,  $R_I = 4$ ,  $Z = \{z_1, \dots, z_4\}$ . Используя определение псевдоэквивалентных ОЛЦ, можно найти разбиение  $\Pi_C = \{B_1, B_2, B_3\}$ , где  $B_1 = \{\alpha_1\}$ ,  $B_2 = \{\alpha_2, \alpha_3, \alpha_4\}$ ,  $B_3 = \{\alpha_5, \alpha_6\}$ . Отметим, что  $\alpha_7 \notin C'$ . Анализ разбиения  $\Pi_C$  показывает, что  $I = 3$ ,  $R_C = 2$ ,  $\tau = \{\tau_1, \tau_2\}$ ,  $\Psi = \{D_6, D_7\}$ . Кроме того, предполагается использование в Ст и Рг информационных входов типа  $D$ , что соответствует современным тенденциям реализации УУ [10]. В вершинах ГСА  $\Gamma_1$  записаны  $N = 7$  микроопераций, т.е.  $|Y'| = 9$ .

Линейная адресация микрокоманд соответствует условию (1) и выполняется тривиальным образом [6]. Для ГСА  $\Gamma_1$  имеем  $A(b_1) = 00000, \dots, A(b_{20}) = 10100$ .

Рассмотрим целесообразность разбиения множества  $Y'$  для приведенного примера. Так как  $R_A = 5$ , необходимо выбрать конфигурацию ВБП  $32 \times 4$ , т.е.  $t_1 = 4$ . Из условия (8) следует, что для реализации  $N + 2 = 9$  выходов БМО требуется  $n_1 = 3$  ВБП. Поскольку  $R_I = 4$ , для реализации БДВК выбирается конфигурация  $16 \times 8$ , т.е.  $t_F = 8$ . Из выражения (7) следует, что  $t_0 = 1$ , при этом условие (9) выполняется. Таким образом, если одна из микроопераций  $y_n \in Y$  может быть отнесена в множество  $Y_I$ , то число ВБП в схеме БМО уменьшится до  $n_1 = 2$ .

Из анализа операторных вершин ГСА  $\Gamma_1$  следует, что микрооперация  $y_1 \in Y$  есть в вершинах  $b_1, b_3, b_{12}$  и  $b_{16}$ . Эти вершины являются элементами множества  $I(\Gamma_1)$ . Итак, найдено разбиение  $Y_I = \{y_1\}$ ,  $Y_R = \{y_2, \dots, y_7\}$ .

Кодирование входов вершин  $b_q \in I(\Gamma)$  может быть выполнено произвольным образом. Один из возможных вариантов приведен в табл. 1. В данной таблице и далее символ  $U_i(\Gamma_j)$  означает, что модель КМУУ  $U_i$  используется для интерпретации ГСА  $\Gamma_j$ .

**Таблица 1**

$I_g^k$	$b_q$	$K(I_g^k)$	$I_g^k$	$b_q$	$K(I_g^k)$
$I_1^1$	$b_1$	0000	$I_4^2$	$b_{14}$	0110
$I_2^1$	$b_3$	0001	$I_5^1$	$b_{16}$	0111
$I_2^2$	$b_5$	0010	$I_5^2$	$b_{17}$	1000
$I_2^3$	$b_6$	0011	$I_6^1$	$b_{18}$	1001
$I_3^1$	$b_7$	0100	$I_7^1$	$b_{20}$	1010
$I_4^1$	$b_{12}$	0101	–	–	1011

Кодирование классов  $B_i \in \Pi_C$  также выполняется произвольным образом. Пусть для КМУУ  $U_3(\Gamma_1)$  приняты следующие коды:  $K(B_1) = 00$ ,  $K(B_2) = 01$  и  $K(B_3) = 10$ .

Для спецификации ВБП предлагается процедура построения:

- 1) системы формул перехода для входов ОЛЦ  $\alpha_g \in C'$ ;
- 2) системы обобщенных формул перехода;
- 3) таблицы ВБП;
- 4) системы (3) на основе таблицы ВБП.

Система формул перехода [1] для рассматриваемого примера имеет следующий вид:

$$b_2 \rightarrow x_1 x_3 b_5 \vee x_1 \bar{x}_3 b_3 \vee \bar{x}_1 x_2 b_7 \vee \bar{x}_1 \bar{x}_2 b_{12};$$

$$b_6 \rightarrow x_3 x_5 b_6 \vee x_1 \bar{x}_5 b_{16} \vee \bar{x}_3 x_4 b_{18} \vee \bar{x}_3 \bar{x}_4 b_{14};$$

$$b_{11} \rightarrow x_3 x_5 b_6 \vee x_1 \bar{x}_5 b_{16} \vee \bar{x}_3 x_4 b_{18} \vee \bar{x}_3 \bar{x}_4 b_{14};$$

$$\begin{aligned}
b_{15} &\rightarrow x_3x_5b_6 \vee x_1\bar{x}_5b_{16} \vee \bar{x}_3x_4b_{18} \vee \bar{x}_3\bar{x}_4b_{14}; \\
b_{17} &\rightarrow x_5b_{17} \vee \bar{x}_5b_{20}; \\
b_{15} &\rightarrow x_5b_{17} \vee \bar{x}_5b_{20}.
\end{aligned} \tag{11}$$

Для построения системы обобщенных формул перехода [6] необходимо заменить вершины  $b_q \in O(\Gamma)$  обозначениями классов  $B_i \in \Pi_C$ , в которые входят такие ОЛЦ. Если при этом система типа (11) содержит  $i$  одинаковых формул, то  $(i-1)$  из них удаляются. Вершины в левой части формул необходимо заменить обозначениями соответствующих входов. Например, из системы (11) имеем

$$\begin{aligned}
B_1 &\rightarrow x_1x_3I_2^2 \vee x_1\bar{x}_3I_2^1 \vee \bar{x}_1x_2I_3^1 \vee \bar{x}_1\bar{x}_2I_4^1; \\
B_2 &\rightarrow x_3x_5I_2^3 \vee x_3\bar{x}_5I_5^1 \vee \bar{x}_3x_4I_6^1 \vee \bar{x}_3\bar{x}_4I_4^2; \\
B_3 &\rightarrow x_5I_5^2 \vee \bar{x}_5I_7^1.
\end{aligned} \tag{12}$$

Таблица БКВ строится по системе обобщенных формул перехода и имеет следующие столбцы:  $B_i, K(B_i), I_g^k, K(I_g^k), X_h, Z_h, h$ . Столбец  $Z_h$  содержит переменные  $z_r \in Z$ , равные единице в коде  $K(I_g^k)$ , для строки  $h$  таблицы, где  $h = 1, \dots, H_0$ . Для БКВ КМУУ  $U_3(\Gamma_1)$  данная таблица имеет  $H_0 = 10$  строк (табл. 2).

Связь табл. 2 с системой (12) очевидна, при этом коды  $K(I_g^k)$  взяты из табл. 1. Функции  $z_r \in Z$  формируются в виде

$$z_r = \bigvee_{h=1}^{H_0} C_{rh} F_h \quad (r=1, \dots, R_I), \tag{13}$$

где  $C_{rh}$  — булева переменная, равная единице, если и только если переменная  $z_r$  записана в строке  $h$  таблицы. Термы  $F_h$  системы (13) определяются следующим образом:

$$F_h = \left( \bigwedge_{r=1}^{R_C} \tau_r^{I_{rh}} \right) X_h \quad (h=1, \dots, H_0). \tag{14}$$

**Таблица 2**

$B_i$	$K(B_i)$	$I_g^k$	$K(I_g^k)$	Множество условий, $X_h$	$z_r \in Z_h$	Номер строки, $h$
$B_1$	00	$I_2^2$	0010	$x_1x_3$	$z_3$	1
		$I_2^1$	0001	$x_1\bar{x}_3$	$z_4$	2
		$I_3^1$	0100	$\bar{x}_1x_2$	$z_2$	3
		$I_4^1$	0101	$\bar{x}_1\bar{x}_2$	$z_2z_4$	4
$B_2$	01	$I_2^3$	0011	$x_3x_5$	$z_3z_4$	5
		$I_5^1$	0111	$x_3\bar{x}_5$	$z_2z_3z_4$	6
		$I_6^1$	1001	$\bar{x}_3x_4$	$z_1z_4$	7
		$I_4^2$	0110	$\bar{x}_3\bar{x}_4$	$z_2z_3$	8
$B_3$	10	$I_5^2$	1000	$x_5$	$z_1$	9
		$I_7^1$	1010	$\bar{x}_5$	$z_1z_3$	10

В выражении (14)  $I_{rh} \in \{0, 1\}$  — значение  $r$ -го разряда кода  $K(B_i)$  из строки  $h$  таблицы,  $\tau_r^0 = \bar{\tau}_r$ ,  $\tau_r^1 = \tau_r$  ( $r=1, \dots, R_C$ ). Так, из табл. 2, например, имеем



$$z_1 = F_7 \vee F_9 \vee F_{10} = \bar{\tau}_1 \tau_2 \bar{x}_3 x_4 \vee \tau_1 \bar{\tau}_2;$$

$$z_2 = F_3 \vee F_4 \vee F_6 \vee F_8 = \bar{\tau}_1 \bar{\tau}_2 \bar{x}_1 \vee \bar{\tau}_1 \tau_2 x_3 \bar{x}_5 \vee \bar{\tau}_1 \tau_2 \bar{x}_3 \bar{x}_4.$$

Для спецификации БДВК необходимо построить таблицу со столбцами:  $I_g^k$ ,  $K(I_g^k)$ ,  $A(I_g^k)$ ,  $K(B_i)$ ,  $Y_I$ ,  $h$ , где  $k = 1, \dots, 2^{R_I}$ . При этом код  $K(I_g^k)$  рассматривается как адрес ВВП, а остальная информация является выходной. Для КМУУ  $U_3(\Gamma_1)$  данная таблица имеет 16 строк. Здесь показаны только первые 11, остальные слова ВВП для БДВК не используются (табл. 3). Например, вторая строка таблицы соответствует вершине  $b_3 \in E_1$ . Этой вершине соответствует адрес  $A(b_3) = 00010$ , который заносится в столбец  $A(I_g^k)$ . Данная вершина входит в ОЛЦ  $\alpha_2 \in B_2$ , поэтому в столбце  $K(B_i)$  записывается код  $K(B_2)$ . В вершине  $b_3$  записана микрооперация  $y_1$ , поэтому в столбце  $Y_I$  во второй строке пишется единица. Подобный анализ осуществляется произвольно для всех вершин  $b_q \in I(\Gamma_1)$ . Для вершин-выходов ОЛЦ  $\alpha_g \notin C'$  код  $K(B_i)$  не имеет значения, поэтому столбец  $K(B_i)$  содержит код 00 для 11-й строки табл. 3.

**Таблица 3**

$I_g^k$	$K(I_g^k)$	$A(I_g^k)$	$K(B_i)$	$Y_I$	Номер строки, $h$
$I_1^1$	0000	00000	00	1	1
$I_2^1$	0001	00010	01	1	2
$I_2^2$	0010	00100	01	0	3
$I_2^3$	0011	00101	01	0	4
$I_3^1$	0100	00110	01	0	5
$I_4^1$	0101	01011	01	1	6
$I_4^2$	0110	01101	01	0	7
$I_5^1$	0111	01111	10	1	8
$I_5^2$	1000	10000	10	0	9
$I_6^1$	1001	10001	10	0	10
$I_7^1$	1010	10011	00	0	11

Для спецификации блока БМО необходимо построить таблицу со столбцами  $b_q$ ,  $A(b_q)$ ,  $y_0$ ,  $Y_R$ ,  $y_E$ ,  $h$ , имеющую  $2^{R_A}$  строк. Заполнение строки, соответствующей вершине  $b_q \in E_1$ , выполняется следующим образом:

- 1) если  $b_q \neq O_g$ , то в столбец  $y_0$  заносится единица;
- 2) если  $b_q = b_E$ , то в столбец  $y_E$  заносится единица;
- 3) если микрооперация  $y_n \in Y_R$  записана в вершине  $b_q$ , то в столбце  $Y_R$  соответствующий бит равен единице.

В табл. 4 представлен фрагмент таблицы спецификации — первые восемь строк, определяющие БМО КМУУ  $U_3(\Gamma_1)$ .

Синтез логической схемы КМУУ  $U_3$  сводится к реализации системы (13) на логических элементах и реализации таблиц вида табл. 3, 4 на встроенных блоках памяти. Для решения этой задачи необходимо использовать стандартные пакеты [7, 8]. В данной статье этот этап не рассматривается.

**Таблица 4**

$b_q$	$A(b_q)$	$y_0$	$Y_R$	$y_E$	$h$
$b_1$	00000	1	1000000	0	1
$b_2$	00001	0	010000	0	2
$b_3$	00010	1	001000	0	3
$b_4$	00011	1	000100	0	4
$b_5$	0100	1	000010	0	5
$b_6$	00101	0	100001	0	6
$b_7$	00110	1	010100	0	7
$b_8$	00111	1	001000	0	8

#### ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРЕДЛОЖЕННЫХ МЕТОДОВ

Для исследования сравнительной эффективности КМУУ  $U_1-U_3$  разработано программное обеспечение, являющееся расширением системы ATOMIC [11]. Для исследований была создана библиотека тестовых ГСА, представленных в формате, приведенном в [1]. Программные модули генерируют модели КМУУ на языке Verilog. Далее эти описания являются основой для синтеза схемы КМУУ с использованием системы XST фирмы Xilinx [8]. В табл. 5 приведены результаты исследований для моделей классического автомата Мура, память которого реализована на регистре, а система микроопераций — на ВВП, а также КМУУ  $U_1$  и  $U_2$ . Эти данные получены для микросхемы XC2VP30 (Xilinx Virtex-II Pro family) [8]. Исследования проводились для 150 различных ГСА, но в табл. 5 приведены результаты только для 15 из них. Здесь указаны число логических условий  $L$ , число микроопераций  $N$  и число операторных вершин  $M$  для каждой тестовой ГСА. Символом FSM обозначен классический автомат Мура. Для каждого исследуемого УУ указано используемое число ЛЭ табличного типа и число ВВП. Отметим, что результаты исследований в значительной мере зависят от методов функциональной декомпозиции и кодирования состояний автомата. Существенно на них влияет также число классов псевдоэквивалентных ОЛЦ: чем меньше классов, тем лучше результат. Таким образом, результаты синтеза зависят от многих факторов (методов декомпозиции и кодирования, характеристик алгоритма управления, параметров элементного базиса).

Результаты исследований показывают, что для линейных ГСА применение модели КМУУ целесообразно, так как позволяет уменьшить число используемых ЛЭ. Напомним, что согласно определению 5 ГСА Г является линейной, если выполняется условие  $M/G \geq 2$ .

Как видно из табл. 5, переход от модели FSM к модели КМУУ  $U_1$  может сопровождаться ростом числа ЛЭ. Этот факт объясняется тем, что ЛЭ используются и для реализации счетчика. При этом число ЛЭ в схеме Ст зависит как от разрядности адреса микрокоманды, так и от используемого типа переноса между разрядами. Анализ табл. 5 показывает, что число ЛЭ в схеме КМУУ  $U_3$  всегда меньше числа ЛЭ для моделей FSM и  $U_1$ . При этом число ВВП на единицу больше, что объясняется необходимостью реализации БДВК. Отметим, что модель  $U_3$  является частным случаем модели  $U_2$ , поэтому в табл. 5 используется только одно обозначение ( $U_3$ ), хотя результаты приведены для той модели, которая может быть реализована для данной ГСА.

**Таблица 5**

Номер ГСА	Результаты синтеза								
	Характеристики ГСА			FSM		$U_1$		$U_3$	
	$L$	$M$	$N$	ЛЭ	ВБП	ЛЭ	ВБП	ЛЭ	ВБП
1	10	9	85	80	1	83	1	34	2
2	10	9	89	51	1	85	1	33	2
3	13	12	92	126	1	94	1	42	2
4	11	10	71	100	1	96	1	37	2
5	9	10	72	81	1	88	1	32	2
6	8	9	53	47	1	52	1	28	2
7	8	12	68	49	1	71	1	29	2
8	65	18	153	642	1	283	1	211	2
9	35	152	100	361	5	176	5	117	6
10	52	222	151	482	7	208	7	173	8
11	52	152	151	486	5	207	5	171	6
12	52	452	151	486	13	204	13	172	14
13	11	42	154	114	2	58	2	41	3
14	11	42	188	108	2	76	2	52	3
15	14	44	207	134	2	71	2	56	3

В табл. 6 представлены результаты сравнения числа ЛЭ в схемах УУ типа FSM и  $U_1$  и числа ЛЭ КМУУ  $U_2$ , полученные на основе табл. 5. Как видим, в среднем использование модели КМУУ  $U_3$  позволяет уменьшить количество ЛЭ на 59% по сравнению с их числом для классического автомата Мура и на 39% по сравнению с количеством ЛЭ в КМУУ  $U_2$ . Такие же результаты были получены и для всех 150 ГСА. Заметим, что результаты будут иными, если используются другие системы синтеза. Однако из табл. 5 следует, что модель  $U_3$  является эффективной для реализации схем УУ по линейным алгоритмам управления.

**Таблица 6**

Соотношение	Результат соотношения для ГСА															Среднее значение, %
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
$FSM/U_3$	58	63	67	63	60	40	41	67	67	64	65	64	65	52	59	59
$U_1/U_3$	57	61	55	62	64	46	60	26	34	17	18	16	30	32	22	39

**ЗАКЛЮЧЕНИЕ**

В работе рассматривается совместное применение методов структурной декомпозиции блока адресации микрокоманд и преобразования кодов входов ОЛЦ в коды классов псевдоэквивалентных ОЛЦ и адреса микрокоманд. Предлагаемый метод ориентирован на уменьшение числа логических элементов в схеме КМУУ, реализуемой в базисе ПЛИС по линейной ГСА. Эта цель достигается за счет увеличения числа регулярных функций, для реализации которых можно использовать встроенные блоки памяти. Кроме того, кодирование

классов псевдоэквивалентных ОЛЦ дает возможность уменьшить число аргументов в нерегулярных функциях, реализуемых на ЛЭ табличного типа.

Проведенные исследования показали, что предложенный метод позволяет уменьшить число ЛЭ по сравнению как с классической моделью автомата Мура, так и с КМУУ с общей памятью. Выигрыш зависит от многих факторов, например от числа входов ОЛЦ и классов псевдоэквивалентных ОЛЦ. Выигрыш увеличивается по мере уменьшения величин этих параметров для ГСА с одинаковым числом операторных вершин.

Дальнейшее уменьшение числа ЛЭ связано с разработкой методов кодирования входов ОЛЦ и классов псевдоэквивалентных ОЛЦ. Это является основным направлением дальнейших исследований. Другое направление — адаптация разработанных методов к особенностям ПЛИС, макроячейки которых основаны на элементах программируемой матричной логики [10].

#### СПИСОК ЛИТЕРАТУРЫ

1. Baranov S. Logic and system design of digital systems. — Tallinn: TUT Press, 2008. — 266 p.
2. Scholl C. Functional decomposition with application to FPGA. — Boston: Kluwer Acad. Publ., 2001. — 288 p.
3. Rawski H., Jozwiak L., Luba T. Functional decomposition with an efficient input support selection for sub-functions based on information relationships measures // J. System Archit. — 2001. — 47, N 3. — P. 137–155.
4. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. — СПб.: БХВ — СПб., 2002. — 636 с.
5. Maxfield S. The design warrior's guide to FPGAs. — Amsterdam: Elsevier, 2004. — 541 p.
6. Barkalov A., Titarenko L. Logic synthesis for compositional microprogram control units. — Berlin: Springer, 2008. — 273 p.
7. Altera Corporation. FPGA, CPLD and Structured ASIC — <http://www.altera.com>.
8. Xilinx Inc. FPGA, CPLD Solutions. — <http://www.xilinx.com>.
9. Баркалов А.А., Титаренко Л.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС. — Донецк: УНИТЕХ, 2009. — 336 с.
10. Соловьев В.В., Климович А. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем. — М.: Горячая линия — Телеком, 2008. — 376 с.
11. Wisniewski R. Synthesis of compositional microprogram control units for programmable devices. — Zielona Gora: UZG Press, 2009. — 104 p.

*Поступила 28.09.2009*