ИНФОРМАТИКА

2006 **июль-сентябрь** **№ 3**

681.324

А.А. Баркалов, Л.А. Титаренко, М. Колопеньчик

ОПТИМИЗАЦИЯ ЕМКОСТИ УПРАВЛЯЮЩЕЙ ПАМЯТИ УСТРОЙСТВА УПРАВЛЕНИЯ С РАЗДЕЛЕНИЕМ КОДОВ

Предлагается метод проектирования композиционного микропрограммного устройства управления с разделением кодов, основанный на применении специального преобразователя адресов для формирования адреса микрокоманды на основе ее представления в виде пары <код операторной линейной цепи – код компоненты>. Такой подход позволяет использовать все положительные качества метода разделения кодов независимо от характеристик интерпретируемой граф-схемы алгоритма. Предлагаемый метод позволяет уменьшить емкость управляющей памяти по сравнению со всеми известными методами синтеза подобных устройств управления. Приведен пример применения этого метода.

Введение

Одним из методов реализации устройства управления любой цифровой системы является применение модели композиционного микропрограммного устройства управления (КМУУ) [1, 2]. В КМУУ система микроопераций реализуется с помощью запоминающих устройств, а система функций адресации микрокоманд – на элементах произвольной логики [2]. Такой подход вполне соответствует современным системам на кристалле (SoC-System-on-chip) [3, 4], в которых есть средства для реализации произвольной логики (FPGA, CPLD) и памяти (DMB – dedicated memory blocks). Стремление к увеличению числа реализуемых цифровой системой функций вызывает необходимость оптимизации ее блоков. В случае КМУУ оптимизация может быть достигнута за счет применения метода разделения кодов [2], однако разделение кодов целесообразно только при сохранении минимальной разрядности адреса микрокоманд. В противном случае резко увеличивается число DMB, необходимых для реализации системы микроопераций. В настоящей работе предлагается метод, позволяющий использовать разделение кодов и сохранить, а в ряде случаев и уменьшить, емкость памяти КМУУ по сравнению с известными методами [1, 2].

1. Основные определения и общие положения

Пусть управляющий алгоритм цифровой системы представлен в виде граф-схемы алгоритма (ГСА) Γ [5] с множеством вершин $B = \{b_0, b_E\} \cup B_1 \cup B_2$ и множеством дуг $E = \{\langle b_i, b_j \rangle \mid b_i, b_j \in B\}$. Здесь b_0 – начальная вершина; b_E – конечная вершина; B_1 – множество операторных вершин, содержащих наборы микроопераций (микрокоманд) $Y_q \subseteq Y$, где $Y = \{y_1, \ \ldots, \ y_N\}$; B_2 – множество условных вершин, содержащих элементы множества логических условий $X = \{x_1, \ \ldots, \ x_L\}$. При этом под микрооперациями понимаются переменные, инициирующие выполнение элементарных операций обработки информации операционным автоматом цифровой системы [1]. Введем ряд определений [1, 2], необходимых для дальнейшего изложения материала.

Определение 1. Операторной линейной цепью (ОЛЦ) ГСА Γ называется конечная последовательность операторных вершин $\alpha_g = \langle \alpha_{g1}, \ \ldots, \ b_{gF_g} \rangle$, такая, что для любой пары соседних компонент вектора α_g существует дуга $\langle b_{gi}, \ b_{gi+1} \rangle \in E$, где $i = 1, \ldots, F_g - 1$.

Определение 2. Входом ОЛЦ α_g называется вершина $b_q \in B_1$, такая, что существует дуга $\langle b_t, \ b_q \rangle \in E$, где $b_t = b_0$, или $b_t \in B_2$, или $b_t \notin D^g$ ($D^g \subseteq B_1$ – множество вершин, входящих в ОЛЦ α_g).

Определение 3. Выходом ОЛЦ α_g называется вершина $b_q \in B_1$, такая, что существует дуга $\langle b_q, \ b_t \rangle \in E$, где $b_t = b_E$, или $b_t \in B_2$, или $b_t \notin D^g$.

Произвольная ОЛЦ α_g может иметь более одного входа, пусть I_g^j – j-й вход ОЛЦ α_g. Каждая ОЛЦ α_g имеет только один выход O_g, входящий в множество выходов ОЛЦ ГСА Γ, которое обозначается как $O(\Gamma)$.

Пусть для ГСА Γ сформировано множество ОЛЦ $C = \{\alpha_1, \ \ldots, \ \alpha_G\}$, удовлетворяющее условию

$$\begin{aligned} &\left|D^i \cap D^j\right| = 0 \ (i \neq j, i, j \in \{1, \ \ldots, \ G\}); \\ &B_1 = D^1 \cup D^2 \cup \ldots \cup D^G; \\ &G \to \min. \end{aligned} \tag{1}$$

Назовем вершины $b_q \in D^g \ (g = 1, \ldots, G)$ компонентами ОЛЦ $\alpha_g \in C$. Пусть F_g – число компонент в ОЛЦ $\alpha_g \in C$ и $F_{max} = \max(F_1, \ \ldots, \ F_G)$. Закодируем ОЛЦ $\alpha_g \in C$ R_1-разрядными кодами $K(\alpha_g)$, где $R_1 =]\log_2 F_{max}[$, причем выполним кодирование так, чтобы для соседних компонент $b_{gi}, b_{gi+1} \ (i = 1, \ \ldots, \ F_g - 1)$ соблюдалось условие

$$K(b_{gi+1}) = K(b_{gi}) + 1 \ (g = 1, \ \ldots, \ G). \tag{2}$$

В этом случае адрес $A(b_t)$ микрокоманды $Y(b_t)$, записанной в вершине $b_t \in B_1$, такой, что $b_t \in D^g$, представляется в виде

$$A(b_t) = K(\alpha_g) * K(b_t), \tag{3}$$

где $*$ – знак конкатенации. Представление адреса микрокоманды (3) называется разделением кодов [2], и ему соответствует КМУУ U_1 (рис. 1).

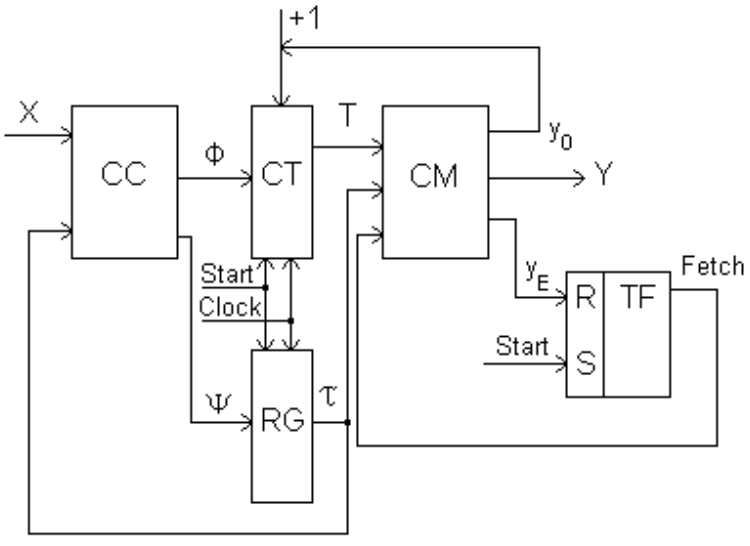

Рис. 1. Структурная схема КМУУ U_1 с разделением кодов

На рис. 1 комбинационная схема СС реализует систему функций возбуждения счетчика СТ

$$\Phi = \Phi(\tau, X) \tag{4}$$

и систему функций возбуждения регистра RG

$$\psi = \psi(\tau, X), \tag{5}$$

где τ – внутренние переменные, кодирующие ОЛЦ $\alpha_g \in C$ и $|\tau| = R_1$. Компоненты ОЛЦ $\alpha_g \in C$ кодируются внутренними переменными T и $|T| = R_2$. Управляющая память СМ хранит микрооперации Y, а также внутренние переменные управления синхронизацией y_0 и выборкой из управляющей памяти y_E.

Устройство U_1 на рис. 1 функционирует следующим образом. По сигналу Start=1 в счетчик СТ и регистр RG заносятся нулевые коды, что соответствует адресу $A(b_q)$, где $\langle b_o, b_q \rangle \in E$, одновременно триггер TF устанавливается в единичное состояние и сигнал Fetch разрешает выборку микрокоманд из СМ. Очередная микрокоманда $Y(b_q)$ считывается из СМ. Если $b_q \neq O_g$ $(g = \overline{1, G})$, то одновременно с микрооперациями Y формируется переменная y_0. При $y_0 = 1$ по сигналу Clock к содержимому СТ прибавляется единица, что соответствует адресации микрокоманд в пределах одной ОЛЦ, содержамое RG при этом не меняется. Если $b_q = O_g$ $(g = \overline{1, G})$, то переменная y_0 не формируется и по сигналу Clock в RG заносится код очередной ОЛЦ, определяемый функциями Ψ, а в СТ заносится код компоненты, определяемый функциями Φ. При достижении адреса $A(b_t)$, где $\langle b_t, b_E \rangle \in E$, формируется переменная y_E, триггер TF обнуляется и выборка микрокоманд из СМ прекращается.

Такая организация КМУУ позволяет использовать методы кодирования ОЛЦ, основанные на результатах [6], и с помощью методов оптимизации автомата Мура [2] минимизировать число FPGA в схеме СС. Однако при выполнении условия

$$R_1 + R_2 > R, \tag{6}$$

где $R =]\log_2 M[$, $M = |B_1|$, емкость управляющей памяти СМ резко увеличивается по сравнению с минимальным значением

$$V_{min} = 2^R (N + 2), \tag{7}$$

что приводит к росту числа DMB, необходимых для ее реализации.

В настоящей работе предлагается метод организации КМУУ, позволяющий не только сохранить, но даже и уменьшить требуемую емкость СМ по сравнению с V_{min} при применении метода разделения кодов.

2. Основная идея предлагаемого метода

Пусть ГСА Γ включает Q различных наборов микроопераций $Y_q \subseteq Y$. Закодируем микрокоманду Y_q кодом $B(Y_q)$ разрядности $R_3 =]\log_2 Q[$ и используем переменные $z_r \in Z$ для такого кодирования, где $|Z| = R_3$. Будем рассматривать код $B(Y_q)$ как адрес микрокоманды Y_q $(q = 1, \ldots, Q)$ в управляющей памяти СМ. Пусть вершина $b_t \in B_1$ содержит микрокоманду $Y_q = Y(b_t)$. Построим таблицу преобразования адресов (3) в коды $B(Y_q)$, соответствующую преобразованию $\Lambda : T \cup \tau \to Z$. В этом случае КМУУ с разделением кодов может быть представлено в виде КМУУ U_2 (рис. 2).

В устройстве U_2 схема местного управления LCS формирует внутренние переменные

$$y_0 = f_1(T, \tau); \tag{8}$$

$$y_E = f_2(T, \tau), \tag{9}$$

преобразователь адресов АТ реализует систему функций

$$Z = Z(T, \tau), \tag{10}$$

соответствующую преобразованию Λ. Таким образом, в КМУУ U_2 схема АТ формирует адреса микрокоманд, а схема LCS управляет работой счетчика СТ и выборкой микрокоманд из СМ.

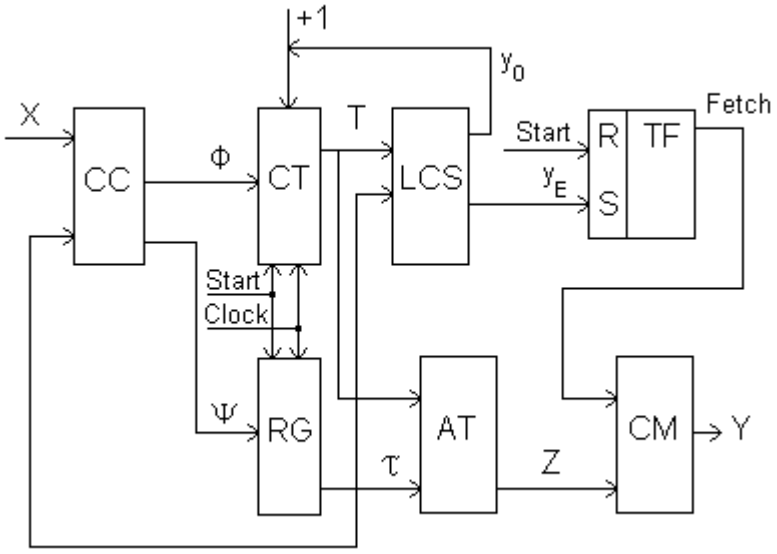

Рис. 2. Структурная схема устройства U_2

При выполнении условия (6) такая организация позволяет использовать все позитивные качества метода преобразования кодов, а при выполнении условия

$$R_3 < R \tag{11}$$

требуемая емкость СМ уменьшается по сравнению с V_{min}.

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

– преобразование исходной ГСА Γ;
– формирование множества С ОЛЦ преобразованной ГСА $\Gamma(U_2)$;
– кодирование операторных линейных цепей $\alpha_g \in C$;
– кодирование операторных вершин;
– адресацию микрокоманд с использованием метода разделения кодов;
– кодирование наборов микроопераций переменными $z_r \in Z$;
– формирование содержимого управляющей памяти;
– формирование таблицы переходов КМУУ;
– формирование таблицы преобразователя адресов;
– формирование таблицы схемы местного управления LCS;
– реализацию схемы КМУУ в заданном элементном базисе.

3. Пример применения предложенного метода

Рассмотрим пример применения предложенного метода для синтеза КМУУ U_2 (Γ_1) по ГСА Γ_1 (рис. 3).

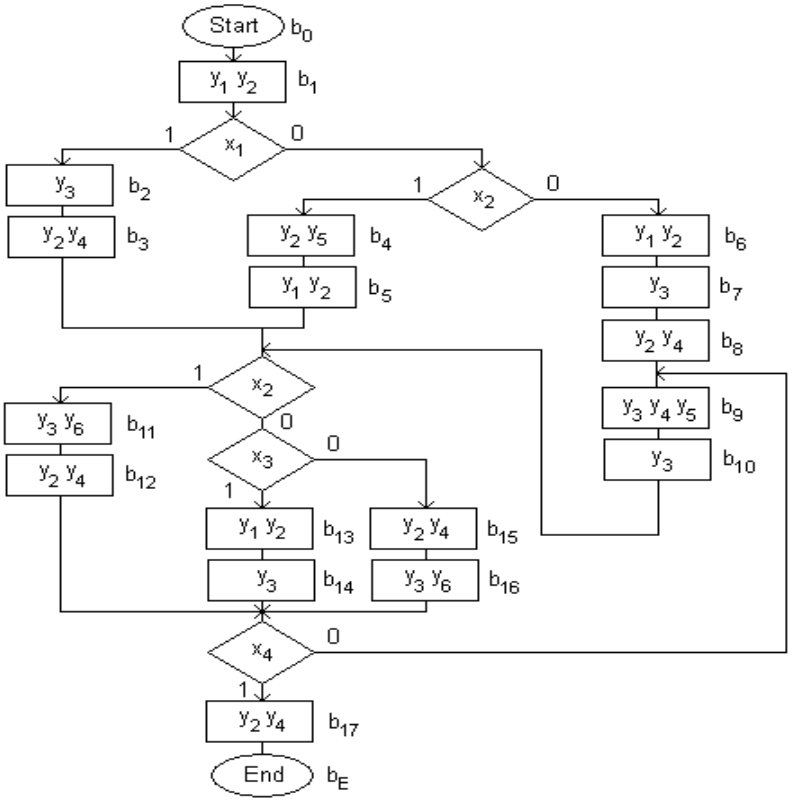

Рис. 3. Исходная ГСА Γ_1

Преобразование исходной ГСА. Преобразование сводится к введению дополнительных операторных вершин, а также к введению дополнительных переменных в вершины исходной ГСА. Если существует дуга $\langle b_0, b_q \rangle \in E$, где $b_q \in B_2$, то в ГСА Γ вводится дополнительная вершина $b_t \in B_1$ и дуга $\langle b_0, b_q \rangle$ заменяется дугами $\langle b_0, b_t \rangle$ и $\langle b_t, b_q \rangle$. Если существует дуга $\langle b_q, b_E \rangle$, где $b_q \in B_1$, то в вершину b_q вводится переменная y_E (в нашем примере – в b_{17}). Если существует дуга $\langle b_q, b_E \rangle$, где $b_q \in B_2$, то в ГСА Γ вводится вершина $b_t \in B_1$ с y_E, а дуга $\langle b_q, b_E \rangle$ заменяется дугами $\langle b_q, b_t \rangle$ и $\langle b_t, b_E \rangle$.

Для рассматриваемого примера структура преобразованной ГСА $\Gamma_1(U_2)$ совпадает со структурой исходной ГСА, поэтому сохраним в дальнейшем обозначение Γ_1.

Формирование множества ОЛЦ. Применение процедуры из работы [2] приводит к множеству $C = \{\alpha_1, \ldots, \alpha_8\}$, где $\alpha_1 = \langle b_1 \rangle$, $I_1^1 = O_1 = b_1$; $\alpha_2 = \langle b_2, b_3 \rangle$, $I_2^1 = b_2$, $O_2 = b_3$; $\alpha_3 = \langle b_4, b_5 \rangle$, $I_3^1 = b_4$, $O_3 = b_5$; $\alpha_4 = \langle b_6, \ldots, b_{10} \rangle$, $I_4^1 = b_6$, $I_4^2 = b_9$, $O_4 = b_{10}$; $\alpha_5 = \langle b_{11}, b_{12} \rangle$, $I_5^1 = b_{11}$, $O_5 = b_{12}$; $\alpha_6 = \langle b_{13}, b_{14} \rangle$, $I_6^1 = b_{13}$, $O_6 = b_{14}$; $\alpha_7 = \langle b_{15}, b_{16} \rangle$, $I_7^1 = b_{15}$, $O_7 = b_{16}$; $\alpha_8 = \langle b_{17} \rangle$, $I_8^1 = O_8 = b_{17}$. Итак, $G = 8$, $F_{max} = 5$, $O(\Gamma_1) = \{b_1, b_3, b_5, b_{10}, b_{12}, b_{14}, b_{16}, b_{17}\}$, $R_1 = 3$, $T = \{T_1, T_2, T_3\}$, $R_2 = 3$, $\tau = \{\tau_1, \tau_2, \tau_3\}$, $M = 17$, $R = 5$, следовательно, условие (6) выполняется и использование предлагаемого метода целесообразно.

Кодирование операторных линейных цепей. В рассматриваемом КМУУ этот этап может быть выполнен по аналогии с кодированием псевдоэквивалентных состояний автомата Мура [7], что позволяет уменьшить аппаратурные затраты в схеме СС. Чтобы не усложнять пример, закодируем ОЛЦ $\alpha_g \in C$ тривиальным образом: $K(\alpha_1) = 000, \ldots, K(\alpha_8) = 111$.

Кодирование операторных вершин. Присвоим первым компонентам ОЛЦ $\alpha_g \in C$ нулевой код разрядности $R_2 = 3$: $K(b_1) = K(b_2) = K(b_4) = K(b_6) = K(b_{11}) = K(b_{13}) = K(b_{15}) = K(b_{17}) = 000$. Тогда в соответствии с условием (2) имеем $K(b_3) = K(b_5) = K(b_7) = K(b_{12}) = K(b_{14}) = K(b_{16}) = 001$, $K(b_8) = 010$, $K(b_9) = 011$, $K(b_{10}) = 100$.

Адресация микрокоманд. Полученные коды операторных вершин и ОЛЦ $\alpha_g \in C$ позволяют найти адреса (3). Например, $K(\alpha_1) = 000$ и $K(b_1) = 000$, следовательно, $A(b_1) = 000000$; $K(\alpha_4) = 011$ и $K(b_8) = 010$, поэтому $A(b_8) = 011010$ и т. д.

Кодирование наборов микроопераций. Анализ ГСА Γ_1 показывает, что в ней имеется $Q = 6$ попарно различных наборов микроопераций, где $Y_1 = \{y_1, y_2\}$, $Y_2 = \{y_3\}$, $Y_3 = \{y_2, y_4\}$, $Y_4 = \{y_2, y_5\}$, $Y_5 = \{y_3, y_4, y_5\}$, $Y_6 = \{y_3, y_6\}$, для кодирования которых достаточно, чтобы $R_3 =]\log_2 6[= 3$. Таким образом, множество переменных $Z = \{z_1, z_2, z_3\}$. Закодируем наборы $Y_q \subseteq Y$ произвольным образом: $B(Y_1) = 000, \ldots, B(Y_6) = 101$.

Формирование содержимого управляющей памяти. Данный этап сводится к формированию таблицы с входами $B(Y_q)$ и выходами $y_n \in Y_q$. Для КМУУ $U_2(\Gamma_1)$ эта таблица имеет $Q = 6$ строк (табл.1).

Таблица 1

Содержимое управляющей памяти КМУУ $U_2(\Gamma_1)$

$B(Y_q)$	Микрооперации	$B(Y_q)$	Микрооперации
000	$y_1\ y_2$	011	$y_2\ y_5$
001	y_3	100	$y_3\ y_4\ y_5$
010	$y_2\ y_4$	101	$y_3\ y_6$

Формирование таблицы переходов КМУУ. Таблица задает функции (4), (5) и может быть сформирована на основе системы формул перехода [5] для вершин $b_t \in O(\Gamma)$. Если $\langle b_t, b_E \rangle \in E$, то переходы для вершины $b_t \in B_1$ не рассматриваются. В этом случае система формул перехода имеет вид

$$O_1 \to x_1 I_2^1 \vee \overline{x}_1 x_2 I_3^1 \vee \overline{x}_1 \overline{x}_2 I_4^1;$$

$$O_2, O_3, O_4 \to x_2 I_5^1 \vee \overline{x}_2 x_3 I_6^1 \vee \overline{x}_2 \overline{x}_3 I_7^1;$$

$$O_5, O_6, O_7 \to x_4 I_8^1 \vee \overline{x}_4 I_4^2.$$

Фрагмент таблицы переходов КМУУ $U_2(\Gamma_1)$ соответствует формуле для O_1 (табл. 2). Здесь α_g – ОЛЦ, соответствующая выходу O_g из левой части формулы перехода; I_q^i – вход ОЛЦ из правой части формулы перехода; $A(I_q^i)$ – адрес входа I_q^i, представленный в форме (3); X_h – набор входных сигналов, определяющих переход $\langle O_g, I_q^i \rangle$; Φ_h – набор функций возбуждения триггеров СТ; Ψ_h – набор функций возбуждения регистра RG; $h = 1, \ldots, H$ – номер перехода. Отметим, что регистр RG и счетчик СТ имеют информационные входы типа D [2].

Таблица 2

Фрагмент таблицы переходов КМУУ $U_2(\Gamma_1)$

α_g	$K(\alpha_g)$	I_q^i	$A(I_q^i)$	X_h	Φ_h	Ψ_h	h
α_1	000	I_2^1	001000	x_1	–	D_3	1
		I_3^1	010000	$\bar{x}_1\, x_2$	–	D_2	2
		I_4^1	011000	$\bar{x}_1\, x_2$	–	$D_2\, D_3$	3

Формирование таблицы преобразователя адресов. Эта таблица задает закон формирования системы (10) и включает столбцы b_m, $A(b_m)$, Y_q, $B(Y_q)$, Z_m, m. Здесь Z_m – переменные $z_r \in Z$, принимающие единичное значение в коде $B(Y_q)$; $m = 1, \ldots, M$ – номер строки таблицы. Таблица АТ формируется тривиальным образом и в случае КМУУ $U_2(\Gamma_1)$ включает $M = 17$ строк, первые пять из которых показаны в табл. 3.

Таблица 3

Фрагмент таблицы преобразователя кодов КМУУ $U_2(\Gamma_1)$

b_m	$A(b_m)$	Y_q	$B(Y_q)$	Z_m	m
b_1	000000	Y_1	000	–	1
b_2	001000	Y_2	001	z_3	2
b_3	001001	Y_3	010	z_2	3
b_4	010000	Y_4	011	$z_2\, z_3$	4
b_5	010001	Y_1	000	–	5

Формирование таблицы схемы местного управления. Эта таблица задает закон формирования переменных (8), (9) и включает столбцы b_m, $A(b_m)$, y_0, y_E, m. Напомним, что переменная y_0 включается в вершины $b_q \notin O(\Gamma)$, а переменная y_E включается в вершины b_t, такие, что $\langle b_t, b_E \rangle \in E$. В рассматриваемом примере данная таблица имеет 17 строк и строится тривиальным образом. Например, для b_2: $A(b_2) = 001000$, $y_0 = 1$, $y_E = 0$; для b_9: $A(b_9) = 011011$, $y_0 = 1$, $y_E = 0$; для b_{17}: $A(b_{17}) = 111000$, $y_0 = 1$, $y_E = 1$.

Реализация схемы КМУУ U2. На этом этапе формируются функции (4), (5) по таблице переходов КМУУ, функции (8), (9) по таблице схемы местного управления и функции (10) по таблице преобразователя кодов. Синтез схемы сводится к реализации полученных систем функций на FPGA и к реализации управляющей памяти на DMB на основе таблицы содержимого управляющей памяти. В настоящее время имеются эффективные методы решения подобных задач [4, 7], которые в данной работе не рассматриваются.

Заключение

Применение метода преобразования адресов позволяет использовать разделение кодов при синтезе КМУУ вне зависимости от характеристик исходной ГСА, определяющих разрядности кодов ОЛЦ, компонент ОЛЦ и минимальную разрядность адреса микрокоманды. При этом сохраняется возможность оптимального кодирования псевдоэквивалентных ОЛЦ, что уменьшает аппаратурные затраты в схеме СС по сравнению с произвольным кодированием ОЛЦ. Отметим, что применение преобразователя адреса приводит к увеличению времени цикла КМУУ U_2 по сравнению с соответствующим параметром КММУ U_1. Однако такой подход дает потенциальную возможность уменьшения емкости управляющей памяти КМУУ U_2 по сравнению со всеми известными методами реализации КМУУ [2]. Исследования авторов пока-

зали, что при выполнении условия (6) и уменьшении емкости управляющей памяти за счет введения преобразователя адреса предлагаемый метод позволяет на 14–17 % уменьшить аппаратурные затраты по сравнению и с известными методами реализации КМУУ.

Список литературы

1. Баркалов, А.А. Синтез микропрограммных устройств управления / А.А. Баркалов, А.В. Палагин. – Киев: ИК НАН Украины, 1997. – 136 с.

2. Баркалов, А.А. Синтез устройств управления на программируемых логических устройствах / А.А. Баркалов. – Донецк: ДонНТУ, 2002. – 262 с.

3. Грушницкий, Р.И. Проектирование систем на микросхемах программируемой логики / Р.И. Грушницкий, А.Х. Мурсаев, Е.П. Угрюмов. – СПб.: БХВ, 2002. – 636 с.

4. Synteza układów cyfrowych: Praca zbiorowa pod redakcją prof. Tadeusza Łuby. – Warszawa: WKŁ, 2003. – 318 s.

5. Baranov, S. Logic Synthesis for Control Automata / S. Baranov. – Kluwer Academic Publishers, 1994.

6. Закревский, А.Д. Основы логического проектирования. Кн. 2. Оптимизация в булевом пространстве / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – Минск: ОИПИ НАН Беларуси, 2004. – 240 с.

7. Соловьев, В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем / В.В. Соловьев. – М.: Горячая линия – Телеком, 2001. – 636 с.

Поступила 16.01.06

Зеленогурский университет,
Польша, Зеленая Гура, Подгорная, 50
e-mail: a.barkalov@iie.uz.zgora.pl
l.titarenko@iie.uz.zgora.pl

A.A. Barkalov, L.A. Titarenko, M. Kolopenczyk

OPTIMIZATION OF CONTROL MEMORY SIZE OF CONTROL UNIT WITH CODES SHARING

A method of design of compositional microprogram control unit with codes sharing is proposed. The proposed method is based on application of special address transformer to form an address of microinstruction on the base of its representation as a pair <code of operational linear chain, code of component>. This approach permits to use all positive features of codes sharing independently on characteristics of interpreted flow-chart of algorithm. The proposed method permits to decrease the size of control memory in comparison with all known methods of such control unit design. An example of a proposed method application is given.