

А.А. Баркалов, Л.А. Титаренко, А.С. Лаврик

Оптимизация схемы адресации микрокоманд в устройстве управления с разделением кодов

Предложен метод уменьшения числа логических элементов табличного типа в схеме композиционного микропрограммного устройства управления с разделением кодов. Метод основан на кодировании входов операторных линейных цепей, позволяющем однозначно определить коды классов псевдоэквивалентных цепей. Приведен метод синтеза и пример его применения.

A method of decrease of the number of logic elements LUT is suggested for a compositional microprogram control unit with code sharing. The method is based on the encoding of the inputs of operational linear chains, when the codes of classes of pseudoequivalent linear chains are determined. The synthesis method and an example of its application are given.

Запропоновано метод зменшення числа логічних елементів табличного типу в схемі композиційного мікропрограмного упорядкування управління з розділенням кодів. Метод засновано на кодуванні входів операторних лінійних ланцюгів, який дозволяє однозначно визначити коди класів псевдоеквівалентних ланцюгів. Наведено метод синтезу та приклад його застосування.

Введение. Один из методов реализации устройства управления любой цифровой системы – использование модели композиционного микропрограммного устройства управления (КМУУ) [1, 2], где система микроопераций реализуется с использованием 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 как произвольной логики, так и памяти. 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], позволяющего использовать известные методы, ориентированные на автомат Мура [5]. В данной статье рассматривается метод уменьшения числа *LUT (look-up table)* элементов в схеме КМУУ с разделением кодов, реализуемой на *FPGA (field-programmable gate arrays)* с блоками встроенной памяти *EMB (embedded memory blocks)*. Метод основан на преобразовании кодов входов операторных линейных цепей (ОЛЦ) в ко-

ды ОЛЦ. При этом кодирование входов осуществляется таким образом, чтобы классы псевдоэквивалентных ОЛЦ представлялись одним интервалом кодирующего пространства. Алгоритм управления представляется в виде граф-схем алгоритма (ГСА) [6].

Основные определения и общие положения

Пусть управляющий алгоритм цифровой системы представлен в виде граф-схемы алгоритма Γ [6] с множеством вершин $B = \{b_0, b_E\} \cup E_1 \cup E_2$ 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, E_1 – множество операторных вершин, содержащих наборы микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$, E_2 – множество условных вершин, содержащих элементы множества логических условий $X = \{x_1, \dots, x_L\}$. Используется ряд определений, взятых из [1] и необходимых для дальнейшего изложения материала.

Определение 1. Операторной линейной цепью ГСА Γ называется конечная последовательность операторных вершин $\alpha_g = \langle b_{g1}, \dots, b_{gF_g} \rangle$, такая что для любой пары соседних компонент

Ключевые слова: устройство управления, операторная линейная цепь, преобразование кодов, *LUT*, *EMB*.

вектора α_g существует дуга $\langle b_{gi}, b_{gi+1} \rangle \in E$ ($i=1, \dots, F_{g-1}$).

Определение 2. Входом ОЛЦ α_g называется вершина $b_q \in E_1$ такая, что существует дуга $\langle b_i, b_q \rangle \in E$, где $b_i \notin D^g$, $D^g \subseteq E_1$ – множество вершин, входящих в ОЛЦ α_g .

Определение 3. Выходом ОЛЦ α_g называется вершина $b_q \in E_1$ такая, что существует дуга $\langle b_q, b_i \rangle \in E$, где $b_i \notin D^g$.

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 , обозначаемое как $O(\Gamma)$. Пусть входы ОЛЦ α_g ГСА Γ образуют множество $I(\Gamma)$.

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

$$\begin{aligned} |D^i \cap D^j| &= 0 (i \neq j; i, j \in \{1, \dots, G\}); \\ E_1 &= D^1 \cup D^2 \cup \dots \cup D^G; \\ G &\rightarrow \min. \end{aligned} \quad (1)$$

Пусть F_g – число компонент в ОЛЦ $\alpha_g \in C$ и $F_{\max} = \max(F_1, \dots, F_G)$. Закодируем ОЛЦ $\alpha_g \in C$ R_1 -разрядными кодами $K(\alpha_g)$, где

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

Закодируем компоненты $b_q \in D^g$ двоичными кодами $K(b_q)$ разрядности

$$R_2 = \lceil \log_2 F_{\max} \rceil, \quad (3)$$

причем выполним кодирование так, чтобы для соседних компонент b_{gi}, b_{gi+1} ($i=1, \dots, F_g-1$) выполнялось условие

$$K(b_{gi+1}) = K(b_{gi}) + 1 \quad (g=1, \dots, G). \quad (4)$$

В этом случае адрес $A(b_i)$ микрокоманды $Y(b_i)$, записанной в вершине $b_i \in E_1$, такой, что $b_i \in D^g$ представляется в виде

$$A(b_i) = K(\alpha_g) * K(b_i), \quad (5)$$

где $*$ – знак конкатенации. Представление адреса микрокоманды (5) называется разделением кодов [2] и ему соответствует КМУУ U_1 (рис. 1).

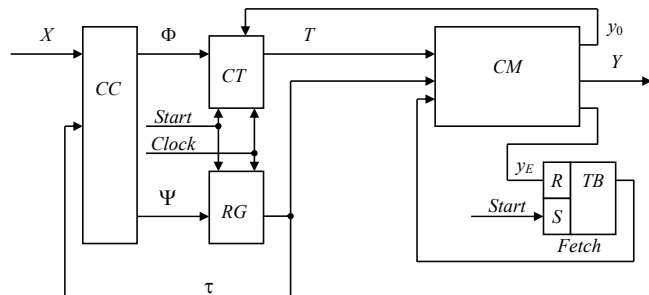


Рис. 1. Структурная схема КМУУ с разделением кодов

Здесь комбинационная схема CC реализует систему функций возбуждения счетчика CT

$$\Phi = \Phi(\tau, X) \quad (6)$$

и систему функций возбуждения регистра RG

$$\Psi = \Psi(\tau, X), \quad (7)$$

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

Устройство управления U_1 функционирует следующим образом. По сигналу $Start = 1$ в счетчик CT и регистр RG заносятся нулевые коды, что соответствует адресу $A(b_q)$, где $\langle b_0, b_q \rangle \in E$, одновременно триггер TF устанавливается в единичное состояние, и сигнал $Fetch$ разрешает выборку микрокоманд из CM . Очередная микрокоманда $Y(b_q)$ считывается из CM . Если $b_q \neq O_g$ ($g = \overline{1, G}$), то одновременно с микрооперациями Y формируется сигнал y_0 . При $y_0 = 1$ по сигналу $Clock$ к содержимому CT прибавляется единица, что соответствует адресации микрокоманд в пределах одной ОЛЦ, содержимое RG при этом не

меняется. Если $b_q = O_g(g = \overline{1, G})$, то сигнал y_0 не формируется и по сигналу $Clock$ в RG заносится код очередной ОЛЦ, определяемый функциями Ψ , а в CT заносится код компоненты, определяемый функциями Φ . При достижении адреса $A(b_i)$, где $\langle b_i, b_E \rangle \in E$, формируется сигнал y_E , триггер TF обнуляется и выборка микрокоманд из CM прекращается.

Основную сложность при реализации схемы КМУУ на $FPGA$ представляет нерегулярность функций Φ и Ψ . Функции называются нерегулярными, если они определены на менее чем 50% возможных входных наборов [5]. Для реализации нерегулярных функций используются LUT элементы, обладающие небольшим количеством входов (до пяти–шести) [4]. Это приводит к увеличению числа уровней в схеме CC и, как следствие, к уменьшению быстродействия КМУУ. Для устранения этого недостатка необходимо уменьшить как число нерегулярных функций, так и число аргументов в них. В статье предлагается метод решения этой задачи, основанный на оптимальном кодировании входов ОЛЦ.

Основная идея предлагаемого метода

Пусть $\Pi_C = \{B_1, \dots, B_l\}$ – разбиение множества $C' \subseteq C$ на классы псевдоэквивалентных ОЛЦ, под которыми понимаются ОЛЦ, выходы которых соединены со входом одной и той же вершины ГСА Γ [2]. При этом ОЛЦ $\alpha_g \in C'$, если $\langle O_g, b_E \rangle \notin E$. Поставим в соответствие каждому входу $I_g \in I(\Gamma)$ двоичный код $K(I_g^i)$ разрядности

$$R_3 = \lceil \log_2 M_o \rceil \quad (8)$$

и используем для кодирования переменные $v_r \in V$, где $|V| = R_3$, $M_o = |I(\Gamma)|$. Закодируем входы так, чтобы коды входов всех ОЛЦ $\alpha_g \in B_i$ составляли один обобщенный интервал пространства кодирования. Выполним кодирование так, чтобы эти интервалы определялись посредством

$$R_4 = \lceil \log_2 I \rceil \quad (9)$$

переменных $v_r \in V$. Полученные интервалы можно рассматривать как коды $K(B_i)$ классов $B_i \in \Pi_C$. Отметим, что для подобного кодирования можно использовать методы из [7] либо метод *ESPRESSO* [8].

Использование предложенных методов позволяет перейти к модели КМУУ U_2 (рис. 2).

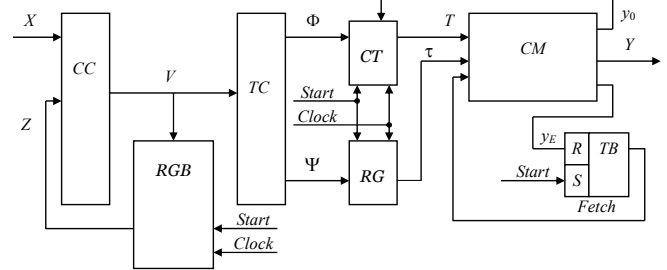


Рис. 2. Структурная схема КМУУ U_2

В КМУУ U_2 преобразователь кодов TC преобразовывает коды классов $K(I_g^i)$ в адреса входов, представленные в виде (5). При этом блок TC генерирует регулярные функции

$$\Phi = \Phi(V), \quad (10)$$

$$\Psi = \Psi(V). \quad (11)$$

Таким образом, схема TC реализуется на EMB . Регистр кодов классов RGB хранит коды $K(B_i)$, представляемые переменными $z_r \in Z$, где $|Z| = R_4$. Информация в RGB загружается, если $y_0 = 0$. Схема CC теперь реализует нерегулярные функции

$$V = V(Z, X). \quad (12)$$

Принцип функционирования КМУУ U_2 очевиден.

В силу выполнения условия

$$R_3 < R_1 + R_2, \quad (13)$$

число нерегулярных функций уменьшается, а при выполнении условия

$$R_4 < R_1 \quad (14)$$

уменьшается число аргументов в них (по сравнению с КМУУ U_1). При этом можно ожидать уменьшение числа как LUT элементов, так и уровней в схеме CC . Отметим, что блок TC может формировать адрес микрокоманды без

выделения кодов $K(\alpha_g)$ и $K(b_q)$. Однако при этом разрядность счетчика увеличивается до

$$R_0 = \lceil \log_2 |E_1| \rceil, \quad (15)$$

что приводит к увеличению времени такта КМУУ.

Модели U_1 и U_2 можно использовать только для ГСА Γ , для которой выполняется два условия:

$$|E_1|/G \geq 2, \quad (16)$$

$$R_1 + R_2 = R_0. \quad (17)$$

При выполнении (16) ГСА Γ является линейной, а при выполнении (17) КМУУ имеет такую же емкость схемы SM , как и эквивалентный автомат Мура [2].

В настоящей статье предлагается метод синтеза КМУУ U_2 , включающий следующие этапы.

- Формирование множеств C , C' и Π_C для ГСА Γ .

- Кодирование ОЛЦ $\alpha_g \in C$ и их компонент.
- Кодирование входов $I_g^j \in I(\Gamma)$.
- Формирование таблицы переходов КМУУ.
- Спецификация преобразователя кодов TC .
- Спецификация управляющей памяти КМУУ.
- Реализация схемы КМУУ в данном элементном базисе.

Пример применения предложенного метода

Пусть некоторая ГСА Γ_1 включает $|E_1| = 42$ операторные вершины, распределенные следующим образом по ОЛЦ $\alpha_g \in C$: $\alpha_1 = \langle b_1, \dots, b_7 \rangle$, $\alpha_2 = \langle b_8, \dots, b_{13} \rangle$, $\alpha_3 = \langle b_{14}, \dots, b_{21} \rangle$, $\alpha_4 = \langle b_{22}, \dots, b_{26} \rangle$, $\alpha_5 = \langle b_{27}, \dots, b_{32} \rangle$, $\alpha_6 = \langle b_{33}, b_{34}, b_{35} \rangle$, $\alpha_7 = \langle b_{36}, \dots, b_{39} \rangle$ и $\alpha_8 = \langle b_{40}, b_{41}, b_{42} \rangle$. Пусть $C' = \{\alpha_1, \dots, \alpha_6\}$ и $\Pi_C = \{B_1, B_2, B_3\}$, где $B_1 = \{\alpha_1, \alpha_2, \alpha_3\}$, $B_2 = \{\alpha_4, \alpha_5\}$, $B_3 = \{\alpha_6\}$. Пусть ОЛЦ $\alpha_g \in C$ имеет следующие входы: $I_1^1 = b_1$, $I_1^2 = b_4$, $I_1^3 = b_6$, $I_2^1 = b_8$, $I_2^2 = b_{10}$, $I_3^1 = b_{14}$, $I_3^2 = b_{18}$, $I_4^1 = b_{22}$, $I_4^2 = b_{23}$, $I_5^1 = b_{27}$, $I_5^2 = b_{29}$, $I_6^1 = b_{33}$, $I_6^2 = b_{35}$, $I_5^1 = b_{27}$ и $I_5^1 = b_{27}$. Очевидно, что множество $O(\Gamma_1)$ включает последние компоненты каждой из ОЛЦ $\alpha_g \in C$.

Итак, для ГСА Γ_1 имеем $R_0 = 6$, $R_1 = 3$, $F_{\max} = 8$, $R_2 = 3$, $G = 8$. Отношение $|E_1|/G \approx 5$, т.е. условие (16) выполняется и ГСА Γ_1 линейна. Условие (17) также удовлетворяется, и модель U_2 может быть использована. Закодируем ОЛЦ $\alpha_g \in C$ тривиальным образом: $K(\alpha_1) = 000, \dots, K(\alpha_8) = 111$. Компоненты ОЛЦ $\alpha_g \in C$ кодируются следующим образом: первая компонента имеет код 000, вторая – 001 и т.д. После кодирования можно найти адреса $A(b_q)$ микрокоманд, соответствующих вершинам $b_q \in B_1$ и представленных в виде (5). Например, вершина b_{10} – третья компонента ОЛЦ α_2 , следовательно $A(b_{10}) = K(\alpha_2) * K(b_{10}) = 001010$ и т.д.

Наиболее сложный – этап кодирования входов ОЛЦ. Для ГСА Γ_1 $M_0 = 15$ и $R_3 = 4$. Один из вариантов кодирования входов показан в карте Карно на рис. 3.

Из рис. 3 следует, что $K(B_1) = 0***$ (включает вершины от b_1 до b_{18}), $K(B_2) = 11**$ (вершины $b_{22} - b_{29}$) и $K(B_3) = 10**$ (вершины b_{33}, b_{35}). Отметим, что коды входов ОЛЦ $\alpha_7, \alpha_8 \in C$ рассматриваются как несущественные наборы. Итак, $R_4 = 2$ и число переменных в схеме СС КМУУ $U_2(\Gamma_1)$ на единицу меньше, чем в КМУУ $U_1(\Gamma_1)$. Здесь $U_i(\Gamma_j)$ означает, что модель U_i используется для интерпретации ГСА Γ_j . Кроме того, число нерегулярных функций уменьшается от $R_1 + R_2 = 6$ до $R_3 = 4$.

		$v_1 v_2$			
		00	01	11	10
$v_3 v_4$	00	b_1	b_{10}	b_{22}	b_{33}
	01	b_4	b_{14}	b_{23}	b_{35}
	11	b_6	b_{18}	b_{27}	b_{36}
	10	b_8	*	b_{29}	b_{40}

Рис. 3. Кодирование входов ОЛЦ ГСА Γ_1

Для формирования системы (12) необходимо построить таблицу переходов КМУУ U_2 , которая в свою очередь формируется по системе обобщенных формул перехода [2, 5]. Пусть для ГСА Γ_1 получена следующая система:

$$\begin{aligned}
B_1 &\rightarrow x_1 I_2^2 \vee x_1 x_2 I_4^1 \vee x_1 x_2 x_3 I_5^1 \vee x_1 x_2 x_3 I_6^2; \\
B_2 &\rightarrow x_3 x_4 I_1^1 \vee x_3 x_4 x_5 I_2^1 \vee x_3 x_4 x_5 I_3^1 \vee x_3 x_6 I_6^1; \\
B_3 &\rightarrow x_4 I_3^1 \vee x_4 x_5 I_4^2 \vee x_4 x_5 x_7 I_5^2 \vee x_4 x_5 x_7 x_8 I_7^2 \vee x_4 x_5 x_7 x_8 I_8^1.
\end{aligned} \quad (18)$$

Таблица переходов КМУУ имеет столбцы B_i , $K(B_i)$, I_g^j , $K(I_g^j)$, X_h , V_h , h , при этом в h -й строке в столбце V_h записаны функции $v_r \in V$, соответствующие единичным битам кода $K(I_g^j)$. Число строк H_0 равняется числу термов в системе вида (18). Фрагмент переходов КМУУ $U_2(\Gamma_1)$, соответствующий формуле для B_1 , приведен в таблице.

B_i	$K_A(B_i)$	I_g^i	$K(I_g^i)$	X_h	V_h	h
B_1	0***	I_2^2	0100	x_1	v_2	1
		I_4^1	1100	$\overline{x_1 x_2}$	$v_1 v_2$	2
		I_5^1	1111	$\overline{x_1 x_2 x_3}$	$v_1 v_2 v_3 v_4$	3
		I_6^2	1001	$\overline{x_1 x_2 x_3}$	$v_1 v_4$	4

Связь таблицы с системой (18) и рис. 3 очевидна. Используя таблицу, с учетом минимизации можно получить следующие фрагменты формул для функций (12):

$$\begin{aligned}
v_1 &= \overline{z_1 x_1}; \quad v_2 = \overline{z_1 x_1} \vee \overline{z_1 x_1 x_2} \vee \overline{z_1 x_1 x_2 x_3}; \\
v_3 &= \overline{z_1 x_1 x_2 x_3}; \quad v_4 = \overline{z_1 x_1 x_2}.
\end{aligned} \quad (19)$$

Для спецификации блока TC необходимо построить таблицу со столбцами $K(I_g^i)$, Φ_h , Ψ_h , h . Столбцы Φ_h и Ψ_h содержат коды ОЛЦ $\alpha_g \in C$ и компоненты $b_q \in D^g$, соответствующие входу I_g^i . Число строк в этой таблице равняется M_0 . Для спецификации блока CM необходимо построить таблицу, вход которой – адрес микрокоманды $Y(b_q)$, а выход – соответствующие микрооперации и переменные y_0 и y_E . Эти этапы выполняются по известной методике [2] и в статье не рассматриваются.

Для реализации схемы КМУУ используются стандартные пакеты [9, 10]. Отметим, что схемы CC , CT , RG , RGB и TF реализуются на LUT элементах, а блоки TC и CM – на EMB . Этот этап в статье не рассматривается.

Заключение. Предложенный метод оптимизации числа LUT элементов в схеме КМУУ с разделением кодов

позволяет уменьшить число нерегулярных функций и их аргументов в сравнении с базовой моделью КМУУ.

Исследования показали, что при выполнении условий (14), (16) и (17) число LUT элементов уменьшается в пределе на 32%, а число уровней – на три–четыре. При этом число блоков EMB в схеме КМУУ увеличивается только на один. Блоки EMB недостаточно эффективно используются современными пакетами, поэтому увеличение числа используемых EMB не критично.

Дальнейшее направление исследований – разработка эффективного алгоритма кодирования входов, позволяющего уменьшить число аргументов в выходных функциях схемы формирования кодов входов. Для этой цели необходимо модифицировать алгоритм *ESPRESSO*.

Для повышения эффективности метода необходимо разработать алгоритм адресации микрокоманд КМУУ, уменьшающий число ОЛЦ, адреса выходов которых следует преобразовывать. Это относится к дальнейшим исследованиям, как и проверка возможности использования метода для случая ПЛИС, а также в «системах–на–кристалле» [10].

1. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. – К.: ИК НАН Украины, 1997. – 156 с.
2. Barkalov A., Titarenko L. Logic Synthesis for Compositional Microprogram Control Units. – Berlin: Springer, 2008. – 272 p.
3. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб.: БХВ – Петербург, 2002. – 608 с.
4. Maxfield C. The Design Warrior's Guide to FPGAs. – Amsterdam: Elsevier, 2004. – 541 p.
5. Баркалов А.А., Титаренко Л.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС. – Донецк: Технопарк ДонНТУ УНИТЕХ, 2009. – 336 с.
6. Baranov S. Logic Synthesis for Control Automata. – Boston: Kluwer Acad. Publ., 1994. – 312 p.
7. Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д. Основы логического проектирования. Кн. 2. Оптимизация в булевом пространстве. – Минск: Объединенный ин-т проблем информатики, 2004. – 240 с.
8. DeMicheli G. Synthesis and Optimization of Digital Circuits. – New-York: McGraw-Hill, 1994. – 636 p.
9. <http://www.altera.com>
10. <http://www.xilinx.com>

Поступила 03.03.2010

Тел. для справок: (057) 301-07-23 (Донецк)

E-mail: A.Barkalov@iie.uz.zgora.pl

© А.А. Баркалов, Л.А. Титаренко, А.С. Лаврик, 2010