

УДК 681.234

А.А. БАРКАЛОВ¹, И.Я. ЗЕЛЕНЕВА², А.С. ЛАВРИК²¹University of Zielona Gora, Польша²Донецкий национальный технический университет, Украина

УМЕНЬШЕНИЕ АППАРАТУРНЫХ ЗАТРАТ В СХЕМЕ АДРЕСАЦИИ МИКРОКОМАНД УСТРОЙСТВА УПРАВЛЕНИЯ С РАЗДЕЛЕНИЕМ КОДОВ

В работе предложен метод уменьшения аппаратных затрат, ориентированный на композиционные микропрограммные устройства управления с разделением кодов, реализованные на микросхемах типа CPLD. Метод основан на большом коэффициенте объединения по входу макроячеек ПМЛ, что позволяет использовать более одного источника для кодов классов ОЛЦ. Приведен пример реализации предложенного метода.

Ключевые слова: композиционное микропрограммное устройство управления, псевдоэквивалентные операторные линейные цепи, разделение кодов.

Введение

Известно, что одной из важнейших частей любой цифровой системы является устройство управления (УУ) [1]. В случае представления алгоритма управления в виде линейной граф-схемы алгоритма (ГСА), можно использовать модель композиционного микропрограммного устройства управления (КМУУ) с разделением кодов [4]. В настоящее время для реализации логических схем устройств управления широко используются программируемые логические интегральные схемы (ПЛИС) типа CPLD (Complex Programmable Logic Devices) [6]. Такие ПЛИС включают в себя макроячейки программируемой матричной логики (ПМЛ), обладающие большим коэффициентом объединения по входу [3,4]. Высокая стоимость этого базиса продвигает актуальную задачу сокращения аппаратных затрат в схеме. При ее решении следует учитывать не только специфику интерпретируемого алгоритма управления, но и особенности элементного базиса. В таком случае, благодаря большому коэффициенту объединения по входу макроячеек ПМЛ, коды классов псевдоэквивалентных операторных линейных цепей (ОЛЦ) могут быть представлены более чем одним источником. В данной работе предлагается метод уменьшения аппаратных затрат в схеме КМУУ, основанный на использовании двух источников кодов.

1. Особенности КМУУ с разделением кодов

Пусть ГСА Γ представлена множеством вершин V и дуг E . Пусть $V = \{b_0, b_E\} \cup E_1 \cup E_2$, где

b_0 – начальная вершина, b_E – конечная вершина, E_1 – множество операторных вершин, где $|E_1| = M$, E_2 – множество условных вершин. Вершина $b_q \in E_1$ содержит набор микроопераций (МО) $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ – множество микроопераций [1]. Каждая вершина содержит элементы множества логических условий.

Сформируем набор операторных линейных цепей $C = \{\alpha_1, \dots, \alpha_G\}$ для ГСА Γ , где каждая ОЛЦ $\alpha_g \in C$ – это последовательность операторных вершин, причем каждой паре ее соседних вершин соответствует одна из дуг ГСА. Каждая ОЛЦ $\alpha_g \in C$ имеет только один выход O_g и произвольное количество входов. Определения ОЛЦ, их входов выходов могут быть найдены в [4].

Назовем ГСА Γ линейной ГСА (ЛГСА), если выполняется следующее условие:

$$\frac{M}{G} \geq 2. \quad (1)$$

Каждая вершина $b_q \in E_1$ соответствует микрооперации M_q , хранимой в управляющей памяти (УП) КМУУ по адресу A_q . Для адресации микрокоманд достаточно

$$R = \lceil \log_2 M \rceil \quad (2)$$

бит. Пусть ОЛЦ $\alpha_g \in C$ включает F_g компонент и пусть $Q = \max(F_1, \dots, F_G)$.

Закодируем ОЛЦ $\alpha_g \in C$ двоичными кодами

$K(\alpha_g)$ разрядности R_G , где

$$R_G = \lceil \log_2 G \rceil. \quad (3)$$

Закодируем каждый компонент ОЛЦ $\alpha_g \in C$ двоичными кодами $K(b_q)$ разрядности R_Q , где

$$R_Q = \lceil \log_2 Q \rceil. \quad (4)$$

Пусть компоненты закодированы таким образом, что

$$K(b_{g_{i+1}}) = K(b_{g_i}) + 1, \quad (5)$$

где $i = 1, \dots, F_g - 1, g = 1, \dots, G$. Пусть для кодирования компонент ОЛЦ используются переменные $T_r \in T$, а для кодирования ОЛЦ – переменные $\tau_r \in \tau$, где $|\tau| = R_G, |T| = R_Q$. В случае, если выполняется условие

$$R_G + R_Q = R, \quad (6)$$

ЛГСА Γ может быть представлена моделью КМУУ с разделением кодов U_1 (рис. 1).

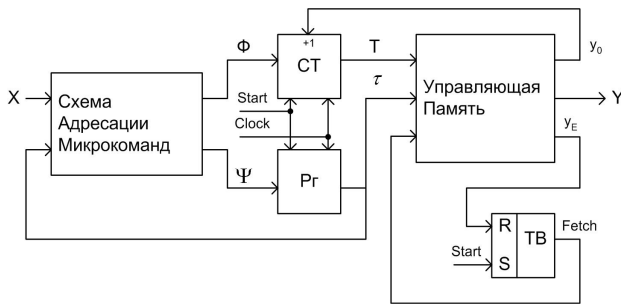


Рис. 1. Структурная схема КМУУ U_1

В КМУУ U_1 , схема адресации микрокоманд (САМ) реализует систему функций возбуждения счетчика СТ и регистра Рг:

$$\begin{aligned} \Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X). \end{aligned} \quad (7)$$

При этом адрес микрокоманды Ml_q представляется в виде:

$$A(b_q) = K(\alpha_g) * K(b_q), \quad (8)$$

где вершина b_q – это компонент ОЛЦ $\alpha_g \in C$, а “*” – знак операции конкатенации.

КМУУ U_1 работает следующим образом. По сигналу $Start = 1$ начальный адрес (все нули) заносится в Рг и СТ. В тоже время, триггер выборки ТВ устанавливается в единичное состояние. При этом $Fetch = 1$, что разрешает выборку микрокоманд из УП. Каждая ячейка УП содержит микрооперации $y_n \in Y$ и специальные переменные y_0 и y_E . Если $y_0 = 1$, то к текущему содержимому СТ прибавляется единица и адресуется следующий компонент текущей ОЛЦ. Иначе – значения СТ и Рг

загружаются из САМ. Первый случай соответствует переходу из любого компонента ОЛЦ кроме выхода. Второй случай соответствует переходу из выхода ОЛЦ.

Если $y_E = 1$, триггер выборки обнуляется, сигнал $Fetch = 0$ и работа КМУУ прекращается. Это соответствует переходу из вершины $b_q \in E_1$, где $\langle b_q, b_E \rangle \in E$.

Отметим, что ОЛЦ $\alpha_i, \alpha_j \in C$ являются псевдоэквивалентными ОЛЦ [4], если их выходы соединены с одной и той же вершиной ГСА Γ .

Аппаратурные затраты в логической схеме САМ могут быть уменьшены введением специального блока преобразователя кодов (ПК), преобразующего коды ОЛЦ в коды классов псевдоэквивалентных ОЛЦ [4]. Однако этот блок потребляет некоторые ресурсы микросхемы.

В данной статье предлагается использование свободных ячеек УП для представления кодов классов ПОЛЦ (первый источник), а также использование регистра Рг в качестве второго источника кодов ПОЛЦ.

2. Основная идея метода

Пусть $C_1 \subseteq C$ будет множеством ОЛЦ, где $\alpha_g \in C_1$, если ее выход не соединен с конечной вершиной b_E . Пусть $\Pi_C = \{V_1, \dots, V_1\}$ – разбиение множества C_1 на классы ПОЛЦ. Закодируем ОЛЦ $\alpha_g \in C_1$ таким образом, чтобы большинство классов $V_i \in \Pi_C$ было представлено одним интервалом R_G -мерного булева пространства. Для этого может быть использован широко известный алгоритм ESPRESSO [5]. Пусть $\Pi_C = \Pi_A \cup \Pi_B$, где $V_i \in \Pi_A$, если класс представлен одним интервалом, иначе $V_i \in \Pi_B$.

Если выполняется условие

$$\Pi_B = \emptyset, \quad (9)$$

то регистр Рг является источником всех кодов $K(V_i)$ для классов $V_i \in \Pi_C$. В этом случае САМ реализует H_0 термов, где H_0 – число переходов эквивалентного автомата Мили, что является минимально возможным числом переходов [4].

Отметим, что схемы блоков САМ, СТ, Рг и ТВ реализуются на макроячейках ПМЛ. Для реализации УП требуются внешние ППЗУ, имеющие t выходов, где $t \in \{1, 2, 4, 8, 16\}$ [2, 3]. При использовании унитарного кодирования микроопераций [6], каждое слово в УП состоит из $N+2$ бит. Число 2 добавляется

к N для учета дополнительных переменных y_0 и y_E (рис. 1).

Если каждое ППЗУ имеет t выходов и не менее M слов, то достаточно K_0 микросхем для реализации УП, где

$$K_0 = \left\lceil \frac{N+2}{t} \right\rceil. \quad (10)$$

Очевидно, что остается R_0 свободных выходов микросхем ППЗУ, где

$$R_0 = K_0 t - N - 2. \quad (11)$$

Эти выходы могут быть использованы для кодирования классов $V_i \in \Pi_B$, где

$$R_B = \left\lceil \log_2 (|\Pi_B| + 1) \right\rceil. \quad (12)$$

Единица добавляется к $|\Pi_B|$ для обозначения ситуации $V_i \notin \Pi_B$. Если выполняется условие

$$R_0 \geq R_B, \quad (13)$$

то все классы $V_i \in \Pi_B$ могут быть представлены с помощью УП. Иначе может быть представлено лишь I_{CM} классов, где

$$I_{CM} = 2^{R_0}. \quad (14)$$

Оставшиеся классы $V_i \in \Pi_B$ должны быть помещены в Π_A и представлены P_T . В обоих случаях, для интерпретации ЛГСА Γ предлагается КМУУ U_2 (рис. 2).

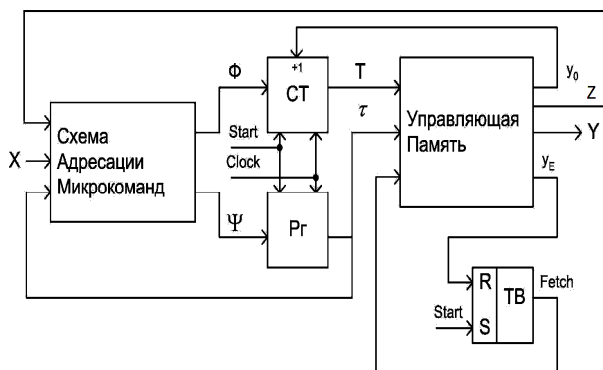


Рис. 2. Структурная схема КМУУ U_2

В КМУУ U_2 , блок САМ реализует функции:

$$\begin{aligned} \Phi &= \Phi(\tau, Z, X), \\ \Psi &= \Psi(\tau, Z, X), \end{aligned} \quad (15)$$

где переменные $\tau_i \in Z$ используются для кодирования классов $V_i \in \Pi_B$, $|Z| = R_0$. В зависимости от переменных τ и T управляющая память реализует функции Z, Y, y_0 и y_E .

Принципы действия КМУУ U_1 и U_2 практически идентичны.

В работе предлагается следующий алгоритм построения КМУУ U_2 :

1. Построение множеств C, C_1 и Π_C для ЛГСА Γ .
2. Кодирование ОЛЦ $\alpha_g \in C$ и их компонентов.
3. Построение разбиений Π_A и Π_B .
4. Кодирование классов $V_i \in \Pi_B$.
5. Построение таблицы переходов КМУУ.
6. Спецификация управляющей памяти.
7. Реализация схемы КМУУ в данном элементном базисе.

3. Пример применения метода

Пусть для некоторой ГСА Γ_1 сформированы множества

$$C = \{\alpha_1, \dots, \alpha_8\}, C_1 = \{\alpha_1, \dots, \alpha_7\}, \Pi_C = \{V_1, \dots, V_4\},$$

где $V_1 = \{\alpha_1\}$, $V_2 = \{\alpha_2, \alpha_3, \alpha_4\}$, $V_3 = \{\alpha_5, \alpha_6\}$, $V_4 = \{\alpha_7\}$,

при этом

$$\alpha_1 = \langle b_1, b_2, b_3 \rangle, \alpha_2 = \langle b_4, b_5, b_6, b_7 \rangle,$$

$$\alpha_7 = \langle b_{20}, b_{21} \rangle, \alpha_3 = \langle b_8, b_9, b_{10} \rangle, \alpha_4 = \langle b_{11}, \dots, b_{14} \rangle,$$

$$\alpha_5 = \langle b_{15}, b_{16} \rangle, \alpha_6 = \langle b_{17}, b_{18}, b_{19} \rangle, \alpha_8 = \langle b_{22}, b_{23}, b_{24} \rangle.$$

Итак, $G = 8$, $R_G = 3$, $Q = 4$, $R_Q = 2$, $M = 24$,

$R = 5$. Таким образом, условие (6) выполняется и метод разделения кодов может быть применен. Отметим, что условие (1) выполняется, следовательно, ГСА Γ_1 является линейной.

Закодируем ОЛЦ $\alpha_g \in C$ как показано на рис. 3, используя переменные $\tau_i \in \tau$, где $|\tau| = 3$.

	$\tau_2 \tau_3$			
τ_1	00	01	11	10
0	α_1	α_2	α_3	α_4
1	α_5	α_6	α_7	α_8

Рис. 3. Коды ОЛЦ для ГСА Γ_1 .

Компоненты ОЛЦ $\alpha_g \in C$ закодированы тривиально [4]: первый компонент имеет код 00, второй – 01 и так далее, для соответствия условию (5). Переменные T_1, T_2 используются для кодирования компонентов.

Из рис. 3 видно, что классы B_1, B_3 и B_4 представлены одним интервалом булева пространства. Эти интервалы представлены кодами $K(B_i)$: $K(B_1) = 000$, $K(B_3) = 10^*$, $K(B_4) = 111$. Класс $B_2 \in \Pi_C$ представлен интервалами 00^* and 01^* . Этот анализ показывает, что $\Pi_A = \{B_1, B_3, B_4\}$ и $\Pi_B = \{B_2\}$ ($R_B = 1$).

Пусть для ГСА Γ_1 $N=13$ и пусть для реализации УП используются микросхемы ППЗУ с $t=4$. В таком случае $K_0=4$, $R_0=1$ и условие (13) выполняется. Переменная $z_1 \in Z$ используется для кодирования класса B_2 .

Пусть $K(B_2)=1$, где $z_1=0$ показывает ситуацию $B_i \in \Pi_B$.

Пусть переходы для классов $B_i \in \Pi_C$ задаются следующей системой обобщенных формул перехода (ОФП) [4]:

$$\begin{aligned} B_1 &\rightarrow x_1 b_4 \vee \overline{x_1} x_2 b_6 \vee \overline{x_1} x_2 x_3 b_8 \vee \overline{x_1} x_2 x_3 b_{14}; \\ B_2 &\rightarrow x_2 b_{15} \vee x_2 x_3 b_{17} \vee x_2 x_3 x_4 b_{20} \vee x_2 x_3 x_4; \\ B_3 &\rightarrow x_5 b_{19} \vee \overline{x_5} b_{11}; \\ B_4 &\rightarrow x_1 b_{17} \vee \overline{x_1} b_{22}; \end{aligned} \quad (16)$$

Эта система является основой для построения таблицы переходов, имеющей следующие столбцы: B_i , $K_A(B_i)$, $K_B(B_i)$, b_q , $A(b_q)$, X_h , Ψ_h , Φ_h , h . Здесь $K_A(B_i)$ – это код $K(B_i)$, где $B_i \in \Pi_A$; $K_B(B_i)$ – код для $B_i \in \Pi_B$; X_h – входной сигнал получаемый из ОФП; Ψ_h – набор выходных функций загружающих код $K(\alpha_g)$ в Рг; Φ_h – набор выходных функций загружающих код $K(b_q)$ в СТ; $h=1, \dots, N_0$ номер перехода. В этом случае $\Psi = \{D_1, D_2, D_3\}$, $\Phi = \{D_4, D_5\}$. Определим несколько адресов микроопераций. Например, вершина b_{20} является первым компонентом ОЛЦ $\alpha_7 \in C$. Следовательно, $A(b_{20})=11100$. По аналогии мы можем найти $A(b_{11})=01000$, $A(b_{15})=10000$, $A(b_{15})=10100$ и $A(b_{19})=10110$.

Пусть символ $U_i(\Gamma_j)$ используется для обозначения интерпретации ГСА Γ_j с помощью модели КМУУ U_i ($i=1, 2$).

Часть таблицы переходов для классов $B_2, B_3 \in \Pi_C$ показана в табл. 1.

Таблица 1
Часть таблицы переходов для КМУУ $U_2(\Gamma_1)$

B_i	$K_A(B_i)$	$K_B(B_i)$	b_q	$A(b_q)$	X_h	Ψ_h	Φ_h	h
	$\tau_1 \tau_2 \tau_3$	z_1						
B_1	***	1	b_{15}	10000	x_2	D_1	-	1
			b_{17}	10100	$\overline{x_2} x_3$	$D_1 D_3$	-	2
			b_{20}	11100	$\overline{x_2} x_3 x_4$	$D_1 D_2 D_3$	-	3
B_2	10*	0	b_{11}	01000	$\overline{x_2} x_3 x_4$	D_2	-	4
			b_{19}	10110	x_5	$D_1 D_3$	D_4	5
			b_{11}	01000	$\overline{x_5}$	D_2	-	6

Эта таблица используется для построения системы (15). Например, следующие выражения получены из табл. 1:

$$\begin{aligned} D_2 &= z_1 \overline{x_2} x_3 \vee \tau_1 \overline{\tau_2} z_1 x_5 \quad (\text{после минимизации}); \\ D_4 &= \tau_1 \tau_2 z_1 x_5. \end{aligned}$$

Управляющая память КМУУ $U_2(\Gamma_1)$ специфицируется используя широко известные методы [4]. Отметим, что $N_0=12$ что определено общим количеством термов в системе (16). В случае КМУУ $U_1(\Gamma_1)$ имеем $N=20$. Таким образом применение предложенного метода позволяет сократить это значение в 1.67 раз. Ожидается что уменьшение количества макроячеек ПМЛ в логической схеме блока САМ будет иметь тот же порядок [4].

Заключение

Предложенный метод направлен на уменьшение числа макроячеек ПМЛ в схеме адресации микрокоманд КМУУ с разделением кодов. Это возможно благодаря таким главным факторам как большой коэффициент объединения по входу современных макроячеек ПМЛ, а также естественная избыточность микросхем ППЗУ, обусловленная тем, что количество их выходов ограничено определенным множеством чисел; существование классов псевдоэквивалентных ОЛЦ.

Исследования показали, что количество макроячеек уменьшается до 12% для КМУУ $U_2(\Gamma_1)$ в сравнении с эквивалентным КМУУ $U_1(\Gamma_1)$. Стоит отметить, что данный метод применим лишь для интерпретации линейных ГСА при условии выполнения (6).

Дальнейшие направления наших исследований связаны с применением предложенного подхода для реализации КМУУ на микросхемах FPGA [5].

Литература

1. Baranov S. *Logic Synthesis for Control Automata* / S. Baranov. – Kluwer Academic Publishers, 1994. – 312 p.
2. *Altera devices overview* [Електрон. ресурс]. – Режим доступу к ресурсу: <http://www.altera.com>.
3. *Xilinx CPLDs* [Електрон. ресурс]. – Режим доступу к ресурсу: <http://www.xilinx.com>.
4. Barkalov A. *Logic Synthesis for Compositional Microprogram Control Units* / A. Barkalov, L. Titarenko. – Berlin: Springer, 2008. – 272 p.
5. Maxfield C. *The Design Warrior's Guide to FPGAs* / C. Maxfield. – Amsterdam: Elsevier, 2004. – 541 p.
6. Соловьев В.В. *Проектирование цифровых схем на основе программируемых логических интегральных схем* / В.В. Соловьев. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.

Поступила в редакцию 1.02.2010

Рецензент: д-р техн. наук, проф., проф. кафедры АСУ Ю.А. Скобцов, Донецкий национальный технический университет «ДонНТУ», Донецк, Украина.

ЗМЕНШЕННЯ АПАРАТУРНИХ ВИТРАТ У СХЕМІ АДРЕСАЦІЇ МІКРОКОМАНД ПРИБРОЮ КЕРУВАННЯ З РОЗДІЛЕННЯМ КОДІВ

О.О. Баркалов, І.Я. Зеленьова, О.С. Лаврик

В роботі запропоновано метод зменшення апаратних витрат, орієнтований на композиційні мікропрограмні пристрої керування з розділенням кодів, що реалізовані на мікросхемах типу CPLD. Метод базується на великому коефіцієнті об'єднання за входом у макроосередків ПМЛ, що дозволяє використовувати більше ніж одне джерело для кодів класів ОЛЛ. Наведено приклад використання запропонованого методу, а також результати експериментів.

Ключові слова: композиційний мікропрограмний пристрій керування, псевдоеквівалентний операторний лінійний ланцюг, розділення кодів.

REDUCTION OF ADDRESSING CIRCUIT HARDWARE AMMOUNT FOR CONTROL UNIT WITH CODE SHARING

A.A. Barcalov, I.Y. Zelenyova, A.S. Lavrik

The method of hardware reduction is proposed which is oriented on compositional microprogram control units with code sharing and PAL-based CPLD chips. The method is based on a wide fan-in of PAL macrocells allowing using more than one source for codes of operational linear chains. An example of proposed method application is given.

Keywords: compositional microprogram control unit, pseudoequivalent operational linear chain, code sharing.

Баркалов Александр Александрович – д-р техн. наук, проф., проф. кафедри комп'ютерної інженерії, Донецкий национальный технический университет «ДонНТУ», Украина, проф. Университета Зеленогурского, Польша, e-mail: A.Barkalov@iie.uz.zgora.pl.

Зеленева Ирина Яковлевна – канд. техн. наук, доц., доц. кафедри комп'ютерної інженерії, Донецкий национальный технический университет «ДонНТУ», Донецк, Украина.

Лаврик Александр Сергеевич – ассистент кафедры комп'ютерної інженерії, Донецкий национальный технический университет «ДонНТУ», Донецк, Украина.