

ОПТИМИЗАЦИЯ КОМПОЗИЦИОННОГО МИКРОПРОГРАММНОГО УСТРОЙСТВА УПРАВЛЕНИЯ С ЭЛЕМЕНТАРНЫМИ ОПЕРАТОРНЫМИ ЛИНЕЙНЫМИ ЦЕПЯМИ

Alexander Barkalov, Remigiusz Wiśniewski, Roman Babakov
University Zelenogursky Institution of Informatics and Electronics, Poland
Донецкий национальный технический университет, кафедра ЭВМ
E-mail: A.Barkalov@iie.uz.zgora.pl

Abstract

Barkalov A., Wiśniewski R., Babakov R. Optimization of compositional microprogram control unit with elementary operational linear chains. The method of optimization of hardware amount of compositional microprogram control unit is proposed. The method is based on partition of the set of elementary operational linear chains by the classes of pseudoequivalent operational linear chains. The method of design and its application have been given.

Введение

Устройство управления цифровой системы может быть реализовано как композиционное микропрограммное устройство управления (КМУУ) [1], включающее микропрограммный автомат (МПА) адресации микрокоманд и микропрограммное устройство управления с естественной адресацией микрокоманд. В настоящее время для реализации логических схем устройств управления широко применяются программируемые логические устройства (ПЛУ) [2, 3]. Этот базис обладает значительной стоимостью и поэтому актуальной является задача уменьшения стоимости реализации схемы устройства управления. Одним из методов решения этой задачи является применение принципа разделения кодов [4] в КМУУ. Однако этот принцип применим только в том случае, если при этом не увеличивается число разрядов в адресе микрокоманды. В противном случае это приводит к увеличению объема управляющей памяти и уменьшению эффективности ее использования. В статье предлагается метод проектирования КМУУ с разделением кодов, позволяющий экономить объем управляющей памяти. Предлагаемый метод ориентирован на использование элементарных операторных линейных цепей [1] исходной граф-схемы алгоритма.

Основные определения и идея метода

Пусть граф-схема алгоритма Γ содержит Q операторных вершин, принадлежащих множеству $B = \{b_1, \dots, b_Q\}$ и пусть E – множество дуг ГСА. Введем некоторые определения, необходимые для будущего.

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

Определение 2. Вершина $b_q \in B$ называется входом ОЛЦ α_g , если существует дуга $\langle b_i, b_q \rangle \in E$, где b_i – начальная или условная вершина граф-схемы алгоритма Γ или операторная вершина, не входящая в ОЛЦ α_g .

Определение 3. Операторная линейная цепь α_g называется элементарной ОЛЦ, если она имеет только один вход.

Определение 4. Вершина $b_q \in B$ называется выходом ОЛЦ α_g , если существует дуга $\langle b_q, b_i \rangle \in E$, где b_i - условная или конечная вершина граф-схемы алгоритма Γ или операторная вершина, не входящая в ОЛЦ α_g .

Обозначим адрес микрокоманды, соответствующей вершине $b_i \in B$ как $A(b_i)$. Представим адрес $A(b_i)$ в виде конкатенации

$$A(b_i) = K(\alpha_g) * K(b_i) \tag{1}$$

где $K(\alpha_g)$ - код ОЛЦ $\alpha_g \in C = \{\alpha_1, \dots, \alpha_G\}$ и $K(b_i)$ - код компоненты ОЛЦ $\alpha_g \in C$, соответствующей вершине $b_i \in B$. Представление (1) называется представлением с разделением кодов[4]. Пусть все ОЛЦ $\alpha_g \in C$ являются элементарными ОЛЦ. В этом случае устройство управления может быть реализовано как КМУУ U_1 (Рис. 1).

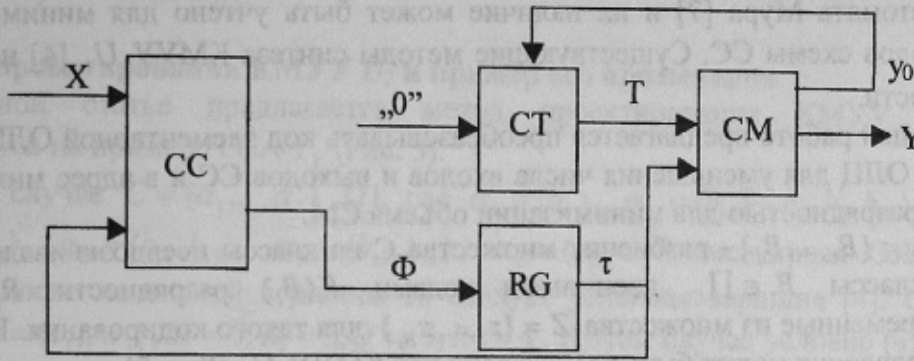


Рис. 1. Структурная схема композиционного микропрограммного устройства управления U_1

Здесь комбинационная схема CC формирует функции возбуждения регистра памяти RG

$$\Phi = \Phi(\tau, X) \tag{2}$$

где $\tau = \{\tau_1, \dots, \tau_R\}$ - множество внутренних переменных для кодирования элементарных ОЛЦ $\alpha_g \in C$, $R = \lceil \log_2 G \rceil$, $X = \{x_1, \dots, x_L\}$ - множество логических условий из условных вершин исходной граф-схемы алгоритма Γ . Счетчик CT хранит код $K(b_i)$ компоненты, имеющий $R_1 = \lceil \log_2 M \rceil$ разрядов, где M - максимальное число компонент в элементарной ОЛЦ $\alpha_g \in C$. Если $b_q = pr_i \alpha_g$, $b_i = pr_{in} \alpha_g$, где $i \in \{1, \dots, F_g - 1\}$, то содержимое счетчика CT увеличивается на единицу по сигналу $y_0 = 1$. Это соответствует режиму естественно адресации микрокоманд [5]:

$$K(b_i) = K(b_q) + 1 \tag{3}$$

Сигнал y_0 и микрооперации $y_n \in Y = \{y_1, \dots, y_N\}$ формируются управляющей памятью СМ, которая реализуется на ППЗУ[5]. Если выход элементарной ОЛЦ $\alpha_g \in C$ достигнут, то сигнал y_0 не формируется и автомата адресации, состоящий из схемы СС и RG, загружает код следующей элементарной ОЛЦ в регистр RG. В этом случае счетчик СТ обнуляется, что соответствует коду первой компоненты очередной элементарной ОЛЦ. Код компоненты $K(b_i)$ представляется внутренними переменными $T_r \in T = \{T_1, \dots, T_R\}$.

Такой подход имеет смысл при выполнении условия

$$R + R_1 = R_2 \tag{4}$$

Здесь $R_2 = \lceil \log_2 Q \rceil$ - минимальное число разрядов, достаточное для адресации микрокоманд. Если условие (4) нарушается, то это приводит к резкому увеличению числа БИС в управляющей памяти.

Определение 5. Элементарные ОЛЦ $\alpha_i, \alpha_g \in C$ называются псевдоэквивалентными ОЛЦ, если формулы переходов [6] для их выходов совпадают.

Псевдоэквивалентные элементарные ОЛЦ соответствуют псевдоэквивалентным состояниям автомата Мура [7] и их наличие может быть учтено для минимизации числа входов и выходов схемы СС. Существующие методы синтеза КМУУ U_1 [4] не используют этой возможности.

В настоящей работе предлагается преобразовывать код элементарной ОЛЦ в код блока элементарных ОЛЦ для уменьшения числа входов и выходов СС и в адрес микрокоманды с минимальной разрядностью для минимизации объема СМ.

Пусть $\Pi_C = \{B_1, \dots, B_l\}$ - разбиение множества C на классы псевдоэквивалентных ОЛЦ. Закодируем классы $B_i \in \Pi_C$ двоичными кодами $K(B_i)$ разрядности $R_3 = \lceil \log_2 l \rceil$ и используем переменные из множества $Z = \{z_1, \dots, z_{R_3}\}$ для такого кодирования. В этом случае устройство управления может быть реализовано как КМУУ U_2 (Рис. 2).

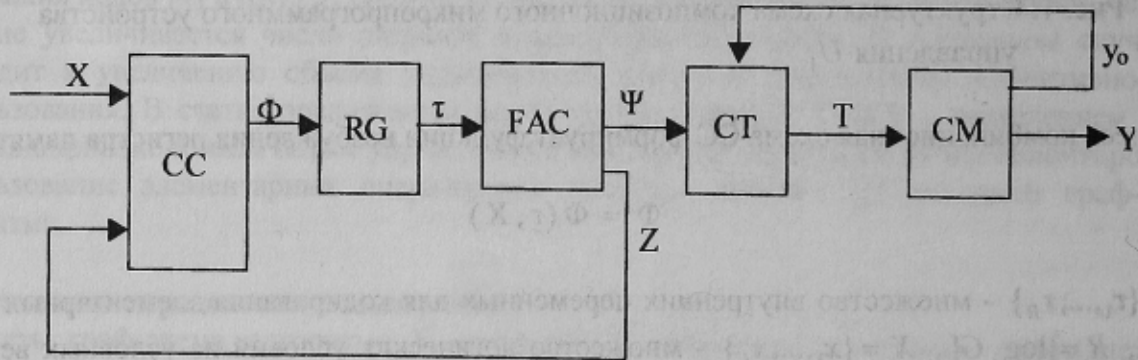


Рис. 2. Структурная схема композиционного микропрограммного устройства управления U_2

Здесь схема СС формирует систему функций

$$\Phi = \Phi(Z, X) \tag{5}$$

для загрузки кода $K(\alpha_g)$ в регистр RG. Если условие

$$R_3 < R \tag{6}$$

выполняется, то система (5) имеет меньше входов и выходов, чем система (2). Схема формирования адреса микрокоманды и кода блока псевдоэквивалентных элементарных ОЛЦ (FAC) преобразовывает код $K(\alpha_g)$ в код блока, где $\alpha_g \in B_i$, и в адрес $A(b_q)$, где $b_q = pr_1 \alpha_g$. Режим адресации (3) теперь реализуется как режим

$$A(b_i) = A(b_q) + 1 \tag{7}$$

где $b_q = pr_1 \alpha_g$, $b_i = pr_{in} \alpha_g$, $i \in \{1, \dots, F_g - 1\}$, $g=1, \dots, G$.

Схема FAC реализует функции

$$\Psi = \Psi(\tau) \tag{8}$$

$$Z = Z(\tau) \tag{9}$$

которые могут быть реализованы с использованием ППЗУ.

Очевидно, что КМУУ U_2 имеет меньшую производительность, чем КМУУ U_1 и его применение имеет смысл только в том случае, если критерием оптимальности схемы КМУУ является минимум стоимости.

Метод проектирования КМУУ U_2 и пример его применения

В данной статье предлагается метод проектирования КМУУ U_2 , который иллюстрируется на примере ГСА Γ_1 (Рис. 3).

В этом случае $C = \{\alpha_1, \dots, \alpha_7\}$ [1], где $\alpha_1 = \langle b_0 \rangle$, $\alpha_2 = \langle b_1, b_4, b_7, b_9, b_{10} \rangle$, $\alpha_3 = \langle b_2, b_5 \rangle$, $\alpha_4 = \langle b_3, b_6, b_8 \rangle$, $\alpha_5 = \langle b_{11}, b_{13} \rangle$, $\alpha_6 = \langle b_{12}, b_{14} \rangle$, $\alpha_7 = \langle b_{15} \rangle$. Элементарная ОЛЦ α_1 введена потому, что после начальной вершины b_0 следует условная вершина [4]. Следовательно, $G = 7, R = 3, \tau = \{\tau_1, \dots, \tau_3\}, M = 5, R_1 = 3, Q = 15, R_2 = 4$. В этом случае условие (4) нарушается и применение КМУУ U_2 имеет смысл.

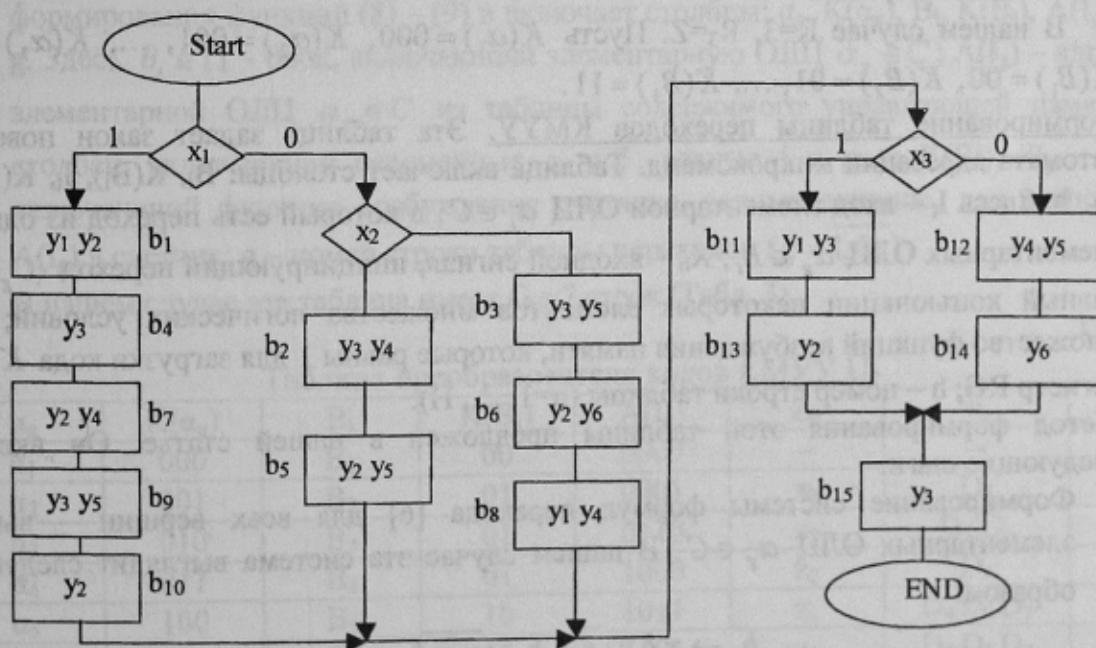


Рис.3. Исходная граф-схема алгоритма Γ_1

Предложенный метод синтеза включает следующие шаги:

1. Естественная адресация микрокоманд и формирование содержимого управляющей памяти. Этот шаг выполняется, используя подход [1]. Для нашего примера содержимое СМ приведено в Табл. 1.

Таблица 1

Содержимое управляющей памяти КМУУ U_2

Address	Microoperations	Comment	Address	Microoperations	Comment
0000	----	$b_0 I_1$	1000	$y_0 y_3 y_5$	$b_3 I_4$
0001	$y_0 y_1 y_2$	$b_1 I_2$	1001	$y_0 y_2 y_6$	b_6
0010	y_3	b_4	1010	$y_1 y_4$	$b_8 O_4$
0011	$y_0 y_2 y_4$	b_7	1011	$y_0 y_1 y_3$	$b_{11} I_5$
0100	$y_0 y_3 y_5$	b_9	1100	y_2	$b_{13} O_5$
0101	y_2	$b_{10} O_2$	1101	$y_0 y_4 y_5$	$b_{12} I_6$
0110	$y_0 y_3 y_4$	$b_2 I_3$	1110	y_6	$b_{14} O_6$
0111	$y_2 y_5$	$b_5 O_3$	1111	$y_3 y_к$	$b_{15} I_7 O_7$

Здесь I_g, O_g означает вход и выход элементарной ОЛЦ $\alpha \in C$ соответственно, сигнал y_0 добавляется во все вершины, которые не являются выходами элементарных ОЛЦ. Для организации режима останова сигнал y_k вводится в вершину b_{15} , связанную с конечной вершиной b_k .

2. Формирование разбиения Π_C . Данное разбиение формируется тривиальным образом по исходной граф-схеме алгоритма Г. Один блок разбиения Π_C включает элементарные ОЛЦ, выходы которых связаны с входом одной и той же вершины граф-схемы алгоритма Г.

В нашем случае $\Pi_C = \{B_1, \dots, B_4\}$ где $B_1 = \{\alpha_1\}$, $B_2 = \{\alpha_2, \alpha_3, \alpha_4\}$, $B_3 = \{\alpha_5, \alpha_6\}$, $B_4 = \{\alpha_7\}$. Следовательно, $I=4, R_3=Z$.

3. Кодирование элементарных ОЛЦ и их классов. Кодирование выполняется тривиальным образом.

В нашем случае $R=3, R_3=Z$. Пусть $K(\alpha_1) = 000, K(\alpha_2) = 001, \dots, K(\alpha_7) = 110, K(B_1) = 00, K(B_2) = 01, \dots, K(B_4) = 11$.

4. Формирование таблицы переходов КМУУ. Эта таблица задает закон поведения автомата адресации микрокоманд. Таблица включает столбцы: $B_i, K(B_i), I_i, K(I_i), X_h, \Phi_h, h$. Здесь I_i – вход элементарной ОЛЦ $\alpha_i \in C$, в который есть переход из одной из элементарных ОЛЦ $\alpha_g \in B_i$; X_h – входной сигнал, инициирующий переход $\langle O_g, I_i \rangle$ и равный конъюнкции некоторых элементов множества логических условий; Φ_h – множество функций возбуждения памяти, которые равны 1 для загрузки кода $K(\alpha_i)$ в регистр RG; h – номер строки таблицы ($h=1, \dots, H$).

Метод формирования этой таблицы предложен в нашей статье. Он включает следующие шаги:

- Формирование системы формул перехода [6] для всех вершин – выходов элементарных ОЛЦ $\alpha_g \in C$. В нашем случае эта система выглядит следующим образом:

$$b_0 \rightarrow x_1 b_1 \vee \overline{x_1} x_2 b_2 \vee \overline{x_1} \overline{x_2} b_3;$$

$$b_5, b_8, b_{10} \rightarrow x_3 b_{11} \vee \overline{x_3} b_{12};$$

$$b_{13}, b_{14} \rightarrow b_{15}; b_{15} \rightarrow b_k.$$

- Замена вершин в левой части каждой формулы перехода обозначением класса $B_i \in \Pi_C$, такого что $\alpha_g \in B_i$, и замена вершин в правой части формул обозначениями соответствующих входов.

Для нашего примера формируется следующая преобразованная система формул перехода:

$$\begin{aligned} B_1 &\rightarrow x_1 I_2 \vee \overline{x_1} x_2 I_3 \vee \overline{x_1} \overline{x_2} I_4 \\ B_2 &\rightarrow x_3 I_5 \vee \overline{x_3} I_6 \\ B_3 &\rightarrow I_7, \quad B_4 \rightarrow b_K \end{aligned} \quad (10)$$

- Тривиальное преобразование системы типа система (10) в таблицу переходов КМУУ U_2 .

В нашем случае эта таблица имеет $N=6$ строк.

Эта таблица не включает переход $B_y \rightarrow b_K$, так как он формируется автоматически при $u_K=1$.

Таблица 2

Таблица переходов КМУУ U_2

B_i	$K(B_i)$	I_t	$K(\alpha_t)$	X_h	Φ_h	h
B_1	00	I_2	001	x_1	D_3	1
		I_3	010	$\overline{x_1} x_2$	D_2	2
		I_4	011	$\overline{x_1} \overline{x_2}$	$D_2 D_3$	3
B_2	01	I_5	100	x_3	D_1	4
		I_6	101	$\overline{x_3}$	$D_1 D_3$	5
B_3	10	I_7	110	1	$D_1 D_2$	6

- Формирование таблицы преобразователя кодов. Эта таблица является основой для формирования функций (8) – (9) и включает столбцы: $\alpha_g, K(\alpha_g), B_i, K(B_i), A(I_g), Z_g, \Psi_g, g$. Здесь $B_i \in \Pi$ - блок, включающий элементарную ОЛЦ $\alpha_g \in C$; $A(I_g)$ – адрес входа элементарной ОЛЦ $\alpha_g \in C$ из таблицы содержимого управляющей памяти; Z_g – столбец, включающий переменные $z_r \in Z$, равные 1 в коде $K(B_i)$; Ψ_g – столбец, включающий функции возбуждения счетчика, равные единице, для записи адреса $A(I_g)$ в счетчик; g – номер строки таблицы переходов ($g = \overline{1, G_0}$).

В нашем случае эта таблица имеет $G_0=7$ строк (Табл. 3).

Таблица 3

Таблица преобразователя кодов КМУУ U_2

α_g	$K(\alpha_g)$	B_i	$K(B_i)$	$A(I_g)$	Z_g	Ψ_g	g
α_1	000	B_1	00	0000	--	--	1
α_2	001	B_2	01	0001	z_2	D_7	2
α_3	010	B_3	01	0110	z_2	$D_5 D_6$	3
α_4	011	B_4	01	1000	z_2	D_4	4
α_5	100	B_5	10	1011	z_1	$D_4 D_6 D_7$	5
α_6	101	B_6	10	1101	z_1	$D_4 D_5 D_7$	6
α_7	110	B_7	11	1111	$z_1 z_2$	$D_4 D_5 D_6 D_7$	7

Отметим, что функции из столбца Ψ_g имеют индексы от $from R+1=4$ до $R+R_1=7$.

6. Синтез функциональной схемы КМУУ U_2 . Схема СС реализуется на ПЛУ с использованием системы (5), которая формируется по таблице переходов. Эта система представляется в следующем виде:

$$\varphi_{\Gamma} = \bigvee_{h=1}^H C_{\Gamma h} V_h X_h \quad (\Gamma = 1, \dots, R) \quad (11)$$

Здесь $C_{\Gamma h}$ – булева переменная, равная 1, если и только если функция $\varphi_{\Gamma}=1$ в h -ой строке таблицы, V_h – конъюнкция переменных $z_{\Gamma} \in Z$, соответствующая коду $K(B_i)$ класса $B_i \in \Pi_C$ из h -ой строки таблицы ($h=1, \dots, H$).

В нашем случае из табл.2 мы имеем, например, $D_1 = \overline{z_1 z_2 x_3} \vee \overline{z_1 z_2} = \overline{z_1 z_2} \vee \overline{z_1 z_2 x_3}$.

Схема FАС реализуется, используя ППЗУ и таблицу преобразователя кодов, схема СМ реализуется, используя ППЗУ и таблицу содержимого управляющей памяти. Проблемы, связанные с проектированием подобных схем, рассмотрены в работах [2, 3] и в данной статье не рассматриваются.

Заключение

Применение предложенного метода преобразования кода элементарной операторной линейной цепи в код класса псевдоэквивалентных элементарных ОЛЦ и адрес входа элементарной ОЛЦ позволяет применять метод разделения кодов для произвольной граф-схемы алгоритма. Эффективность метода возрастает по мере роста отношения суммарной разрядности кода элементарной ОЛЦ и кода ее компоненты к минимальной разрядности адреса микрокоманды и по мере уменьшения числа классов псевдоэквивалентных элементарных ОЛЦ в исходной граф-схеме алгоритма. Применение этого метода позволяет минимизировать количество входов, выходов и термов микропрограммного автомата Мили, выполняющего адресацию микрокоманд, по сравнению со всеми известными методами реализации композиционных микропрограммных устройств управления. Необходимо отметить, что использование преобразователя кодов увеличивает время цикла КМУУ и предлагаемый метод применим только в случае, если критерием эффективности проектируемой схемы является минимум аппаратных затрат. Исследования авторов показали, что выигрыш в стоимости может достигать 20-25% по сравнению с известными методами проектирования КМУУ.

Литература

1. Баркалов А. А., Палагин А.В. Синтез микропрограммных устройств управления. – Киев: ИК НАН Украины, 1997.- 136 с.
2. Соловьев В.В. Проектирование цифровых систем с использованием программируемых логических интегральных схем. – Москва: Горячая линия - телеком, 2001. – 636 с.
3. Грушницкий Р.И., Мурзаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – Петербург:БХВ– 2002. – 636 с.
4. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах. – Донецк: ДНТУ, 2002. – 262 с.
5. Баркалов А.А. Синтез операционных устройств. – Донецк: ДНТУ, 2003. – 306 с.
6. Baranov S. Logic Synthesis for Control Automata. – Kluwer Academic Publishers, 1994. – 312 p.
7. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура// Кибернетика и системный анализ. – 1998, №1. – С. 65-72 .