

Синтез устройства управления с разделением кодов и модификацией операторных линейных цепей

Баркалов А.А., Красичков А.А., Мирошкин А.Н.
Донецкий национальный технический университет, каф. ЭВМ
A.Barkalov@iie.uz.zgora.pl

Abstract

Barkalov A.A., Krasichkov A.A., Miroshkin A.N. Syntheses of control unit with code sharing and modification of operational linear chains. The method of design for compositional microprogram control units with code sharing is proposed. The method is oriented on reduction in the number of PAL macrocells in the combinational part of control unit. The method is based on modification of each operational linear chains by some additional control microinstructions, which contain codes of the classes of pseudoequivalent chains. Proposed method is illustrated with example.

Введение

При проектировании цифровых управляющих устройств (УУ) одной из актуальных задач является соблюдение ограничений на аппаратные затраты при его реализации [1]. Для реализации линейных алгоритмов управления, доля операторных вершин в которых больше 75%, целесообразно использовать УУ класса композиционных микропрограммных устройств управления (КМУУ) [2]. В настоящее время базис CPLD (complex programmable logic devices) [3,4] широко используется для реализации схем устройств управления [5,6]. Проблема минимизации аппаратных затрат решается путем преобразования исходных дизъюнктивных нормальных форм (ДНФ) функций автомата с целью минимизации числа термов в них, а, следовательно, и необходимого для их реализации количества макроячеек программируемой логики (ПМЛ). В настоящей работе предлагается один из путей решения этой проблемы: реализация управляющего устройства в виде КМУУ с разделением кодов [2].

Целью исследования является снижение аппаратных затрат в комбинационной схеме КМУУ путем модификации исходного алгоритма управления и использования классов псевдоэквивалентных операторных линейных цепей.

Задачей исследования является разработка метода синтеза КМУУ, позволяющего уменьшить число макроячеек ПМЛ в схеме формирования функций возбуждения памяти. При этом алгоритм управления представляется в виде граф-схемы алгоритма (ГСА) [7].

Особенности синтеза КМУУ с разделением кодов.

Обозначим в некоторой ГСА Γ множество вершин $B = \{b_0, b_E\} \cup E_1 \cup E_2$ и множество дуг E . В этом множестве b_0 – начальная вершина, b_E – конечная вершина, E_1 – множество операторных вершин, E_2 – множество условных вершин. Операторная вершина $b_q \in E_1$ содержит набор микроопераций $Y(b_q) \subseteq Y$, где $Y = \{y_1, \dots, y_N\}$ – множество микроопераций (выходных сигналов). Условная вершина $b_q \in E_2$ содержит один элемент множества логических условий $X = \{x_1, \dots, x_L\}$ (входных сигналов). Под линейной понимается ГСА, в которой число операторных вершин превосходит 75% общего числа вершин.

Операторная линейная цепь (ОЛЦ) является последовательностью операторных вершин граф-схемы алгоритма. Сформируем множество ОЛЦ $C = \{\alpha_1, \dots, \alpha_G\}$. При этом каждая пара соседних вершин ОЛЦ $\alpha_g \in C$ соответствует некоторой дуге из множества E [2]. Каждая ОЛЦ $\alpha_g \in C$ имеет произвольное число входов и только один выход O_g . Формальные определения ОЛЦ, их входов и выходов можно найти в [2]. Отметим, что каждая вершина $b_q \in E_1$ соответствует микрокоманде MI_q , хранимой в управляющей памяти (УП) по адресу A_q . Для адресации микрокоманд достаточно

$$R = \lceil \log_2 M \rceil \quad (1)$$

бит, где $M = |E_1|$. Пусть каждая ОЛЦ $\alpha_g \in C$ включает F_g компонент и пусть $Q = \max(F_1, \dots, F_G)$. Закодируем каждую ОЛЦ $\alpha_g \in C$ двоичным кодом $K(\alpha_g)$ разрядности

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

а каждую компоненту $b_q \in E_1$ – двоичным кодом $K(b_q)$ разрядности

$$R_2 = \lceil \log_2 Q \rceil. \quad (3)$$

Используем для кодирования ОЛЦ элементы $\tau_r \in \mathcal{T}$, а для кодирования компонент – элемент $T_r \in T$, где $|\mathcal{T}| = R_1$, $|T| = R_2$. Кодирование компонент выполняется в естественном порядке, то есть

$$K(b_g^i) = K(b_g^{i-1}) + 1, \quad (4)$$

где $g = 1, \dots, G$, $i = 1, \dots, F_g$. Пусть для ГСА Г выполняется условие

$$R_1 + R_2 = R. \quad (5)$$

В этом случае для её интерпретации может быть использована модель КМУУ с разделением кодов (Рис.1), обозначаемая в дальнейшем символом U_1 .

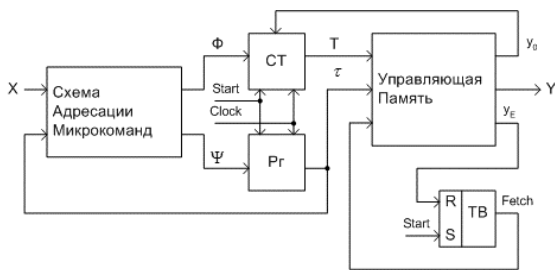


Рисунок 1 – Структурная схема КМУУ U_1

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

$$\begin{aligned} \Phi &= \Phi(\tau, X), \\ \Psi &= \Psi(\tau, X). \end{aligned} \quad (6)$$

При этом адрес микрокоманды MI_q представляется в виде

$$A(b_q) = K(\alpha_g) * K(b_q), \quad (7)$$

где b_q входит в ОЛЦ $\alpha_g \in C$, * - знак операции конкатенации.

По сигналу Start в Рг и СТ заносится начальный адрес микропрограммы, а триггер выборки ТВ устанавливается в единичное состояние. При этом сигнал Fetch=1, что разрешает выборку микрокоманд из УП. Если считанная микрокоманда не соответствует выходу ОЛЦ, то одновременно с микрооперациями $Y(b_q)$ формируется сигнал y_0 . Если $y_0=1$, то к содержимому СТ прибавляется единица и адресуется следующая компонента текущей ОЛЦ. Если выход ОЛЦ достигнут, то $y_0=0$. При этом адрес входа следующей ОЛЦ формируется схемой САМ. При достижении окончания микропрограммы формируется сигнал y_E , триггер ТВ обнуляется и выборка микрокоманд прекращается.

Число термов в схеме САМ может быть уменьшено путем введения преобразователя кодов ОЛЦ в коды классов псевдоэквивалентных ОЛЦ (ПОЛЦ) [2].

Операторные линейные цепи $\alpha_i, \alpha_j \in C$ называются псевдоэквивалентными, если их выходы связаны со входом одной и той же вершины ГСА Г. Однако реализация такого преобразователя требует дополнительных ячеек CPLD.

В настоящей работе предлагается использовать свободные ячейки управляющей памяти с целью снижения аппаратных затрат в схеме устройства.

Основная идея метода модификации ГСА

Пусть $C_1 \subset C$ – множество ОЛЦ, выходы которых не связаны с вершиной $b_E \in b$. Найдем разбиение $\Pi_C = \{B_1, \dots, B_I\}$ множества C_1 на классы ПОЛЦ, где I – количество классов ПОЛЦ. Пусть для каждой ОЛЦ $\alpha_g \in C_1$ выполняется условие

$$2^{R_2} > F_g, \quad (8)$$

Закодируем классы $B_i \in \Pi_C$ двоичными кодами $K(B_i)$ разрядности

$$R_3 = \lceil \log_2 I \rceil \quad (9)$$

и используем для кодирования элементы множества Z , где $|Z| = R_3$. Введем в каждую ОЛЦ $\alpha_g \in C_1$ дополнительную микрокоманду MC_q , содержащую $y_0=0$ и код $K(B_i)$, где $\alpha_g \in B_i$. При этом все микрокоманды MI_q содержат $y_0=1$.

В этом случае для интерпретации ГСА Γ предлагается КМУУ U_2 (Рис.2)

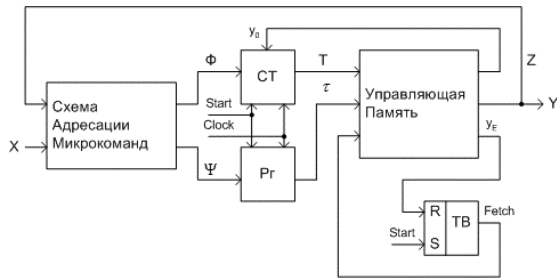


Рисунок 2 – Структурная схема КМУУ U_2

В КМУУ U_2 схема САМ реализует функции

$$\Phi = \Phi(Z, X), \quad (10)$$

$$\Psi = \Psi(Z, X), \quad (11)$$

а остальные элементы имеют то же назначение, что и для КМУУ U_1 .

Функции (10)-(11) генерируются, если из УП выбирается управляющая микрокоманда. В этом случае операционный автомат цифровой системы не получает микрооперации и находится в ждущем состоянии. Этого можно достичь, например, путем введения зависимости синхронизации операционного автомата от сигнала y_0 .

Отметим, что все элементы КМУУ U_2 (кроме УП) реализуются в составе кристалла CPLD. Для реализации УП необходимо использовать внешние ППЗУ.

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

1. Формирование множеств C, C_1, P_C для ГСА Γ .
2. Ввод дополнительных компонент в ОЛЦ $\alpha_g \in C_1$
3. Кодирование ОЛЦ, их компонент и классов.
4. Формирование содержимого управляющей памяти.
5. Формирование таблицы переходов КМУУ.

6. Синтез логической схемы КМУУ.

Пример применения предложенного метода

Пусть ГСА Γ_1 характеризуется множествами ОЛЦ

$$C = \{\alpha_1, \dots, \alpha_7\},$$

$$C_1 = \{\alpha_1, \dots, \alpha_6\},$$

$$P_C = \{B_1, B_2, B_3\},$$

где $B_1 = \{\alpha_1\},$

$$B_2 = \{\alpha_2, \alpha_3\},$$

$$B_3 = \{\alpha_4, \alpha_5, \alpha_6\},$$

$$\alpha_1 = \langle b_1, b_2, b_3 \rangle,$$

$$\alpha_2 = \langle b_4, b_5 \rangle,$$

$$\alpha_3 = \langle b_6, b_7, b_8 \rangle,$$

$$\alpha_4 = \langle b_9, b_{10}, b_{11} \rangle,$$

$$\alpha_5 = \langle b_{12}, b_{13}, b_{14} \rangle,$$

$$\alpha_6 = \langle b_{15}, b_{16} \rangle,$$

$$\alpha_7 = \langle b_{17}, b_{18}, b_{19}, b_{20} \rangle.$$

Следовательно, $G=7, R_1=3,$

$\tau = \langle \tau_1, \tau_2, \tau_3 \rangle, Q=4, R_2=2, T = \{T_1, T_2\},$

$M=20, R=5$ и условия (5) и (8) выполняются. Это означает, что применение предложенного метода имеет смысл. Отметим, что $R_3=2, Z = \{z_1, z_2\}.$

После введения дополнительных компонент в ОЛЦ $\alpha_g \in C_1$ имеем:

$$\alpha_1 = \langle b_1, b_2, b_3, MC_1 \rangle,$$

$$\alpha_2 = \langle b_4, b_5, MC_2 \rangle,$$

$$\alpha_3 = \langle b_6, b_7, b_8, MC_3 \rangle,$$

$$\alpha_4 = \langle b_9, b_{10}, b_{11}, MC_4 \rangle,$$

$$\alpha_5 = \langle b_{12}, b_{13}, b_{14}, MC_5 \rangle,$$

$$\alpha_6 = \langle b_{15}, b_{16}, MC_6 \rangle.$$

Закодируем ОЛЦ $\alpha_g \in C$

произвольным образом: $K(\alpha_1) = 000, \dots, K(\alpha_7) = 110$. Для выполнения условия (4) присвоим первой компоненте каждой ОЛЦ $\alpha_g \in C$ код 00, второй – 01, третьей – 10 и четвертой – 11. Адреса микрокоманд КМУУ $U_2(\Gamma_1)$ представлены в табл. 1.

Таблица 1. Адреса микрокоманд КМУУ $U_2(\Gamma_1)$

		$\tau_1 \tau_2 \tau_3$						
		000	001	010	011	100	101	110
$T_1 T_2$	00	b_1	b_4	b_6	b_9	b_{12}	b_{15}	b_{17}
	01	b_2	b_5	b_7	b_{10}	b_{13}	b_{16}	b_{18}
	10	b_3	MC_2	b_8	b_{11}	b_{14}	MC_6	b_{19}
	11	MC_1	*	MC_3	MC_4	MC_5	*	b_{20}

Здесь и далее символы $U_i(\Gamma_j)$ означают, что ГСА Γ_j интерпретируется КМУУ U_i . Из рис.3 имеем, например, $A(b_1) = 00000$, $A(b_8) = 01010$, $A(MC_2) = 00110$.

Закодируем классы $B_i \in \Pi_C$ следующим образом: $K(B_1) = 00$, $K(B_2) = 01$, $K(B_3) = 10$. Пусть микрооперации $y_n \in Y$ распределены по вершинам $b_q \in E_1$ следующим образом:

$$\begin{aligned} Y(b_1) &= Y(b_5) = \{y_1, y_2\}, \\ Y(b_2) &= Y(b_6) = Y(b_{14}) = \{y_3\}, \\ Y(b_3) &= Y(b_7) = Y(b_{12}) = \{y_1, y_4\}, \\ Y(b_4) &= Y(b_9) = Y(b_{13}) = \{y_2, y_3\}, \\ Y(b_8) &= Y(b_{15}) = Y(b_{17}) = \{y_5\}, \\ Y(b_{10}) &= Y(b_{16}) = Y(b_{19}) = \{y_4\}, \\ Y(b_{11}) &= Y(b_{18}) = \{y_1, y_5\}, \\ Y(b_{20}) &= \{y_6\} \end{aligned}$$

В этом случае содержимое управляющей памяти КМУУ $U_2(\Gamma_1)$ показано в табл. 2:

Таблица 2. Содержимое управляющей памяти КМУУ $U_2(\Gamma_1)$

		$\tau_1 \tau_2 \tau_3$						
		000	001	010	011	100	101	110
$T_1 T_2$	00	y_0	y_0	y_0	y_0	y_0	y_0	y_0
		y_1	y_2	y_3	y_2	y_1	y_0	y_5
		y_2	y_3	y_3	y_3	y_4	y_5	y_5
	01	y_0	y_0	y_0	y_0	y_0	y_0	y_0
y_3		y_1	y_1	y_4	y_2	y_4	y_1	
y_2	y_4	y_4	y_4	y_3	y_3	y_5	y_5	
10	y_0	z_2	y_0	y_0	y_0	z_1	y_0	
	y_1		y_5	y_1	y_3		y_4	
y_4			y_5	y_5				
11	-	*	z_2	z_1	z_1	*	y_E	
							y_6	

Здесь используется тривиальный метод формирования содержимого УП.

Пусть переходы из выходов ОЛЦ ГСА Γ_1 представлены следующей системой обобщенных формул переходов [2]:

$$\begin{aligned} B_1 &\rightarrow x_1 b_4 \vee \overline{x_1 x_2} b_6 \vee \overline{x_1 x_2} \overline{b_{11}}; \\ B_2 &\rightarrow x_2 x_3 b_9 \vee x_2 x_3 \overline{b_{13}} \vee \overline{x_2 x_4} b_{12} \vee \overline{x_2 x_4} \overline{b_{15}}; \\ B_3 &\rightarrow x_4 b_{17} \vee \overline{x_4 x_5} b_{11} \vee \overline{x_4 x_5} \overline{b_{20}}. \end{aligned} \quad (12)$$

Подобная система является основой для формирования таблицы переходов КМУУ U_2 , включающей столбцы: $B_i, K(B_i), b_q, A(b_q),$

X_h, Ψ_h, Φ_h, h . Здесь X_h - конъюнкция входных переменных, определяющая переход из класса B_i на микрокоманду, соответствующую вершине $b_q \in E_1$; Ψ_h - набор функций возбуждения, формирующей в РГ код $K(\alpha_g)$, где b_q входит в $\alpha_g \in C$; Φ_h - набор функций возбуждения, формирующей в СТ код $K(b_q)$; h - номер перехода, где $h = 1, \dots, H_2(\Gamma_j)$. Индекс 2 подчеркивает, что речь идет о КМУУ $U_2(\Gamma_j)$. Число $H_2(\Gamma_j)$ совпадает с числом термов в системе вида (12). В нашем случае $H_2(\Gamma_1) = 10$ и часть таблицы для класса $B_3 \in \Pi_C$ показана в табл.3:

Таблица 3. Фрагмент таблицы переходов КМУУ $U_2(\Gamma_1)$

B_i	$K(B_i)$	b_q	$A(b_q)$	X_h	Ψ_h	Φ_h	h
B_3	10	b_{17}	11000	x_4	D_1 D_2	-	1
		b_{11}	01110	$\overline{x_4 x_5}$	D_2 D_3	D_4	2
		b_{20}	11011	$\overline{x_4 x_5}$	D_1 D_2	D_4 D_5	3

Связь между системой (12), адресами из Рис.3 и табл.1 очевидна. Из табл.1 имеем (с учетом минимизации) фрагменты систем (10) и (11):

$$\begin{aligned} D_1 &= \overline{z_1 z_2} x_4 \vee \overline{z_1 z_2} \overline{x_4 x_5}; \\ D_2 &= \overline{z_1 z_2}; \quad D_3 = \overline{z_1 z_2} \overline{x_4 x_5}; \\ D_4 &= \overline{z_1 z_2} x_4; \quad D_5 = \overline{z_1 z_2} \overline{x_4 x_5}. \end{aligned} \quad (13)$$

Реализация логической схемы КМУУ $U_2(\Gamma_j)$ сводится к реализации систем (10)-(11) на макроячейках ПМЛ, и реализации управляющей памяти на ППЗУ. В нашем случае для реализации УП используется содержимое УП из рис.4

Отметим, что для рассматриваемой ГСА Γ_1 имеем $H_1(\Gamma_1) = 20$ и $\eta = H_1(\Gamma_1)/H_2(\Gamma_1) = 2$. Как показывают исследования, число термов в системах (10)-(11) приблизительно в η раз

меньше, чем в системе (6). Естественно, для определения числа макроячеек ПМЛ в схеме адресации микрокоманд необходимо знать число термов в одной макроячейке. Однако, отношение числа макроячеек также пропорционально параметру η [8].

Заключение

Предлагаемый метод модификации операторных линейных цепей КМУУ ориентирован на уменьшение числа макроячеек ПМЛ в схеме формирования адреса микрокоманд. При этом число микросхем ППЗУ, реализующих управляющую память, совпадает с этим числом для базовой структуры U_1 КМУУ с разделением кодов.

Метод основан на кодировании классов псевдоэквивалентных ОЛЦ, что позволяет уменьшить длину таблицы переходов по сравнению с этим параметром эквивалентного КМУУ U_1 .

Недостатком предлагаемого метода является увеличение числа тактов выполнения алгоритма в сравнении с КМУУ U_1 . Однако, побочным эффектом при уменьшении числа макроячеек ПМЛ может являться уменьшение числа уровней комбинационной схемы и, следовательно, уменьшение времени такта. Таким образом, окончательный вывод о среднем времени выполнения алгоритма может быть сделан только после реализации схем КМУУ $U_1(\Gamma_j)$ и $U_2(\Gamma_j)$. Исследования данного метода на произвольном наборе ГСА показали, что число макроячеек уменьшается до 30 %, а число уровней – на 2-3. Напомним, что применение этого метода возможно только для линейных ГСА при условии, что введение дополнительных микрокоманд не увеличивает разрядность адреса в сравнении с КМУУ U_1 .

Научная новизна предложенного метода заключается в использовании особенностей КМУУ (наличие классов псевдоэквивалентных ОЛЦ) и алгоритма управления (возможность введения дополнительных компонент ОЛЦ) для уменьшения числа макроячеек ПМЛ в схеме адресации микрокоманд.

Практическая значимость метода заключается в уменьшении аппаратных затрат при реализации схемы управляющего устройства, а также в улучшении его временных характеристик за счет уменьшения количества уровней схемы.

Дальнейшие направления исследований связаны с возможностью применения предложенного метода для базиса FPGA (Field-programmable logic arrays) [9], а также с разработкой САПР для синтеза схем композиционных микропрограммных устройств управления.

Литература

1. De Micheli G. Synthesis and Optimization of Digital Circuits. – NY: McGraw-Hill, 1994. – 636pp.
2. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах: Донецк ДНТУ, 2002 – 262 с.
3. MAX II CPLD Family: The Lowest-Power, Lowest-Cost CPLDs:
<http://www.altera.com/products/devices/cpld/max2/mx2-index.jsp>.
4. CoolRunner-II 1.8 Volt CPLD:
http://www.xilinx.com/products/silicon_solutions/cplds/coolrunner_series/coolrunner_ii_cplds/overview.htm.
5. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608 с.
6. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
7. Baranov S. Logic Synthesis for Control Automata – Boston: Kluwer Academic Publishers, 1994 – 312pp.
8. Баркалов А.А., Бабаков Р.М., Ахмад Бадер. Исследование аппаратных характеристик автомата Мили с кодированием фрагмента микроопераций по VHDL- моделям // Искусственный интеллект. – 2007, №1, - С. 117-122.
9. Maxfield C. The Design Warriors Guide for FPGA – Amsterdam : Elseteir, 2004. –541pp.