

СИНТЕЗ АВТОМАТА МУРА С ПРЕОБРАЗОВАНИЕМ РАСШИРЕННЫХ КОДОВ СОСТОЯНИЙ

Предлагается метод уменьшения числа макроячеек программируемой матричной логики в схеме микропрограммного автомата Мура. Метод основан на представлении кода состояния в виде конкатенации кодов класса псевдоэквивалентных состояний и состояния внутри этого класса. Такой подход позволяет устранить прямую зависимость между состояниями и микрооперациями. Для формирования микроопераций используется специальный преобразователь кода. В результате оптимизируется как блок формирования функций возбуждения, так и блок формирования микроопераций. Приводится пример применения предложенного метода.

1. Введение

Практически каждая цифровая система включает устройство управления, координирующее работу всех ее блоков [1]. Одной из широко используемых моделей устройства управления является микропрограммный автомат (МПА) Мура [2,3]. В настоящее время для реализации схем быстродействующих МПА применяются программируемые логические интегральные схемы (ПЛИС), макроячейки которых включают блоки программируемой матричной логики (ПМЛ) [4,5]. Одной из особенностей ПМЛ является ограниченное число промежуточных шин (термов), что вызывает необходимость минимизации реализуемых функций [6,7]. Схема МПА Мура состоит из блока формирования функций возбуждения памяти (БВП) и блока формирования микроопераций (БМО), которые реализуются на ПМЛ. Известные методы синтеза МПА Мура [2] позволяют оптимизировать аппаратные затраты только в одном из блоков МПА. В настоящей работе предлагается метод, позволяющий оптимизировать число ПМЛ в обоих блоках МПА Мура. Метод основан на

введении дополнительного преобразователя кодов и нестандартном представлении кодов состояний автомата.

Целью исследования является оптимизация схемы МПА Мура за счёт преобразования расширенных кодов состояний в коды наборов микроопераций.

Задача исследования состоит в разработке метода синтеза МПА Мура, позволяющего уменьшить число ПМЛ в схеме автомата.

2. Общее положение и основная идея предлагаемого метода

Пусть алгоритм управления задан в виде граф-схемы алгоритма (ГСА) Γ , отмеченной состояниями автомата Мура [2]. Состояния МПА образуют множество $A = \{a_1, \dots, a_M\}$, а для их кодирования используются внутренние переменные $T_r \in T = \{T_1, \dots, T_R\}$. При этом число внутренних переменных определяется формулой

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

В условных вершинах ГСА записаны логические условия (ЛЮ) $x_l \in X = \{x_1, \dots, x_L\}$. В операторных вершинах ГСА записаны наборы микроопераций $Y_q \subseteq Y$, где $q = \overline{1, Q}$, а $Y = \{y_1, \dots, y_N\}$ – множество микроопераций.

Представление алгоритма управления в виде ГСА используется для построения прямой структурной таблицы (ПСТ) [2], имеющей столбцы: a_m – исходное состояние МПА; $K(a_m)$ – код состояния $a_m \in A$ разрядности R ; a_s – состояние перехода; $K(a_s)$ – код состояния a_s ; X_h – выходной сигнал, определяющий переход из a_m в a_s ; Φ_h – набор функций возбуждения регистра памяти (RG), определяющий переход из $K(a_m)$ в $K(a_s)$; h – номер строки ПСТ, где $h = \overline{1, N_1}$. В столбце a_m записывается номер микроопераций (НМО) $Y(a_m) = Y_q$, формируемых в состоянии $a_m \in A$. Как правило, для реализации RG используется D-триггер. Поэтому множество функций возбуждения Φ содержит элементы D_r ($r = \overline{1, R}$). Эта таблица является основой для формирования системы функций

$$\Phi = \Phi(T, X), \quad (2)$$

$$Y = Y(T). \quad (3)$$

Система (2) – (3) определяет логическую схему МПА Мура, представленную на рис. 1. В этой схеме блок БВП реализует систему (2), а блок БМО – систему (3). Условимся в дальнейшем обозначать схему на рис. 1 символом U_1 . При этом $U_1(\Gamma_j)$ означает, что автомат Мура U_1 реализует ГСА Γ_j . Сигнал Start используется для установки МПА в начальное состояние, а сигнал Clock – для задания момента изменения содержимого RG.

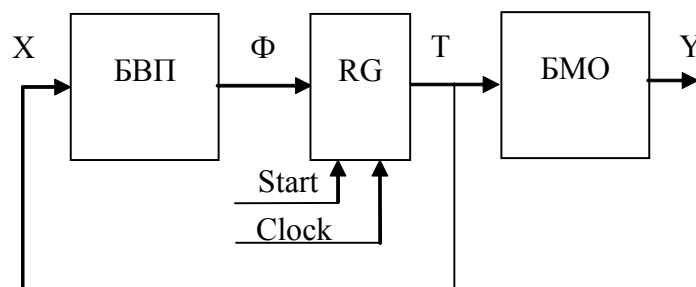


Рис. 1. Структурная схема МПА U_1

В МПА Мура число переходов N значительно превосходит число N_0 переходов эквивалентного МПА Мили [2]. Однако соответствующее кодирование состояний позволяет уменьшить этот параметр, что уменьшает число ПМЛ в блоке БВП [8]. Для уменьшения числа ПМЛ в блоке БМО также используются различные методы кодирования состояний [3,8]. Однако не существует эффективного метода кодирования состояний, который одновременно оптимизирует оба блока.

В настоящей работе предлагается метод синтеза МПА Мура, позволяющий максимально уменьшить число ПМЛ в блоках БВП и БМО. Этот метод заключается в следующем.

Пусть множество A разбивается на I классов псевдоэквивалентных состояний, образующих разбиение $\Pi_A = \{B_1, \dots, B_I\}$. Напомним, что состояния a_i, a_j называются псевдоэквивалентными, если выходы отмеченных ими вершин ГСА связаны со входом одной и той же вершины [3]. Закодируем классы $B_i \in \Pi_A$ кодами $K(B_i)$, имеющими R_1 разрядов:

$$R_1 = \lceil \log_2 I \rceil. \quad (4)$$

Используем для кодирования классов переменные $\tau_r \in \tau$, где $|\tau| = R_1$.

Пусть класс $B_i \in \Pi_A$ содержит M_i состояний и пусть

$$M_0 = \max(M_1, \dots, M_I). \quad (5)$$

Закодируем состояния $a_m \in B_i$ кодом $C(a_m)$ разрядности

$$R_2 = \lceil \log_2 M_0 \rceil \quad (6)$$

и используем для кодирования переменные $T_r \in T$, где $|T| = R_2$.

В этом случае состояние $a_m \in A$ однозначно определяется расширенным кодом

$$K(a_m) = K(B_i) * C(a_m), \quad (7)$$

где $*$ – знак конкатенации, $m = \overline{1, M}$, $i = \overline{1, I}$.

Подставим в соответствие каждому набору $Y_q \subseteq Y$ двоичный код $K(Y_q)$ разрядности

$$R_3 = \lceil \log_2 Q \rceil. \quad (8)$$

Используем для кодирования НМО переменные $z_r \in Z$, где $|Z| = R_3$. В этом случае для интерпретации ГСА Γ предлагается МПА Мура U_2 (рис. 2).

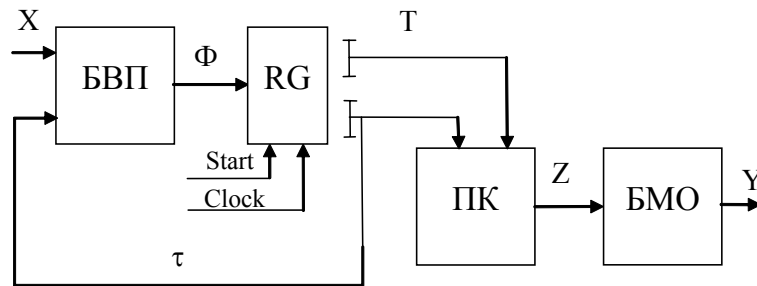


Рис. 2. Структурная схема МПА U_2

3. Метод синтеза автомата Мура U_2

В автомате U_2 блок БВП реализует систему функций

$$\Phi = \Phi(\tau, X), \quad (9)$$

включающую $R_1 + R_2$ элементов. Блок преобразователя кодов (ПК) формирует переменные $z_r \in Z$. Для этого реализуется система функций

$$Z = Z(\tau, T). \quad (10)$$

Блок БМО реализует микрооперации $y_n \in Y$, как функции системы

$$Y = Y(Z). \quad (11)$$

Такой подход даёт следующие преимущества по сравнению с МПА U_1 :

– число термов в системе (9) гарантировано равняется H_0 при любом кодировании классов $B_i \in \Pi_A$;

– кодирование наборов $Y_q \subseteq Y$ может быть выполнено таким образом, чтобы минимизировать число ПМЛ в блоке БМО.

Недостатками являются увеличение числа функций в системе Φ при выполнении условия

$$R_1 + R_2 > R, \quad (12)$$

а также наличие блока ПК. Отметим, что кодирование классов $B_i \in \Pi_A$ и состояний $a_m \in B_i$ можно выполнить так, чтобы уменьшить число ПМЛ в блоке ПК. При выполнении условия

$$R_3 < R \quad (13)$$

уменьшаются аппаратные затраты в блоке БМО, если он реализуется на заказных матрицах [9] либо макроячейках памяти [3].

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

1. Формирование разбиения Π_A .
2. Формирование наборов микроопераций $Y_q \subseteq Y$.
3. Оптимальное кодирование НМО.
4. Кодирование классов $B_i \in \Pi_A$ и состояний $a_m \in B_i$.
5. Формирование прямой структурной таблицы МПА U_2 .
6. Формирование систем функций (9) – (10).
7. Реализация схемы МПА в заданном элементном базисе.

Рассмотрим пример синтеза МПА $U_2(\Gamma_1)$, где ГСА Γ_1 представлена на рис. 3.

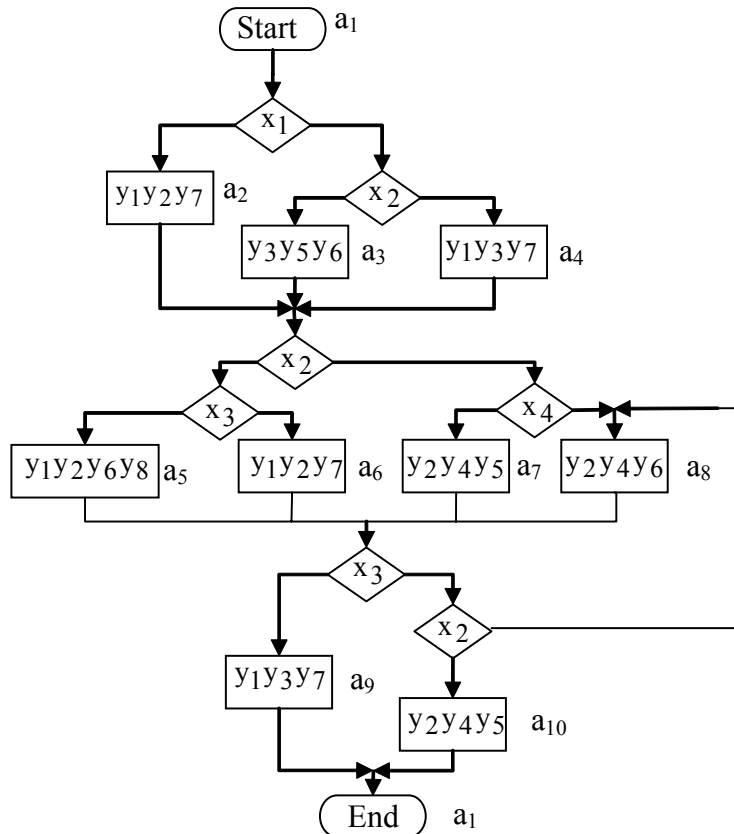


Рис. 3. Граф-схема алгоритма Γ_1 , отмеченная состояниями автомата Мура

Как видно из рис. 3, автомат $U_2(\Gamma_1)$ имеет $M = 10$ состояний, что дает $R = 4$. Исходя из определения псевдоэквивалентных состояний [3], можно найти множество

$\Pi_A = \{B_1, \dots, B_4\}$. Здесь $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5, \dots, a_8\}$, $B_4 = \{a_9, a_{10}\}$. Итак, $I = 4$, $R_1 = 2$.

В операторных вершинах ГСА Γ_1 находятся $Q = 7$ различных НМО: $Y_1 = Y(a_1) = \emptyset$, $Y_2 = \{y_1, y_2, y_7\}$, $Y_3 = \{y_3, y_5, y_6\}$, $Y_4 = \{y_1, y_3, y_7\}$, $Y_5 = \{y_1, y_2, y_6, y_8\}$, $Y_6 = \{y_2, y_4, y_5\}$, $Y_7 = \{y_2, y_4, y_6\}$. Таким образом, для кодирования НМО достаточно $R_3=3$ переменные и $Z = \{z_1, z_2, z_3\}$. Отметим, что условие (13) в данном случае выполняется.

Будем понимать под «оптимальным» такое кодирование НМО, когда система (11) имеет минимально возможное число термов. На рис. 4 представлен один из вариантов такого кодирования.

$z_1 \backslash z_2 z_3$	00	01	11	10
0	Y_1	Y_3	Y_4	*
1	Y_7	Y_6	Y_2	Y_5

Рис. 4. Оптимальные коды наборов микроопераций

С учётом этих кодов имеем систему (14) для автомата $U_2(\Gamma_1)$:

$$\begin{aligned}
 y_1 &= Y_2 \vee Y_4 \vee Y_5 = z_2; y_2 = Y_2 \vee Y_6 \vee Y_5 \vee Y_7 = z_1; \\
 y_3 &= Y_3 \vee Y_4 = \bar{z}_1 z_3; y_4 = Y_6 \vee Y_7 = z_1 \bar{z}_2; \\
 y_5 &= Y_3 \vee Y_6 = \bar{z}_2 z_3; y_6 = Y_3 \vee Y_5 \vee Y_7 = \bar{z}_1 \bar{z}_2 z_3 \vee z_1 \bar{z}_3; \\
 y_7 &= Y_2 \vee Y_4 = z_2 z_3; z_8 = Y_5 = z_2 \bar{z}_3.
 \end{aligned} \tag{14}$$

Классы $B_i \in \Pi_A$ и состояния $a_m \in B_i$ необходимо закодировать так, чтобы уменьшить число термов в системе (10). При этом оптимизируется число ПМЛ в схеме блока ПК. Построим таблицу, отражающую связь между кодами наборов МО и состояниями автомата $U_2(\Gamma_1)$ (табл. 1).

Таблица 1

$K(Y_q)$	001	011	100	101	110	111
a_m	a_3	a_4, a_9	a_8, a_{10}	a_7	a_5	a_2, a_6
Y_q	Y_3	Y_4	Y_7	Y_6	Y_5	Y_2

В каждом столбце табл.1 первый разряд кода $K(Y_q)$ соответствует переменной z_1 , второй – z_2 и третий – z_3 . Это позволяет построить следующую систему функций:

$$\begin{aligned}
 z_1 &= a_2 \vee a_5 \vee a_6 \vee a_7 \vee a_8 \vee a_{10}; \\
 z_2 &= a_2 \vee a_4 \vee a_6 \vee a_9; \\
 z_3 &= a_2 \vee a_3 \vee a_4 \vee a_6 \vee a_7 \vee a_9.
 \end{aligned} \tag{15}$$

Состояния $a_m \in A$ должны быть закодированы так, чтобы у всех $a_m \in B_i$ код $K(B_i)$ совпадал.

Оптимизация системы (15) – это типичная задача символьного кодирования. Методы решения подобных задач можно найти в [1]. Один из вариантов оптимального кодирования классов $B_i \in \Pi_A$ и состояний $a_m \in B_i$ представлен на рис. 5.

Используя коды из рис. 5, можно получить следующую систему (16) из системы (15):

$$\begin{aligned}
 z_1 &= \tau_2 \bar{T}_1 \vee \tau_1 \tau_2 \vee \tau_1 T_1 \bar{T}_2; \\
 z_2 &= \bar{T}_1 T_2 \vee \bar{\tau}_2 T_2; \\
 z_3 &= T_2 \vee \bar{\tau}_1 \tau_2.
 \end{aligned} \tag{16}$$

T_1T_2		00	01	11	10
$\tau_1\tau_2$	00	a_1	*	*	*
	01	*	a_2	a_4	a_3
	11	a_5	a_6	a_7	a_8
	10	*	*	a_9	a_{10}

Рис. 5. Кодирование классов и состояний автомата $U_2(\Gamma_1)$

Если для реализации блока ПК используются ПМЛ, имеющие $q = 3$ терма, то каждая функция $z_r \in Z$ реализуется на одной макроячейке ПМЛ.

Для построения ПСТ необходимо получить систему обобщённых формул перехода [3]. В случае МПА $U_2(\Gamma_1)$ эта система имеет следующий вид:

$$\begin{aligned}
 B_1 &\rightarrow x_1a_2 \vee \bar{x}_1x_2a_3 \vee \bar{x}_1\bar{x}_2a_4; \\
 B_2 &\rightarrow x_2x_3a_5 \vee x_2\bar{x}_3a_6 \vee \bar{x}_2x_4a_7 \vee \bar{x}_2\bar{x}_4a_8; \\
 B_3 &\rightarrow x_3a_9 \vee x_2\bar{x}_3a_{10} \vee \bar{x}_2\bar{x}_3a_8; \\
 B_4 &\rightarrow a_1.
 \end{aligned} \tag{17}$$

Эта система очевидным образом преобразовывается в таблицу переходов (табл. 2).

Таблица 2

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Φ_h	h
B_1	00	a_2	0101	x_1	D_2D_4	1
		a_3	0110	$\bar{x}_1x_2$	D_2D_3	2
		a_4	0111	$\bar{x}_1\bar{x}_2$	$D_2D_3D_4$	3
B_2	01	a_5	1100	x_2x_3	D_1D_2	4
		a_6	1101	$x_2\bar{x}_3$	$D_1D_2D_4$	5
		a_7	1111	$\bar{x}_2x_4$	$D_1D_2D_3D_4$	6
		a_8	1110	$\bar{x}_2\bar{x}_4$	$D_1D_2D_3$	7
B_3	11	a_9	1011	x_3	$D_1D_3D_4$	8
		a_{10}	1010	$\bar{x}_3x_2$	D_1D_3	9
		a_8	1110	$\bar{x}_3\bar{x}_2$	$D_1D_2D_3$	10

Из табл. 2 строится система (9), которая затем минимизируется для реализации на макроячейках ПМЛ. Пусть F_h – терм, соответствующий строке h ПСТ. Например, $F_1 = \bar{\tau}_1\bar{\tau}_2x_1$, $F_{10} = \tau_1\tau_2\bar{x}_3\bar{x}_2$. Из табл. 2 имеем, например, $D_1 = F_4 \vee \dots \vee F_{10}$. После минимизации получаем окончательную функцию $D_1 = \tau_2$. Аналогичным образом можно получить остальные функции системы (9).

Последний этап синтеза связан с реализацией схем блоков МПА по полученным системам функций. Этот этап достаточно полно рассмотрен в литературе [6,7]. Имеются промышленные пакеты [4,5], ориентированные на решение подобных задач. В этой связи данный этап в этой статье не рассматривается.

Отметим, что ПСТ автомата $U_1(\Gamma_1)$ имеет $H = 23$ строки. При этом каждая функция $D_r \in D$ будет иметь до 23 термов. Применение предложенного метода уменьшает этот параметр в 2,3 раза. При этом число функций $D_r \in \Phi$ будет одинаковым для $U_1(\Gamma_1)$ и $U_2(\Gamma_2)$ из-за нарушения условия (12).

4. Заключение

Предлагаемый в работе метод синтеза МПА Мура ориентирован на уменьшение аппаратных затрат в схеме автомата. Этот метод позволяет гарантированно уменьшить число макроячеек ПМЛ в схемах формирования функций возбуждения памяти и микроопераций. Однако это связано с введением дополнительного блока ПК, который потребляет некоторые ресурсы кристалла ПЛИС и вносит задержку в схему МПА. Эти недостатки можно уменьшить путем оптимального кодирования расширенных кодов состояний.

Исследование эффективности предложенного метода на стандартных примерах [10] показали, что предложенный метод позволяет уменьшить аппаратные затраты на 37 % (в среднем) для 98,2 % примеров. При этом быстроедействие МПА U_2 практически совпадает с этим параметром МПА U_1 . Это объясняется уменьшением числа уровней в схемах блоков МПА U_2 .

Научная новизна предложенного метода заключается в использовании классов псевдоэквивалентных состояний и кодов состояний в этих классах для уменьшения числа макроячеек ПМЛ в схеме МПА Мура.

Практическая значимость метода заключается в уменьшении аппаратных затрат в схеме МПА, что позволяет получить более дешёвые схемы по сравнению с известными из литературы аналогами.

Дальнейшее направление развития этой работы связано с возможностью применения предложенного метода для базиса FPGA [11] и заказных матриц [12].

Список литературы: 1. *De Micheli G.* Synthesis and Optimization of Digital Circuits. №4.: McGraw-Hill, 1994. 636 p. 2. *Baranov S.* Logic and System Design of Digital Systems. Tallinn: TUT Press, 2008. 267 p. 3. *Barkalov A., Titarenko L.* Logic Synthesis for FSM-Based Control Units. Springer. Lectures Notes in Electrical Engineering. №53 Berlin: Verlag Heidelberg, 2009. 233 p. 4. *Altera Corporation* webpage. <http://www.altera.com> 5. *Xilinx Corporation* webpage. <http://www.xilinx.com> 6. *Соловьев В.В.* Проектирование цифровых систем на основе программируемых логических интегральных схем. М.: Горячая линия – Телеком, 2007. 636 с. 7. *Kania D.* Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL. Gliwice: Politechnika. 2004. 212 str. 8. *Barkalov A., Titarenko L., Chmielewski S.* Hardware reduction for Moore FSM implemented with CPLD // KEiT PAN, 2009. 212 p. 9. *Barkalov A., Titarenko L., Hebda O.* Matrix implementation of Moore FSM with expansion of coding space // Pomiar Automatyka Kontrola. vol.56. 2010. № 7. 694 – 696 p. 10. *Yang S.* Logic Synthesis and Optimization Benchmarks user guide. Technical report №1991 IWLS-UG Saeyang. – Microelectronics center of North Carolina. 11. *Maxfield C.* The Design Warrior's Guide to FPGAs. Amsterdam: Elsevier, 2004. 541 p. 12. *Smith M.* Application Specific Integrated Circuits. Boston: Addison-Wesley, 1997.

Поступила в редколлегию 22.11.2010

Баркалов Александр Александрович, д-р техн. наук, профессор Института информатики и электроники Университета Зеленогурского. Адрес: ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, e-mail: A.Barkalov@iie.uz.zgora.pl.

Титаренко Лариса Александровна, д-р. техн. наук, профессор Института информатики и электроники Университета Зеленогурского. Адрес: ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, e-mail: L.Titarenko@iie.uz.zgora.pl.

Хебда Елена Петровна, магистр, аспирант Института информатики и электроники Университета Зеленогурского. Адрес: ul. prof. Z. Szafrana 2, 65-516 Zielona Góra, e-mail: O.Shapoval@iie.uz.zgora.pl.

Солдатов Константин Альбертович, магистр, аспирант кафедры компьютерной инженерии Донецкого национального технического университета. Адрес: Украина, 83000, Донецк, ул. Артема, 56, e-mail: K.Soldatov@cs.dgtu.donetsk.ua