

С.А. Цололо

## Реализация автомата Мура в базисе гибридных *FPGA*

Предложенный метод реализации автомата Мура в базисе гибридных *FPGA* направлен на уменьшение аппаратных затрат в логической схеме автомата Мура, основан на использовании двух источников кодов псевдоэквивалентных состояний.

The proposed method is directed at reducing instrumental coast of Moore FSM logic circuit based on using two source of codes of pseudoequivalent states.

Запропоновано метод реалізації автомата Мура в базисі гібридних *FPGA*, спрямований на зменшення апаратних витрат в логічній схемі автомата Мура, заснований на використанні двох джерел кодів псевдоеквівалентних станів.

**Введение.** Модель микропрограммного автомата (МПА) Мура часто используется при реализации схем устройств управления [1, 2]. Как известно, устройства управления – важная часть цифровых схем [3]. В настоящее время программируемые логические интегральные схемы (ПЛИС) типа *FPGA* (*field programmable gate arrays*) – популярный базис для реализации сложных цифровых систем [4, 5]. При этом большое значение имеет уменьшение площади кристалла, занимаемого схемой любого блока системы, в том числе и МПА [6].

Одним из направлений в технологии *FPGA* есть так называемые гибридные *FPGA* [7, 8]. В состав подобных ПЛИС входят табличные элементы типа *LUT* (*look-up table*) и встроенные программируемые логические матрицы (*PLA*, *programmable logic array*). В литературе практически отсутствуют методы синтеза устройств управления, ориентированные на гибридные *FPGA*.

В настоящей статье предлагается метод синтеза МПА Мура в базисе гибридных *FPGA*. Этот метод – адаптация идей [9, 10] к особенностям гибридных *FPGA*.

Цель исследования – разработка метода синтеза МПА Мура, учитывающего особенности автомата (наличие классов псевдоэквивалентных состояний) и элементного базиса.

Задача исследования – уменьшение аппаратных затрат в схеме формирования функций возбуждения памяти автомата.

При этом алгоритм управления, реализуемый автоматом, представлен в виде граф-схемы алгоритма (ГСА) [1].

### Основные положения и особенности реализации автомата Мура

Пусть алгоритм управления цифровой системы представлен ГСА  $\Gamma = \Gamma(B, E)$ , где  $B = \{b_0, b_E\} \cup E_1 \cup E_2$  – множество вершин,  $E = \{ \langle b_q, b_t \rangle | b_q, b_t \in B \}$  – множество дуг. Здесь  $b_0$  – начальная вершина ГСА,  $b_E$  – конечная вершина ГСА,  $E_1$  – множество операторных вершин,  $E_2$  – множество условных вершин. В вершинах  $b_q \in E_1$  записываются наборы микроопераций  $Y(b_q) \subseteq Y$ , где  $Y = \{y_1, \dots, y_N\}$  – множество микроопераций операционного автомата цифровой системы [1]. В вершинах  $b_q \in E_2$  записываются элементы множества логических условий  $X = \{x_1, \dots, x_L\}$ . Начальная и конечная вершины ГСА соответствуют состоянию  $a_1 \in A = \{a_1, \dots, a_M\}$ , где  $A$  – множество состояний автомата Мура, а каждая вершина  $b_q \in E_1$  соответствует одному из элементов множества  $A$  [3]. Логическая схема МПА Мура задается системой уравнений

$$\Phi = \Phi(T, X), \quad (1)$$

**Ключевые слова:** автомат Мура, гибридные *FPGA*, *PLA*, *LUT*, синтез.

$$Y = Y(T), \quad (2)$$

где  $T = \{T_1, \dots, T_R\}$  – множество внутренних переменных, кодирующих состояния  $a_m \in A$ ,  $R = \lceil \log_2 M \rceil$ ;  $\Phi = \{D_1, \dots, D_R\}$  – множество функций возбуждения триггеров памяти состояний. Системы (1)–(2) формируются на основе прямой структурной таблицы (ПСТ) со столбцами:  $a_m$  – текущее состояние;  $K(a_m)$  – код состояния  $a_m \in A$ ;  $a_s$  – состояние перехода;  $K(a_s)$  – код состояния  $a_s \in A$ ;  $X_h$  – конъюнкция некоторых элементов множества  $X$  (или их отрицаний), определяющая переход  $\langle a_m, a_s \rangle$ ;  $\Phi_h$  – набор функций возбуждения памяти МПА, принимающих единичное значение для переключения памяти из  $K(a_m)$  в  $K(a_s)$ ;  $h = 1, \dots, H_1(\Gamma)$  – номер строки таблицы. В столбце  $a_m$  записывается набор микроопераций  $Y(a_m) \subseteq Y$ , формируемых в состоянии  $a_m \in A$ . Естественно, что  $Y(a_m) = Y(b_q)$ , где вершина  $b_q \in E_1$  отмечена состоянием  $a_m \in A$ .

Системы (1)–(2) определяют модель  $U_1$  автомата Мура (рис. 1), включающую блок переходов (БП), блок микроопераций (БМО) и регистр состояний РГ.

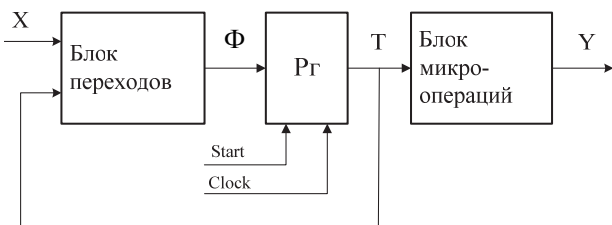


Рис. 1. Структурная схема МПА Мура  $U_1$

В схеме на рис. 1 блок БП реализует функции (1), а блок БМО – функции (2). Коды состояний  $a_m \in A$  хранятся в регистре, который обнуляется сигналом *Start* и переключается по сигналу *Clock*.

Как правило, число переходов  $H_1(\Gamma)$  больше числа переходов  $H_0(\Gamma)$  эквивалентного автомата Мили [1], что приводит к росту аппара-

турных затрат в схеме МПА Мура в сравнении с этим показателем эквивалентного автомата Мили. Параметр  $H_1(\Gamma)$  можно уменьшить, благодаря наличию псевдоэквивалентных состояний (ПЭС) МПА Мура [11]. Состояния  $a_m, a_s \in A$  называется ПЭС, если выходы соответствующих им вершин соединены с входом одной и той же вершины ГСА  $\Gamma$ . Пусть  $\Pi_A = \{B_1, \dots, B_I\}$  – разбиение множества  $A$  на классы ПЭС ( $I \leq M$ ). Построим систему функций

$$B_i = \bigvee_{m=1}^I C_{mi} A_m \quad (i = 1, \dots, I), \quad (3)$$

где  $C_{mi}$  – булева переменная, равная единице, если и только если  $a_m \in B_i$ ,  $A_m$  – конъюнкция внутренних переменных  $T_r \in T$ , соответствующая коду  $K(a_m)$  состояния  $a_m \in A$ . Закодируем состояния  $a_m \in A$  так, чтобы любая функция системы (3) представлялась одним конъюнктивным термом. Назовем такое кодирование оптимальным кодированием состояний.

Такой подход ведет к модели  $U_2$ , структура которой совпадает со структурой модели  $U_1$ , но число термов совпадает с  $H_0(\Gamma)$ . Однако такое кодирование не всегда возможно [2] из-за особенностей ГСА. Например, для  $R = 2$ ,  $B_1 = \{a_1\}$ ,  $B_2 = \{a_2, a_3, a_4\}$  оптимальное кодирование состояний, сокращающее параметр  $H_2(\Gamma)$  до  $H_0(\Gamma)$ , невозможно. Здесь мы вводим обозначение  $H_i(\Gamma_j)$ , что определяет число строк в модели  $U_i$  при интерпретации ГСА  $\Gamma_j$ .

Число строк ПСТ гарантированно равняется  $H_0(\Gamma)$ , если использовать следующий подход. Поставим в соответствие классу  $B_i \in \Pi_A$  двоичный код  $K(B_i)$  разрядности  $R_B = \lceil \log_2 I \rceil$  и используем переменные  $\tau_r \in \tau$  для такого кодирования, где  $|\tau| = R_B$ . В этом случае МПА Мура представляется в виде структуры  $U_3$  (рис. 2).

В МПА  $U_3$  БП формирует функции

$$\Phi = \Phi(\tau, X), \quad (4)$$

а блок преобразователя кодов (БПК) реализует систему функций

$$\tau = \tau(T). \quad (5)$$

При этом код  $K(B_i)$  класса  $B_i \in \Pi_A$  формируется на основе кодов состояний  $a_m \in B_i$ . Как и ранее, блок БМО реализует систему (2).

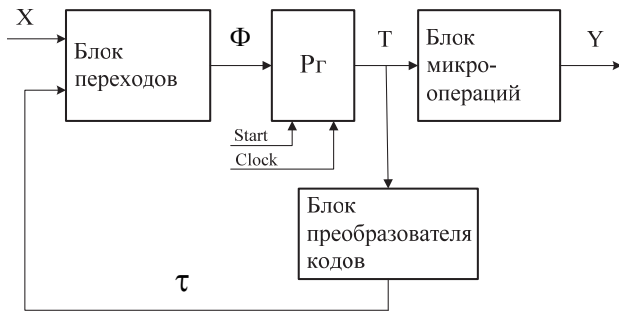


Рис. 2. Структурная схема МПА Мура  $U_3$

В [11] показано, что  $H_3(\Gamma) = H_0(\Gamma)$ . Недостаток модели  $U_3$  – наличие БПК, потребляющего некоторые ресурсы ПЛИС. В статье предложен метод синтеза МПА Мура, позволяющий сохранить положительные и устранить отрицательные качества модели  $U_3$ .

Как уже отмечалось, метод ориентирован на технологию гибридных *FPGA*. В этом случае предлагается реализовать схему БП на встроенных *PLA*, а схемы РГ, БМО и БПК – на *LUT* элементах. При этом необходимо уменьшать число термов в системе (4), т.е. БПК необходимо оставить. Однако число *LUT* элементов можно уменьшить, если использовать несколько источников кодов классов ПЭС.

#### Основная идея предлагаемого метода

Закодируем состояния  $a_m \in A$  оптимальным образом. Пусть  $T(B_i)$  – число термов в функции  $B_i \in \Pi_A$ . Представим множество  $\Pi_A$  в виде объединения множеств  $\Pi_B$  и  $\Pi_C$ . При этом распределение классов выполняется следующим образом:

$$\begin{aligned} (T(B_i) = 1) &\rightarrow B_i \in \Pi_B; \\ (T(B_i) > 1) &\rightarrow B_i \in \Pi_C. \end{aligned} \quad (6)$$

Очевидно, что преобразованию подлежат только коды состояний  $a_m \in B_i$  для блоков  $B_i \in \Pi_C$ . Поставим в соответствие каждому классу  $B_i \in \Pi_C$  двоичный код  $K(B_i)$  разрядности

$$R_c = \lceil \log_2(I_c + 1) \rceil, \quad (7)$$

где  $I_c = |\Pi_C|$ . Назначение единицы в формуле (7) объясним позже.

Пусть следующие условия выполняются для *PLA*, выходящей в состав гибридной *FPGA*:

$$L + R + R_c \leq S, \quad (8)$$

$$H_0(\Gamma) \leq q. \quad (9)$$

В (8) – (9) входит число входов  $S$  и термов  $q$  блока *PLA*. При выполнении (8) – (9) БП реализуется в виде одного блока *PLA*.

В настоящей статье предлагается следующая модель  $U_3$  автомата Мура (рис. 3).

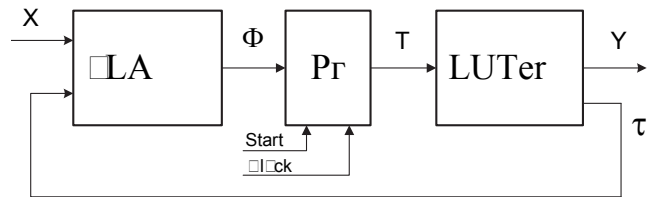


Рис. 3. Структурная схема МПА Мура  $U_4$

В МПА  $U_4$  блок *PLA* формирует функции

$$\Phi = \Phi(T, \tau, X). \quad (10)$$

Блок *LUTer* состоит из *LUT* элементов и формирует выходные сигналы (2) и функции (5). В МПА  $U_4$  имеются два источника кодов классов ПЭС. Коды классов  $B_i \in \Pi_B$  содержатся в регистре РГ, а коды классов  $B_i \in \Pi_C$  формируются блоком *LUTer*.

Для идентификации источника кода ПЭС необходим специальный код. Условимся, что этот код определяется выражением

$$\tau_r = 0 \quad (r = \overline{1, R_c}). \quad (11)$$

Наличием этого кода определяется необходимость единицы в выражении (7).

В статье предлагается метод синтеза МПА Мура  $U_4$  по отмеченной ГСА. Метод включает следующие этапы:

- формирование разбиения  $\Pi_A = \{B_1, \dots, B_7\}$ ;
- оптимальное кодирование состояний  $a_m \in A$ ;
- формирование множеств  $\Pi_B$  и  $\Pi_C$ ;
- кодирование классов  $B_i \in \Pi_C$ ;
- формирование содержимого блока  $PLA$ ;
- формирование таблицы блока  $LUTer$ ;
- формирование системы функций, задающих схему МПА;
- реализация схемы в заданном элементном базисе.

Рассмотрим применение предложенного метода.

### Пример применения предложенного метода

В целях экономии используем задание автомата не в виде ГСА, а в виде системы обобщенных формул перехода (ОФП) [6, 7]. Пусть автомат Мура  $U_1(\Gamma_1)$  определяется следующей системой ОФП:

$$\begin{aligned}
 B_1 &\rightarrow x_1 a_2 \vee \bar{x}_1 a_3; \\
 B_2 &\rightarrow x_2 a_4 \vee \bar{x}_2 x_3 a_5 \vee \bar{x}_2 \bar{x}_3 a_6; \\
 B_3 &\rightarrow x_3 a_6 \vee \bar{x}_3 x_4 a_8 \vee \bar{x}_3 \bar{x}_4 a_1; \\
 B_4 &\rightarrow x_5 a_8 \vee \bar{x}_5 x_6 a_{11} \vee \bar{x}_5 \bar{x}_6 a_{13}; \\
 B_5 &\rightarrow a_{10}; \\
 B_6 &\rightarrow x_1 a_{12} \vee \bar{x}_1 a_1; \\
 B_7 &\rightarrow x_4 a_1 \vee \bar{x}_4 x_5 a_7 \vee \\
 &\vee \bar{x}_4 \bar{x}_5 x_6 a_{11} \vee \bar{x}_4 \bar{x}_5 \bar{x}_6 a_{13}.
 \end{aligned} \tag{12}$$

Пусть при этом получено разбиение  $\Pi_A = \{B_1, \dots, B_7\}$ , где  $B_1 = \{a_1\}$ ,  $B_2 = \{a_2, a_3\}$ ,  $B_3 = \{a_4\}$ ,  $B_4 = \{a_5, a_6, a_7\}$ ,  $B_5 = \{a_8, a_9\}$ ,  $B_6 = \{a_{10}\}$ ,  $B_7 = \{a_{11}, a_{12}, a_{13}\}$ .

Пусть система микроопераций автомата  $U_1(\Gamma_1)$  представляется следующей системой уравнений:

$$\begin{aligned}
 y_1 &= a_2 \vee a_3 \vee a_{12}; \\
 y_2 &= a_4 \vee a_8 \vee a_9 \vee a_{10}; \\
 y_3 &= a_3 \vee a_7 \vee a_8 \vee a_{11}; \\
 y_4 &= a_5 \vee a_{11}; \\
 y_5 &= a_6 \vee a_7 \vee a_8;
 \end{aligned} \tag{13}$$

$$\begin{aligned}
 y_6 &= a_4 \vee a_6 \vee a_{10}; \\
 y_7 &= a_3 \vee a_{11} \vee a_{12} \vee a_{13}.
 \end{aligned}$$

Система (13) строится тривиальным образом: если микрооперация  $y_n \in A$  формируется в состоянии  $a_m \in A$ , то в уравнение для  $y_n$  включается терм  $a_m$ .

Итак, для автомата  $U_1(\Gamma_1)$  имеем  $M = 13$ ,  $R = 4$ ,  $T = \{T_1, \dots, T_4\}$ ,  $\Phi = \{D_1, \dots, D_4\}$ ,  $I = 7$ ,  $N = 8$ . Сформируем систему (3), имеющую следующий вид:

$$\begin{aligned}
 B_1 &= a_1; \\
 B_2 &= a_2 \vee a_3; \\
 B_3 &= a_4; \\
 B_4 &= a_5 \vee a_6 \vee a_7; \\
 B_5 &= a_8 \vee a_9; \\
 B_6 &= a_{10}; \\
 B_7 &= a_{11} \vee a_{12} \vee a_{13}.
 \end{aligned} \tag{14}$$

Один из возможных результатов оптимального кодирования состояний для автомата  $U_1(\Gamma_1)$  показан картой Карно на рис. 4.

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	00	$a_1$	$a_2$	*	$a_4$
	01	$a_5$	*	$a_6$	$a_{10}$
	11	$a_{11}$	$a_3$	$a_7$	$a_8$
	10	$a_{13}$	$a_{12}$	*	$a_9$

Рис. 4. Коды состояний автомата Мура  $U_1(\Gamma_1)$

Анализ этой карты показывает, что классы  $B_1, B_3, B_5, B_6 \in \Pi_B$  и  $B_2, B_4, B_7 \in \Pi_C$ . Таким образом,  $I_C = 3$ ,  $R_C = 2$ ,  $\tau = \{\tau_1, \tau_2\}$ . Закодируем классы  $B_i \in \Pi_C$  следующим образом:  $K(B_2) = 11$ ,  $K(B_4) = 01$ ,  $K(B_7) = 10$ . Таким образом, чем больше состояний входит в класс  $B_i \in \Pi_C$ , тем меньше единиц содержит код этого класса. Отметим, что код 00 соответствует ситуации  $B_i \in \Pi_B$ . Из карты Карно имеем  $K(B_1) = 0000$ ,  $K(B_3) = 001*$ ,  $K(B_5) = 1*10$ ,  $K(B_6) = 0110$ . Теперь переходим к модели  $U_4(\Gamma_1)$ .

Таблица блока *PLA* Мура  $U_1(\Gamma)$  включает столбцы  $B_i$ ,  $K(B_i)$ ,  $a_s$ ,  $K(a_s)$ ,  $X_h$ ,  $\Phi_h$ ,  $h$ . Столбец  $K(B_i)$  разбивается на подстолбцы  $\Pi_B$  и  $\Pi_C$ , чтобы показать источник кодов. Для автомата  $U_1(\Gamma_1)$  эта таблица включает  $H_4(\Gamma_1) = 18$  строк, что определяется числом термов в системе (12). Фрагмент преобразованной ПСТ для классов  $B_1 \in \Pi_B$  и  $B_2 \in \Pi_C$  содержит пять строк (табл. 1).

**Таблица 1.** Таблица блока *PLA* автомата Мура  $U_1(\Gamma_1)$

$B_i$	$K(B_i)$		$a_s$	$K(a_s)$	$X_h$	$\Phi_h$	$h$
	$\Pi_B$	$\Pi_C$					
$B_1$	0000	00	$a_2$	0001	$x_1$	$D_4$	1
			$a_3$	1101	$\bar{x}_1$	$D_1 D_2 D_4$	2
$B_2$	****	11	$a_4$	0010	$x_2$	$D_3$	3
			$a_5$	0100	$\bar{x}_2 x_3$	$D_2$	4
			$a_6$	0111	$\bar{x}_2 \bar{x}_3$	$D_2 D_3 D_4$	5

Связь этой таблицы с системой (12) и кодами классов и состояний очевидны. Отметим, что при  $B_i \in \Pi_B$  столбец  $\Pi_C$  содержит код 00, а при  $B_i \in \Pi_C$  содержимое столбца  $\Pi_B$  игнорируется и может быть любым, что отмечено знаками \*. Эта таблица – основа для формирования системы  $\Phi = \Phi(T, \tau, X)$ , задающей блок *PLA*. Например, из табл. 1 с учетом минимизации имеем:

$$D_1 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{\tau}_1 \bar{\tau}_2 \bar{x}_1 \text{ (строка 2);}$$

$$D_2 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{\tau}_1 \bar{\tau}_2 \bar{x}_1 \vee \tau_1 \tau_2 \bar{x}_2 \text{ (строки 2, 4, 5);}$$

$$D_3 = \tau_1 \tau_2 x_2 \vee \tau_1 \tau_2 \bar{x}_3 \text{ (строки 3, 5);}$$

$$D_4 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{\tau}_1 \bar{\tau}_2 \bar{x}_1 \vee \tau_1 \tau_2 \bar{x}_2 \bar{x}_3 \text{ (строки 1, 2, 5). (15)}$$

Формирование содержимого блока микроопераций сводится к формированию таблицы со столбцами  $a_m$ ,  $K(a_m)$ ,  $Y(a_m)$ ,  $\tau_m$ ,  $m$ . Здесь  $Y(a_m) \subseteq Y$  – набор микроопераций, формируемый в состоянии  $a_m \in A$ . Эта информация содержится в операторных вершинах ГСА  $\Gamma$ , а в нашем примере задана системой (13). Столбец  $\tau_m$  содержит переменные  $\tau_r \in \tau$ , равные единице в коде  $K(B_i)$ , формируемом блоком *LUTer* и представленном в табл. 2.

Пусть *LUT* элементы имеют  $S_L = 4$  входа. В этом случае для реализации блока *LUTer* достаточно  $N + R_C = 9$  элементов. В общем случае применение предложенного метода целесообразно, если

$$S_L \geq R. \quad (16)$$

**Таблица 2.** Таблица блока *LUTer* автомата  $U_4(\Gamma_1)$

$a_m$	$K(a_m)$	$Y(a_m)$	$\tau_m$	$m$	$a_m$	$K(a_m)$	$Y(a_m)$	$\tau_m$	$M$
$a_1$	0000	–	–	1	$a_8$	1110	$y_2 y_3 y_5$	–	8
$a_2$	0001	$y_1$	$\tau_1 \tau_2$	2	$a_9$	1010	$y_2$	–	9
$a_3$	1101	$y_1 y_3 y_7$	$\tau_1 \tau_2$	3	$a_{10}$	0110	$y_2 y_6$	–	10
$a_4$	0010	$y_2 y_6$	–	4	$a_{11}$	1100	$y_3 y_4 y_7$	$\tau_1$	11
$a_5$	0100	$y_4$	$\tau_2$	5	$a_{12}$	1001	$y_1 y_7$	$\tau_1$	12
$a_6$	0111	$y_5 y_6$	$\tau_2$	6	$a_{13}$	1000	$y_7$	$\tau_1$	13
$a_7$	1111	$y_3 y_5$	$\tau_2$	7	–	****	***	*	*

Реализация схемы МПА  $U_4(\Gamma_1)$  сводится к соответствующему программированию блоков *PLA* и *LUT* элементов. Этот этап в нашей статье не рассматривается.

Отметим, что для автомата  $U_1(\Gamma_1)$  прямая структурная таблица имеет  $H_1(\Gamma_1) = 37$  строк. Это приводит к нарушению условия (9) для блоков *PLA APEX 20K*. Для данных блоков  $q = 32$ . Нарушение условия (9) резко усложняет процесс синтеза и ведет к увеличению числа *PLA* в схеме автомата.

**Заключение.** Технология гибридных *FPGA* требует адаптации методов синтеза МПА, ориентированных на базис *FPGA*. Отсутствие встроенных блоков памяти вызывает необходимость реализации системы микроопераций на табличных элементах типа *LUT*. При выполнении условия (16) для реализации каждой микрооперации достаточно использовать только один *LUT*. Система функций возбуждения памяти реализуется на встроенных блоках *PLA*.

В статье предлагается метод уменьшения аппаратных затрат в схеме МПА Мура, основанный на использовании двух источников классов псевдоэквивалентных состояний. Это возможно, благодаря большому числу входов *PLA*. Например, в гибридных *FPGA APEX20K* фирмы *Altera* этот параметр равен 32.

Применение предложенного метода целесообразно при выполнении условий (8), (9) и (16). При выполнении условий (8) – (9) схема формирования функций возбуждения памяти реализуется на одном блоке *PLA*. При выполнении условия (16) блок микроопераций включает минимальное число *LUT* элементов. Анализ стандартных ГСА из библиотеки [12] показал, что условия (8), (9) и (16) выполняются для 87 процентов примеров (для *FPGA APEX 20K*).

Научная новизна предложенного метода заключается в учете особенностей автомата Мура и элементного базиса гибридных *FPGA* для уменьшения аппаратурных затрат в схеме автомата.

Практическая значимость метода заключается в снижении стоимости схемы автомата Мура на гибридных *FPGA* в сравнении с известными аналогами.

1. *Baranov S.* Logic Synthesis for Control Automata. – Boston: Kluwer Acad. Publ., 1994. – 312 p.
2. *DeMicheli G.* Synthesis and Optimization of Digital Circuits. – New York: McGraw-Hill, 1994. – 636 p.
3. *Соловьев В.В.* Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.

4. *Грушницкий Р.И., Мурсаев А.Х., Угрюмов Е.П.* Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ – Петербург, 2002. – 608 с.
5. *Skliarova I., Sklyarov U., Sudnitson A.* Design of FPGA-based circuits using Hierarchical Finite State Machines. – Tallinn: TUT Press, 2012. – 240 p.
6. *Czerwinski R., Kania D.* Finite State Machine Logic Synthesis for complex programmable Logic Devices. – Berlin: Springer, 2013. – 172 p.
7. *Kabiani A., Brown S.* The Hybrid Field Programmable Architecture // IEEE Design & Test of Computers. – 1999. – 16, N 4. – P. 74–83.
8. *Altera Corporation APEX20K PLD Family Data Sheet (2004).* – www.altera.com.
9. *Баркалов А.А., Цололо С.А.* Оптимизация схемы автомата Мура в составе системы на кристалле // Радиоэлектроника и информатика. – 2007. – № 1. – С. 35–39.
10. *Баркалов А.А., Цололо С.А.* Оптимизация числа макроячеек *PAL* в схеме автомата Мура // УСиМ. – 2008. – № 2. – С. 54–59.
11. *Баркалов А.А.* Принципы оптимизации логической схемы микропрограммного автомата Мура // Кибернетика и системный анализ. – 1998. – № 1. – С. 65–72.
12. *Yang S.* Logic Synthesis and optimization benchmarks user guide. – Microelectronics Center of North Carolina, 1991. – 43 p.

Поступила 08.07.2014  
Тел. для справок: +38 062 301-0723 (Донецк)  
E-mail: s.solos@gmail.com  
© С.А. Цололо, 2014