

# КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

A.A. Barkalov, G. Bazydlo,  
Y.E. Vizor, A.V. Matvienko,  
L.A. Titarenko

## IMPLEMENTING CIRCUIT OF CONTROL UNIT WITH TWO SOURCES OF CODES

*The method is proposed for reducing the number of LUT elements in the circuit of Moore finite state machine. The method is based on the using two sources of pseudoequivalent states.*

*Key words: Moore finite state machine, FPGA, LUT, EMB, synthesis.*

*Пропонується метод зменшення числа LUT елементів у схемі автомата Мура. Метод заснований на використанні двох джерел кодів класів псевдоеквівалентних станів.*

*Ключові слова: мікропрограмний автомат Мура, FPGA, LUT, EMB, синтез.*

*Предлагается метод уменьшения числа LUT элементов в схеме автомата Мура. Метод основан на использовании двух источников кодов классов псевдоэквивалентных состояний.*

*Ключевые слова: микропрограммный автомат Мура, FPGA, LUT, EMB, синтез.*

© А.А. Баркалов, Г. Базыдло,  
Я.Е. Визор, А.В. Матвиенко,  
А.А. Титаренко, 2014

УДК 004.274

А.А. БАРКАЛОВ, Г. БАЗЫДЛО, Я.Е. ВИЗОР,  
А.В. МАТВИЕНКО, Л.А. ТИТАРЕНКО

## РЕАЛИЗАЦИЯ СХЕМЫ УСТРОЙСТВА УПРАВЛЕНИЯ С ДВУМЯ ИСТОЧНИКАМИ КОДОВ

**Введение.** При реализации схем устройств управления (УУ), часто используется модель микропрограммного автомата (МПА) Мура [1, 2]. Важной задачей, возникающей при синтезе УУ, является уменьшение числа логических элементов в схеме УУ [3, 4]. Решение этой задачи позволяет уменьшить число межсоединений в схеме и потребляемую мощность. В настоящее время для реализации схем цифровых систем используются микросхемы *FPGA* (field-programmable logic arrays) [5, 6]. В настоящей работе рассматривается метод уменьшения площади кристалла, занимаемой схемой МПА Мура, ориентированный на *FPGA*. При этом алгоритм управления представлен в виде граф-схемы алгоритма Г [1].

Как правило *FPGA* включают элементы табличного типа *LUT* (look-up table) и встроенные блоки памяти *EMB* (embedded memory blocks) [5, 6]. Число входов  $S$  логических элементов *LUT* ограничено ( $S \leq 6$ ). Существует возможность связи выхода *LUT* с программируемым триггером. Блоки *EMB* обладают свойством реконфигурации, что позволяет менять число ячеек памяти ( $V$ ) и их выходов ( $t_F$ ). При этом емкость блока  $V_0$  остается постоянной. Существуют следующие конфигурации *EMB*:  $16K \times 1$ ,  $8K \times 2$ ,  $4K \times 4$ ,  $2K \times 8$ ,  $1K \times 16$ ,  $512 \times 32$  [5, 6]. Это означает, что  $S_A \in \{14, 13, 12, 11, 10, 9\}$  и  $t_F \in \{1, 2, 4, 8, 16, 32\}$ , где  $S_A$  – число адресных разрядов блока.

При синтезе схем УУ необходимо учитывать как особенности модели МПА, так и элементного базиса. В настоящей работе

предлагается использовать такие особенности МПА Мура, как наличие классов псевдоэквивалентных состояний (ПЭС) и регулярность системы микроопераций [1]. Первая особенность позволяет использовать более, чем один источник кодов состояний [3, 4]. Вторая – позволяет использовать блоки *EMB* для реализации системы микроопераций. Небольшое число входов логических элементов *LUT* требует модификации структур МПА и методов их синтеза по сравнению с их аналогами [7, 8]

**Реализация МПА Мура на *FPGA*.** Автомат Мура можно охарактеризовать множествами  $X, Y, A$  и двумя функциями – переходов и выходов [1]. Здесь  $X=\{x_1, \dots, x_L\}$  – множество логических условий (ЛУ),  $Y=\{y_1, \dots, y_N\}$  – множество микроопераций (МО) и  $A=\{a_1, \dots, a_M\}$  – множество состояний. Как правило, в МПА выделяется начальное состояние  $a_1 \in A$ . Функции переходов и выходов представляются прямой структурной таблицей (ПСТ), имеющей следующие столбцы:

$a_m$  – 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  $K(a_s)$ ;  $h = 1, \dots, H$  – номер строки таблицы. В столбце  $a_m$  записывается набор микроопераций  $Y(a_m) \subseteq Y$ , формируемых в состоянии  $a_m \in A$ . Для кодирования состояний  $a_m \in A$  используется множество внутренних переменных  $T = \{T_1, \dots, T_R\}$ , где  $R = \lceil \log_2 M \rceil$ . Функции возбуждения памяти МПА образуют множество  $\Phi = \{D_1, \dots, D_R\}$ , т. е. при синтезе используются триггеры типа *D* [9].

Логическая схема МПА задается системой уравнений

$$\Phi = \Phi(T, X), \tag{1}$$

$$Y = Y(T). \tag{2}$$

Системы (1) – (2) формируются на основе прямой структурной таблицы по правилам [1] и определяют модель *PY* автомата Мура (рис. 1).

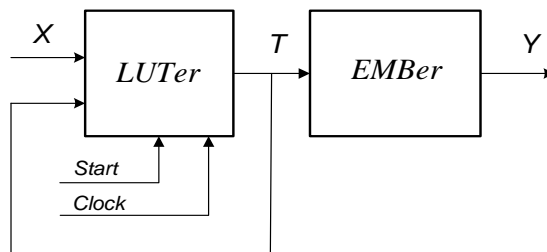


РИС. 1. Структурная схема *PY* автомата Мура

В этой схеме блок *LUTer* состоит из *LUT* элементов, реализующих (1). В состав блока входят *R* триггеров, обнуляемых по сигналу *Start*. Изменение кода  $K(a_m)$  происходит по сигналу синхронизации *Clock*. Выходы блока *LUTer* представляют собой внутренние переменные  $T_r \in T$ . Блок *EMBer* состоит из встроенных блоков *EMB*, реализующих систему (2).

Как правило, число переходов  $H_1(\Gamma)$  больше числа переходов  $H_0(\Gamma)$  эквивалентного автомата Мили [1]. Это приводит к росту аппаратных затрат в схеме МПА Мура, по сравнению с этим показателем для эквивалентного автомата Мили. Параметр  $H_1(\Gamma)$  можно уменьшить, благодаря наличию псевдоэквивалентных состояний (ПЭС) МПА Мура [10]. Состояния  $a_m, a_s \in A$  называются 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

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

где булевская переменная  $C_{mi}$  равна 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  $P_0Y$  автомата, структура которой совпадает со структурой  $PY$  автомата, но число термов совпадает с  $H_0(\Gamma)$ . Однако такое кодирование не всегда возможно [10] из-за особенностей ГСА.

Для уменьшения числа строк ПСТ можно использовать преобразование кодов состояний  $a_m \in A$  в коды классов псевдоэквивалентных состояний  $K(B_i)$ . 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 для представления МПА используется модель  $P_cY$  (рис. 2).

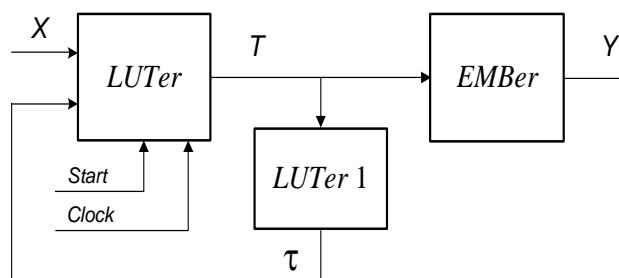


РИС. 2. Структурная схема  $P_cY$  автомата Мура

Здесь блок *LUTer* реализует систему функций  $\Phi = \Phi(\tau, X)$ .

Блок *LUTer1* – это преобразователь кодов  $K(a_m)$  в коды классов  $K(B_i)$ . *LUTer1* реализует систему функций

$$\tau = \tau(T). \quad (4)$$

В работе [10] показано, что  $H_3(\Gamma) = H_0(\Gamma)$ . Однако блок *LUTer1* потребляет некоторые ресурсы *FPGA*. Предлагаемый метод, позволяет сохранить положительные качества *PCY* автомата и удалить блок *LUTer1*.

**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:

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

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

$$R_C = \lceil \log_2(I_C) \rceil,$$

где  $I_C = |\Pi_C|$ .

Пусть следующее условие выполняется для блоков *EMB*:

$$\left\lceil \frac{N}{t_F} \right\rceil = \left\lceil \frac{N + R_C + 1}{t_F} \right\rceil. \quad (5)$$

При этом функции (4) могут быть реализованы блоком *EMBer*, что устраняет необходимость в блоке *LUTer1*.

Представим множество  $\Phi$  в виде объединения множеств  $\Phi_1$  и  $\Phi_2$ , где  $\Phi_1 \cap \Phi_2 = \emptyset$ . Этим множествам соответствуют функции:

$$\Phi_1 = \Phi_1(T, X), \quad (6)$$

$$\Phi_2 = \Phi_2(\tau, X). \quad (7)$$

На рис. 3 показана структурная схема предлагаемого *P<sub>2</sub>Y* автомата.

В этой схеме блок *LUTer1* реализует систему (6), а блок *LUTer2* – систему (7). Блок *EMBer* реализует системы функций (2) и (4). Блок мультиплексора *MX* реализуется на *LUT*. Блок *MX* служит для выбора функций  $\Phi_1$  и  $\Phi_2$  в качестве кода состояния перехода. При этом выбор определяется переменной  $y_M$ :

$$T = y_M \Phi_1 \vee \bar{y}_M \Phi_2.$$

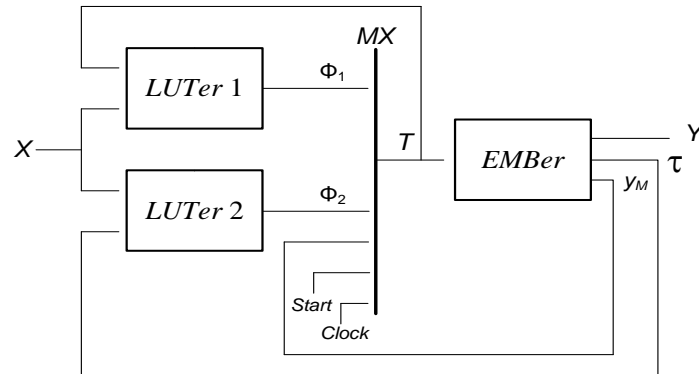


РИС. 3. Структурная схема  $P_2Y$  автомата Мура

Наличие единицы в (5) объясняется необходимостью резервирования выхода  $EMBer$  для переменной  $y_m$ .

Далее приведен метод синтеза предложенного  $P_2Y$  автомата Мура по исходной ГСА Г.

**Предлагаемый метод синтеза.** Данный метод синтеза включает следующие этапы:

1. Формирование разбиения  $\Pi_A = \{B_1, \dots, B_l\}$ .
2. Оптимальное кодирование состояний  $a_m \in A$ .
3. Формирование множеств  $\Pi_B$  и  $\Pi_C$ .
4. Кодирование классов  $B_i \in \Pi_C$ .
5. Формирование таблиц блоков  $LUTer1$  и  $LUTer2$ .
6. Формирование таблицы блока  $EMBer$ .
7. Формирование системы уравнений для блока  $MX$ .
8. Реализация схемы в заданном элементном базисе.

Рассмотрим пример применения предложенного метода. В целях экономии используем задание автомата не в виде ГСА, а в виде системы обобщенных формул перехода (ОФП) [6, 10]. Пусть автомат Мура  $S_1$  определяется следующей системой ОФП:

$$\begin{aligned}
 B_1 &\rightarrow x_1 a_2 \vee \overline{x_1 a_3}; & B_4 &\rightarrow x_5 a_8 \vee \overline{x_5 x_6 a_{11}} \vee \overline{\overline{x_5 x_6 a_{13}}}; \\
 B_2 &\rightarrow x_2 a_4 \vee \overline{x_2 x_3 a_5} \vee \overline{\overline{x_2 x_3 a_6}}; & B_5 &\rightarrow a_{10}; \\
 B_3 &\rightarrow x_3 a_6 \vee \overline{x_2 x_4 a_8} \vee \overline{\overline{x_3 x_4 a_1}}; & B_6 &\rightarrow \overline{x_1 a_{12}} \vee \overline{x_1 a_1}; \\
 B_7 &\rightarrow x_4 a_1 \vee \overline{x_4 x_5 a_7} \vee \overline{\overline{x_4 x_5 x_6 a_{11}}} \vee \overline{\overline{x_4 x_5 x_6 a_{13}}}.
 \end{aligned}$$

Пусть при этом получено разбиение  $\Pi_A = \{B_1, \dots, B_l\}$ , 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  $S_1$  включает  $N = 13$  элементов. Очевидно, что для  $S_1$  имеем сле-

дующие множества и параметры:  $A = \{a_1, \dots, a_{13}\}$ ,  $M = 13$ ,  $R = 4$ ,  $\Phi = \{D_1, \dots, D_4\}$ ,  $T = \{T_1, \dots, T_4\}$ ,  $I = 7$ . Сформируем систему (3), которая в случае автомата  $S_1$  имеет следующий вид:

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

Один из возможных вариантов оптимального кодирования состояний для автомата  $S_1$  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  $S_1$

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:  $K(B_2) = 00$ ,  $K(B_4) = 01$ ,  $K(B_7) = 10$ . Из карты Карно имеем  $K(B_1) = 0000$ ,  $K(B_3) = 001*$ ,  $K(B_5) = 1*10$ ,  $K(B_6) = 0110$ .

Пусть для реализации схем используется *FPGA*, в состав которой входят *EMB* с конфигурацией  $16 \times 16$ . В этом случае условие (5) выполняется и модель  $P_2Y$  автомата  $S_1$  может быть использована.

Таблицы блоков *LUTer1* и *LUTer2* включают следующие столбцы:  $B_i, K(B_i), a_s, K(a_s), X_h, \Phi_h^1, h$ . В табл. 1 и табл. 2 приведены фрагменты таблиц блоков *LUTer1* и *LUTer2* соответственно автомата  $S_1$ .

Как следует из этих таблиц, можно найти следующие множества  $\Phi_1 = \{D_1^1, \dots, D_4^1\}$  и  $\Phi_2 = \{D_1^2, \dots, D_4^2\}$ .

Формирование содержимого *EMBer* сводится к формированию таблицы со столбцами  $a_m, K(a_m), Y(a_m), \tau_m, u_m, m$ . Здесь  $Y(a_m) \subseteq Y$  – набор микроопераций формируемый в состоянии  $a_m \in A$ . Эта информация содержится в операторных вершинах ГСА Г. Столбец  $\tau_m$  содержит переменные  $\tau_r \in \tau$ . В столбце  $u_m$  записывается 1, если  $a_m \in B_i$  и  $B_i \in \Pi_B$ . Эта таблица строится элементарным образом и мы ее не рассматриваем.

ТАБЛИЦА 1. Фрагмент таблицы блока  $LUTer1 P_2Y$  автомата  $S_1$

$B_i$	$K(B_i)$	$a_s$	$K(a_s)$	$X_h$	$\Phi_h^1$	$h$
	$T_1 T_2 T_3 T_4$					
$B_1$	0000	$a_2$	0001	$x_1$	$D_4^1$	1
		$a_3$	1101	$\bar{x}_1$	$D_1^1 D_2^1 D_4^1$	2
$B_3$	001X	$a_6$	0111	$x_3$	$D_2^1 D_3^1 D_4^1$	3
		$a_8$	1110	$\bar{x}_3 x_4$	$D_1^1 D_2^1 D_3^1$	4
		$a_1$	0000	$\bar{x}_3 \bar{x}_4$	–	5

ТАБЛИЦА 2. Фрагмент таблицы блока  $LUTer2 P_2Y$  автомата  $S_1$

$B_i$	$K(B_i)$	$a_s$	$K(a_s)$	$X_h$	$\Phi_h^2$	$h$
	$\tau_1 \tau_2$					
$B_2$	00	$a_4$	0010	$x_2$	$D_3^2$	1
		$a_5$	0100	$\bar{x}_2 x_3$	$D_2^2$	2
		$a_6$	0111	$\bar{x}_2 \bar{x}_3$	$D_2^2 D_3^2 D_4^2$	3
$B_4$	01	$a_8$	1110	$x_5$	$D_1^2 D_2^2 D_3^2$	4
		$a_{11}$	1100	$\bar{x}_5 x_6$	$D_1^2 D_2^2$	5
		$a_{13}$	1000	$\bar{x}_5 \bar{x}_6$	$D_1^2$	6

Система уравнений блока  $MX$  содержит  $R$  уравнений, имеющих следующий вид:

$$D_r = y_M D_r^1 \vee \bar{y}_M D_r^2 \quad (r = 1, R).$$

В рассматриваемом примере  $R = 4$ . Следовательно, для реализации блока  $MX$  достаточно 4 элементов  $LUT$ , имеющих по 3 входа.

Последний этап предлагаемого метода сводится к использованию стандартных пакетов [6] и не рассматривается в данной работе.

**Заключение.** Особенности технологии  $FPGA$  требуют адаптации к ним известных методов синтеза МПА. Для уменьшения числа  $LUT$  элементов в схеме нужно использовать характерные черты как модели МПА, так и элементного базиса. Это уменьшает как число  $LUT$  элементов, так и число межсоединений в схеме автомата.

В работе предлагается метод снижения аппаратных затрат в схеме МПА Мура. Метод основан на использовании двух источников кодов классов псевдоэквивалентных состояний. Такой подход позволяет уменьшить число термов в

системе функций возбуждения памяти. Часть термов включает в себя конъюнкции кодов состояний, имеющих меньше  $R$  переменных. Это ведет к дополнительному уменьшению числа  $LUT$  элементов в схеме автомата. При выполнении условия (5) система микроопераций реализуется на одном блоке  $EMB$ .

Недостатком предложенного метода является наличие мультиплексора  $MX$ . Этот блок вносит дополнительную задержку к времени распространения сигнала, по сравнению с  $PY$  автоматом. Однако уменьшение числа уровней в схеме формирования функций  $\Phi_1$  и  $\Phi_2$  (по сравнению с  $\Phi$ ) может компенсировать этот недостаток.

Для сравнения различных методов синтеза МПА используется библиотека стандартных ГСА [9]. Анализ этой библиотеки показал, что условие (5) выполняется для 95 % автоматов, т. е. для 95 % стандартных примеров система микроопераций реализуется на одном блоке встроенной памяти. Дальнейшее направление наших исследований заключается в разработке САПР для синтеза  $P_2Y$  автоматов.

Научная новизна предложенного метода заключается в учете особенностей автомата Мура и элементного базиса  $FPGA$  для уменьшения аппаратных затрат в схеме автомата.

Практическая значимость заключается в уменьшении стоимости схемы автомата Мура на  $FPGA$  по сравнению с известными из литературы аналогами.

1. *Baranov S.* Logic Synthesis for Control Automata. – Dordrecht: Kluwer Academic Publishers, 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. *Skliarova I., Sklyarov V., Sudnitson A.* Design of FPGA-based circuits using Hierarchical Finite State Machines. – Tallinn: TUT Press, 2012. – 240 p.
5. *Грушницкий Р.И., Мурсаев А.Х., Узрюмов Е.П.* Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608 с.
6. *Sklyarov V., Skliarova I., Barkalov A., Titarenko L.* Synthesis and Optimization of FPGA-based Systems. – Berlin: Springer, 2014. – 432 p.
7. *Баркалов А.А., Цололо С.А.* Оптимизация схемы автомата Мура в составе системы на кристалле // Радиоэлектроника и информатика. – 2007. – № 1. – С. 35 – 39.
8. *Баркалов А.А., Цололо С.А.* Оптимизация числа макроячеек PAL в схеме автомата Мура // Управляющие системы и машины. – 2008. – № 2. – С. 54 – 59.
9. *Yang S.* Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina. – 1991. – 43 p.
10. *Баркалов А.А.* Принципы оптимизации логической схемы микропрограммного автомата Мура // Кибернетика и системный анализ. – 1998, № 1. – С. 65 – 72.

Получено 01.09.2014