

УДК 004.274

**А.А. Баркалов (д-р техн. наук, проф.)<sup>1,2</sup>,  
Р.В. Мальчева (канд. техн. наук, доц.)<sup>2</sup>, А.А. Баркалов<sup>3</sup>**

- 1) Университет Зеленогурский, Польша,  
2) ГБУЗ «Донецкий национальный технический университет», г. Донецк  
кафедра компьютерной инженерии  
3) Институт кибернетики НАН Украины, г. Донецк  
E-mail: [raisa@cs.dgtu.donetsk.ua](mailto:raisa@cs.dgtu.donetsk.ua)

## УМЕНЬШЕНИЕ ЧИСЛА LUT-ЭЛЕМЕНТОВ В СХЕМЕ АВТОМАТА МИЛИ

*Предложен метод уменьшения аппаратных затрат в схеме микропрограммного автомата Мили, ориентированный на технологию FPGA. Метод основан на использовании модели PR-автомата и реализации системы микроопераций на встроенных блоках памяти ЕМВ. Такой подход позволяет уменьшить число LUT элементов в схеме автомата. Приведены условия применения предложенного метода.*

**Ключевые слова:** микропрограммный автомат, PR-автомат, FPGA, LUT, ЕМВ, синтез.

### Введение

Модель микропрограммного автомата (МПА) Мили широко применяется для реализации схем устройств управления [1]. В настоящее время для реализации цифровых систем используется базис программируемых логических интегральных схем типа FPGA (field-programmable gate arrays) [2,3]. Как правило, микросхемы FPGA включают элементы табличного типа, называемые LUT (look-up table), и встроенные блоки памяти ЕМВ (embedded memory blocks) [4,5]. Одной из важных проблем, возникающих при реализации схем МПА в базисе FPGA, является проблема уменьшения цикла LUT элементов [6]. Решение этой задачи позволяет уменьшить число межсоединений в схеме. Это, в свою очередь, приводит к увеличению быстродействия и уменьшению потребления энергии [7]. Одним из путей решения этой задачи является использование ЕМВ для реализации системы микроопераций [8]. Однако это связано с кодированием наборов и использования части кристалла для формирования кодирующих переменных. В настоящей работе предлагается использовать модель PR-автомата [9] для реализации схем МПА Мили на FPGA. Эта модель была предложена в 90-х годах и ориентирована на базис программируемых логических матриц. Таким образом, метод синтеза должен быть адаптирован к особенностям FPGA.

*Целью исследования* является уменьшения числа LUT элементов в схеме МПА Мили за счет применения модели PR-автомата.

*Задачей исследования* является адаптация методов синтеза PR-автоматов к особенностям базиса FPGA.

### Реализации автомата Мили на FPGA

Схема МПА Мили задается двумя системами функций [1]:

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

$$Y = Y(T, X), \tag{2}$$

где  $X = \{x_1, \dots, x_L\}$  - множество входных переменных (логических условий);  $Y = \{y_1, \dots, y_N\}$  - множество выходных переменных (микроопераций);  $T = \{T_1, \dots, T_R\}$  - множество внутренних переменных, кодирующих состояние  $a_m \in A$ , где  $A = \{a_1, \dots, a_M\}$  - множество состояний автомата;  $\Phi = \{D_1, \dots, D_R\}$  - множество функций возбуждения памяти МПА.

Число переменных  $T_r \in T$  определяется как

$$R = \lceil \log_2 M \rceil. \tag{3}$$

При реализации схем МПА Мили на FPGA системы (1) и (2) реализуются на LUT элементах. Обозначим совокупность LUT элементов символом LUTer, полученная схема МПА Мили приведена на рис.1.

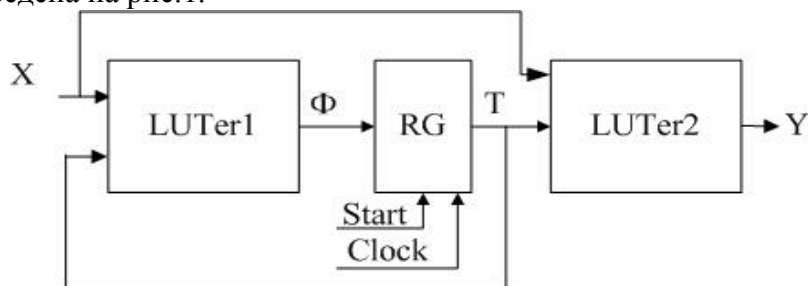


Рисунок 1 – Структурная схема автомата Мили на LUT элементах

В схеме блок LUTer1 реализует систему (1), а блок LUTer2 – систему (2). Регистр RG хранит код  $K(a_m)$  состояния МПА. По сигналу Start в RG записывается нулевой код начального состояния  $a_m \in A$ . По сигналу Clock происходит смена кодов состояния в RG.

При синтезе схем на FPGA производится раздельная минимизация, декомпозиция и реализация каждой функции систем (1) и (2) [10]. При этом блоки схем не имеют общих LUT элементов, что помогает упростить межсоединения.

Одним из путей уменьшения числа LUT элементов в схеме Мили является кодирование наборов микроопераций (НМО)  $Y_t \subseteq Y$  ( $t=1, T_0$ ). Наборы  $Y_t \subseteq Y$  формируются при переходах между состояниями автомата. Каждому набору ставится в соответствие двоичный код  $K(Y_t)$  разрядности

$$R_Y = \lceil \log_2 T_0 \rceil. \tag{4}$$

Для кодирования НМО используются дополнительные переменные  $z_r \in Z$ , где  $(|Z| = R_Y)$ . Обозначим схему, приведенную на рис.1, символом  $U_1$ . Применение кодирования НМО приводит к модели  $U_2$  (рис.2).

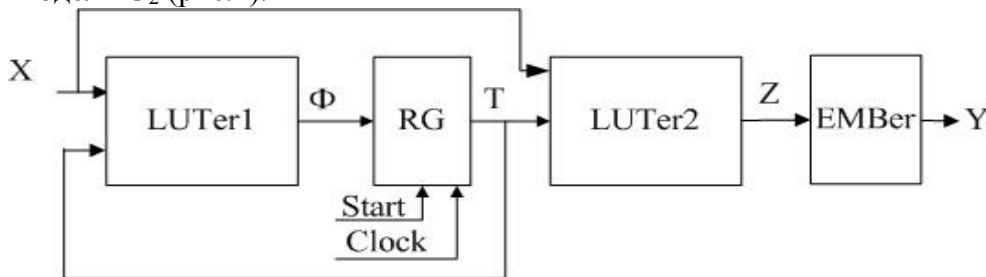


Рисунок 2 – Структурная схема автомата Мили  $U_2$

В автомате  $U_2$  блок LUTer2 реализует систему функций

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

Блоки LUTer1 и LUTer2 в автоматах  $U_1$  и  $U_2$  срабатывают параллельно. Блок EMBer состоит из встроенных блоков памяти, реализующих систему функций

$$Y = Y(Z). \quad (6)$$

Недостатком модели  $U_2$  является наличие блока LUTer2. Введение блока LUTer2 имеет смысл, если выполняется условие

$$R_Y \ll N. \quad (7)$$

В противном случае аппаратные затраты в блоках LUTer2 моделей  $U_1$  и  $U_2$  могут быть достаточно близкими. Кроме того, блок EMBer вносит дополнительную задержку во время формирования выходных функций.

Для устранения блока LUTer2 предлагается использовать модель PR-автомата [9]. Рассмотрим особенности реализации PR-автомата на FPGA.

#### Реализация PR-автомата на FPGA

В PR-автомате набор  $Y_t \subseteq Y$  определяется парой состояний  $\langle a_m, a_s \rangle$ , где  $a_m \in A$  - исходное состояние и  $a_s \in A$  - состояние перехода [9]. Очевидно, элементы этой пары должны быть представлены различными переменными. В настоящей работе предлагается структурная схема автомата  $U_3$  (рис.3).

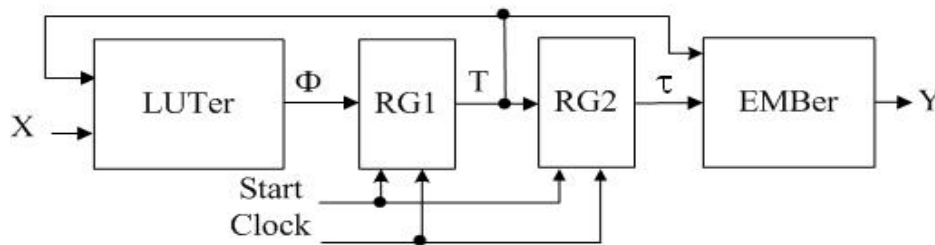


Рисунок 3 – Структурная схема автомата Мили  $U_3$

В автомате  $U_3$  регистр RG1 хранит код  $K(a_m)$  исходного состояния, а регистр RG2 – код  $K(a_s)$  состояния перехода. Для представления кодов в RG2 используются переменные  $\tau_r \in \tau$ , где  $|\tau|=R$ . По сигналу Start в оба регистра записывается код начального состояния  $a_1 \in A$ . В каждом такте схема LUTer формирует функции (1), определяющие код состояния перехода. По сигналу Clock код исходного состояния переписывается в RG2, а код состояния перехода записывается в RG1. Блок EMBer формирует выходные функции, определяемые следующим образом:

$$Y = Y(T, \tau). \quad (8)$$

Функционирование продолжается аналогичным образом до записи в RG1 кода состояния  $a_1 \in A$ . Дальнейшая работа МПА возможна только по очередному сигналу Start.

Сравнение автоматов  $U_1 - U_3$  позволяет сделать следующие выводы:

1. В автомате  $U_3$  блок LUTer реализует только  $R$  функций, что является минимально возможной величиной.
2. Автомат  $U_3$  обладает большим быстродействием, чем автомат  $U_2$ . Это связано с исключением промежуточного блока LUTer2.
3. Для устойчивой работы автомата  $U_1$  необходим дополнительный  $N$ -разрядный регистр для записи микроопераций [1]. В автомате  $U_2$  роль дополнительного регистра выполняет RG2.

В настоящей работе предлагается метод синтеза автомата  $U_3$  по граф-схеме алгоритма (ГСА). Метод включает следующие этапы:

1. Формирование множества состояний  $A$ .
2. Кодирование состояний  $a_m \in A$ .
3. Формирование прямой структурной таблицы (ПСТ) автомата Мили.

4. Формирование таблицы блока LUTer.
5. Формирование таблицы блока EMVer.
6. Реализация схемы автомата в заданном базисе.

**Особенности реализации PR-автомата на FPGA**

Рассмотрим пример синтеза PR-автомата для некоторого автомата  $S_1$ , заданного прямой структурной таблицей (табл.1).

Таблица 1

Прямая структурная таблица автомата  $S_1$

$a_m$	$K(a_m)$	$a_s$	$K(a_s)$	$X_h$	$Y_h$	$\Phi_h$	$h$
$a_1$	00	$a_2$	01	$x_1$	$y_1y_2$	$D_2$	1
		$a_3$	10	$\bar{x}_1$	$y_3$	$D_1$	2
$a_2$	01	$a_3$	10	$x_2$	$y_3$	$D_2$	3
		$a_4$	11	$\bar{x}_2$	$y_2y_4$	$D_1D_2$	4
$a_3$	10	$a_2$	01	$x_1$	$y_2$	$D_2$	5
		$a_3$	10	$\bar{x}_1 x_3$	$y_1y_4$	$D_1$	6
		$a_4$	11	$\bar{x}_1 \bar{x}_3$	$y_5$	$D_1D_2$	7
$a_4$	11	$a_1$	00	1	$y_3$	-	8

Эта таблица строится по правилам [1], которые мы здесь не рассматриваем. Этапы 1 и 2 также выполняются по методике [1]. Рассмотрим этапы 4 и 5 предлагаемого метода.

Таблица блока LUTer строится по исходной ПСТ. Обе таблицы включают  $N=8$  строк. Для построения таблицы блока LUTer достаточно удалить из исходной ПСТ столбец  $Y_h$ , содержащий наборы  $Y_t \subseteq Y$  (табл.2). Табл.2 используется для формирования функций (1).

Таблица 2

Таблица блока LUTer PR-автомата  $S_1$

$a_m$	$K(a_m)$	$a_s$	$K(a_s)$	$X_h$	$\Phi_h$	$h$
$a_1$	00	$a_2$	01	$x_1$	$D_2$	1
		$a_3$	10	$\bar{x}_1$	$D_1$	2
$a_2$	01	$a_3$	10	$x_2$	$D_1$	3
		$a_4$	11	$\bar{x}_2$	$D_1D_2$	4
$a_3$	10	$a_2$	01	$x_1$	$D_2$	5
		$a_3$	10	$\bar{x}_1 x_2$	$D_1$	6
		$a_4$	11	$\bar{x}_1 \bar{x}_2$	$D_2D_1$	7
$a_4$	11	$a_1$	00	1	—	8

Для уменьшения числа LUT элементов каждая из функций  $D_r \in \Phi$  должна быть проминимизирована. Например, с учетом минимизации из табл.2 можно получить следующее уравнение:  $D_1 = \bar{T}_1 \bar{T}_2 \bar{x}_1 \vee \bar{T}_1 T_2 \vee T_1 \bar{T}_2 \bar{x}_1 = \bar{T}_2 \bar{x}_1 \vee \bar{T}_1 T_2$ . Это уравнение может быть реализовано на одном LUT элементе, имеющем не менее трёх входов. Для построения таблицы блока EMVer необходимо удалить из исходной ПСТ столбцы  $X_h$  и  $\Phi_h$ . Это дает следующую промежуточную таблицу (табл.3).

Таблица блока EMVer должна содержать  $N_1$  строк, где  $N_1 = 4^R$ . В рассматриваемом примере  $N_1 = 16$ . При этом только  $N=8$  содержат микрооперации (табл.4).

Таблица 3

Промежуточная таблица PR-автомата  $S_1$ 

$a_m$	$K(a_m)$	$a_s$	$K(a_s)$	$Y_h$	$h$
$a_1$	00	$a_2$	01	$y_1y_2$	1
		$a_3$	10	$y_3$	2
$a_2$	01	$a_3$	10	$y_3$	3
		$a_4$	11	$y_2y_4$	4
$a_3$	10	$a_2$	01	$y_2$	5
		$a_3$	10	$y_1y_4$	6
		$a_4$	11	$y_5$	7
$a_4$	11	$a_1$	00	$y_3y_4$	8

Таблица 4

Таблица блока EMBer PR-автомата  $S_1$ 

$K(a_s)$		$K(a_m)$		$Y$					$h$	Прим.
$\tau_1$	$\tau_2$	$T_1$	$T_2$	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$		
0	0	0	0	0	0	0	0	0	1	
0	0	0	1	0	0	0	0	0	2	
0	0	1	0	0	0	0	0	0	3	
0	0	1	1	0	0	1	1	0	4	8
0	1	0	0	1	1	0	0	0	5	1
0	1	0	1	0	0	0	0	0	6	
0	1	1	0	0	1	0	0	0	7	5
0	1	1	1	0	0	0	0	0	8	
1	0	0	0	0	0	1	0	0	9	2
1	0	0	1	0	0	1	0	0	10	3
1	0	1	0	1	0	0	1	0	11	6
1	0	1	1	0	0	0	0	0	12	
1	1	0	0	0	0	0	0	0	13	
1	1	0	1	0	1	0	1	0	14	4
1	1	1	0	0	0	0	0	1	15	7
1	1	1	1	0	0	0	0	0	16	

Эта таблица отображает состояния регистров  $RG_1$  и  $RG_2$  после осуществления перехода  $\langle a_m, a_s \rangle$ . При этом регистр  $RG_1$  содержит код состояния перехода  $a_s \in A$ , а регистр  $RG_2$  -- код исходного состояния  $a_m \in A$ . Около некоторых строк табл.4 записаны числа, соответствующие номерам строк промежуточной ПСТ (табл.3). Это помогает найти связи между исходной ПСТ и таблицей блока EMBer.

Последний этап предложенного метода связан с построением VHDL-моделей устройства управления и использованием стандартных потоков автоматизированного синтеза [5,6]. В данной статье этот этап не рассматривается.

Рассмотрим условия целесообразности использования предложенного метода. Современные микросхемы FPGA содержат блоки EMB с изменяемой конфигурацией [5,6]. Такие блоки имеют постоянную емкость  $V_0$  для различного числа ячеек  $V$  и выходов  $t_F$ :

$$V_0 = V \cdot t_F. \quad (9)$$

Число ячеек в блоке EMB определяется выражением (9). Число выходов  $t_F$  может принимать некоторые фиксированные значения, образующие множество  $O$  [5,6].

$$O = \{1, 2, 4, 8, 16, 32, 64\}.$$

Параметр  $t_F$  можно определить как ближайший элемент из множества  $O$ , удовлетворяющий условию

$$t_F \geq \left\lceil \frac{V_0}{4^R} \right\rceil. \quad (10)$$

После определения величины  $t_F$  можно найти число блоков ЕМВ, необходимое для реализации схемы блока ЕМВег:

$$n = \left\lceil \frac{N}{t_F} \right\rceil. \quad (11)$$

Если число  $n$  не превышает числа свободных блоков ЕМВ, то применение предложенного метода имеет смысл.

### **Заключение**

Предложенный метод реализации автомата Мили позволяет уменьшить число LUT элементов в схеме. Метод основан на применении модели PR-автомата и замене части LUT элементов схемой, содержащей встроенные блоки памяти. Основной особенностью PR-автомата является устранение непосредственной зависимости между входами (логическими условиями) и выходами (наборами микроопераций) автомата. Такой подход позволяет представить выходные функции в виде таблиц, реализуемых с использованием блоков ЕМВ.

Авторы проанализировали стандартные автоматы, представленные в библиотеке [11]. Оказалось, что для реализации системы микроопераций достаточно от одного до трёх блоков ЕМВ, имеющих конфигурацию  $1K \times 16$  (бит). При этом число LUT элементов в схеме автомата уменьшается, в среднем, на 28%.

Научная новизна предложенного метода заключается в адаптации метода синтеза PR-автомата к особенностям базиса FPGA.

Практическая значимость метода заключается в уменьшении LUT элементов в схеме МПА по сравнению с известными из литературы аналогами.

Дальнейшее направление работы связано с разработкой метода кодирования состояний, позволяющего уменьшить число LUT элементов в схеме формирования функций возбуждения памяти автомата.

### **Список использованной литературы**

1. Baranov S. Logic and System Design of Digital Systems / S. Baranov. - Tallinn: TUT Press, 2008. – 312 pp.
2. Grout I. Digital Systems Design with FPGAs and CPLDs / I. Grout – Amsterdam: Elsevier, 2008.–406 pp.
3. Barkalov A. Logic Synthesis for FSM – based Control Units / A. Barkalov, L Titarenko. – Berlin: Springer, 2009. – 233 pp.
4. [Электронный ресурс]. – Режим доступа: <http://www.altera.com>.
5. [Электронный ресурс]. – Режим доступа: <http://www.xilinx.com>.
6. Sklyarov V. Reconfigurable models of finite state machines and their implementation in FPGAs / V. Sklyarov // Journal of System Architecture. – 2002, V.47, №2. – PP. 1043-1064.
7. Chaudhury S. Genetic algorithm-based FSM synthesis with area-power trade-offs / S. Chaudhury, K. Sistla, S. Chattopadhyay // Integrated VLSI Journal. – 2009, V.42, №3. – pp. 376-384.
8. Barkalov A. Reduction in the number of LUT elements for control units with code shaving / A. Barkalov, L. Titarenko, J. Bieganowski // Journal of Applied Mathematics and Computer Science. – 2010. - V.12, №4. – PP. 751-761.

9. Баркалов А.А. Оптимизация логической схемы автомата Мили на ПЛИМ / А.А. Баркалов, D.K. Das // Автоматика и вычислительная техника. – 1992. - №3. – С. 90-94.
10. Scholl C. Function Decomposition with Application to FPGA Synthesis / C. Sholl. – Norwell: Kluwer Academic Publishers, 2001. – 418 p.
11. Yang S. Logic Synthesis and Optimization Benchmarks User Guide / S. Yang. – Technical Report. Microelectronic Center of North Carolina, 1991. – 44 p.

Надійшла до редакції:  
27.03.2013

Рецензент:  
д-р техн. наук, проф. Скобцов Ю.О.

**О.О. Баркалов, Р.В. Мальчева, О.О. Баркалов**

**Університет Зеленогурський, ДВНЗ «Донецький національний технічний університет»**

**Зменшення кількості LUT-елементів у схемі автомата Мілі.** Запропонований метод зменшення апаратних витрат у схемі мікропрограмного автомата Мілі, який орієнтований на технологію FPGA. Метод заснований на використанні моделі PR-автомата й реалізації системи мікрооперацій на вбудованих блоках пам'яті ЕМВ. Такий підхід дозволяє зменшити кількість LUT елементів у схемі автомата. Наведені умови застосування запропонованого метода. Виконаний аналіз реалізації системи мікрооперацій з використанням блоків ЕМВ, що мають конфігурацію 1К×16 (біт), для стандартних автоматів.

**Ключові слова:** мікропрограмний автомат, PR-автомат, FPGA, LUT, ЕМВ, синтез.

**A.A. Barkalov, R.V. Malcheva, A.A. Barkalov**

**University of Zielona, Donetsk National Technical University**

**Reduction in the Number of LUTs in Mealy Finite State Machine's Logic Circuit.** The model of Mealy finite state machine (FSM) is widely used for implementing the control units. Nowadays, the field – programmable gate arrays (FPGA) are applied for implementing complex digital systems. As a rule, the FPGAs include look-up table (LUT) elements and embedded memory blocks (EMB). One of the important problems connected with FSM design is the reduction of the number of LUTs in an FSM's logic circuit. The solution of this problem allows decreasing the number of interconnections among the LUTs. In turn, it leads to increasing of the performance and decreasing of the power dissipation. Using EMBs instead of LUTs is one of the possible ways for solving this problem. In the case of Mealy FSM, the system of microoperations could be implemented with EMBs. But it leads to the encoding of collections of microoperations and using some resources of a chip for generating these additional variables. A method is proposed for reducing the hardware amount in logic circuit of Mealy FSM. The method targets the technology of FPGA. The method is based on using the model of PR-automaton and implementing the system of microoperations with embedded memory blocks. This approach allows reducing the number of LUTs in the FSM's circuit. The conditions are shown for using the proposed method. The example of FSM synthesis is given with applying the proposed approach. The analysis of application of EMBs with the configuration 1Kx16 (bits) for implementing the system of microoperations for the standard benchmarks is done. An application of proposed method allows the average decrease for the number of LUTs up to 28%. The scientific novelty of the proposed method is reduced to adaptation of the design method for PR-automaton to the specifics of FPGAs. The practical meaning of the method is determined by reducing for the number of LUTs in an FSM logic circuit in comparison with known methods. The further direction of the research is connected with development of state assignment methods leading to decreasing of the number of LUTs in the circuit of LUTer.

**Keywords:** finite-state-machine, PR-automaton, FPGA, LUT, EMB, synthesis.