



**Ключевые слова:** *автомат Мура, ГСА, псевдоэквивалентные состояния, FPGA.*

#### **ВВЕДЕНИЕ**

Модель микропрограммного автомата (МПА) Мура широко используется при реализации устройств управления цифровых систем [1]. При проектировании схем МПА на СБИС важной задачей является уменьшение ресурсов кристалла, используемых схемой [2, 3]. Решение этой задачи позволяет уменьшить энергопотребление и в ряде случаев повысить быстродействие устройства [4]. Методы решения этой задачи во многом зависят от особенностей СБИС.

В настоящее время для реализации цифровых систем широко применяется СБИС типа FPGA (Field Programmable Gate Arrays) [5, 6]. Большинство микросхем FPGA включают три основных компонента: табличные логические элементы типа LUT (Look-up table), встроенные блоки памяти ЕМВ (Embedded memory blocks) и программируемую матрицу межсоединений. Как известно [7], LUT-элементы используются для реализации систем булевых функций (СБФ), представленных в аналитической форме, а ЕМВ — в табличной форме. Важной особенностью LUT-элементов является ограниченное число входов (порядка шести), что вызывает необходимость уменьшения числа аргументов и термов в реализуемых функциях. Выходы LUT-элементов связаны с входами программируемых триггеров, поэтому они идеально подходят для реализации функций возбуждения памяти МПА [3]. Особенностью ЕМВ является возможность изменения числа адресных входов и разрядности ячеек памяти при сохранении постоянной емкости блока [8]. Блоки ЕМВ идеально подходят для реализации системы микроопераций МПА Мура [3].

Особенностью МПА Мура является наличие псевдоэквивалентных состояний (ПЭС) [9]. Использование классов ПЭС позволяет уменьшить требуемое число LUT-элементов в схеме МПА [3]. Вторая особенность МПА Мура — зависимость выходных функций автомата только от его состояний [1]. Это позволяет использовать блоки ЕМВ для реализации системы микроопераций, что приводит к дальнейшему уменьшению числа LUT-элементов.

В настоящей статье рассматриваются два подхода к уменьшению числа LUT-элементов и блоков ЕМВ в схеме автомата Мура. Рассматриваемые методы модифицируют идеи, изложенные в работе [10]. Для представления реализуемых алгоритмов управления используется язык граф-схем алгоритмов (ГСА) [1].

## ОСНОВНЫЕ ПОЛОЖЕНИЯ И ИДЕИ ПРЕДЛАГАЕМЫХ МЕТОДОВ

Пусть по ГСА  $\Gamma$  построена прямая структурная таблица (ПСТ) со столбцами [3]:  $a_m, K(a_m), a_s, K(a_s), X_h, \Phi_h, h$ . Здесь  $a_m$  — исходное состояние автомата,  $a_m \in A$ , где  $A = \{a_1, \dots, a_M\}$  — множество состояний МПА;  $K(a_m)$  — код состояния  $a_m$  разрядности  $R = \lceil \log_2 M \rceil$ ;  $a_s, K(a_s)$  — соответственно состояние перехода и его код;  $X_h$  — конъюнкция некоторых элементов множества логических условий (ЛУ)  $X = \{x_1, \dots, x_L\}$  (или их отрицаний), определяющая переход из  $a_m$  в  $a_s \in A$ ;  $\Phi_h$  — набор функций возбуждения триггеров регистра состояний (Рг) автомата, принимающих единичные значения для переключения регистра из  $K(a_m)$  в  $K(a_s)$ ,  $\Phi \subseteq \Phi = \{D_1, \dots, D_R\}$ ;  $h = 1, H$  — номер перехода. Для кодирования состояний  $a_m$  используются внутренние переменные, образующие множество  $T = \{T_1, \dots, T_R\}$ . В столбце  $a_m$  ПСТ записывается набор микроопераций  $Y(a_m)$ , формируемый в состоянии  $a_m$ . При этом  $Y(a_m) \subseteq Y$ , где  $Y = \{y_1, \dots, y_N\}$  — множество микроопераций (МО). Эта таблица является основой для формирования систем функций

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

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

Функции (1), (2) задают схему МПА Мура, структура которой приведена на рис. 1. Условимся такой МПА Мура обозначать символом  $U_1$ . В МПА  $U_1$  схема формирования функций возбуждения (СФВ) реализует систему (1), а схема формирования микроопераций (СМО) — систему (2). По сигналу  $\text{Start} = 1$  в регистр Рг записывается нулевой код, что соответствует начальному состоянию МПА. Переключение регистра (смена состояний МПА) происходит по фронту импульса синхронизации  $\text{Clock}$ .

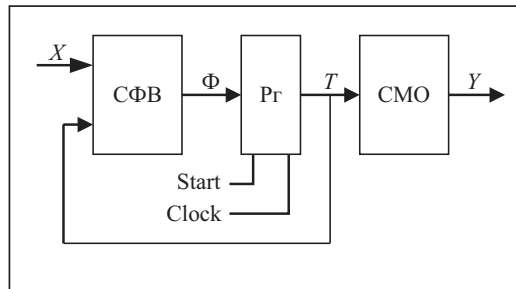


Рис. 1. Структурная схема МПА Мура  $U_1$

Как правило, регистр Рг имеет информационные входы типа  $D$  [3]. Поэтому множество  $\Phi$  было определено как  $\{D_1, \dots, D_R\}$ .

При реализации МПА  $U_1$  на FPGA блок СФВ реализуется на LUT-элементах, а блок СМО — на встроенных блоках памяти.

Недостатком автомата  $U_1$  является значительное число термов ( $H_1$ ), реализуемых СФВ. Как правило, параметр  $H_1$  значительно превосходит число термов  $H_0$  в системе (1) для эквивалентного автомата Мили [1]. Это приводит к тому, что схема СФВ автомата Мура потребляет значительно больше LUT-элементов, чем схема СФВ эквивалентного автомата Мили. Для уменьшения параметра  $H_1$  (и соответственно числа LUT-элементов в схеме СФВ) возможно использование таких методов, как оптимальное кодирование состояний или введение блока преобразователя кодов [3]. При оптимальном кодировании нет гарантии того, что число термов станет равно  $H_0$ , а при реализации с блоком преобразователя кодов схема будет занимать дополнительные ресурсы кристалла. В данной работе рассматриваются методы, гарантирующие уменьшение числа термов в системе (1) до  $H_0$  без введения преобразователя кодов.

Одной из особенностей МПА Мура является наличие псевдоэквивалентных состояний [8], т.е. состояний с одинаковыми переходами под воздействием одинаковых входных сигналов. Такие состояния соответствуют операторным вершинам [1] алгоритма управления, выходы которых связаны со входом одной и той же вершины алгоритма.

Пусть  $\Pi_A$  — разбиение множества  $A$  на классы псевдоэквивалентных состояний. Закодируем классы  $B_i \in \Pi_A$  двоичными кодами  $K(B_i)$  разрядности

$$R_B = \lceil \log_2 I \rceil. \quad (3)$$

Пусть исходная ГСА  $\Gamma$  включает  $Q$  попарно различных наборов микроопераций (НМО)  $Y_q \subseteq Y$ . Закодируем набор  $Y_q$  двоичным кодом  $K(Y_q)$  разрядности

$$R_Y = \lceil \log_2 Q \rceil. \quad (4)$$

Предположим также, что операторная вершина  $b_t$  ГСА  $\Gamma$  соответствует состоянию  $a_m \in B_i$  и в ней записан набор микроопераций  $Y_q$ . Тогда код состояния  $a_m \in A$  можно представить в виде

$$K(a_m) = K(B_i) * K(Y_q), \quad (5)$$

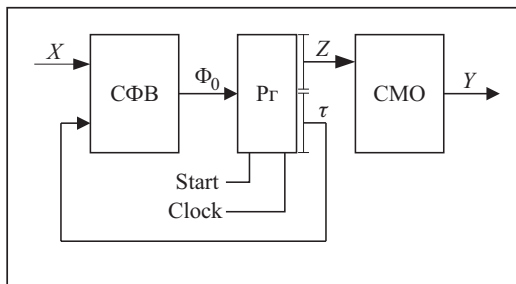


Рис. 2. Структурная схема МПА Мура  $U_2$

где символ  $*$  обозначает операцию конкатенации. При этом предполагается, что выполнена минимизация состояний автомата Мура [1] и наборы микроопераций являются попарно различными для состояний одного класса.

Представление кодов состояний в виде (5) называется расширением кодов состояний. Такое представление позволяет перейти к структурной схеме МПА Мура  $U_2$  (рис. 2), называемого автоматом с расширением кодов состояний [10].

В МПА  $U_2$  схема СФВ реализует систему

$$\Phi_0 = \Phi_0(\tau, X). \quad (6)$$

Первые  $R_B$  элементов  $D_r \in \Phi_0$  формируют в регистре коды классов  $B_i \in \Pi_A$ , представленные переменными  $\tau_r \in \tau$ , где  $|\tau| = R_B$ . Последующие  $R_Y$  элементов  $D_r \in \Phi_0$  формируют коды наборов микроопераций  $Y_q \subseteq Y$ . Для кодирования наборов используются переменные  $z_r \in Z$ .

Как правило, выполняются следующие условия:

$$R_B + R_Y > R; \quad (7)$$

$$R_B < R; \quad (8)$$

$$R_Y < R. \quad (9)$$

При выполнении условия (7) число функций, реализуемых блоком СФВ автомата  $U_2$ , будет больше, чем для автомата  $U_1$ . Однако этот недостаток компенсируется тем, что число термов в системе (6) гарантировано равно  $H_0$ . Эти термы определяются строками преобразованной прямой структурной таблицы МПА, о которой говорится в следующем разделе статьи. Кроме того, при выполнении условия (8) уменьшается число аргументов в функциях (6). Эти два фактора позволяют уменьшить как число LUT-элементов в схеме СФВ, так и число их уровней (т.е. быстродействие МПА увеличивается).

При выполнении условия (9) уменьшается требуемое число адресных входов блоков ЕМВ. При этом требуются блоки, имеющие  $V_0$  ячеек памяти,

$$V_0 = 2^{R_Y}. \quad (10)$$

Как известно, возможные числа выходов  $t_F$  блока ЕМВ фиксированы и образуют множество  $O_F = \{1, 2, 4, 8, 18, 36\}$  [8]. При этом емкость блока постоянна [8]. Например, возможны следующие конфигурации ЕМВ:  $16K \times 1$ ,  $8K \times 2$ ,  $4K \times 4$ ,  $2K \times 8$ ,  $1024 \times 18$ ,  $512 \times 36$  [5, 6]. Из этого следует, что при уменьшении разрядности адреса на единицу число выходов удваивается. Пусть  $\Delta R = R - R_Y$ , тогда число микроопераций, реализуемых одним блоком ЕМВ автомата  $U_2$ , превышает это число для автомата  $U_1$  в  $n_Y$  раз, где

$$n_Y = 2^{\Delta R}. \quad (11)$$

Исследования авторов показали, что при  $\Delta R > 0$  число блоков ЕМВ в схеме СМО автомата  $U_2$  уменьшается до  $n_Y$  раз по сравнению с этим параметром эквивалентного автомата  $U_1$ .

Для реализации схемы блока СМО необходимо

$$n_0 = \left\lceil \frac{N}{t_F} \right\rceil \quad (12)$$

блоков ЕМВ. При этом вполне возможна ситуация, когда выходы одного из блоков используются неэффективно. Предположим, например, что в блоке ЕМВ число выходов  $t_F = 16$ . Однако только три выхода блока задействованы для реализации микроопераций  $y_n \in Y$ . В этом случае предлагаем использовать смешанную реализацию блока СМО. Поясним эту концепцию.

Представим множество  $Y$  в виде объединения множеств  $Y^1$  и  $Y^2$ ,  $(Y^1 \cap Y^2) = \emptyset$ . При этом множество  $Y^1$  включает  $N_1$  элементов, где

$$N_1 = t_F * (n_0 - 1). \quad (13)$$

Очевидно, множество  $Y^2$  включает оставшиеся  $N_2$  микроопераций, где

$$N_2 = N - t_F * (n_0 - 1). \quad (14)$$

Схема модели автомата со смешанной памятью (МПА  $U_3$ ) показана на рис. 3.

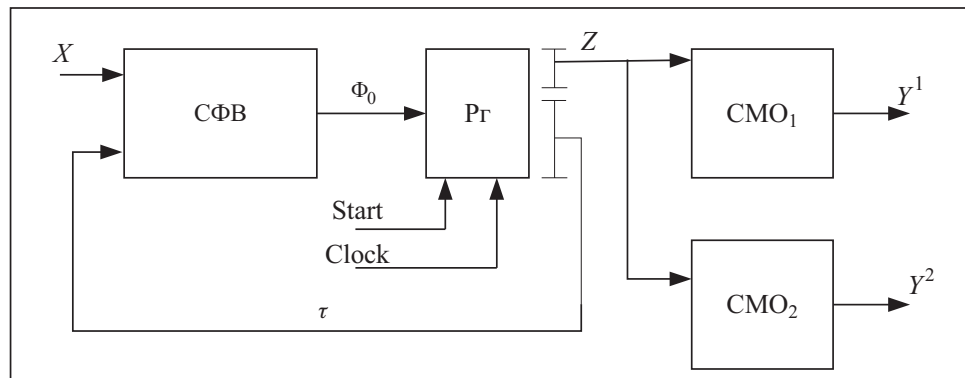


Рис. 3. Структурная схема МПА Мура  $U_3$

В автомате  $U_3$  блок СМО<sub>1</sub> реализует функции  $y_n \in Y^1$ , которые представлены таблицами истинности. Блок СМО<sub>2</sub> реализует микрооперации  $y_n \in Y^2$ , которые представляются в дизъюнктивной нормальной форме (ДНФ). Разбиение множества  $Y$  на классы  $Y^1$  и  $Y^2$  целесообразно выполнить так, чтобы ДНФ функций  $y_n \in Y^2$  имели минимально возможное число термов. Очевидно, функции  $y_n \in Y^2$  реализуются на LUT-элементах.

#### МЕТОДЫ СИНТЕЗА АВТОМАТОВ С РАСШИРЕНИЕМ КОДОВ СОСТОЯНИЙ И СМЕШАННОЙ ПАМЯТЬЮ

В данной работе предлагается метод синтеза МПА Мура  $U_2$  по граф-схеме алгоритма Г. Метод включает следующие этапы:

- 1) отметка ГСА Г и формирование множества состояний  $A$ ;
- 2) разбиение множества  $A$  на классы псевдоэквивалентных состояний;
- 3) кодирование классов  $B_i \in \Pi_A$ ;
- 4) кодирование наборов микроопераций (НМО)  $Y_q \subseteq Y$ ;
- 5) формирование преобразованной прямой структурной таблицы МПА;
- 6) формирование системы функций  $\Phi_0$ ;
- 7) построение таблицы схемы формирования микроопераций;
- 8) реализация схемы МПА в базисе FPGA.

Первый этап выполняется по известной методике [1], когда каждая операторная вершина отмечается отдельным состоянием.

Второй этап выполняется тривиальным образом, используя определение псевдоэквивалентных состояний [3]. Напомним, что состояния  $a_m, a_s \in A$  называются псевдоэквивалентными, если отмеченные ими операторные вершины ГСА связаны со входом одной и той же вершины. При таком подходе длина ПСТ (число строк таблицы) гарантировано равна  $H_0$  и коды классов  $B_i \in \Pi_A$  не влияют на длину ПСТ. Следовательно, коды  $K(B_i)$  могут быть выбраны произвольным образом. Результат кодирования НМО не влияет на число блоков ЕМВ в схеме СМО, и кодирование НМО может быть выполнено тривиальным образом.

Последний этап предлагаемого метода связан с использованием стандартных САПР [5, 6], например пакета WebPack фирмы Xilinx. Этот этап достаточно сложный, и его описание выходит за рамки данной статьи.

Предлагаемый метод синтеза МПА Мура  $U_3$  включает те же этапы, что и ранее рассмотренный метод. Однако этап 4 имеет несколько подэтапов:

- а) оптимальное кодирование наборов микроопераций;
- б) разбиение множества  $Y$  на классы  $Y^1$  и  $Y^2$ ;
- в) формирование ДНФ функций  $y_n \in Y^2$ .

Отметим, что этап 7 выполняется только для микроопераций  $y_n \in Y^1$ .

Для выполнения этапа 4 необходимо определить параметр  $N_2$ , используя формулы (12) и (14). Затем наборы  $Y_q \subseteq Y$  кодируются так, чтобы не менее  $N_2$  функций представлялись одним интервалом  $R_Y$ -мерного пространства каждая. Такое кодирование назовем оптимальным. Далее  $N_2$  функций  $y_n \in Y$  помещаются в класс  $Y^2$ , а их ДНФ формируются по карте Карно, содержащей коды наборов микроопераций.

Рассмотрим примеры применения изложенных методов.

### ПРИМЕРЫ ПРИМЕНЕНИЯ РАССМОТРЕННЫХ МЕТОДОВ

Пусть алгоритм управления задан ГСА  $\Gamma_1$  (рис. 4), а символ  $U_i(\Gamma_j)$  означает, что модель  $U_i$  ( $i = \overline{1,3}$ ) автомата Мура используется для реализации схемы по алгоритму  $\Gamma_1$ . Рассмотрим пример синтеза автомата  $U_2(\Gamma_1)$ . Анализ ГСА  $\Gamma_1$  позволяет получить следующие множества и параметры:  $A = \{a_1, \dots, a_8\}$ ,  $M = 8$ ,  $R = 3$ ;  $T = \{T_1, T_2, T_3\}$ ;  $\Pi_A = \{B_1, \dots, B_4\}$ ,  $B_1 = \{a_1\}$ ,  $B_2 = \{a_2, a_3, a_4\}$ ,  $B_3 = \{a_5, a_6\}$ ,  $B_4 = \{a_7, a_8\}$ ,  $I = 4$ ,  $\tau = \{\tau_1, \tau_2\}$ ,  $R_B = 2$ ,  $Y = \{y_1, \dots, y_5\}$ ,  $N = 5$ ;  $X = \{x_1, \dots, x_4\}$ ,  $L = 4$ . Пусть среди возможных конфигураций блоков памяти, используемых в качестве элементарного базиса, имеется конфигурация  $8 \times 6$  бит. Тогда  $t_F = 6$  и  $t_F > N$ . Следовательно, для реализации автомата Мура по ГСА  $\Gamma_1$  может быть использована модель  $U_2$ .

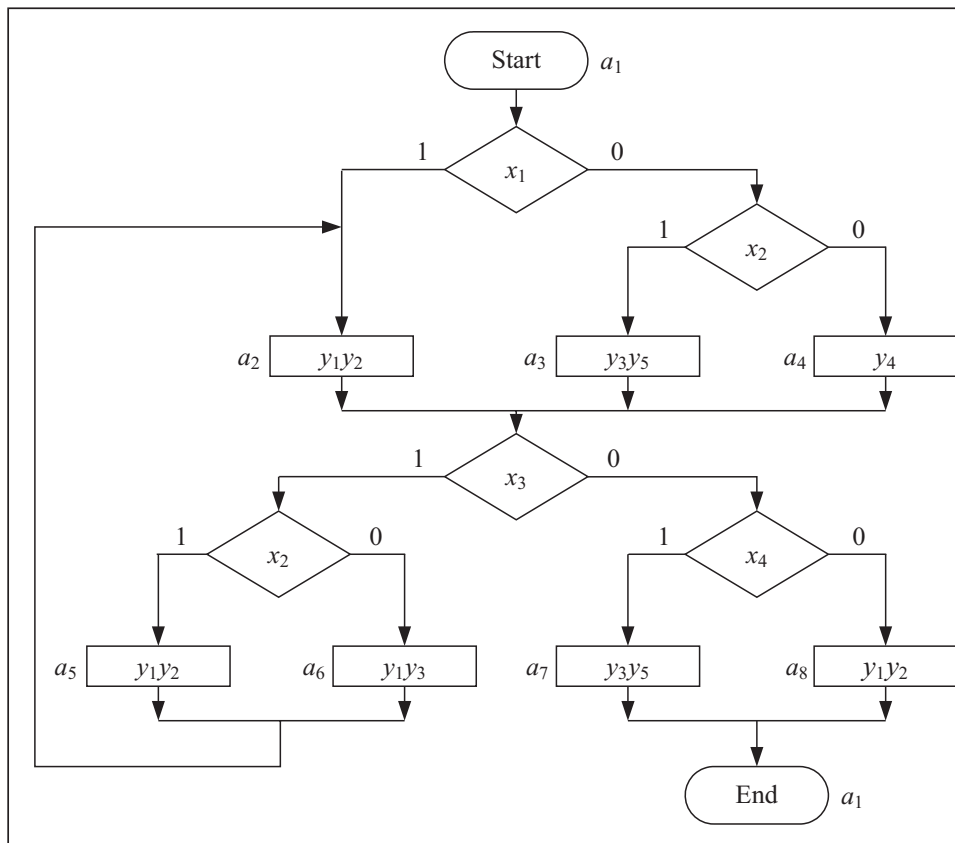


Рис. 4. Исходная граф-схема алгоритма  $\Gamma_1$

В вершинах ГСА  $\Gamma_1$  находится  $Q = 5$  различных наборов микроопераций:  $Y_1 = \emptyset$ ,  $Y_2 = \{y_1, y_2\}$ ,  $Y_3 = \{y_3, y_5\}$ ,  $Y_4 = \{y_4\}$ ,  $Y_5 = \{y_1, y_3\}$ . Набор  $Y_1$  соответствует состоянию  $a_1 \in A$ , и его код должен формироваться по сигналу Start. Для кодирования НМО достаточно  $R_Y = 3$  переменных, образующих множество  $Z = \{z_1, z_2, z_3\}$ . Коды в карте Карно (рис. 5) отражают частотный принцип [1]: чем чаще некоторый объект встречается, тем меньше единиц содержит его код. Используем этот же принцип при кодировании классов  $B_i \in \Pi_A$ . В результате получим следующие коды классов:  $K(B_1) = 00$ ,  $K(B_2) = 01$ ,  $K(B_3) = 10$ ,  $K(B_4) = 11$ . При этом код класса  $B_1 = \{a_1\}$  должен быть нулевым, так как он за-

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

Рис. 5. Коды НМО

гружается в регистр Pr по сигналу Start. На основе кодов  $K(B_i)$  и  $K(Y_q)$  согласно (5) можно получить расширенные коды состояний автомата  $U_2(\Gamma_1)$  (табл. 1).

Для формирования преобразованной ПСТ автомата Мура необходимо построить систему обобщенных формул перехода [3]. Для автомата  $U_2(\Gamma_1)$  эта система включает четыре уравнения:

$$\begin{aligned}
 B_1 &\rightarrow x_1 a_2 \vee \overline{x_1 x_2 a_3} \vee \overline{x_1 x_2 a_4}; \\
 B_2 &\rightarrow x_3 x_2 a_5 \vee \overline{x_3 x_2 a_6} \vee \overline{x_3 x_4 a_7} \vee \overline{x_3 x_4 a_8}; \\
 B_3 &\rightarrow a_2; \quad B_4 \rightarrow a_1.
 \end{aligned}
 \tag{15}$$

Таблица 1

$a_m$	$\tau_1$	$\tau_2$	$z_1$	$z_2$	$z_3$	$a_m$	$\tau_1$	$\tau_2$	$z_1$	$z_2$	$z_3$	$a_m$	$\tau_1$	$\tau_2$	$z_1$	$z_2$	$z_3$	$a_m$	$\tau_1$	$\tau_2$	$z_1$	$z_2$	$z_3$
$a_1$	0	0	0	0	0	$a_3$	0	1	0	1	0	$a_5$	1	0	0	0	1	$a_7$	1	1	0	1	0
$a_2$	0	1	0	0	1	$a_4$	0	1	1	0	0	$a_6$	1	0	1	0	1	$a_8$	1	1	0	0	1

Таблица 2

$B_i$	$K(B_i)$	$a_s$	$K(a_s)$	$X_h$	$\Phi_h$	$h$
$B_1$	00	$a_2$	01001	$\overline{x_1}$	$D_2 D_5$	1
		$a_3$	01001	$\overline{x_1 x_2}$	$D_2 D_4$	2
		$a_4$	01100	$\overline{x_1 x_2}$	$D_2 D_3 D_4$	3
$B_2$	01	$a_5$	10001	$x_3 \overline{x_2}$	$D_1 D_5$	4
		$a_6$	10101	$\overline{x_3 x_2}$	$D_1 D_3 D_5$	5
		$a_7$	11010	$\overline{x_3 x_4}$	$D_1 D_2 D_4$	6
		$a_8$	11001	$\overline{x_3 x_4}$	$D_1 D_2 D_5$	7
$B_3$	10	$a_2$	01001	1	$D_2 D_5$	8
$B_4$	11	$a_1$	00000	1	—	9

С использованием системы (15) формируется преобразованная ПСТ автомата Мура  $U_2(\Gamma_1)$  (табл. 2). Связь табл. 2 с системой (15) очевидна.

Таблица имеет  $H_2 = 9$  термов, из которых восемь термов входят в функции  $D_r \in \Phi_0 = \{D_1, \dots, D_5\}$ . По табл. 2 формируется система (6). Например,  $D_1 = B_2 = \overline{\tau_1 \tau_2}$  (с учетом минимизации),  $D_2 = B_1 \vee B_2 \overline{x_3} \vee B_3$  (с учетом минимизации),  $D_3 = B_1 \overline{x_1 x_2} \vee B_2 \overline{x_3 x_2}$  и т.д. Отметим, что функция  $D_1$  зависит от двух переменных, функция  $D_2$  — от трех, функция  $D_3$  — от пяти, функции  $D_4$  и  $D_5$  зависят от шести переменных. При этом автомат  $U_2(\Gamma_1)$  характеризуется параметром  $H_1 = 20$ , а функции  $D_1 - D_3$  зависят от семи переменных ( $\tau_1, \dots, \tau_3, x_1, \dots, x_4$ ). Если используются LUT-элементы, имеющие число входов  $S = 4$ , то для реализации функций  $D_1$  и  $D_2$  необходим один LUT-элемент, для  $D_3$  — три элемента (два уровня), для  $D_4$  — три элемента (два уровня), для  $D_5$  — пять элементов (два уровня). Таким образом, для реализации схемы СФВ автомата  $U_2(\Gamma_1)$  необходимо 13 LUT-элементов (два уровня). Для автомата  $U_1(\Gamma_1)$  необ-

ходимо 27 LUT-элементов (четыре уровня). Таким образом, предлагаемый подход позволяет для данного случая получить схему с меньшими (в 2,25 раза) аппаратными затратами и в два раза большим быстродействием, чем для схемы СФВ автомата  $U_1(\Gamma_1)$ . Естественно, величина выигрыша зависит от числа входов LUT-элементов.

Для построения таблицы содержимого ЕМВ схемы СМО достаточно преобразовать карту Карно (см. рис. 5). Таблица содержит адрес набора, задаваемого переменными  $z_r \in Z$ , и унитарный код НМО  $Y_q \subseteq Y$  (табл. 3).

**Таблица 3**

Адрес, $z_1 z_2 z_3$	Содержимое, $y_1 y_2 y_3 y_4 y_5$	Адрес, $z_1 z_2 z_3$	Содержимое, $y_1 y_2 y_3 y_4 y_5$
0 0 0	0 0 0 0 0	1 0 0	0 0 0 1 0
0 0 1	1 1 0 0 0	1 0 1	1 0 1 0 0
0 1 0	0 0 1 0 1	1 1 0	0 0 0 0 0
0 1 1	0 0 0 0 0	1 1 1	0 0 0 0 0

Дальнейшее построение схемы МПА требует использования некоторых стандартных пакетов [5, 6]. Этот этап в данной статье не рассматривается.

Далее изложим ситуацию, когда вместо конфигурации  $8 \times 6$  блок памяти имеет конфигурацию  $8 \times 4$ . Из (12) следует, что  $n_0 = 2$ . При этом ресурсы одного ЕМВ использованы полностью, а у другого ЕМВ используется только один выход. Итак, целесообразно применять модель  $U_3(\Gamma_1)$ , в которой  $N_1 = 4$  и  $N_2 = 1$ .

Поскольку МО  $y_5$  входит только в набор  $Y_3$ , то из рис. 3 следует формула  $y_5 = z_2$ . Таким образом,  $Y^1 = \{y_1, \dots, y_4\}$  и  $Y^2 = \{y_5\}$ . Далее блок СМО<sub>1</sub> задается таблицей, получаемой из табл. 3 путем удаления столбца  $y_5$ . Схема блока СМО автомата  $U_3(\Gamma_1)$  приведена на рис. 6.

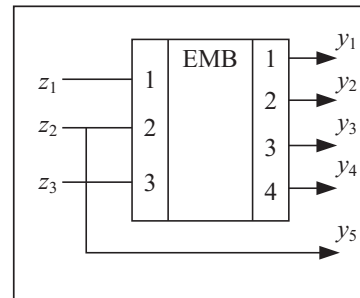


Рис. 6. Схема блока формирования микроопераций автомата  $U_3(\Gamma_1)$

Отметим, что схемы блоков СФВ для автомата  $U_2(\Gamma_1)$  и  $U_3(\Gamma_1)$  совпадают.

### ЗАКЛЮЧЕНИЕ

В настоящей статье рассматриваются два метода уменьшения количества ресурсов кристалла FPGA, потребляемых схемой микропрограммного автомата Мура. В основе этих методов находится кодирование наборов микроопераций и классов псевдоэквивалентных состояний. При выполнении условия  $R_Y < R$  схема формирования микроопераций требует меньше встроенных блоков памяти ЕМВ, чем для известных моделей автомата. Дополнительное уменьшение этого параметра возможно благодаря изложенному методу смешанной памяти. При этом часть микроопераций реализуется схемой, состоящей из LUT-элементов.

Использование классов псевдоэквивалентных состояний позволяет уменьшить число аргументов и термов в системах функций возбуждения памяти МПА.



При выполнении условия  $R < R_A + R_Y$  предлагаемый метод приводит к увеличению числа функций возбуждения. Однако этот недостаток компенсируется уменьшением числа аргументов и термов в этих функциях.

Исследования эффективности предложенных методов на стандартных примерах [11] показали, что аппаратные затраты уменьшаются в среднем на 42 % по сравнению с автоматом Мура  $U_1$ . Побочным положительным эффектом предложенных методов является увеличение быстродействия автомата, что наблюдалось в 92 % рассмотренных случаев.

#### СПИСОК ЛИТЕРАТУРЫ

1. Baranov S. Logic and system design of digital systems. — Tallinn: TUT Press, 2008. — 328 p.
2. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. — М.: Горячая линия – Телеком, 2001. — 636 с.
3. Barkalov A., Titarenko L. Logic synthesis for FSM-based control units. — Berlin: Springer, 2009. — 233 p.
4. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. — СПб.: БХВ, 2002. — 608 с.
5. FPGAs, CPLD, and ASIC from Altera. — <http://www.altera.com>.
6. FPGAs and CPLD Solutions from Xilinx. — [www.xilinx.com](http://www.xilinx.com).
7. DeMicheli G. Synthesis and optimization of digital circuits. — New York: McGraw Hill, 1994. — 541 p.
8. Максфилд К. Проектирование на ПЛИС. Курс молодого бойца. — М.: Додэка-21, 2007. — 408 с.
9. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура // Кибернетика и системный анализ. — 1998. — № 1. — С. 65–72.
10. Баркалов А.А., Мальчева Р.В., Солдатов К.А. Синтез автомата Мура с идентификацией состояний // Наук. праці Донецьк. нац. техн. ун-ту. Сер. «Інформатика, кібернетика та обчислювальна техніка» (ІКОТ-2009). Вип. 10 (153). — Донецьк: ДВНЗ «ДонНТУ», 2009. — С. 180–183.
11. Yang S. Logic synthesis and optimization benchmarks user guide. Technical report, №1991 — IWLS-UG-Saryang.-Microelectronics center of North Carolina, 1991. — 45 p.

*Поступила 27.06.2012*