

СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ КАК СРЕДСТВО ОПТИМИЗАЦИИ СХЕМЫ АВТОМАТА МИЛИ НА ПЛИС

Ключевые слова: микропрограммный автомат Мили, ПЛИС, структурная декомпозиция, встроенные блоки памяти.

ВВЕДЕНИЕ

Модель микропрограммного автомата (МПА) Мили [1] часто используется для синтеза устройств управления [2]. Развитие микроэлектроники привело к появлению разнообразных семейств программируемых логических интегральных схем (ПЛИС) [3, 4]. При синтезе схем МПА в базисе ПЛИС возникает ряд оптимизационных проблем, одна из них — уменьшение числа макроячеек ПЛИС в логической схеме [5]. Методы решения этой задачи зависят от особенностей модели автомата, алгоритма управления и элементного базиса [6]. В данной работе рассматривается задача реализации схемы МПА Мили в базисе ПЛИС типа FPGA (field-programmable gate array) [7, 8]. Подобные ПЛИС состоят из элементов табличного типа (look-up table — LUT) и встроенных блоков памяти (embedded memory block — ЕМВ). Условимся в дальнейшем под ПЛИС понимать FPGA, а также введем обозначения ЛЭ (логический элемент) для LUT и ВБП (встроенный блок памяти) для ЕМВ.

Как правило, ЛЭ реализуют таблицу истинности произвольной булевой функции, имеющей 4–6 аргументов [4]. Это ограничение приводит к необходимости функциональной декомпозиции [5] реализуемых систем булевых функций (СБФ). При этом схема приобретает многоуровневый характер, что увеличивает время такта в случае реализации МПА на ПЛИС. Для уменьшения аппаратных затрат целесообразно использовать метод структурной декомпозиции [6], основанный на увеличении числа структурных уровней схемы МПА. В данной работе рассматривается метод, заключающийся в кодировании некоторых объектов МПА (например, наборов микроопераций) и их последующем декодировании с использованием ВБП. Это приводит к сбалансированному использованию различных ресурсов ПЛИС (ЛЭ и ВБП), а также к уменьшению части кристалла, занятой схемой МПА [9]. В настоящей работе рассматриваются методы структурной декомпозиции схемы МПА Мили, основанной на методе преобразования объектов [10]. Эти методы адаптируются к особенностям ПЛИС со встроенными блоками памяти.

1. ОСНОВНЫЕ ПОЛОЖЕНИЯ

Микропрограммный автомат Мили представляется вектором [1]

$$S = \langle X, Y, A, \delta, \lambda, a_1 \rangle, \quad (1)$$

где $X = \{x_1, \dots, x_L\}$ — множество входных сигналов (логических условий), $Y = \{y_1, \dots, y_N\}$ — множество выходных сигналов (микроопераций), $A = \{a_1, \dots, a_M\}$ — множество внутренних состояний. Для синтеза схемы МПА состояния $a_m \in A$ кодируются двоичными кодами $K(a)_m$ разрядности R , при этом для кодирования используются внутренние переменные $T_r \in T$. Функция переходов δ определяет состояние перехода, а функция выходов λ — выход-

ной сигнал МПА. В начальный момент времени автомат находится в начальном состоянии $a_1 \in A$. На практике функция δ представляется системой функций возбуждения элементов памяти МПА

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

где $\Phi = \{D_1, \dots, D_R\}$. Аналогично функция λ представляется системой выходных функций

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

Пусть МПА Мили представлен прямой структурной таблицей (ПСТ) [1] со столбцами: $a_m, K(a)_m, a_s, K(a)_s, X_h, Y_h, \Phi_h, h$. Здесь $a_m \in A$ — исходное состояние МПА; $a_s \in A$ — состояние перехода; X_h — входной набор, определяющий переход $\langle a_m, a_s \rangle$ и определяемый конъюнкцией некоторых элементов (или их отрицаний) множества X ; $Y_h \subseteq Y$ — набор выходных функций, формируемый на переходе $\langle a_m, a_s \rangle$; $\Phi_h \subseteq \Phi$ — набор функций возбуждения, принимающих единичное значение для переключения памяти МПА из $K(a)_m$ в $K(a)_s$; $h \in \{1, \dots, H\}$ — номер перехода. Каждая строка ПСТ представляется конъюнктивным термом

$$F_h = X_h \cdot \bigwedge_{r=1}^R T_r^{\ell_{mr}} \quad (h=1, \dots, H), \quad (4)$$

где $\ell_{mr} \in \{0, 1\}$ — значение r -го разряда кода $K(a)_m$ состояния $a_m \in A$, записанного в h -й строке ПСТ, $T_r^0 = \bar{T}_r$, $T_r^1 = T_r$ ($r=1, \dots, R$). При этом функции (2) и (3) представляются в следующем виде:

$$D_r = \bigvee_{h=1}^H C_{rh} F_h \quad (r=1, \dots, R), \quad (5)$$

$$y_n = \bigvee_{h=1}^H C_{nh} F_h \quad (n=1, \dots, N), \quad (6)$$

где $C_{rh} (C_{nh})$ — булева переменная, равная единице, если и только если в h -й строке ПСТ записан символ $D_r (y_n)$.

На рис. 1 приведена структурная схема МПА Мили, реализуемого на основании систем (5), (6). В этой модели комбинационная схема (КС) реализует функции

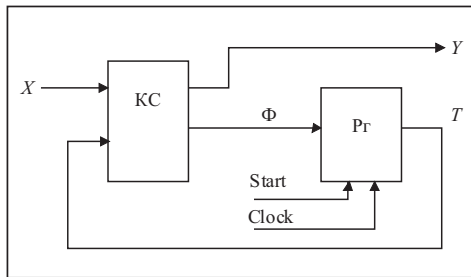


Рис. 1. Схема автомата Мили

(2), (3), а регистр (Pr) используется как память состояний. Сигнал Start применяется для установки МПА в начальное состояние $a_1 \in A$, а сигнал Clock — для изменения содержимого Pr. Теоретически эта модель приводит к реализации (схеме автомата) с максимальным быстродействием. Условимся в дальнейшем обозначать эту модель символом S_1 .

2. РЕАЛИЗАЦИЯ СХЕМ МПА НА ПЛИС

При использовании ПЛИС типа FPGA схема КС и регистр Pr реализуются на ЛЭ, имеющих S входов ($S \leq 6$). Пусть $f_i \in \Phi \cup Y$ — булева функция, реализуемая схемой КС, $L(f_i)$ — число логических условий в термах (4), входящих в дизъюнктивную нормальную форму (ДНФ) функции f_i ($i=1, \dots, R+N$). Если выполняется условие

$$L(f_i) + R > S, \quad (7)$$

то для реализации функции f_i необходимо использовать методы функциональной декомпозиции [11]. При этом функция f_i реализуется в виде многоуровневой схемы (рис. 2).

Здесь $T_H^i \subseteq T, X_H^i \subseteq X$ — множества внутренних переменных и логических условий, используемых соответственно для реализации подфункций H_i ;

$T_U^i \subseteq T, X_U^i \subseteq X$ — множества внутренних переменных и логических условий, используемых соответственно для реализации функции f_i . При этом

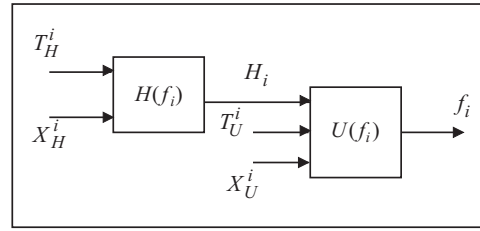


Рис. 2. Многоуровневая схема функции f_i

$$|H_i| + |T_U^i| + |X_U^i| \leq S \quad (8)$$

и могут выполняться условия

$$\begin{aligned} X_H^i \cap X_U^i &\neq \emptyset; \\ T_H^i \cap T_U^i &\neq \emptyset. \end{aligned} \quad (9)$$

Если выполняется условие

$$|T_H^i| + |X_H^i| \leq S, \quad (10)$$

то результирующая схема имеет два уровня. В противном случае необходимо решать задачу функциональной декомпозиции для функций H_i . Таким образом, результирующая схема может иметь произвольное число уровней.

Отметим, что число уровней в схеме КС, как и число ЛЭ в ней, в значительной степени зависит от используемого метода кодирования состояний [5, 12]. Эта проблема в настоящей статье не рассматривается, а для кодирования используются двоичные коды разрядности

$$R = \lceil \log_2 M \rceil. \quad (11)$$

Назовем булеву функцию f регулярной относительно ВБП (r -функция), если число ее аргументов не превышает числа адресных входов ВБП. В противном случае назовем функцию нерегулярной относительно ВБП (i -функция). Условимся в дальнейшем выражение «относительно ВБП» опускать и называть функции соответственно регулярными и нерегулярными. Для реализации r -функции можно использовать ВБП, в то время как i -функция реализуется на логических элементах (в случае ПЛИС) [13]. В автоматах средней сложности [1] $R \approx 6, L \approx 30, H \approx 2000$, при этом имеется около 10^{12} различных входных наборов для систем функций Φ и Y . Следовательно, функции (2), (3) относятся к классу i -функций. Таким образом, для реализации схемы МПА S_1 используются только логические элементы. При этом нет возможности использовать ВБП, т.е. нет сбалансированности в использовании ресурсов кристалла ПЛИС.

3. СТРУКТУРНАЯ ДЕКОМПОЗИЦИЯ СХЕМЫ АВТОМАТА

Пусть $Y_q \subseteq Y$ — набор микроопераций, записанный в некоторой строке ПСТ, где $q \in \{1, \dots, Q\}$. Поставим в соответствие каждому набору Y_q двоичный код $K(Y_q)$ размерности

$$R_Y = \lceil \log_2 Q \rceil. \quad (12)$$

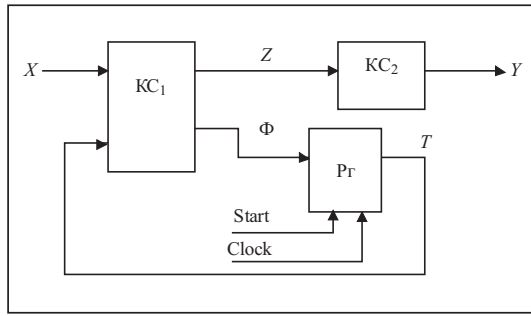


Рис. 3. Схема автомата Мили S_2

Используем для кодирования наборов $Y_q \subseteq Y$ переменные $z_r \in Z$, где $|Z| = R_Y$. При этом автомат S_1 преобразовывается в PY -автомат [6], обозначаемый в дальнейшем символом S_2 (рис. 3).

В МПА S_2 схема KC_1 реализует функции (2) и формирует переменные $z_r \in Z$, представленные в виде

$$Z = Z(X, T). \quad (13)$$

Комбинационная схема KC_2 реализует микрооперации $y_n \in Y$, представленные в виде

$$Y = Y(Z). \quad (14)$$

Поскольку параметр R_Y определялся согласно (12), функции (14) относятся к классу r -функций и могут быть реализованы на ВБП. Таким образом, число i -функций уменьшается от $k_1 = R + N$ (автомат S_1) до $k_2 = R + R_Y$.

Рассмотренный метод кодирования наборов микроопераций относится к методам структурной декомпозиции. При этом увеличивается число структурных уровней схемы, но уменьшается число ЛЭ в ней (по сравнению с автоматом S_1). Теперь схему МПА можно реализовать на ПЛИС с меньшим числом ЛЭ (т.е. на более дешевой ПЛИС), либо сэкономленные ЛЭ можно использовать для реализации других блоков цифровой системы, работающих под управлением МПА Мили. Очевидно, что экономия числа ЛЭ тем больше, чем меньше i -функций необходимо реализовать. Абсолютный минимум числа i -функций достигается при использовании метода преобразования кодов объектов [10].

Назовем состояния $a_m \in A$ и наборы микроопераций $Y_q \subseteq Y$ объектами МПА Мили. Основной идеей метода преобразования кодов объектов является представление, например кодов состояний (вторичных объектов) как некоторых функции от наборов микроопераций (первичных объектов) и дополнительных переменных, обеспечивающих однозначность такого представления [10]. Рассмотрим этот метод и порождаемую им структуру МПА Мили с максимальным кодированием наборов микроопераций.

Пусть $A(Y_q) \subseteq A$ — множество состояний МПА, при переходе в которые формируется набор $Y_q \subseteq Y$, где $m_q = |A(Y_q)|$. Пусть $K = \max(m_1, \dots, m_Q)$, тогда для однозначной идентификации любого из состояний $a_m \in A(Y_q)$ достаточно K идентификаторов, образующих множество $I = \{I_1, \dots, I_K\}$. Поставим в соответствие состоянию $a_m \in A(Y_q)$ пару $\alpha_{mq} = \langle I_{mq}, Y_q \rangle$, где идентификатор I_{mq} однозначно определяет состояние $a_m \in A$ в множестве $A(Y_q)$. Закодируем наборы $Y_q \subseteq Y$ двоичными кодами размерности (12), а идентификаторы $I_k \in I$ — двоичными кодами $K(I_k)$ разрядности

$$R_I = \lceil \log_2 K \rceil \quad (15)$$

и используем для их кодирования переменные $v_r \in V$, где $|V| = R_I$. Теперь пара α_{mq} однозначно определяет код

$$C(a_m)_q = K(Y_q) * K(I_{mq}), \quad (16)$$

где $*$ — знак конкатенации. Такой подход приводит к модели автомата Мили S_3 (рис. 4).

В автомате S_3 схема KC_1 реализует системы (13) и

$$V = V(X, T), \quad (17)$$

схема KC_2 реализует систему (14), а схема KC_3 формирует функции Φ , представленные в виде

$$\Phi = \Phi(V, Z). \quad (18)$$

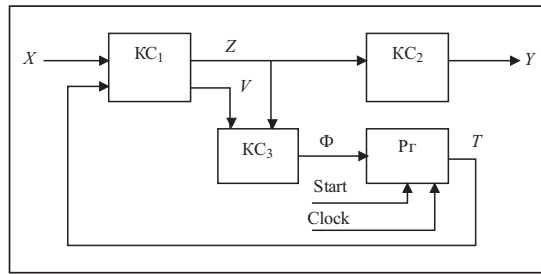


Рис. 4. Схема автомата Мили S_3

Как показывает анализ, для автоматов средней сложности [1] функции (18) могут быть реализованы на ВБП современных ПЛИС [7, 8]. Теперь число i -функций определяется как $k_3 = R_Y + R_I$. Очевидно, что $k_3 < k_2$ при выполнении условия

$$R > R_I. \quad (19)$$

Чем больше разность $R - R_I$, тем больший выигрыш дает переход от модели S_2 к модели S_3 .

Аналогичным образом можно получить модель S_4 , основанную на преобразовании кодов состояний в коды наборов микроопераций. Пусть $Y(a_m)$ — множество наборов микроопераций, формируемых при переходе в состояние $a_m \in A$, и пусть $|Y(a_m)| = Q_m$. Если $U = \max(Q_1, \dots, Q_M)$, то для однозначной идентификации любого набора $Y_q \in Y(a_m)$ достаточно U идентификаторов, образующих множество $I_0 = \{I_1, \dots, I_U\}$. Поставим в соответствие каждому набору $Y_q \in Y(a_m)$ пару $\alpha_{qm} = \langle I_{qm}, a_m \rangle$, где идентификатор I_{qm} однозначно определяет набор $Y_q \subseteq Y$ в множестве $Y(a_m)$. Закодируем наборы $Y_q \subseteq Y$ двоичными кодами размерности (12), а идентификаторы $I_u \in I_0$ — двоичными кодами разрядности

$$R_U = \lceil \log_2 U \rceil \quad (20)$$

и используем для их кодирования переменные $w_r \in W$, где $|W| = R_U$. Теперь пара α_{qm} однозначно определяет код

$$C(Y_q)_m = K(a)_m * K(I_{qm}), \quad (21)$$

где $*$ — знак конкатенации. Этот подход приводит к модели автомата Мили S_4 (рис. 5).

В автомате S_4 схема KC_1 реализует системы (2) и

$$W = W(X, T), \quad (22)$$

схема KC_3 реализует систему (13), а схема KC_2 — систему

$$Z = Z(T, W). \quad (23)$$

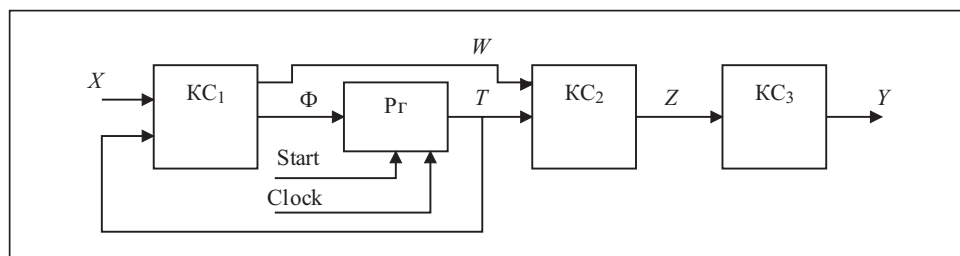


Рис. 5. Схема автомата Мили S_4

Как и в случае автомата S_3 , схемы блоков KC_2 и KC_3 автомата S_4 могут быть реализованы на ВБП. Таким образом, число i -функций теперь определяется как $k_4 = R + R_U$. Если

$$R_U < R_Y, \quad (24)$$

то автомат S_4 является альтернативой автомату S_2 . Отметим, что выбор конкретной модели из множества $\{S_2, S_3, S_4\}$ определяется характеристиками алгоритма управления и элементного базиса.

4. РЕАЛИЗАЦИЯ НА ПЛИС АВТОМАТОВ С ПРЕОБРАЗОВАНИЕМ КОДОВ ОБЪЕКТОВ

Характерной особенностью современных ПЛИС [7, 8] является наличие ВБП, для которых можно выбирать число слов N_W и их разрядность L_W при некоторой фиксированной емкости V_0 :

$$V_0 = N_W \cdot L_W. \quad (25)$$

Наиболее часто $V_0 = 4$ Кб, при этом блок памяти может быть организован как ОЗУ или ПЗУ со следующими характеристиками: $4K \times 1$, $2K \times 2$, $1K \times 4$, 512×8 , 256×16 . По мере роста стоимости микросхемы увеличивается и параметр V_0 . В настоящее время существует ВБП, например, с характеристиками от 512×36 до $16K \times 1$ (бит) [7, 8]. Эта особенность может использоваться при реализации схем МПА, основанных на преобразовании кодов объектов.

Пусть $O = \{1, 2, 4, 8, 16\}$ — множество фиксированных выходов ВБП, включающее элементы L_W из (25). Для реализации схемы KC_3 автомата S_3 необходимо

$$N_W(S_3)_3 = 2^{R_Y + R_I} \quad (26)$$

слов ВБП. При этом ВБП будет иметь

$$L_W(S_3)_3 = V_0 / N_W(S_3)_3 \quad (27)$$

выходов, определяемых множеством O . При выполнении условия

$$L_W(S_3)_3 > R \quad (28)$$

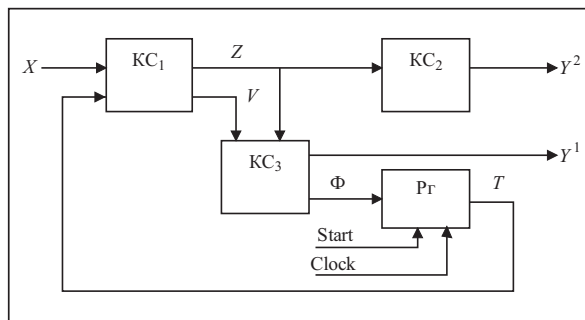


Рис. 6. Схема автомата Мили S_5

часть микроопераций $y_n \in Y$ реализуются схемой KC_3 . Пусть эти микрооперации образуют множество Y^1 . Теперь блок KC_2 реализует оставшиеся микрооперации, образующие множество Y^2 . Естественно, что $Y^1 \cap Y^2 = \emptyset$ и $Y^1 \cup Y^2 = Y$. Распределение микроопераций по множествам Y^1 и Y^2 не имеет значения.

На рис. 6 приведена структурная схема МПА S_5 , основанная на разбиении множества микроопераций в МПА S_3 .

Аналогичным образом может быть преобразована и модель S_4 , в которой необходимо

$$N_W(S_4)_2 = 2^{R + R_U} \quad (29)$$

слов ВБП для реализации схемы KC_2 . При этом число выходов ВБП определяется как

$$L_W(S_4)_2 = V_0 / N_W(S_4)_2. \quad (30)$$

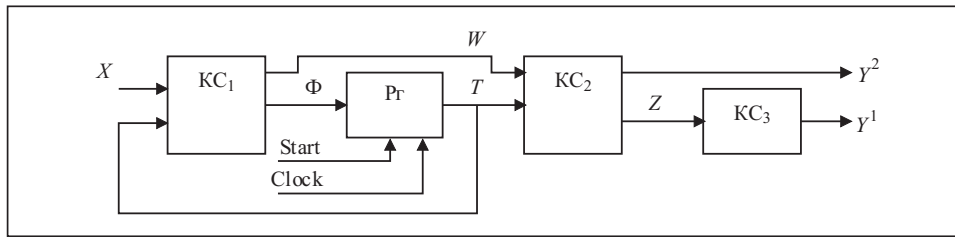


Рис. 7. Схема автомата Мили S_6

При выполнении условия

$$L_W(S_4)_2 > R_Y \quad (31)$$

микрооперации $y_n \in Y^1$ реализуются схемой KC_2 . Оставшиеся $N - (L_W(S_4)_2 - R_Y)$ микроопераций образуют множество Y^2 и реализуются схемой KC_3 . Этот подход преобразовывает автомат S_4 в автомат S_6 (рис. 7).

В настоящей работе предлагается метод синтеза автомата S_5 , заданного прямой структурной таблицей. Метод включает следующие этапы:

- 1) кодирование наборов микроопераций $Y_q \subseteq Y$;
- 2) однозначная идентификация состояний $a_m \in A$ кодами (16);
- 3) преобразование исходной ПСТ;
- 4) разбиение множества микроопераций;
- 5) спецификация блока KC_2 ;
- 6) спецификация блока KC_3 ;
- 7) реализация схемы автомата в заданном элементном базисе.

В этой работе метод синтеза автомата S_6 не рассматривается; он является предметом дальнейших исследований. Рассмотрим пример синтеза схемы автомата S_5 , заданного ПСТ (табл. 1).

Таблица 1

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Y_h	Φ_h	h
a_1	000	a_2	001	x_1	$y_1 y_2$	D_3	1
		a_3	010	\bar{x}_1	$y_3 y_8$	D_2	2
a_2	001	a_2	001	x_2	$y_1 y_2$	D_3	3
		a_3	010	$\bar{x}_2 x_3$	y_4	D_2	4
		a_4	011	$\bar{x}_2 \bar{x}_3$	$y_1 y_2$	$D_2 D_3$	5
a_3	010	a_4	011	x_1	$y_2 y_5$	$D_2 D_3$	6
		a_5	100	\bar{x}_1	$y_6 y_9$	D_1	7
a_4	011	a_5	100	1	$y_3 y_7$	D_1	8
a_5	100	a_2	001	$x_2 x_3$	$y_1 y_2$	D_3	9
		a_3	010	$x_2 \bar{x}_3$	$y_3 y_8$	D_2	10
		a_5	100	$\bar{x}_2 x_4$	$y_3 y_7$	D_1	11
		a_1	000	$\bar{x}_2 \bar{x}_4$	—	—	12

5. ПРИМЕР СИНТЕЗА АВТОМАТА S_5

Как следует из табл. 1, автомат S_5 имеет следующие характеристики: $M = 5$, $L = 4$, $N = 9$, $R = 3$, $H = 12$. Память МПА реализована на D -триггерах, что наиболее часто используется на практике [14]. Кроме того, из табл. 1 следует, что автомат S_5 имеет $Q = 7$ различных наборов микроопераций $Y_1 = \emptyset$, $Y_2 = \{y_1, y_2\}$, $Y_3 = \{y_3, y_8\}$, $Y_4 = \{y_4\}$, $Y_5 = \{y_2, y_5\}$, $Y_6 = \{y_6, y_9\}$, $Y_7 = \{y_3, y_7\}$.

Как следует из (12), $R_Y = 3$ и для кодирования наборов используются переменные $z_r \in Z = \{z_1, z_2, z_3\}$. Закодируем наборы $Y_q \subseteq Y$ следующим образом: $K(Y_1) = 000, \dots, K(Y_7) = 110$.

Из табл. 1 можно найти следующие множества $A(Y_q)$: $A(Y_1) = \{a_1\}$, $A(Y_2) = \{a_2, a_4\}$, $A(Y_3) = \{a_3\}$, $A(Y_4) = \{a_3\}$, $A(Y_5) = \{a_4\}$, $A(Y_6) = \{a_5\}$, $A(Y_7) = \{a_5\}$. Следовательно, $m_1 = m_3 = \dots = m_7 = 1$, $m_2 = 2$, $K = 2$, $R_I = 1$, $V = \{v_1\}$, $I = \{I_1, I_2\}$. Пусть $K(I_1) = 0$ и $K(I_2) = 1$. Сформируем пары α_{mq} для нашего примера. Если $m_q = 1$, то идентификатор не нужен и такая пара будет включать символ \emptyset . Для автомата S_5 имеем: $\alpha_{11} = \langle \emptyset, Y_1 \rangle$, $\alpha_{22} = \langle I_1, Y_2 \rangle$, $\alpha_{42} = \langle I_2, Y_2 \rangle$, $\alpha_{33} = \langle \emptyset, Y_3 \rangle$, $\alpha_{34} = \langle \emptyset, Y_4 \rangle$, $\alpha_{45} = \langle \emptyset, Y_5 \rangle$, $\alpha_{56} = \langle \emptyset, Y_6 \rangle$ и $\alpha_{57} = \langle \emptyset, Y_7 \rangle$. В табл. 2 представлены коды $C(a_m)_q$ для автомата S_5 .

Таблица 2

a_m	$C(a_m)_q$	α_{mq}	m	a_m	$C(a_m)_q$	α_{mq}	m
a_1	000*	α_{11}	1	a_4	100*	α_{45}	5
a_2	0010	α_{22}	2	a_4	0011	α_{42}	6
a_3	010*	α_{33}	3	a_5	101*	α_{56}	7
a_3	011*	α_{34}	a_5	4	110*	α_{57}	8

В табл. 2 символ * соответствует неопределенному значению идентификатора в паре α_{mq} . Построим преобразованную ПСТ автомата S_5 (табл. 3).

Таблица 3

a_m	$K(a_m)$	X_h	Z_h	V_h	h	α_{mq}
a_1	000	x_1	z_3	—	1	α_{22}
		\bar{x}_1	z_2	*	2	α_{33}
a_2	001	x_2	z_3	—	3	α_{22}
		$\bar{x}_2 x_3$	$z_2 z_3$	*	4	α_{34}
		$\bar{x}_2 \bar{x}_3$	z_3	v_1	5	α_{42}
a_3	010	x_1	z_1	*	6	α_{45}
		\bar{x}_1	$z_1 z_3$	*	7	α_{56}
a_4	011	1	$z_1 z_2$	*	8	α_{57}
a_5	100	$x_2 x_3$	z_3	—	9	α_{22}
		$x_2 \bar{x}_3$	z_2	*	10	α_{33}
		$\bar{x}_2 x_4$	$z_1 z_2$	*	11	α_{57}
		$\bar{x}_2 \bar{x}_4$	—	*	12	α_{11}

Для построения преобразованной ПСТ необходимо столбцы a_s , $K(a_s)$ и Φ_h исходной ПСТ заменить столбцами Z_h и V_h . В столбце α_{mq} , используемом как комментарий, указывается пара α_{mq} , соответствующая h -й строке ПСТ; в столбце Z_h записываются переменные $z_r \in Z$, равные единице в коде $K(Y_q)$ для пары $\langle I_k, Y_q \rangle$ из h -й строки ПСТ; в столбце V_h содержится код $K(I_k)$ для пары α_{mq} , при этом знак — означает, что $v_r = 0$, знак * — что значение v_r не определено для h -й строки ПСТ ($h \in \{1, \dots, H\}$).

Преобразованная ПСТ используется для формирования функций (13) и (17). Так, из табл. 3 могут быть найдены функции $z_1 = \bar{T}_1 \bar{T}_2 \vee T_1 \bar{T}_2 \bar{T}_3 \bar{x}_2 \bar{x}_4$ и $v_1 = \bar{T}_1 \bar{T}_2 T_3 \bar{x}_2 \bar{x}_3$.

Пусть для реализации схемы МПА используются ПЛИС, содержащие ВБП с $V_0 = 64$ и $Q = \{1, 2, 4, 8\}$. Для реализации блока $КС_3$ необходимо выбрать конфигурацию 16×4 , так как $R_Y + R_I = 4$. Поскольку условие (28) выполняется, то $L_W(S_3)_3 - R = 1$, т.е. один выход схемы $КС_3$ может использоваться для реализации микроопераций. Пусть $Y^1 = \{y_2\}$, тогда $|Y^2| = 8$ и для реализации блока $КС_2$ необходимо выбрать ВБП с конфигурацией 8×8 .

Для спецификации блока $КС_2$ необходимо построить таблицу со столбцами Z_h, Y_h, h , где Z_h содержит код набора $Y_q \subseteq Y$ из столбца Y_h ; $h \in \{1, \dots, H_2\}$ — номер строки. В нашем случае эта таблица имеет $H_2 = 8$ строк (табл. 4).

Таблица 4

Z_h	Y_h								h
	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	
000	0	0	0	0	0	0	0	0	1
001	1	0	0	0	0	0	0	0	2
010	0	1	0	0	0	0	1	0	3
011	0	0	1	0	0	0	0	0	4
100	1	0	0	1	0	0	0	0	5
101	0	0	0	0	1	0	0	1	6
110	0	1	0	0	0	1	0	0	7
111	0	0	0	0	0	0	0	0	8

Для спецификации блока $КС_3$ необходимо построить таблицу со столбцами Z_h, V_h, Φ_h, Y_h, h , где $h \in \{1, \dots, H_3\}$ — номер строки. Столбец Y_h содержит микрооперации $y_n \in Y^1$. Для автомата S_5 эта таблица содержит $H_3 = 16$ строк (табл. 5).

Поясним принцип заполнения этой таблицы. Например, $y_1 \in Y^1$ и $y_1 \in Y_2$. Поэтому бит $y_1 = 1$ для строк 3 и 4 таблицы, соответствующих $K(Y_2)$. Состояние a_2 соответствует коду 0010 (см. табл. 2), поэтому в строке 3 табл. 5 записаны $D_1 = D_2 = D_3 = 1$, что соответствует коду $K(a_2) = 001$. Состояние a_3 соответствует кодам 010* и 011* (см. табл. 2), поэтому его код 010 записан в строках 5, 6, 7 и 8 табл. 5.

Реализация схемы МПА S_5 сводится к реализации систем (13) и (17) на ЛЭ и к реализации таблиц, задающих блоки $КС_2$ и $КС_3$, на ВБП. В настоящее время существуют эффективные методы решения этих задач [3, 4], поэтому этот этап не рассматривается. Отметим, что переход от модели S_3 к модели S_5 позволил использовать только два ВБП с $V_0 = 64$. В схеме, определяемой моделью S_3 , требовалось бы три таких блока. При этом число ЛЭ в схемах блоков $КС_1$ для обеих моделей, определяемых ПСТ (см. табл. 1), одинаково.

Таблица 5

h	Z_h	V_h	Φ_h			Y_h
			D_1	D_2	D_3	y_1
1	000	0	0	0	0	0
2	000	1	0	0	0	0
3	001	0	0	0	1	1
4	001	1	0	1	1	1
5	010	0	0	1	0	0
6	010	1	0	1	0	0
7	011	0	0	1	0	0
8	011	1	0	1	0	0
9	100	0	0	1	1	0
10	100	1	0	1	1	0
11	101	0	1	0	0	0
12	101	1	1	0	0	0
13	110	0	1	0	0	0
14	110	1	1	0	0	0
15	111	0	0	0	0	0
16	111	1	0	0	0	0

6. ИССЛЕДОВАНИЕ ПРЕДЛОЖЕННЫХ МЕТОДОВ СИНТЕЗА

Для определения эффективности предложенных методов синтеза использовалась библиотека LGSynth91 [15], содержащая описания автоматов в формате KISSII. Эти описания являются стандартными тестовыми примерами, используемыми для сравнения различных методов синтеза МПА [5]. Для синтеза схем МПА использовалась система AS [9], расширенная программами для синтеза схемы МПА S_5 . Сравнение проводилось для схем, основанных на одноуровневой модели (автомат S_1), модели с кодированием наборов микроопераций (автомат S_2) и модели с преобразованием кодов наборов микроопераций в коды состояний и разбиением множества микроопераций (автомат S_5). Некоторые результаты исследований предложенных методов показаны в табл. 6. При этом схемы МПА синтезировались на микросхемах Virtex V50 [8] фирмы Xilinx.

В этой таблице в столбце ЛЭ приведено число LUT элементов, требуемых для реализации комбинационной схемы, а в столбце ВБП — число встроенных

Таблица 6

Название теста	Автомат S_1		Автомат S_2		Автомат S_5	
	ЛЭ	ВБП	ЛЭ	ВБП	ЛЭ	ВБП
cse	78	—	85	1	41	1
dk16	68	—	70	1	24	1
ex1	105	—	109	2	87	3
ex2	49	—	46	1	29	1
keyb	90	—	99	1	60	2
kirkman	58	—	138	1	56	1
planet	248	—	155	2	65	4
s1494	205	—	159	2	87	4
s298	951	—	843	1	252	4
sty5	199	—	182	1	78	1

блоков памяти, используемых схемами $КС_2$ и $КС_3$. Необходимо отметить, что процесс синтеза осуществлялся системой XST 8.1i фирмы Xilinx, входящей в пакет ISE 8.1i [8]. При использовании других пакетов результаты могут быть другими. Исследования показали, что предложенный метод практически всегда дает лучшие результаты, чем его прототипы. Исследования проводились для 47 тестовых примеров.

В среднем использование модели S_5 для этих автоматов позволяет на 38 % уменьшить число ЛЭ по сравнению с моделью S_2 и на 42 % — по сравнению с моделью S_1 . Выигрыш достигается за счет использования ВБП и разбиения множества микроопераций. Эти результаты оказались лучше, чем предвиделось при использовании вероятностного подхода, используемого для исследований в [10]. Моделирование, проведенное с помощью этой системы, подтвердило, что предложенные методы позволяют сохранить функциональность исходного автомата.

ЗАКЛЮЧЕНИЕ

В настоящей работе рассмотрена адаптация метода преобразования кодов объектов при реализации схемы автомата Мили в базисе ПЛИС с встроенными блоками памяти. Этот метод относится к методам структурной декомпозиции, которая выполняется перед применением функциональной декомпозиции реализуемых нерегулярных функций. Основная идея метода заключается в увеличении относительного числа регулярных функций, для реализации которых используются ВБП.

В работе рассмотрены две модели автомата Мили, основанные на этом подходе. В модели S_5 коды наборов микроопераций и некоторые дополнительные переменные формируются схемой, реализуемой на ЛЭ. Эти переменные исполь-

зуются для формирования микроопераций и функций возбуждения памяти МПА. Эта часть схемы реализуется на ВБП. В модели S_6 коды состояний и некоторые дополнительные переменные используются для формирования микроопераций. В обоих случаях множество микроопераций разбивается на два класса, что может уменьшить число блоков памяти, используемых для реализации регулярных функций.

В работе исследована эффективность реализации схемы автомата, основанной на модели S_5 , по сравнению с одноуровневой схемой МПА и схемой, основанной на кодировании наборов микроопераций. При этом использовались стандартные тестовые примеры из библиотеки LGSynth91 [15]. Исследования показали, что модель S_5 практически всегда приводит к схемам с меньшим числом ЛЭ. Кроме того, преобразование кодов объектов способствует более сбалансированному использованию ресурсов кристалла.

Дальнейшие направления исследований связаны с разработкой программных средств для синтеза схемы автомата S_6 . Это позволит делать выбор между моделями S_5 и S_6 , который практически невозможно произвести априори. Кроме того, необходимо исследовать временные характеристики схем, основанных на моделях S_1 – S_6 , что позволит выбирать модели, приводящие к схемам с требуемыми характеристиками по аппаратным затратам и быстродействию.

СПИСОК ЛИТЕРАТУРЫ

1. Baranov S. Logic and system design of digital systems. — Tallinn: TUT Press, 2008. — 266 p.
2. Minks P., Eliot I. FSM — based digital design using verilog. — London: John Wiley & Sons, 2008. — 351 p.
3. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. — Петербург: БХВ-Петербург, 2002. — 636 с.
4. Maxfield S. The design warrior's guide to FPGAs. — Amsterdam: Elsevier, 2004. — 541 p.
5. De Micheli G. Synthesis and optimization of digital circuits. — N.Y.: McGraw Hill, 1994. — 626 p.
6. Баркалов А.А., Титаренко Л.А. Синтез микропрограммных автоматов на заказных и программируемых СБИС. — Донецк: УНИТЕХ, 2009. — 336 с.
7. www.altera.com
8. www.xilinx.com
9. Bukowiec A. Synthesis of finite state machines for programmable devices based on multi-level implementation. — Zielona Gora: UZ Press, 2009. — 105 p.
10. Barkalov A., Barkalov A. Design of Mealy finite state machines with transformation of object codes // Appl. Mathematics and Comput. Sci. — 2005. — **15**, N 1. — P. 151–158.
11. Scholl C. Functional decomposition with application to FPGA synthesis. — Boston: Kluwer Academ. Publ., 2001. — 288 p.
12. Czerwinski R., Kania D., Kulisz I. FSM state encoding targeting at logic level minimization // Bulletin of the Polish Academ. Sci. — 2006. — **54**, N 4. — P. 479–487.
13. Barkalov A., Titarenko L. Logic Synthesis for FSM-based control units. — Berlin: Springer, 2009. — 233 p.
14. Соловьев В.В., Климович А.С. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем. — М.: Горячая линия – Телеком, 2008. — 376 с.
15. Yang S. Logic Synthesis and optimization benchmarks user guide. Techn. Rep. N 1991 — IWLS — UG — Saeyang. — Microelectronics Center of North Carolina, 1991. — 43 p.

Поступила 28.09.2009

После доработки 02.02.2010