# СИНТЕЗ МИКРОПРОГРАММНЫХ АВТОМАТОВ НА FPGA

**Красичков А.А.**
Донецький национальный технический университет,
кафедра ЭВМ
E-mail: Krasich@cs.dgtu.donetsk.ua

**Abstract.** ***Krasichkov A.A. The method of Mealy FSM's synthesis on FPGA based on functional decomposition is proposed.*** *Given method allows reducing the number of CLBs in respect to standard methods. An example of application of the proposed method is given.*

**Введение.** В настоящее время микросхемы программируемой логики с архитектурой FPGA (Field Programmable Gate Arrays — программируемые пользователем вентильные матрицы) получили широкое распространение [1]. Данная архитектура кардинально отличается от известных ранее программируемых логических устройств (ПЛУ). В отличие от программируемых логических матриц (ПЛМ) и программируемых матриц логики (ПМЛ), основу которых составляют вентили «И», «ИЛИ» и «НЕ» с фиксированным числом входов [2], микросхемы FPGA содержат большое число одинаковых конфигурационных логических блоков (КЛБ) с небольшим числом входов, соединенных программируемыми трассировочными ресурсами. Каждый КЛБ может реализовать любую булеву функцию (БФ) от R переменных, где R — число его входов. Проектирование микропрограммных автоматов (МПА) в базисе КЛБ требует специфичных методов синтеза, ориентированных на особенности базиса, поскольку число входов КЛБ всегда значительно меньше, чем число аргументов реализуемых БФ, и стандартные методы синтеза, применявшиеся ранее при реализации МПА [3], в данном случае не подходят. В настоящей работе предлагается метод синтеза МПА Мили на FPGA, основанный на функциональной декомпозиции булевых функций.

**Основные положения.** Пусть автомат Мили задан прямой структурной таблицей (ПСТ) со столбцами [4]:

$a_m$ — исходное состояние автомата, $a_m \in A$, где $A=\{a_1,\ldots,a_M\}$ — множество состояний автомата; $K(a_m)$ — код состояния $a_m$ разрядности $R=]\log_2 M[$, для ко-

дирования используются внутренние переменные $T_r \in T=\{T_1,\ldots,T_R\}$; $a_s$, $K(a_s)$ — состояние перехода и его код; $\overline{X_h}$ — входной сигнал, определяющий переход $<a_m,a_s>$ и равный конъюнкции некоторых элементов (или их отрицаний) множества логических условий $X = \{x_1,\ldots, x_L\}$; $Y_h$ — набор микроопераций, формируемых на переходе $<a_m,a_s>$, $Y_h \subseteq Y$, где $Y = \{y_1,\ldots,y_N\}$ — множество микроопераций; $D_h$ — набор функций возбуждения памяти автомата, принимающих единичные значения для переключения памяти из $K(a_m)$ в $K(a_s)$, $D_h \subseteq D$, где $D=\{D_1,\ldots,D_R\}$ — множество функций возбуждения; $h=\overline{1,H}$ — номер перехода. Эта таблица является основной для синтеза автомата Мили.

Традиционно для реализации на ПЛУ используется структура МПА Мили без дешифратора (Рис.1, а).

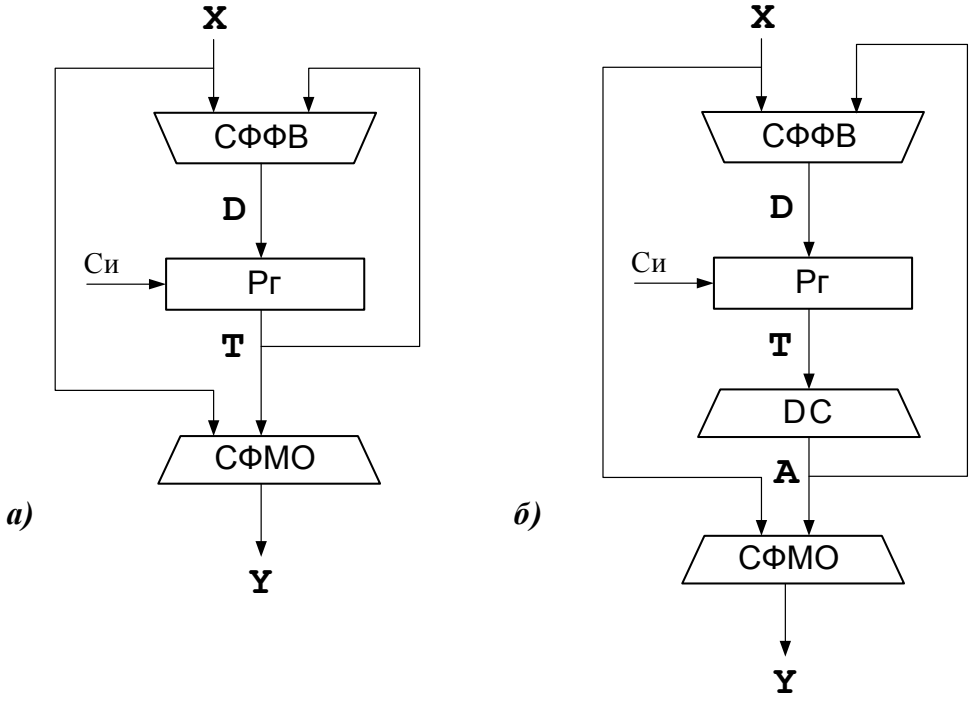


Рисунок 1 — Структуры МПА Мили

Здесь СФФВ — схема формирования функций возбуждения $D_h \subseteq D$, Рг — регистр кода состояния $T$, СФМО — схема формирования микроопераций $Y_h \subseteq Y$. Схемы СФФВ и СФМО строятся по системам функций (1) и (2) соответственно.

$$D_i = \bigvee_{h=1}^{H}\Big(\bigwedge_{r=1}^{R} T_r^{l_{mr}}\Big) \& \bigwedge_{l=1}^{L} x_l^{E_{lh}}, \qquad (1)$$

$$y_n = \bigvee_{h=1}^{H}\Big(\bigwedge_{r=1}^{R} T_r^{l_{mr}}\Big) \& C_{nh} \& \bigwedge_{l=1}^{L} x_l^{E_{lh}}, \qquad (2)$$

где $l_{mr} \in \{0,1,*\}$, $T_r^0 = \overline{T}r, T_r^1 = T_r, T_r^* = 1$, $E_{lh} \in \{0,1,*\}$, $x_l^0 = \overline{x}l, x_l^1 = x_l, x_l^* = 1$, $C_{nh}$ — булева переменная, равная единице, если и только если в h-й строке ПСТ записана переменная $y_n \in Y_h$.

Системы (1)–(2) наиболее эффективно реализуются на ПЛМ или ПМЛ [2]. Однако для реализации на FPGA целесообразнее использовать структуру МПА Мили с дешифратором DC (Рис.1, б), исходя из следующих соображений.

Основой для реализации БФ в FPGA являются КЛБ с $R$ входами [1]. В зависимости от модели ИС FPGA $R = \overline{2,5}$, тогда как число аргументов БФ проектируемых систем на порядок выше. Поэтому, для реализации БФ $P$ аргументов в базисе КЛБ с $R$ входами, необходимо представить ее в виде совокупности подфункций $M$ аргументов, причем $M \le R$ [5]. Приведение функции к такому виду называется функциональной декомпозицией и может быть выполнено множеством методов, обычно трудно реализуемых при больших значениях $P$ [1].

Уравнения системы (1) представляют собой ДНФ с большим числом элементарных конъюнкций в каждом терме. Так, при $L$=10, $M$=1000 ($R$=10) число переменных в каждом терме может достигать $L+R$=20, при этом, число термов может достигать для автоматов реальной сложности несколько тысяч. Для их реализации на FPGA необходимо выполнять декомпозицию, которая, как правило, приводит к значительному ветвлению блоков КЛБ в СФФВ. Оптимизация в этом случае возможна лишь за счет использования совпадающих конъюнктивных термов $\bigwedge_{r=1}^{R} T_r^{l_{mr}}$ для разных разрядов $D_i$.

Для МПА Мили с дешифратором схемы СФФВ и СФМО строятся по системам функций (3) и (4) соответственно.

$$D_i = \bigvee_{h=1}^{H} (\mathrm{A}_{\mathrm{m}}^{\mathrm{h}} \,\&\, \bigwedge_{l=1}^{L} x_l^{E_{lh}}), \tag{3}$$

$$y_n = \bigvee_{h=1}^{H} (\mathrm{A}_{\mathrm{m}}^{\mathrm{h}} \,\&\, C_{nh} \,\&\, \bigwedge_{l=1}^{L} x_l^{E_{lh}}), \tag{4}$$

где $A_m^h$ — конъюнкция внутренних переменных, соответствующая коду исходного состояния $a_m$ из $h$-й строки ПСТ, реализуемая схемой дешифратора DC.

При реализации систем (3)–(4) на FPGA число обратных связей между DC и схемами СФФВ и СФМО в $2^R/R$ раз больше, чем для систем (1)–(2), одна-

ко трассировочные ресурсы FPGA ориентированы на реализацию схем с большим числом межсоединений КЛБ. Кроме того, дешифратор исключает необходимость оптимизации схемы по переменным $T_r$ и уменьшает число аргументов в формулах систем (3)–(4), что упрощает их последующую декомпозицию.

**Пример применения метода.** Пусть МПА Мили задан прямой структурной таблицей (Табл.1) и необходимо реализовать его в виде структуры с дешифратором на FPGA с $R$=4. Полученные из ПСТ уравнения функций возбуждения и микроопераций имеют следующий вид:

$$D_1 = a_1 x_1 x_2 \overline{x}_3 \vee a_2 \overline{x}_4 \vee a_4 \overline{x}_4 x_2 \overline{x}_3 \vee a_3;$$
$$D_2 = a_1 x_1 x_2 x_3 \vee a_1 x_1 \overline{x}_2 \vee a_2 x_4 \vee a_2 \overline{x}_4 \overline{x}_2 \vee a_4 \overline{x}_4 x_2 x_3;$$
$$D_3 = a_1 \overline{x}_1 \vee a_2 x_1 \overline{x}_2 \vee a_4 \overline{x}_4 \overline{x}_2 \vee a_0;$$
$$y_1 = a_0 \vee a_1 \overline{x}_1 \vee a_1 x_1 x_2 \overline{x}_3 \vee a_2 \overline{x}_4 \vee a_4 \overline{x}_4 x_2 \overline{x}_3 \vee a_3;$$
$$y_2 = a_0 \vee a_1 \overline{x}_1 \vee a_1 x_1 \overline{x}_2 \vee a_4 \overline{x}_4 \overline{x}_2;$$
$$y_3 = a_1 x_1 \overline{x}_2 \vee a_4 \overline{x}_4 \overline{x}_2;$$
$$y_4 = a_1 x_1 x_2 x_3 \vee a_4 \overline{x}_4 x_2 x_3.$$

Таблица 1 — Прямая структурная таблица МПА Мили

| $a_m$ | $K(a_m)$ | $a_s$ | $K(a_s)$ | $X_h$ | $Y_h$ | $D_1$ | $D_2$ | $D_3$ | $H$ |
|---|---|---|---|---|---|---|---|---|---|
| $a_0$ | 000 | $a_1$ | 001 | 1 | $y_1,y_2$ | 0 | 0 | 1 | 1 |
| $a_1$ | 001 | $a_1$ | 001 | $\overline{x}_1$ | $y_1,y_2$ | 0 | 0 | 1 | 2 |
| | | $a_2$ | 010 | $x_1 x_2 x_3$ | $y_4$ | 0 | 1 | 0 | 3 |
| | | $a_3$ | 011 | $x_1 \overline{x}_2$ | $y_2,y_3$ | 0 | 1 | 1 | 4 |
| | | $a_4$ | 100 | $x_1 x_2 \overline{x}_3$ | $y_1$ | 1 | 0 | 0 | 5 |
| $a_2$ | 010 | $a_2$ | 010 | $x_4$ | $y_5$ | 0 | 1 | 0 | 6 |
| | | $a_4$ | 100 | $\overline{x}_4$ | $y_1$ | 1 | 0 | 0 | 7 |
| $a_3$ | 011 | $a_4$ | 100 | 1 | $y_1$ | 1 | 0 | 0 | 8 |
| $a_4$ | 100 | $a_0$ | 000 | $x_4$ | $y_5$ | 0 | 0 | 0 | 9 |
| | | $a_2$ | 010 | $x_2 x_3 \overline{x}_4$ | $y_4$ | 0 | 1 | 0 | 10 |
| | | $a_3$ | 011 | $\overline{x}_2 \overline{x}_4$ | $y_2,y_3$ | 0 | 1 | 1 | 11 |
| | | $a_4$ | 100 | $\overline{x}_2 x_3 \overline{x}_4$ | $y_1$ | 1 | 0 | 0 | 12 |

Все уравнения имеют одинаковую структуру, поэтому рассмотрим, например, реализацию на КЛБ функции $y_2$. Для удобства обозначим аргументы функции литерами *a, b, c, d, e* и *f*: $y_2 = a \vee b\bar{c} \vee bc\bar{d} \vee e\bar{d}\,\bar{f}$. Функция зависит от 6 аргументов, поэтому выполнить декомпозицию при помощи карт Карно не представляется возможным [5]. Выполним декомпозицию, применив разложение Шеннона, представив функцию в виде:

$$y_2 = F[a,b,c,G(d,e,f)]. \qquad (5)$$

При выполнении такого представления функция будет реализована на двух КЛБ. Для того чтобы (5) имело место, необходимо выполнить разложение Шеннона для $y_2$ по аргументам *a, b, c* так, чтобы 8 полученных подфункций от аргументов *d, e* и *f* произвольным образом делились на два класса, в каждом из которых функции равны. Разложение представлено в табл.2. При таком выборе аргументов разложение не имеет места, поскольку подфункции разбиваются не на два, а на три класса: $\{g_0, g_1\}$, $\{g_3\}$ и $\{g_2, g_4, g_5, g_6, g_7\}$.

Таблица 2 — Разложение $y_2$ по аргументам *a, b* и *c*

| *abc* | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| *def* | $g_0$ | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ | $g_7$ |
| 000 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 001 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 010 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 011 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 101 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 111 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

Далее необходимо производить разложение по всем оставшимся комбинациям из трех аргументов до тех пор, пока разложение не будет иметь место. Всего в нашем случае существует $C_6^3 = \frac{6!}{3!\cdot(6-3)!} = 20$ различных вариантов разложения. В том случае, если среди всех комбинаций ни одна не привела к искомому разложению, то можно попытаться представить функцию в следующем виде:

$$y_2 = F[G(a,c,d),b,d,e,f], \qquad (6)$$

то есть с общим аргументом *d*. В этом случае функция также будет реализована на двух КЛБ. Выполним разложение Шеннона для $y_2$ по аргументам *b, d, e,f* (Табл.3). В результате разложения получены функции $g_0 - g_{15}$ от аргументов *a, c* и *d*. Звездочками в таблице отмечены несуществующие значения функций $g_i$, поскольку в этих случаях значения аргумента *d* в левой и верхней частях таблицы не совпадают. Используя эти неопределенности можно привести функции к двум классам:

$$\{ g_0, g_1, g_3 - g_7 \} = a;$$
$$\{ g_2, g_8 - g_{15} \} = \overline{\overline{a}\,c\,d}\,.$$

Функция *G* находится как дизъюнкция подфункций $g_i$, относящихся к одному из классов: $G = g_0 \vee g_1 \vee g_3 \vee g_4 \vee g_5 \vee g_6 \vee g_7 = \overline{b}\,\overline{e} \vee \overline{b}\,d \vee \overline{b}\,f$ . Функция $y_2$ имеет следующий вид:

$$y_2 = a \,\&\, G \vee \overline{\overline{a}\,c\,d} \,\&\, \overline{G}\,. \qquad (7)$$

Таблица 3 — Разложение $y_2$ по аргументам *a, c* и *d*

| *bdef* | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *acd* | $g_0$ | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ | $g_7$ | $g_8$ | $g_9$ | $g_{10}$ | $g_{11}$ | $g_{12}$ | $g_{13}$ | $g_{14}$ | $g_{15}$ |
| 000 | 0 | 0 | 1 | 0 | * | * | * | * | 1 | 1 | 1 | 1 | * | * | * | * |
| 001 | * | * | * | * | 0 | 0 | 0 | 0 | * | * | * | * | 1 | 1 | 1 | 1 |
| 010 | 0 | 0 | 1 | 0 | * | * | * | * | 1 | 1 | 1 | 1 | * | * | * | * |
| 011 | * | * | * | * | 0 | 0 | 0 | 0 | * | * | * | * | 0 | 0 | 0 | 0 |
| 100 | 1 | 1 | 1 | 1 | * | * | * | * | 1 | 1 | 1 | 1 | * | * | * | * |
| 101 | * | * | * | * | 1 | 1 | 1 | 1 | * | * | * | * | 1 | 1 | 1 | 1 |
| 110 | 1 | 1 | 1 | 1 | * | * | * | * | 1 | 1 | 1 | 1 | * | * | * | * |
| 111 | * | * | * | * | 1 | 1 | 1 | 1 | * | * | * | * | 1 | 1 | 1 | 1 |

После обратной замены литер на аргументы и упрощения полученного выражения получаем:

$$y_2 = a_0 \vee \overline{x_1 x_2} \,\&\, \overline{G} = a_0 \vee \overline{x_1 x_2} \,\&\, [\overline{a_1} \,\&\, (\overline{a_4} \vee x_2 \vee x_4)]\,.$$

Реализация функции $y_2$ в виде (7) на КЛБ представлена на рис.2.

**Заключение.** Исследования автора показали, что по сравнению с известными методами, предложенный метод приводит к минимальной по числу КЛБ схеме микропрограммного автомата Мили. При этом схема автомата с дешифратором состояний требует в среднем на 15-20% меньше блоков КЛБ, чем ав-

томат без дешифратора. Предлагаемые методы декомпозиции могут быть использованы при создании отечественных САПР цифровых устройств на микросхемах программируемой логики с архитектурой FPGA.

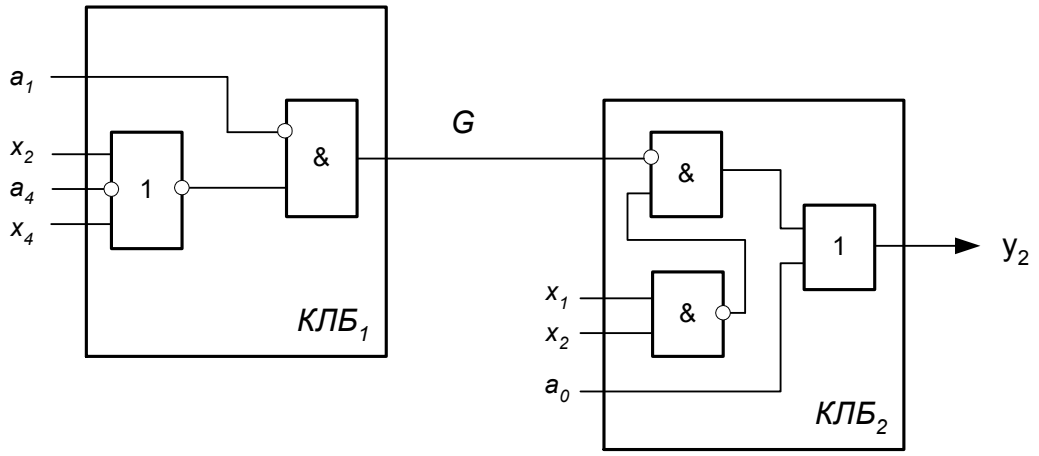


Рисунок 2 — Реализация функции $y_2$ на КЛБ

### *Литература*


1. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. — СПб.: БХВ-Петербург, 2002. — 608с.
2. Соловьев В.В. Проектирование функциональных узлов цифровых систем на программируемых логических устройствах. — Минск: Бестпринт, 1996 — 252с.
3. Скляров В.А. Синтез автоматов на матричных БИС. — Минск: Наука и техника, 1984 — 287с.
4. Баранов С. И. Синтез микропрограммных автоматов. — Л.: Энергия, 1979. — 232с.
5. Баркалов А.А., Красичков А.А. Методы декомпозиции булевых функций.// Научные труды ДонНТУ. Серия: ИКВТ-2002, выпуск 39. — Донецк: ДонНТУ, 2002.



Сдано в редакцию: 28.02.2003г.

Рекомендовано к печати: д.т.н., проф. Баркалов А.А.