

Предложен метод синтеза схемы совмещенного автомата в базе заказных интегральных схем. Метод основан на расширенной матрицы, генерирующей термы систем функций возбуждения памяти и выходных функций. Дополнительная часть матрицы генерирует термы для выходных функций автомата Мура и позволяет уменьшить площадь кристалла по сравнению с площадью двухуровневой схемы автомата. Приведены результаты исследований и пример синтеза схемы автомата.

Ключевые слова: совмещенный автомат, ASIC, синтез, кодирование состояний, матричная схема.

© А.А. Баркалов, Л.А. Титаренко,
Я.Е. Визор, А.В. Матвиенко, 2020

УДК 004.274

DOI:10.34229/2707-451X.20.2.8

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

СИНТЕЗ СОВМЕЩЕННОГО АВТОМАТА В БАЗИСЕ ASIC

Введение. Устройство управления (УУ) – это одно из главных блоков цифровых систем [1, 2]. Часто на практике модель совмещенного микропрограммного автомата (СМПА) используется для спецификации и синтеза схемы УУ [3]. Методы синтеза и оптимизации характеристик схем СМПА во многом зависят от особенностей логических элементов, используемых для имплементации схемы [4 – 7]. В настоящей работе мы рассматриваем случай использования заказных матричных СБИС типа ASIC (*application specific integrated circuit*) [8, 9]. Базис ASIC позволяет получить схемы с оптимальными характеристиками по быстродействию и потребляемой мощности [9].

СМПА совмещают особенности автоматов Мили и Мура [10, 11]. Это позволяет использовать методы оптимизации схем автоматов Мили и Мура для уменьшения площади кристалла, занимаемой схемой СМПА. Как правило, уменьшение площади кристалла ведет к уменьшению потребляемой мощности и времени распространения сигналов [9].

Отметим, что задача реализации схемы СМПА в базисе ASIC до настоящего времени не рассматривалась. Поэтому мы рассмотрим тривиальную реализацию СМПА в базисе ASIC и предложим один из возможных подходов к уменьшению площади кристалла, занимаемой схемой СМПА. Для спецификации поведения СМПА (реализуемого алгоритма управления) используем язык граф-схем алгоритма (ГСА) [1].

Тривиальная реализация СМПА в базисе ASIC. Четыре множества характеризуют СМПА: множество состояний $A = \{a_1, \dots, a_M\}$, множество логических условий (ЛУ) $X = \{x_1, \dots, x_L\}$, множество микроопераций (МО) автомата Мили $Y = \{y_1, \dots, y_{N_1}\}$ и множество МО автомата Мура $V = \{v_1, \dots, v_{N_2}\}$. Для синтеза схемы СМПА необходимо построить прямую структурную таблицу (ПСТ) [1], состоящую из H строк. Строка номер h соответствует вектору $\langle a_m, K(a_m), a_s, K(a_s), V(a_m), X_h, Y_h, \Phi_h, h \rangle$. Здесь a_m – исходное (текущее) состояние СМПА; $K(a_m)$ – код состояния $a_m \in A$; a_s – состояние перехода; $K(a_s)$ – код состояния

$a_s \in A$; $V(a_m)$ – набор микроопераций (НМО), формируемый в состоянии a_m ($V(a_m) \subseteq V$); X_h – конъюнкция ЛУ, определяющая переход $\langle a_m, a_s \rangle$; Y_h – НМО, формируемый при переходе $\langle a_m, a_s \rangle$ ($Y_h \subseteq Y$); Φ_h – набор функций возбуждения памяти (ФВП), принимающих единичное значение для перехода от $K(a_m)$ к $K(a_s)$; h – номер перехода ($h \in \{1, \dots, H\}$).

Для кодирования используются элементы множества внутренних переменных $T = \{T_1, \dots, T_R\}$. Коды состояний хранятся в специальном регистре RG . Как правило, RG состоит из R двухтактных D триггеров [4]. Поэтому множество ФВП состоит из элементов $\Phi = \{D_1, \dots, D_R\}$. Параметр R определяется выражением $R = \lceil \log_2 M \rceil$ [1, 2]. Условимся при кодировании состояний $a_m \in A$ использовать метод максимального кодирования.

ПСТ используется для формирования систем булевых функций (СБФ), задающих схему СМПА:

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

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

$$V = V(T). \tag{3}$$

Системы (1) – (3) определяют структурную схему U_1 , показанную на рис. 1. Она включает три комбинационные схемы и регистр памяти состояний.

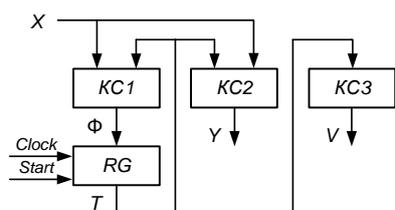


РИС. 1. Структурная схема СМПА U_1

Схема $KC1$ реализует СБФ (1), $KC2$ – СБФ (2) и $KC3$ – СБФ (3). Регистр RG хранит коды состояний. Смена содержимого RG происходит по сигналу синхронизации $Clock$. Сигнал $Start$ инициирует запись в RG кода начального состояния $a_1 \in A$.

Как правило, в ASIC системы булевых функций реализуются на заказных матрицах [9]. Такая матрица представляет собой набор горизонтальных и вертикальных шин, на пересечении которых находятся элементы с односторонней проводимостью. Аргументы функций связаны с вертикальными шинами, а их конъюнкции (термы) – с горизонтальными. Для каждого аргумента необходимы две вертикальные шины, одна для прямого значения аргумента, а другая – для инверсного. При этом каждая горизонтальная шина соответствует распределенной функции И – НЕ [9]. Соединение двух матриц позволяет реализовать систему булевых функций. Будем обозначать символом AND матрицу термов и символом OR матрицу функций (рис. 2).

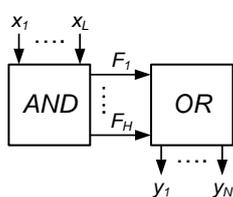


РИС. 2. Реализация СБФ на заказных матрицах

В системе на рис. 2 матрица M_1 (матрица AND) реализует термы F_1, \dots, F_H , образующие множество F . Матрица M_2 (матрица OR) реализует функции y_n , зависящие от термов $F_h \in F$. Сложность матричной реализации можно оценить площадью матриц, измеряемых в условных единицах [1]:

$$S(AND) = 2L \times H; \tag{4}$$

$$S(OR) = H \times N. \tag{5}$$

В выражениях (4), (5) символы $S(AND)$ и $S(OR)$ означают площади соответствующих матриц. Коэффициент 2 в (4) соответствует наличию двух вертикальных шин для каждого аргумента, входящего в термы.

Из анализа выражений (4), (5) следует, что для уменьшения площади кристалла, занимаемой схемой, необходимо уменьшать значения параметров L , N или H . При этом СБФ должна подвергаться совместной минимизации.

Условимся обозначать матрицы символами M_i . Пусть конъюнктивные матрицы имеют внутри символ "&", а дизъюнктивные – "1". Очевидно системы (1) – (3) зависят от одних и тех же переменных. Поэтому тривиальная схема СМПА включает две матрицы и регистр памяти состояний (рис. 3).

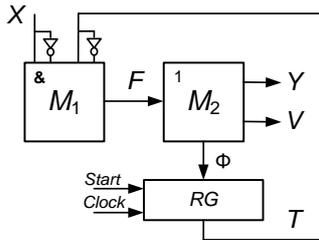


РИС. 3. Тривиальная матричная реализация СМПА U_1

Матрица M_1 реализует термы $F_h \in F$. Каждый терм представляет собой конъюнкцию

$$F_h = \bigwedge_{r=1}^R T_r^{l_{mr}} \cdot X_h, \quad (h = \overline{1, H}). \quad (6)$$

В выражении (6) первый член соответствует конъюнкции A_m , определяемой кодом $K(a_m)$ исходного состояния $a_m \in A$ для h -ой строки ПСТ. При этом $l_{mr} \in \{0, 1\}$ – значение r -го разряда кода $T_r^0 = \overline{T_r}$ и $T_r^1 = T_r$, $r \in \{1, \dots, R\}$.

Отметим, что функции $v_n \in V(a_m)$ зависят только от состояния $a_m \in A$. Поэтому в дизъюнктивные нормальные формы (ДНФ) этих функций входят все термы, содержащие конъюнкцию A_m . Это позволяет генерировать функции $v_n \in V$ на выходах матрицы M_2 .

Площади матриц M_1 и M_2 равны $S(M_1) = 2(L + R) \times H$ и $S(M_2) = (R + N_1 + N_2) \times H$ соответственно, а общая площадь кристалла $S(U_1)$ равна их сумме: $S(U_1) = (2L + 3R + N_1 + N_2) \times H$.

Известно [1], что двухуровневые матричные схемы автоматов обладают значительной избыточностью. Это справедливо и для схемы СМПА U_1 . Для устранения избыточности предлагаем подход, названный нами расширением матрицы термов.

Основная идея предлагаемого метода. Мы предлагаем генерировать термы A_m ($m = \overline{1, H}$) как выходы матрицы M_1 . Эти термы образуют множества $SA = \{A_1, \dots, A_M\}$. По существу, элементы множества SA – это унитарные коды состояний $a_m \in A$. Также предлагаем организовать матрицу M_1 в виде соединения двух матриц. Матрица M_{11} формирует термы (6), а матрица M_{12} – переменные $A_m \in SA$. Это ведет к СМПА U_2 (рис. 4).

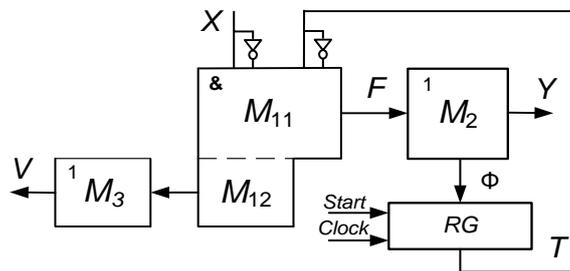


РИС. 4. Структурная схема СМПА U_2

В СМПА U_2 матрица M_2 формирует функции

$$\Phi = \Phi(F); \quad (7)$$

$$Y = Y(F). \quad (8)$$

Матрица M_3 формирует функции

$$V = V(SA). \tag{9}$$

Оценим сложность этой схемы, определив площади входящих в нее матриц:

$$\begin{aligned} S(M_{11}) &= 2(L + R) \times H; & S(M_{12}) &= 2R \times (M_0 + N_0); \\ S(M_2) &= H \times (R + N_1); & S(M_3) &= M_0 \times (N_2 - N_0). \end{aligned} \tag{10}$$

Символ M_0 означает число состояний, для которых $V(a_m) \neq 0$, N_0 – число МО, формируемых матрицей M_{12} .

Предложенный метод тем эффективнее, чем больше разность $S(U_1) - S(U_2)$. Очевидно, выражение $S(U_2)$ определяется, как сумма выражений (10).

Отметим, что для автоматов Мура возможна ситуация, когда входные функции представляются одним термом [6, 7]. В случае СМПА U_2 это позволяет формировать такие МО, как выходы матрицы M_{12} . При этом уменьшается число входов матрицы M_3 . К данной ситуации приводит специальное кодирование состояний (СКС) [6]. Для СКС может быть использован метод [12].

В настоящей работе предлагается метод синтеза СМПА U_2 . Метод включает этапы.

1. Отметка исходной ГСА Γ состояниями автомата Мура.
2. Выполнение специального кодирования состояний.
3. Формирование ПСТ совмещенного автомата.
4. Формирование системы термов для функций (7) – (9).
5. Формирование ДНФ функций (7) и (9).
6. Реализация схемы СМПА.

Пример синтеза схемы СМПА U_2 . Обозначим символом $U_i(\Gamma_j)$ схему СМПА U_i , синтезируемую по ГСА Γ_j . Рассмотрим пример синтеза СМПА $U_2(\Gamma_1)$. Исходная ГСА Γ_1 , отмеченная состояниями автомата Мура [1], показана на рис. 5.

Автомат $U_2(\Gamma_1)$ характеризуется множествами $A = \{a_1, \dots, a_7\}$, $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$ и $V = \{v_1, \dots, v_9\}$. В результате имеем $M = 7$, $L = N_1 = 3$ и $N_2 = 9$.

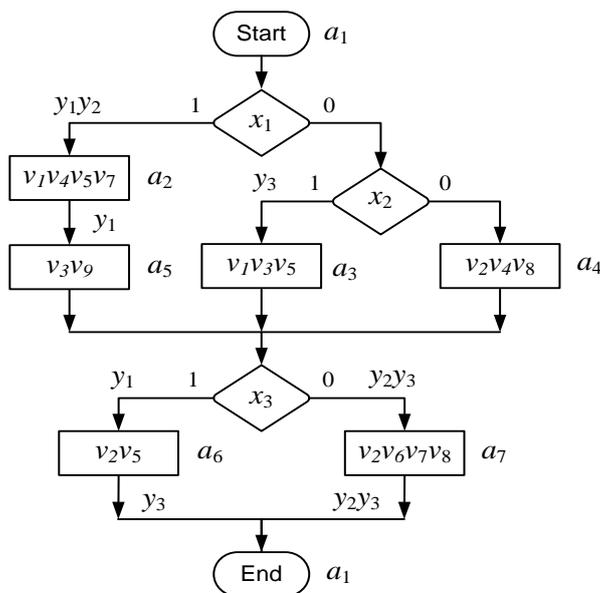


РИС. 5. Исходная ГСА Γ_1

Построим систему функций $V=V(A)$. Для этого достаточно использовать информацию из операторных вершин ГСА. В нашем случае имеем систему

$$\begin{aligned} v_1 &= A_2 \vee A_3; & v_2 &= A_4 \vee A_6 \vee A_7; & v_3 &= A_3 \vee A_5; \\ v_4 &= A_2 \vee A_4; & v_5 &= A_2 \vee A_3 \vee A_6; & v_6 &= A_7; \\ v_7 &= A_2 \vee A_7; & v_8 &= A_4 \vee A_7; & v_9 &= A_3. \end{aligned} \tag{11}$$

Выполним СКС для СМПА $U_2(\Gamma_1)$. Результат СКС показан на рис. 6.

	T_1	T_2		
T_3	00	01	11	10
0	a_1	a_2	a_7	a_4
1	a_5	a_3	*	a_6

РИС. 6. Коды состояний СМПА $U_2(\Gamma_1)$

Автомат $U_2(\Gamma_1)$ имеет $H = 12$ переходов, поэтому его ПСТ (таблица) имеет $H = 12$ строк.

Функции (8) – (9) зависят от термов (6). Эти термы соответствуют строкам ПСТ. Например, $F_1 = \overline{T_1 T_2 T_3} x_1$, $F_2 = \overline{T_1 T_2 T_3} x_1 x_2$ и так далее. Поскольку каждая строка ПСТ включает либо $Y_h \neq \emptyset$, либо $\Phi_h \neq \emptyset$, матрица M_{11} формирует все 12 термов.

Для нахождения термов системы (9), найдем ДНФ функций $v_n \in V$. Используя систему (11), коды (рис. 6) и законы алгебры логики [2], получаем систему

$$\begin{aligned} v_1 &= \overline{T_1} T_2; & v_2 &= T_1; & v_3 &= \overline{T_1} T_3; & v_4 &= \overline{T_1 T_2 T_3} \vee T_1 \overline{T_2 T_3} = A_2 \vee A_4; \\ v_5 &= \overline{T_1 T_2} \vee T_1 T_3 = A_8 \vee A_9; & v_6 &= T_1 T_2; & v_7 &= T_2 \overline{T_3}; & v_8 &= T_1 \overline{T_3}; & v_9 &= T_2 T_3. \end{aligned} \tag{12}$$

Как следует из (12), функции v_4 и v_5 формируются матрицей M_3 , которая имеет входы A_2, A_4, A_8 и A_9 . Остальные функции формируются матрицей M_{12} .

ТАБЛИЦА. ПСТ автомата $U_2(\Gamma_1)$

a_m	$K(a_m)$	a_s	$K(a_s)$	$V(a_m)$	X_h	Y_h	Φ_h	h
a_1	000	a_2	010	–	x_1	$y_1 y_2$	D_2	1
		a_3	011		$\overline{x_1 x_2}$	y_3	$D_2 D_3$	2
		a_4	100		$\overline{x_1 x_2}$	–	D_1	3
a_2	010	a_5	100	$v_1 v_4 v_5 v_7$	1	y_1	D_3	4
a_3	011	a_6	101	$v_1 v_3 v_5$	x_3	y_1	$D_1 D_3$	5
		a_7	110		$\overline{x_3}$	$y_2 y_3$	$D_1 D_2$	6
a_4	100	a_6	101	$v_2 v_4 v_8$	x_3	y_1	$D_1 D_3$	7
		a_7	110		$\overline{x_3}$	$y_2 y_3$	$D_1 D_2$	8
a_5	001	a_6	101	$v_3 v_9$	x_3	y_1	$D_1 D_3$	9
		a_7	110		$\overline{x_3}$	$y_2 y_3$	$D_1 D_2$	10
a_6	101	a_1	000	$v_2 v_5$	1	y_3	–	11
a_7	110	a_1	000	$v_2 v_6 v_7 v_8$	1	$y_2 y_3$	–	12

Система (12) представляет ДНФ (9) для рассматриваемого примера. Формулы для функций (7) – (8) тривиально следуют из ПСТ. Например, из таблицы имеем $D_1 = F_3 \vee F_5 \vee \dots \vee F_{10}$ и $y_1 = F_1 \vee F_4 \vee F_5 \vee F_7 \vee F_9$.

Реализация схемы СМПА сводится к формированию схем отдельных матриц и их межсоединений. Затем должны быть изготовлены маски для реализации этих схем. Этот шаг мы не рассматриваем.

Анализ эффективности предложенного метода. Сравним автоматы $U_1(\Gamma_1)$ и $U_2(\Gamma_1)$. Используя формулу (7) и параметры L, R, H, N_1, N_2 имеем $S(U_1) = 324$ условных единиц площади. Для автомата $U_2(\Gamma_1)$ необходимо использовать формулы (10).

Из выражения (10) $S(M_{11}) = 144$. Анализ системы (12) дает $M_0 = 4$ и $N_0 = 6$. При этом $M_0 \vee 2$ представляет собой выход первого триггера и не формируется на M_{12} . Это дает $S(M_{12}) = 2 \times 3 \times (4+6) = 60$. Далее из (10) имеем $S(M_2) = 12 \times (3+3) = 72$ и $S(M_3) = 4 \times 2 = 8$. Итак, $S(U_2) = 284$. Следовательно, применение предложенного метода в этом простом примере дает выигрыш в 12,4 %.

Для автоматов средней сложности имеем [1]: $L \approx 30, R \approx 7, N_1 + N_2 \approx 50, H \approx 2000, M \approx 100$. Это дает $S(U_1) \approx 262000$ условных единиц площади. Пусть $N_1 = N_2, M_0 \approx 50, N_0 \approx 15$. Тогда $S(M_{11}) = 148000, S(M_{12}) = 910, S(M_2) = 56000, S(M_3) = 500$. Это дает $S(U_2) \approx 205000$ условных единиц площади. Таким образом, переход от модели U_1 к модели U_2 дает экономию, равную 22 % площади кристалла.

Результаты моделирования для стандартных автоматов из библиотеки [13] подтвердили наши теоретические исследования. Выигрыш от перехода к U_2 варьировался от 10 % для простых автоматов и до 26 % для сложных. При этом он увеличивался по мере уменьшения параметра M_0 и увеличения N_0 .

Выводы. Предложенный в работе метод позволяет уменьшить площадь кристалла ASIC, занимаемой схемой СМПА. Для увеличения эффективности метода необходимо использовать специальное кодирование состояний, которое минимизирует число термов в СБФ выходных функций автомата Мура.

Исследования, проведенные на стандартных автоматах [13], показали, что наш метод позволяет уменьшить площадь ASIC от 10 % до 26 %. При этом выигрыш растет по мере роста числа состояний и строк ПСТ.

Исходная ГСА отмечается состояниями автомата Мура. Для автомата Мура характерно наличие псевдо эквивалентных состояний [14]. Используя классы таких состояний возможно уменьшить число строк ПСТ до показателя эквивалентного автомата Мили [6].

В наших дальнейших исследованиях рассмотрим возможность соединения методов, основанных на наличии псевдоэквивалентных состояний, и метода расширения матрицы термов. Кроме того, мы планируем использовать методы структурной декомпозиции [4, 6, 7] для уменьшения площади кристалла ASIC, занимаемой схемой совмещенного автомата. Отметим, что до настоящего времени подобные проблемы рассматривались только для оптимизации схем СМПА в базисе стандартных программируемых логических интегральных схем [3, 10, 11].

Список литературы

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. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В., Горина В.В. Уменьшение числа LUT элементов в схеме совмещенного автомата. *Управляющие системы и машины*. 2016. 3. С. 16–22.

4. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA-based Systems. Berlin: Springer, 2014. 432 p.
5. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. М.: Горячая линия ТЕЛЕКОМ, 2001. 636 с.
6. Barkalov A., Titarenko L. Logic Synthesis for FSM-based Control Units. Berlin: Springer, 2009. 233 p.
7. Barkalov A., Titarenko L., Kolopenczyk M., Mielcarek K., Bazydlo G. Logic Synthesis for FPGA-based Finite State Machines. Berlin: Springer, 2016. 280 p.
8. Smith M. Application Specific Integrated Circuits. Boston: Addison-Wesley, 1997. 632 p.
9. Nababi Z. Embedded Core Design with FPGAs. New York: McGraw-Hill, 2008. 418 p.
10. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Синтез совмещенного микропрограммного автомата в базисе FPGA. *Комп'ютерні засоби, мережі та системи*. 2015. 14. С. 32–39.
11. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Реализация схемы совмещенного микропрограммного автомата в базисе FPGA. *Проблеми інформатизації та управління*. 2015. 3 (51). С. 5–13.
12. Ачасова С.М. Алгоритмы синтеза автоматов на программируемых логических матрицах. М.: Советское радио, 1987. 132 с.
13. Yang S. Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina, 1991. 43 p.
14. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура. *Кибернетика и системный анализ*. 1998. 1. С. 65–72.

Получено 15.06.2020

Баркалов Александр Александрович,

доктор технических наук, профессор Университета Зеленогурского (Польша),
<https://orcid.org/0000-0002-4941-3979>

Титаренко Лариса Александровна,

доктор технических наук, профессор Университета Зеленогурского (Польша),
профессор Харьковского национального университета радиоэлектроники,
<https://orcid.org/0000-0001-9558-3322>

Визор Ярослав Евстафьевич,

кандидат технических наук, старший научный сотрудник
Института кибернетики имени В.М. Глушкова НАН Украины,

Матвиенко Александр Владимирович,

научный сотрудник Института кибернетики имени В.М. Глушкова НАН Украины.
<https://orcid.org/0000-0003-1838-1422>
avmatv@ukr.net

УДК 004.274

О.О. Баркалов¹, Л.О. Титаренко^{1,2}, Я.Є. Візор³, О.В. Матвієнко^{3*}

Синтез суміщеного автомата в базисі ASIC

¹ Університет Зеленогурський, Зелена Гура, Польща

² Харківський національний університет радіоелектроніки, Харків, Україна

³ Інститут кібернетики імені В.М. Глушкова НАН України, Київ

* Листування: avmatv@ukr.net

Вступ. Модель кінцевого автомата широко використовується для завдання поведінки різних послідовних блоків, наприклад, пристроїв управління, які можуть мати вихідні сигнали двох типів – Мілі і Мура. Для синтезу подібних пристроїв можна використовувати модель суміщеного автомата.

При реалізації схем автоматів необхідно оптимізувати його характеристики, такі як апаратні витрати. Методи вирішення цього завдання в значній мірі залежать від використовуваного елементного базису.

У статті запропоновано метод зменшення апаратних витрат у схемі суміщеного автомата, що реалізується в базисі замовних матричних схем. Запропонований метод дозволяє зменшити площу кристала ASIC, займану схемою суміщеного автомата. Метод заснований на розширенні матриці, що генерує терми систем функцій збудження пам'яті і вихідних функцій. Додаткова частина матриці генерує терми для вихідних функцій автомата Мура і дозволяє зменшити площу кристала в порівнянні з площею дворівневої схеми автомата.

Мета роботи. Показати, як розділення матриць схеми автомата дозволяє зменшити результуючу площу схеми. При цьому оцінки витрат апаратури для тривіальної структури автомата і запропонованого підходу визначаються в умовних одиницях площі.

Результати. Запропоновано метод синтезу автомата з розширенням матриці термів. На прикладі показано, як виконувати кроки запропонованого методу синтезу. Для збільшення ефективності методу запропоновано використовувати спеціальне кодування станів, яке мінімізує число термів у системах булевських функцій для виходів автомата Мура. Дослідження, проведені на стандартних автоматах, показали, що запропонований метод призводить до зменшення площі ASIC від 10 % до 26 %. При цьому виґраш зростає за мірою зростання складності автомата.

Висновки. Порівняння з відомими методами синтезу показало, що розширення матриці, що генерує терми систем функцій збудження пам'яті і вихідних функцій, дозволяє зменшити площу кристала, займану схемою суміщеного автомата.

Ключові слова: суміщений автомат, ASIC, синтез, кодування станів, матрична схема.

UDC 004.274

A.A. Barkalov¹, L.A. Titarenko^{1,2}, Y.E. Vizer³, A.V. Matvienko^{3*}

Synthesis of a combined automaton with ASIC

¹ University of Zielona Gora, Poland

² Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

³ V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv

* Correspondence: avmatv@ukr.net

Introduction. The model of a finite state machine is widely used for describing behavior of different sequential blocks, such as control units. It is possible that control units possess output signals having both types of Mealy and Moore automata. A model of the combined automaton can be used to synthesize such devices.

When the automaton circuit is implemented, it is necessary to optimize its characteristics such as hardware amount. The methods of this task solution depend significantly on logic elements used to implement circuits.

In this article, we propose a method of reducing hardware in the circuit of combined automaton implemented with ASIC. In this case, the circuit is implemented using customized matrix circuits. The proposed method allows reducing the chip area occupied by the circuit of the automaton. The method is based on the expansion of the matrix that generates circuit product terms of the systems of input memory functions and output functions of the combined automaton. The additional part of the matrix generates terms for output functions of Moore automaton. It allows reduction of the chip area as compared to the area of the two-level circuit of the combined automaton.

The purpose of the article is to show that the division of circuit matrices allows reducing the resulting matrix area. The hardware amount is estimated for both trivial automaton structure and for the proposed approach. They are determined in conventional units of area.

Results. The method is proposed based on the expansion of the matrix of terms. Using an example, it is shown how to execute the steps of the proposed method. To increase the method efficiency, it is proposed to use a special state assignment that minimizes the number of terms in the systems of Boolean functions of outputs with Moore type. The conducted investigations show that the proposed method allows for reducing the resulting ASIC area from 10% to 26%. The gain increases with the growth of the automaton complexity.

Conclusions. A comparison of the proposed method with some known synthesis methods shows that the expansion of the matrix of terms for systems of input memory functions and output functions allows reducing the chip area occupied by the circuit of the combined automaton.

Keywords: combined automaton, ASIC, synthesis, state encoding, matrix circuit.