

DOI <https://doi.org/10.15407/usim.2020.04.005>
UDC 004.274

О.О. БАРКАЛОВ, доктор техн. наук, професор,
Зеленогурський Університет (Польща),
65246, Зелена Гура, вул. Підгірна, 50, Польща,
a.barkalov@iie.uz.zgora.pl

Л.О. ТПАРЕНКО, доктор техн. наук,
Зеленогурський Університет (Польща),
65246, Зелена Гура, вул. Підгірна, 50, Польща,
L.Titarenko@iie.uz.zgora.pl, D_ts@nure.ua

Я.Є. ВІЗОР, канд. техн. наук, ст. наук. співробітник,
Ін-т кібернетики ім. В.М. Глушкова НАН України,
просп. Глушкова, 40, Київ 03187, Україна,
yaviz@ukr.net

О.В. МАТВІЄНКО, наук. співробітник,
Ін-т кібернетики ім. В.М. Глушкова НАН України,
просп. Глушкова, 40, Київ 03187, Україна,
avmatv@ukr.net

СИНТЕЗ СУМІЩЕНОГО АВТОМАТА ЗІ ЗМЕНШЕННЯМ ПЛОЩІ НАНО-ПЛМ

Пропонується метод зменшення площі схеми суміщеного автомата, в базисі нано-ПЛМ. Метод заснований на оптимальному кодуванні станів автомата Мура, який враховує наявність класів псевдоеквівалентних станів. При цьому виділяється частина схеми, яка реалізує функції автомата Мура. Запропонований метод дає змогу зменшити площу нано-ПЛМ, необхідну для реалізації схеми, в порівнянні з тривіальною дворівневою схемою. Наведено результати досліджень і приклад синтезу схеми автомата.

Ключові слова: суміщений мікропрограмний автомат, синтез, нано-ПЛМ, ASIC, кодування станів.

Вступ

За масового виробництва виробів електроніки широко використовуються замовні надвеликі інтегральні схеми (НВІС) типу *ASIC* (*application specific integrated circuit*) [1, 2]. Для зменшення споживаної енергії та збільшення швидкодії схем у базисі *ASIC* необхідно зменшувати

площу кристала, займаного схемою [1]. Це стосується і схем, поведінка яких задається системами булевих функцій (СБФ).

У 70 — 80-х роках ХХ століття для реалізації СБФ часто використовували програмовані логічні матриці (ПЛМ) [3, 4]. Останнім часом

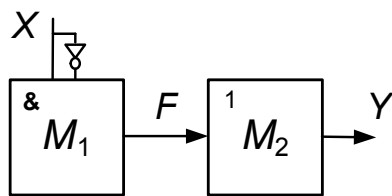


Рис. 1. Реалізація СБФ на нано-ПЛМ

ПЛМ-подібні структури використовуються й у складі *ASIC* [2]. Вони отримали назву нано-ПЛМ [5, 6]. Для оптимізації площі нано-ПЛМ необхідно виконати спільну мінімізацію реалізованих СБФ [7].

Одним з важливих блоків цифрових систем є пристрої управління (ПУ) [8, 9]. Дуже часто для синтезу пристрою управління використовується модель суміщеного мікропрограмного автомата (СМПА) [10, 11]. Однак методів синтезу схем СМПА в базисі нано-ПЛМ на цей час немає. Оскільки ця проблема є важливою, у статті пропонується метод синтезу СМПА на нано-ПЛМ. Як мова специфікації СМПА використовується мова граф-схем алгоритму (ГСА) [9].

Особливості СМПА і нано-ПЛМ

Основною особливістю СМПА є наявність двох типів вихідних змінних [8, 14, 15], які ми називатимемо мікроопераціями (МО). Змінні типу Мілі утворюють множину $Y = \{y_1, \dots, y_{M_1}\}$. Змінні $y_n \in Y$ існують тільки на час переходів між станами автомата [8]. Змінні типу Мура існують практично весь такт роботи автомата [3]. Вони утворюють множину $V = \{v_1, \dots, v_{N_2}\}$. Стани автомата утворюють множину $A = \{a_1, \dots, a_M\}$. Переходи між станами визначаються логічними умовами (ЛУ) $X = \{x_1, \dots, x_L\}$ і поточним станом. Змінні $x_i \in X$ є вхідними змінними СМПА.

Для синтезу схеми СМПА необхідно закодувати стани $a_m \in A$ внутрішніми змінними з множини $T = \{T_1, \dots, T_R\}$. Коди станів $K(a_m)$ зберігаються в спеціальному регістрі *RG*. Як правило, для реалізації схеми *RG* використовуються двотактні тригери типу

D [3,9]. У множині *A* виділяють початковий стан $a_i \in A$, код якого завантажується в *RG* за імпульсом *Start*. Зміна вмісту *RG* можлива при появі певного фронту імпульсу синхронізації *Clock*. Для задання коду стану переходу використовуються елементи множини функцій збудження пам'яті (ФЗП) $\Phi = \{D_1, \dots, D_R\}$.

Домовимося використовувати мінімальну кількість елементів множини *T*. Це число визначається як

$$R = \lceil \log_2 M \rceil. \quad (1)$$

СМПА має особливості автоматів Мілі та Мура, тому для оптимізації схем СМПА можна використовувати методи оптимізації цих обох автоматів. У цій статті ми пропонуємо метод, заснований на наявності класів псевдоеквівалентних станів (ПЕС) автомата Мура [16]. Кожен клас ПЕС відповідає одному стану еквівалентного автомата Мілі [16].

Нано-ПЛМ — це послідовне з'єднання кон'юнктивної матриці M_1 і диз'юнктивної матриці M_2 (рис. 1). Нано-ПЛМ реалізує систему функцій $Y = Y(X)$, що залежать від кон'юнктивних термів $F_h \in F (h = 1, H_0)$.

Терми $F_h \in F$ залежать від прямих і/або інверсних значень вхідних змінних. Цей факт визначає наявність інвертора на рис. 1. Складність схеми оцінюється її площею [8]:

$$\begin{aligned} S(M_1) &= 2L \times H; \\ S(M_2) &= H \times N; \\ S(Y) &= S(M_1) + S(M_2). \end{aligned} \quad (2)$$

У системі (2) символи $S(M_1)$, $S(M_2)$ і $S(Y)$ означають площі матриць M_1 , M_2 та схеми в цілому, відповідно.

Реальну площу схеми з урахуванням міжз'єднань можна визначити, лише знаючи розміри елементів. Тому оцінки (2) мають попередній характер і виражаються в так званих умовних одиницях площі (у.о.п.) [8]. Ми використовуватимемо подібні оцінки для порівняння різних структур схем СМПА.

Як впливає з (2), для зменшення $S(Y)$ необхідно зменшити значення хоча б одного з параметрів *L*, *N* або *H*. Зазвичай пара-

метри L або N є фіксованими. Тому зменшення площі можливо тільки шляхом зменшення параметра H . Отже, необхідно виконати спільну мінімізацію СБФ Y [7].

Тривіальна матрична схема СМПА

У найпростішому випадку схема СМПА реалізується у вигляді двох матриць. Одна з них формує терми, що входять до ДНФ функцій $y_n \in Y$, $v_n \in V$ і $D_r \in \Phi$. Друга матриця реалізує ці функції. Позначимо цю тривіальну схему символом U_1 (рис. 2).

Для реалізації схеми U_1 необхідно побудувати пряму структурну таблицю (ПСТ) СМПА [8]. Цьому етапу передують кодування станів $a_m \in A$ двійковими кодами $K(a_m)$ розмірності (1). За своєю суттю ПСТ є списком переходів СМПА. Кожен перехід представляється рядком із такими стовпчиками: a_m – початковий стан; $K(a_m)$ – код стану $a_m \in A$; $V(a_m)$ – набір мікрооперацій (НМО), що формується в стані $a_m \in A$; a_s – стан переходу; $K(a_s)$ – код стану $a_s \in A$; X_h – вхідний сигнал, що визначає перехід $\langle a_m, a_s \rangle$ і дорівнює кон'юнкції деяких елементів (або їх інверсій) множини X ; Y_h – НМО, що формується на переході $\langle a_m, a_s \rangle$ ($Y_h \subseteq Y$); Φ_h – набір ФЗП, що приймають середнє арифметичне значення для перемикання RG з $K(a_m)$ в $K(a_s)$; h – номер переходу.

Площу схеми U_1 можна оцінити так:

$$S(M_1) = 2(L+R) \times H;$$

$$S(M_2) = (R+N1+N2) \times H;$$

$$S(U_1) = S(M_1) + S(M_2).$$

Параметри L , R , $N1$ і $N2$ є фіксованими для даної ГСА Γ . Тому зменшення площі $S(U_1)$ можливе тільки за рахунок зменшення кількості термів H .

Позначимо символом U_i (Γ_ϕ) той факт, що ГСА Γ_j використовується для реалізації схеми СМПА зі структурою U_i . Розглянемо характеристики СМПА $U_1(\Gamma_1)$, де ГСА Γ_1 наведено на рис. 3. При цьому Γ_1 позначено станами автомата Мура.

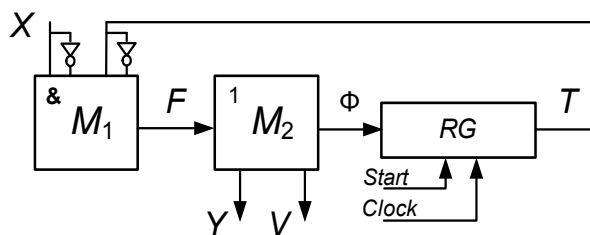


Рис. 2. Структурна схема СМПА U_1

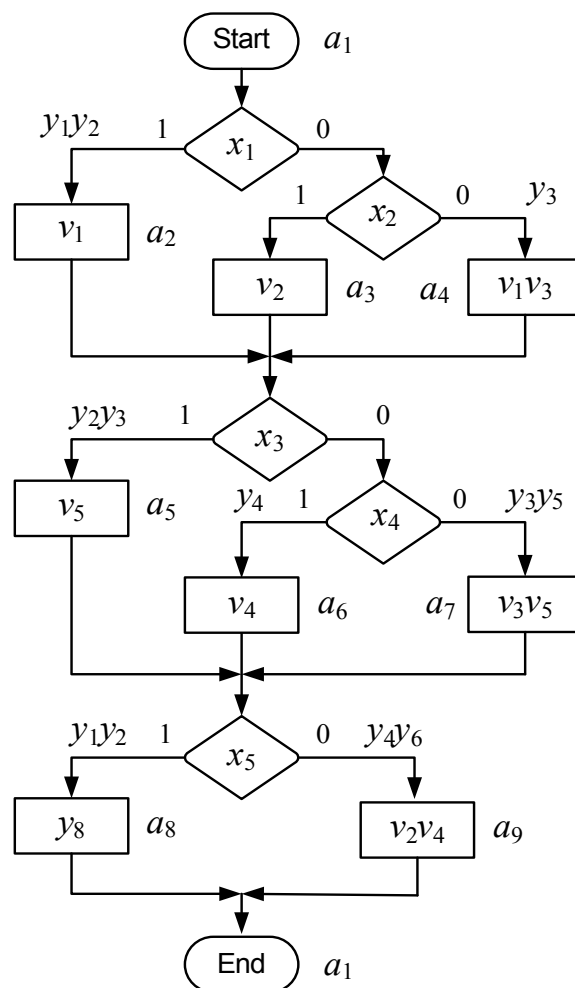


Рис. 3. Початкова ГСА Γ_1

		$T_1 T_2$			
		00	01	11	10
$T_3 T_4$	00	a_1	a_2	a_8	a_5
	01	*	a_3	a_9	a_6
	11	*	*	*	*
	10	*	a_4	*	a_7

Рис. 4. Оптимальні коди станів

З аналізу ГСА Γ_1 можна отримати множини $A = \{a_1, \dots, a_9\}$, $X = \{x_1, \dots, x_3\}$, $Y = \{y_1, \dots, y_6\}$, $V = \{v_1, \dots, v_3\}$. Це уможливорює отримання параметрів $M=9$, $L=N_2=5$, $N_1=6$. Використовуючи (1), маємо число $R=4$, що визначає множини $\Phi = \{D_1, \dots, D_4\}$ і $T = \{T_1, \dots, T_4\}$. Також можна визначити, що автомат має $H=18$ переходів. Отже, затривіальної реалізації цього автомата отримуємо такі площі:

$$S(M_1) = 2 \times (5+4) \times 18 = 324,$$

$$S(M_2) = 18 \times (5+6+5) \times 18 = 288,$$

$$S(U_1) = 612 \text{ (у.о.п.)}.$$

Основна ідея пропонуваного методу

Для автомата Мура характерна наявність станів з однаковими переходами. Такі стани утворюють класи ПЕС [16]. Наприклад, для ГСА Γ_1 можна знайти розбиття $\Pi_A = \{B_1, \dots, B_4\}$ множини A на класи $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5, a_6, a_7\}$ та $B_4 = \{a_8, a_9\}$.

У цій статті ми пропонуємо використовувати класи ПЕС для оптимізації схеми СМПА U_1 . Для цього необхідно закодувати стани так, щоб кожен клас $B_i \in \Pi_A$ було представлено мінімально можливою кількістю інтервалів R -розмірного булевого простору. Таке кодування називається оптимальним [13], а для його виконання можна використовувати метод із праці [4].

Для автомата, відповідного ГСА Γ_1 , один із варіантів оптимального кодування показано на рис. 4. Тут B_1 представлено інтервалом 00^{**} , $B_3 - 01^{**}$, $B_3 - 10^{**}$ и $B_4 - 11^{**}$.

Такий підхід дає змогу зменшити кількість рядків ПСТ до величини H_0 . Зазначмо, що H_0 — це кількість переходів еквівалентного автомата Мілі [16]. Для синтезу автомата Мура з ПЕС необхідно побудувати перетворену ПСТ (ППСТ) [13, 16], у якій вихідні стани $a_m \in A$ замінено класами $B_i \in \Pi_A$.

ППСТ дає змогу отримати терми F_h ($h = \overline{1, H_0}$), що визначають функції Φ і Y . Терми визначаються у такий спосіб:

$$F_h = \bigwedge_{r=1}^R T_r^{l_r} \cdot X_h, \quad (h = \overline{1, H_0}) \quad (3)$$

Перший член виразу (3) визначає кон'юнкцію внутрішніх змінних, відповідно до коду $K(B_i)$. Цей код визначається інтервалом булевого простору, отриманим внаслідок оптимального кодування станів. Змінні $l_r \in \{0, 1, *\}$ відповідають біту r коду $K(B_i)$, $T_r^0 = \overline{T_r}$, $T_r^1 = T_r$ і $T_r^* = 1, (r = \overline{1, R})$.

Тепер матриця M_1 генерує терми (6). Ці терми не входять до функції $v_r \in V$. Отже, для генерації функцій $v_r \in V$ необхідна окрема матрична схема. Це веде до пропонуваного в цій статті СМПА U_2 (рис. 5).

У схемі автомата U_2 матриці $M_1 - M_4$ генерують такі функції:

$$\begin{aligned} M_1: F_0 &= F_0(T, X); \\ M_2: \Phi &= \Phi(F_0); \quad Y = Y(F_0); \\ M_3: A_V &= A_V(T); \\ M_4: V &= V(A_V). \end{aligned} \quad (4)$$

У (4) символ F_0 позначає множину термів виду (3). Тут $T_0 \subseteq T$ — множина внутрішніх змінних, що входять до термів (3), де $|T_0| = R_0$. Символом A_V позначено множину термів, що входять у функції $v_r \in V$, де $|A_V| = H_V$.

Автомат U_2 можна оцінити такими площами:

$$S(M_1) = 2(R_0+L) \times H_0;$$

$$S(M_2) = (R+N1) \times H_0;$$

$$S(M_3) = 2R \times H_V;$$

$$S(M_4) = H_V \times N2.$$

Сума наведених виразів для $S(M_1) - S(M_4)$ дає площу $S(U_2)$.

Очевидно, що запропонований метод доцільно використовувати за виконання умови $S(U_1) > S(U_2)$. Наш аналіз бібліотеки стандартних автоматів [17] показав, що ця умова виконується для 84% усіх прикладів.

У цій статті пропонується метод синтезу схеми СМПА $U_2(\Gamma_j)$, що включає такі етапи:

1. Позначення початкової ГСА станами автомата Мура.
2. Формування розбиття $\Pi_A = \{B_1, \dots, B_l\}$.
3. Оптимальне кодування станів $a_m \in A$.
4. Формування ПСТ автомата $S(\Gamma_j)$.
5. Формування СБФ для матриць $M_1 - M_4$.
6. Реалізація схеми СМПА в базисі нано-ПЛМ.

Розгляньмо приклад синтезу СМПА $U_2(\Gamma_1)$. Зазначмо, що на цей момент нами вже виконано три етапи синтезу для нашого прикладу.

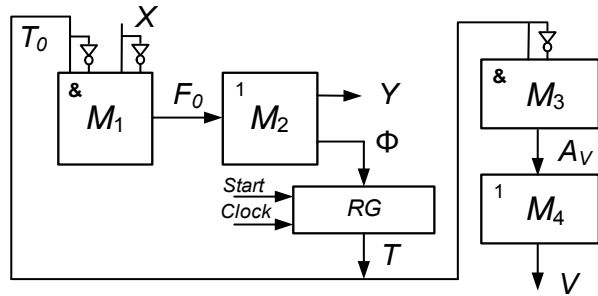


Рис. 5. Структурна схема СМПА U_2

Приклад синтезу схеми СМПА U_2

Починаємо приклад з п. 4, запропонованого методу. Побудуємо систему узагальнених формул переходу [12] для ГСА Γ_1 :

$$B_1 \rightarrow x_1 a_2 \vee \overline{x_1} x_2 a_3 \vee \overline{x_1} x_2 a_4;$$

$$B_2 \rightarrow x_3 a_5 \vee \overline{x_3} x_4 a_6 \vee \overline{x_3} x_4 a_7;$$

$$B_3 \rightarrow x_5 a_8 \vee \overline{x_5} a_9, B_4 \rightarrow a_1.$$

Кожен терм цієї системи відповідає одному рядку ПСТ таблиці. У цій ПСТ міститься $H_0 = 8$ рядків. При цьому переходи зі станів $a_m \in B_4$ не розглядаються, оскільки для них характерні нульові значення ФЗП.

Використовуючи таблицю знайдемо терми F_0 і функції Φ і Y із системи (4). Це виконується у тривіальний спосіб і дає такий результат:

Таблиця. ПСТ автомата $U_2(\Gamma_1)$

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Y_h	Φ_h	h
B_1	00**	a_2	0100	x_1	$y_1 y_2$	D_2	1
		a_3	0100	$\overline{x_1} x_2$	—	$D_2 D_4$	2
		a_4	0110	$\overline{x_1} x_2$	y_3	$D_2 D_3$	3
B_2	01**	a_5	1000	x_3	$y_2 y_6$	D_1	4
		a_6	1001	$\overline{x_3} x_4$	y_4	$D_1 D_4$	5
		a_7	1010	$\overline{x_3} x_4$	$y_3 y_5$	$D_1 D_3$	6
B_3	10**	a_8	1100	x_5	$y_1 y_2$	$D_1 D_3$	7
		a_9	1101	$\overline{x_5}$	$y_4 y_6$	$D_1 D_2 D_4$	8

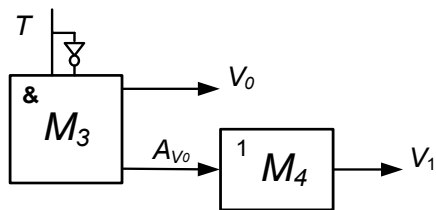


Рис. 6. Матрична реалізація схеми для формування виходів автомата Мура

$$F_1 = \bar{T}_1 \bar{T}_2 x_1, F_2 = \bar{T}_1 \bar{T}_2 \bar{x}_1 x_2, F_3 = \bar{T}_1 \bar{T}_2 \bar{x}_1 \bar{x}_2, \\ F_4 = \bar{T}_1 T_2 x_3, F_5 = \bar{T}_1 T_2 \bar{x}_3 x_4, F_6 = \bar{T}_1 T_2 \bar{x}_3 \bar{x}_4, \\ F_7 = T_1 \bar{T}_2 x_5, F_8 = T_1 T_2 \bar{x}_5, \quad (5)$$

$$y_1 = F_1 \vee F_7, y_2 = F_1 \vee F_4, y_3 = F_3 \vee F_4 \vee F_6, \\ y_4 = F_5 \vee F_8; y_5 = F_6; y_6 = F_8. \quad (6)$$

$$D_1 = F_4 \vee \dots \vee F_8; D_2 = F_1 \vee F_2 \vee F_3 \vee F_7 \vee F_8; \quad (7)$$

$$D_3 = F_3 \vee F_6; D_4 = F_2 \vee F_5 \vee F_8.$$

Система (5) визначає вміст матриці M_1 , системи (6) – (7) – матриці M_2 . Площі цих матриць визначаються як

$$S(M_1) = 2 \times (2 + 5) \times 8 = 112; \\ S(M_2) = 8 \times (4 + 6) = 80. \quad (8)$$

Для формування СБФ для $v_r \in V$ необхідно використовувати інформацію з операторних вершин ГСА Γ_1 . Це дає таку СБФ:

$$v_1 = A_2 \vee A_4 = \bar{T}_1 \bar{T}_2 \bar{T}_4 = F_9, v_2 = A_3 \vee A_9 = T_2 T_4 = F_{10} \\ v_3 = A_4 \vee A_7 = T_3, v_4 = A_6 \vee A_9 = T_1 T_4 = F_{11}; \\ v_5 = A_5 \vee A_7 \vee A_8 = T_1 \bar{T}_4 = F_{12}; \quad (9)$$

Отже, множина $A_v = \{F_9, \dots, F_{12}\}$, що дає значення $M_v = 4$. Функція v_3 формується на виході третього тригера RG . З аналізу системи (9) випливає, що кожна функція $v_r \in V$ визначається лише одним термом. Через цю

особливість для цього прикладу немає матриці M_4 . Площа матриці M_3 визначається як

$$S(M_3) = 2 \times 2 \times 4 = 16. \quad (10)$$

Додавання (8) і (10) дає загальну площу схеми $S(U_2) = 208$ умовних одиниць площі. Цей показник є майже в тричі меншим, ніж для автомата $U_1(\Gamma_1)$.

Аналіз ефективності запропонованого методу

Для розглянутого прикладу, запропонований метод дозволив зменшити площу кристала, займаного схемою автомата $U_1(\Gamma_1)$ в 2,94 рази. Однак така економія можлива далеко не завжди. Для її досягнення необхідним є спільне виконання декількох умов, розглянутих далі.

По-перше, необхідно, щоб кожен клас ПЕС було представлено одним інтервалом простору кодування. Це дає змогу зменшити кількість рядків ПСТ автомата $U_2(\Gamma_j)$ до показника еквівалентного автомата Мілі. Це можливо не завжди через особливості конкретної ГСА. Наприклад, для $\Pi_A = \{B_1, B_2\}$ з $B_1 = \{a_1\}$ і $B_2 = \{a_2, a_3, a_4\}$ такого оптимального рішення немає.

По-друге, для кодування класів необхідно використовувати мінімальну кількість внутрішніх змінних. В ідеальному варіанті це кількість змінних для еквівалентного автомата Мілі (R_0). Виконання цієї умови дає змогу зменшити площу матриці M_1 на $(R - R_0) \times H_0$ одиниць площі.

По-третьє, кодування станів виконується так, щоб кожна МО $v_n \in V$ була представлена лише одним термом. Це дає змогу виключити матрицю M_4 . Однак такий результат є ідеальним. Насамперед кодування має зменшити кількість переходів. Це зменшує площу матриці M_1 , яка займає найбільшу частину загальної площі.

Якщо N_0 функцій $v_r \in V$ представлено одним термом, то матриця M_4 реалізує лише

$(N_2 - N_0)$ функцій. При цьому матриця M_3 формує N_0 термів для функцій $v_r \in V_0$ і M_{V_0} термів для функцій $v_r \in V/V_0$. Тут $V_0 \subseteq V$ є множиною функцій, ДНФ яких представлено одним термом. Це веде до схеми, показаної на рис. 6, де $|A_{V_0}| = M_{V_0}$ і $V_1 = V/V_0$.

Площі цих матриць визначаються у такий спосіб:

$$S(M_3) = 2R \times (N_0 + M_{V_0});$$

$$S(M_4) = M_{V_0} \times (N_2 - N_0).$$

Для автомата $U_2(\Gamma_1)$ ми мали крайній випадок, коли $N_0 = N_2$, що веде до $M_{V_0} = 0$ і відсутності матриці M_4 .

Наші дослідження, здійснені на базі бібліотеки [17], показали, що запропонований метод дозволяє в середньому на 35% зменшити складність матричної схеми СМПА. При цьому зменшення довжини ПСТ до H_0 досягалося для 38 відсотків прикладів, а зменшення входів матриці M_1 до R_0 — для 16 відсотків. Матриця M_4 була відсутня лише для 4-х відсотків прикладів. При цьому максимальне зменшення площі досягало 67%. Зазначимо, що для 14 відсотків прикладів площа не зменшувалася.

ЛІТЕРАТУРА

1. *Smith M.* Application Specific Integrated Circuits. Boston: Addison-Wesley, 1997. 632 p.
2. *Nabati Z.* Embedded Core Design with FPGAs. New York: McGraw-Hill, 2008. 418 p.
3. *Баранов С. И., Скляр В. А.* Цифровые устройства на программируемых БИС с матричной структурой. М.: Радио и связь, 1986. 272 с.
4. *Ачасова С. М.* Алгоритмы синтеза автоматов на программируемых логических матрицах. М.: Советское радио, 1987. 132 с.
5. *S. Baranov, L. Levin, O. Keren, M. Karpovsky.* Designing fault tolerant FSM by nano-PLA. In: Proceeding of 15th International On-Line Testing Symposium, 2009. Lisbon, pp. 216–220.
6. *H. Naemi, A. De Hon.* A Greedy Algorithm for tolerating Crosspoints in Nano PLA design. In: Proceeding of IEEE International Competence on Field-Programmable Technology, 2004. pp. 49–56.
7. *DeMicheli G.* Synthesis and Optimization of Digital Circuits. New York: McGraw-Hill, 1994. 634 p.
8. *Baranov S.* Logic Synthesis for Control Automata. Dordrecht: Kluwer Academic Publishers, 1994. 312 p.
9. *Соловьев В. В.* Проектирование цифровых схем на основе программируемых логических интегральных схем. М.: Горячая линия — ТЕЛЕКОМ, 2001. 636 с.
10. *А.А. Баркалов, Л.А. Титаренко, Я.Е. Визор, А.В. Матвиенко, Горина В.В.* Уменьшение числа LUT элементов в схеме совмещенного автомата. Управляющие системы и машины. 2016, №3. С. 16–22.
11. *Sklyarov V., Sklyarova I., Barkalov A., Titarenko L.* Synthesis and Optimization of FPGA-based Systems. Berlin: Springer, 2014. 432 p.

Висновки

Запропонований метод засновано на оптимальному кодуванні станів автомата Мура [17]. Це уможливило зменшення площі нано-ПЛМ, що реалізують схему суміщеного автомата в базисі *ASIC*. Оптимізація схеми стає можливою завдяки зменшенню кількості рядків таблиці переходів і змінних, що кодують стани. Таке зменшення засновано на використанні класів псевдоеквівалентних станів автомата Мура [12]. Метод орієнтовано на суміщені автомати, стани яких, як відомо, є станами автомата Мура.

Дослідження бібліотеки [17] показало, що запропонований метод уможливило зменшення площі для 86 відсотків усіх стандартних автоматів. При цьому вигреш зростає за зростання кількості станів автомата.

Є й інші методи оптимізації, засновані на використанні класів ПЕС [13]. До них належать методи перетворення кодів станів на коди класів ПЕС і використання декількох джерел кодів класів ПЕС. У наших подальших дослідженнях ми плануємо використовувати ці методи, адаптовані до особливостей суміщених автоматів і нано-ПЛМ.

12. *Barkalov A., Titarenko L.* Logic Synthesis for FSM–based Control Units. Berlin: Springer, 2009. 233 p.
13. *Barkalov A., Titarenko L., Kolopenczyk M., Mielcarek K., Bazydlo G.* Logic Synthesis for FPGA–based Finite State Machines. Berlin: Springer, 2016. 280 p.
14. *А.А. Баркалов, Л.А. Титаренко, Я.Е. Візор, А.В. Матвієнко.* Синтез совмещенного микропрограммного автомата в базе FPGА. Комп’ютерні засоби, мережі та системи. К.: Ін-т кібернетики ім. В.М.Глушкова НАН України, Київ, 2015. Випуск 14, С. 32–39.
15. *А.А. Баркалов, Л.А. Титаренко, Я.Е. Візор, А.В. Матвієнко.* Реализация схемы совмещенного микропрограммного автомата в базе FPGА. Проблеми інформатизації та управління. Збірник наукових праць. Національний авіаційний університет. К., 2015 Вип. 3(51). С 5–13.
16. *Kubica M., Kania D.* Technology mapping oriented to adaptive logic modules. Bulletin of Polish Academy of Sciences, 2019, V. 67, N 5, pp. 947–956.
17. *Yang S.* Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina. 1991, 43 p.

Надійшла 28.05.2020

REFERENCES

1. *Smith, M., 1997.* Application Specific Integrated Circuits. Boston: Addison-Wesley, 632 p.
2. *Nababi, Z., 2008.* Embedded Core Design with FPGAs. New York: McGraw-Hill, 418 p.
3. *Baranov, S. I., Sklyarov, V. A., 1986.* Digital devices on programmable LSI with matrix structure. М.: Radio and communications, 272 p. (In Russian).
4. *Achasova, S. M. 1987.* Algorithms for the synthesis of automata on programmable logic matrices. М.: Soviet Radio, 132 p. (In Russian).
5. *Baranov, S., Levin, L., Keren, O., Karpovsky M., 2009.* “Designing fault tolerant FSM by nano-PLA”. In: Proceeding of 15th International On-Line Testing Symposium, Lisbon, pp. 216–220.
6. *Naemi, H., Hon., A. De., 2004.* “A Greedy Algorithm for tolerating Crosspoints in Nano PLA design”. In: Proceeding of IEEE International Competence on Field–Programmable Technology, pp. 49–56.
7. *DeMicheli, G., 1994.* Synthesis and Optimization of Digital Circuits. New York: McGraw-Hill, 1994. 634 pp.
8. *Baranov, S., 1994.* Logic Synthesis for Control Automata. Dordrecht: Kluwer Academic Publishers, 312 p.
9. *Soloviev, V.V., 2001.* Designing digital circuits based on programmable logic integrated circuits. М.: Hot line – TELECOM, 636 p. (In Russian).
10. *Barkalov, A.A., Titarenko, L.A. Vizor, Y.E. Matvienko, A.V., Gorina V.V., 2016.* “Synthesis of Combined Finite State Machine with FPGAs”. *Upravlsusie sistemy i masiny*, 3, pp. 16-22. (In Russian).
11. *Sklyarov, V., Skliarova, I., Barkalov, A., Titarenko, L., 2014.* Synthesis and Optimization of FPGA-based Systems. Berlin: Springer, 432 p.
12. *Barkalov, A., Titarenko, L., 2009.* Logic Synthesis for FSM–based Control Units. Berlin: Springer, 233 p.
13. *Barkalov, A., Titarenko, L., Kolopenczyk, M., Mielcarek, K., Bazydlo, G., 2016.* Logic Synthesis for FPGA–based Finite State Machines. Berlin: Springer, 280 p.
14. *Barkalov, A.A., Titarenko, L.A., Vizor, Y.E., Matvienko A.V., 2015.* “Synthesis of a combined finite state machine in the FPGA basis”. *Computer tools, networks and systems*. К.: Institute of Cybernetics. V.M.Glushkova NAS of Ukraine, Kyiv, 14, pp. 32-39. (In Russian).
15. *Barkalov, A.A., Titarenko, L.A., Vizor, Y.E., Matvienko A.V., 2015.* “Implementing circuit of combined finite state machine with FPGAs – Problems of informatization and management”. *Collection of scientific works. National Aviation University. Kyiv*, 3 (51), pp. 5-13. (In Russian).
16. *Kubica, M., Kania, D., 2009.* “Technology mapping oriented to adaptive logic modules”. *Bulletin of Polish Academy of Sciences*, 67 (5), pp. 947 – 956.
17. *Yang, S., 1991.* Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina, 43 pp.

Received 28.05.2020

O.O. Barkalov, Doctor in Techn. Sciences, Professor,
University of Zielona Gora (Poland),
Podgorna str., 50, Zielona Gora, 65246, Poland,
a.barkalov@iie.uz.zgora.pl

L.O. Titarenko, Doctor of Technical Sciences,
Professor of the University of Zelenogorsk (Poland),
Podgorna str., 50, Zielona Gora, 65246, Poland
L.Titarenko@iie.uz.zgora.pl, D_ts@nure.ua

Ya. Ye. Vizor, Ph.D., Senior Researcher, Institute of Cybernetics of NAS of Ukraine,
03187, Kiev, Glushkov Avenue, 40, Ukraine,
yaviz@ukr.net

O.V. Matvienko, Researcher Associate, Institute of Cybernetics of NAS of Ukraine,
03187, Kiev, Glushkov Avenue, 40, Ukraine,
matv@online.ua

SYNTHESIS OF CIRCUIT OF COMBINED AUTOMATION WITH REDUCING AREA OF NANO-PLA

Introduction. To reduce the energy consumed and increase the speed of the circuits in the ASIC basis, it is necessary to reduce the crystal area occupied by the circuit. This also applies to PLA-like structures that have recently been used as part of ASICs. They are called nano-PLA and are most often used to implement control devices.

Purpose. Among the models that are used for the synthesis of control devices, a significant place is occupied by the model of combined microprogram automation (CMPA). However, methods for the synthesis of CMPA schemes in the nano-PLA basis are currently lacking. Due to the importance of this problem, we propose a method for the synthesis of CMPA on nano-PLA basis.

Methods. For the specification of CMPA, the language of graph diagrams of the algorithm is used. The method proposed in the article is based on the optimal coding of states of the Moore FSM, taking into account the existence of classes of pseudoequivalent states (PES). In this case, a part of the circuit that implements the functions of the Moore automation is highlighted.

Results. The method allows to reduce the nano-PLS area in comparison with the trivial two-level scheme. Studies conducted on the basis of the standard library showed that the proposed method allows on average 35% decrease in the complexity of the matrix CMPA scheme. At the same time, a decrease in the length of the direct structural table was achieved for 38% of the examples, and a decrease in the number of matrix entries for 16%. Moreover, the maximum decrease in area reached 67%. For 14% of the examples, the area did not decrease.

Resumes. A method allows to reduce the area for 86% of all standard automations. In this case, the gain grows as the number of states of the automation increases.

There are other optimization methods based on the use of PES classes. These include methods for converting status codes into PES class codes and using several sources of PES class codes. In our further studies, it is planned to use these methods adapted to the features of combined automation and nano-PLA.

Keywords: *combined automation, synthesis, nano-PLA, ASIC, state assignment.*