

# ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, КІБЕРБЕЗПЕКА

DOI: 10.18372/2310-5461.44.14314

УДК 004.274

**О. О. Баркалов**, д-р техн. наук, проф.  
Університет Зеленогурський (Польща)  
orcid.org/0000-0002-4941-3979  
e-mail: A.Barkalov@iie.uz.zgora.pl

**Л. О. Титаренко**, д-р техн. наук, проф.  
Університет Зеленогурський (Польща)  
orcid.org/0000-0001-0558-3322  
e-mail: A.Barkalov@iie.uz.zgora.pl

**Я. Є. Візор**, канд. техн. наук, старш. наук. співроб.  
Інститут кібернетики НАН України  
orcid.org/0000-0002-5580-2604  
e-mail: yaviz@ukr.net

**О. В. Матвієнко**, наук. співроб.  
Інститут кібернетики НАН України  
orcid.org/0000-0003-1838-1422  
e-mail: matv@online.ua

## КОДУВАННЯ НАБОРІВ МІКРООПЕРАЦІЙ У ТРЬОХРІВНЕВОМУ СУМІЩЕНОМУ АВТОМАТІ

### Вступ

Одним з головних блоків цифрових систем є пристрій керування (ПУ) [1; 2]. Для підвищення якості цифрових систем (зменшення площі кристала, зниження споживання енергії, підвищення швидкодії) необхідно зменшувати площу кристала надвеликих інтегральних схем (НВІС), займану схемою ПУ [3; 4]. Методи вирішення цієї задачі значною мірою залежать від моделі, що представляє ПУ, і характеристик елементного базису, у якому реалізується схема ПУ [5].

### Аналіз останніх досліджень і публікацій

Сьогодні моделі мікропрограмних автоматів (МПА) широко використовуються для завдання поведінки ПУ [6; 7]. До однієї з таких моделей відноситься суміщений МПА (СМПА). Його характерною рисою є наявність двох типів вихідних сигналів. Вихідні сигнали автомата Мілі існують під час переходів між станами автомата. Вихідні сигнали автомата Мура визначаються тільки станами автомата і тривають майже цілий такт. У силу універсальності цієї моделі ми використовуємо її в даній статті.

Найбільш популярним базисом, використовуваним для реалізації цифрових систем, є мікросхеми FPGA (field-programmable logic arrays) [8].

Основними елементами FPGA, які використовуються для реалізації МПА, є елементи табличного типу LUT (look-up table), програмовані тригери і програмовані міжз'єднання [5].

У цій статті пропонуємо метод зменшення числа елементів LUT і кількості їх рівнів у схемі СМПА. Для завдання поведінки ПУ використовуємо мову граф-схем алгоритму [1].

### Постановка завдання

Для реалізації схеми СМПА необхідно отримати три системи булевих функцій [6; 7]:

$$Y_{M1} = Y_{M1}(T, X); \quad (1)$$

$$Y_{M2} = Y_{M2}(T); \quad (2)$$

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

У виразах (1)–(3) можуть використовуватися такі позначення:  $Y_{M1} \subseteq Y$  — множина мікрооперацій автомата Мілі,  $|Y_{M1}| = N_1$ ;  $Y_{M2} \subseteq Y$  — множина мікрооперацій автомата Мура,  $|Y_{M2}| = N_2$ ;  $\Phi$  — множина функцій збудження пам'яті МПА;  $T$  — множина внутрішніх змінних, використовуваних для кодування станів  $a_m \in A$ , де  $A = \{a_1, \dots, a_M\}$  — множина станів.

При цьому  $Y_{M1} \cup Y_{M2} = Y$ , де  $Y$  — множина мікрооперацій СМПА ( $Y = \{y_1, \dots, y_N\}$ ). Крім того, виконується така умова:  $Y_{M1} \cap Y_{M2} = \emptyset$ .

Для синтезу схеми СМПА необхідно побудувати пряму структурну таблицю (ПСТ). Цьому етапу передують кодування станів, коли кожному стану  $a_m \in A$  ставиться у відповідність двійковий код  $K(a_m)$  розрядності  $R$ :

$$R = \lceil \log_2 M \rceil. \quad (4)$$

Множини  $\Phi$  і  $T$  мають однакову кількість елементів, при цьому  $\Phi = \{D_1, \dots, D_R\}$  і  $T = \{T_1, \dots, T_R\}$ . Стани  $a_m \in A$  кодуються кодами  $K(a_m)$ , які зберігаються у внутрішній пам'яті СМПА. Ця пам'ять представлена регістром  $RG$ . Як правило,  $RG$  має інформаційні входи типу  $D$  [3].

Щоб знайти системи (1)–(3), побудуємо ПСТ поєднаного МПА. Ця таблиця містить такі стовпці:  $a_m$  — початковий стан СМПА;  $K(a_m)$  — код стану  $a_m \in A$ ;  $a_s$  — стан переходу;  $K(a_s)$  — код стану  $a_s \in A$ ;  $X_h$  — вхідний сигнал, що визначає перехід з  $a_m$  в  $a_s$  і рівний кон'юнкції деяких змінних  $x_i \in X$  (або їх заперечень);  $Y_{M1h}$  — набір мікрооперацій (НМО), що формуються на переході  $\langle a_m, a_s \rangle$ ;  $\Phi_h$  — набір функцій  $D_r \in \Phi$ , рівних одиниці для перемикання  $RG$  з  $K(a_m)$  в  $K(a_s)$ ;  $h$  — номер переходу ( $h = \overline{1, H}$ ). У стовпці  $a_m$  записується НМО, що формується в стані  $a_m \in A$ . Ця таблиця формується на основі вихідної ГСА  $G$ , зазначеної станами МПА Мура [6, 7].

Будемо розглядати проблему реалізації схеми СМПА в базисі елементів  $LUT$ . Нехай  $LUTer$  означає блок, що складається з елементів  $LUT$ , тригерів і прогамованих міжз'єднань, показаний на рис. 1.

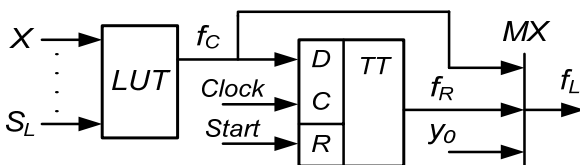


Рис. 1. Логічний елемент блоку  $LUTer$

Елемент  $LUT$  реалізує довільну логічну функцію  $f_C$ , яка залежить від не більше, ніж  $S_L$  аргументів. Вихід  $f_C$  подається на вхід  $D$  двотактного тригера. Імпульс  $Start$  використовується для операції  $f_R := 0$ . Імпульс  $Clock$  ініціює операцію  $f_R := f_C$ . Вихід логічного елемента може бути або комбінаційною ( $f_L = f_C$ ), або реєстровою ( $f_L = f_R$ ). Для вибору типу виходу використовується мультиплексор  $MX$  і внутрішній сигнал

$$y_0: f_L = y_0 f_C \vee \overline{y_0} f_R.$$

Отже, елемент  $LUT$  має тільки один вихід. Тому системі (1) можна ставити у відповідність блок  $LUTer 1$ , системі (2) — блок  $LUTer 2$  і системі (3) — блок  $LUTer T$ . Це призводить до СМПА  $U_1$  (рис. 2). В  $U_1$  регістр  $RG$  розподілений

між тригерами блоку  $LUTer T$ . Тому блок  $LUTer T$  має входи синхронізації ( $Clock$ ) і обнуління ( $Start$ ).

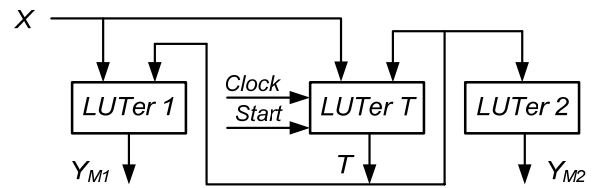


Рис. 2. Структурна схема СМПА  $U_1$

При синтезі СМПА в базисі  $FPGA$  виникає серйозна проблема, пов'язана з обмеженням числа  $S_L$  входів  $LUT$  [9]. Як правило, величина  $S_L$  не перевищує 6 [10; 11]. Аналіз бібліотеки [12] показує, що функції (1) і (3) можуть залежати від  $L + R \approx 30$  аргументів. При цьому виконується умова

$$S_L \ll L + R. \quad (5)$$

Нехай  $L(f_i)$  — число аргументів у функції  $f_i \in \Phi \cup Y$ . Якщо виконується умова

$$L(f_i) \leq S_L, \quad (6)$$

то для реалізації схеми, що відповідає функції  $f_i$ , досить одного елемента  $LUT$ .

Якщо умова (6) виконується для всіх функцій  $\Phi \cup Y$ , то схема  $U_1$  містить  $N + R$  елементів  $LUT$  і тільки один логічний рівень. Така схема має мінімальне число міжз'єднань, споживану потужність і час затримки (максимальна швидкодія). Однак в силу справедливості (5), умова (6) може виконуватися тільки обмеженого числа функцій  $f_i \in \Phi \cup Y$ .

Якщо умова (6) порушується, то відповідні функції представляються у вигляді композиції підфункцій, для кожної з яких виконується умова (6). Для цієї мети використовуються методи функціональної декомпозиції [13; 14]. Їх застосування призводить до багаторівневих схем та мають набагато гірші характеристики порівняно з однорівневими схемами МПА [15].

Якщо умова (6) порушується, то необхідно оптимізувати схему СМПА. Для цього можуть бути використані такі методи [5]:

1. *Оптимальне кодування станів.* Під оптимальним розуміється кодування, що зменшує число аргументів у функціях (1)–(3). Існують сотні методів кодування. Одним з кращих вважається метод *JEDI* [16]. Однак для складних автоматів ( $M > 50$ ) жоден з методів не гарантує виконання умови (6) [17].

2. *Використання вбудованих блоків пам'яті.* Сучасні  $FPGA$  мають в своєму складі блоки пам'яті *EMB* (embedded memory blocks). Один блок *EMB* може замінити сотні елементів  $LUT$

[18]. Однак ці блоки широко використовуються для реалізації операційних блоків цифрових систем [4]. Тому цілком імовірна ситуація, коли для реалізації схеми ПУ залишаються тільки елементи *LUT*.

3. *Структурна декомпозиція*. Цей підхід пов'язаний зі збільшенням кількості структурних рівнів схеми МПА [17]. Кожен рівень характеризується власними вихідними змінними і вихідними функціями. При цьому вихідні функції рівня *i* служать входними змінними рівня *i + 1*. Відомі такі методи структурної декомпозиції, як [5]: кодування логічних умов, кодування наборів мікрооперацій, перетворення об'єктів.

За відсутності блоків *EMB* автомат можна подати у вигляді тривірневої схеми [20; 21], заснованої на подвійному кодуванні станів. У цій роботі ми пропонуємо: 1) адаптувати цей підхід до особливостей моделі СМПА і 2) зменшити кількість елементів *LUT*, використовуючи кодування наборів мікрооперацій (НМО) автомата Мілі.

**Основна мета запропонованого методу**

Нехай задана ГСА  $\Gamma$ , що представляє поведінку СМПА і зазначена станами  $a_m \in A$ . Знайдемо розбиття  $\Pi_A = \{A^1, \dots, A^k\}$  множини  $A$  на класи, для кожного з яких виконується умова:

$$R_k + L_k \leq S \quad (k \in \{1, \dots, K\}). \quad (7)$$

У варіанті (7) символ  $R_k$  означає число змінних, необхідних для кодування станів  $a_m \in A^k$ ,  $L_k$  — число входних змінних  $x_i \in X$ , що визначають переходи з цих станів і утворюють множину  $X^k \subseteq X$ .

Параметр  $R_k$  визначається формулою:

$$R_k = \lceil \log_2(|A^k|) \rceil \quad (k \in \{1, \dots, K\}). \quad (8)$$

Одиниця в формулі (8) додається для коду, відповідного відношення  $a_m \notin A$ . Використовуємо для кодування станів  $a_m \in A^k$  змінні з множини  $\tau^k \subseteq \tau$ , де

$$|\tau| = R_A = R_1 + R_2 + \dots + R_K. \quad (9)$$

Нехай в ГСА  $\Gamma \in Q$  наборів мікрооперацій  $Y_q \subseteq Y_{M1}$ . Закодуємо ці НМО двійковими кодами  $K(Y_q)$ , що мають  $R_Q$  розрядів:

$$R_Q = \lceil \log_2 Q \rceil. \quad (10)$$

Використовуємо для кодування НМО змінні  $z_r \in Z$ , де  $|Z| = R_Q$ .

Кожен клас  $A^k \in \Pi_A$  визначає множини  $X^k, Z^k$  і  $\Phi^k$ . Множина  $Z^k \subseteq Z$  містить змінні  $z_r \in Z$ , що дорівнюють 1 в кодах  $K(Y_q)$  наборів, які формуються при переходах зі станів  $a_m \in A$ . Множина  $\Phi^k \subseteq \Phi$  включає змінні  $D_r \in \Phi$ , що дорівнюють одиниці при переходах зі станів  $a_m \in A^k$ .

Закодуємо стану  $a_m \in A$  кодами  $K(a_m)$  розрядності  $R$ . Тепер кожний стан має два коди. Код  $K(a_m)$  визначає  $a_m \in A$ , а код  $C(a_m) — a_m \in A^k$ . Грунтуючись на такому підході, ми пропонуємо СМПА  $U_2$  (рис. 3).

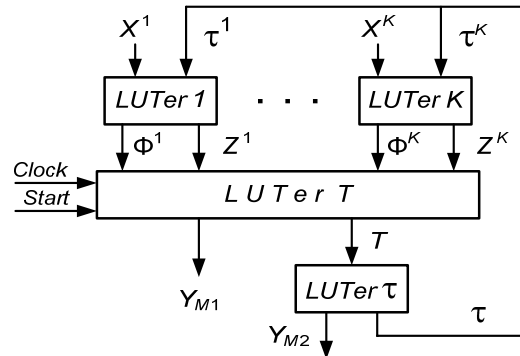


Рис. 3. Структурна схема СМПА  $U_2$

В СМПА  $U_2$  блок *LUTer k* формує функції:

$$\Phi^k = \Phi^k(\tau^k, x^k); \quad (10)$$

$$Z^k = Z^k(\tau^k, x^k). \quad (11)$$

Блок *LUTer T* формує системи функцій мікрооперацій  $y_n \in Y_{M1}$  і внутрішні змінні  $T_r \in T$ . Сигнали *Start* і *Clock* керують розподіленням регістром *RG*. Для формування функцій використовуються такі рівняння:

$$z_r = z_r^1 \vee z_r^2 \vee \dots \vee z_r^K \quad (r = \overline{1, R_Q}); \quad (12)$$

$$T_r = T_r^1 \vee T_r^2 \vee \dots \vee T_r^K \quad (r = \overline{1, R}). \quad (13)$$

Функції  $T_r \in T$  є виходами тригерів, розподілених між елементами *LUT* блоку *LUTer T*. На ці тригера надходять сигнали синхронізації *Clock* і обнулення *Start*.

Блок *LUTer Y* реалізує систему функцій:

$$Y_{M1} = Y_{M1}(Z), \quad (14)$$

а блок *LUTer tau* — систему (2) і функції:

$$\tau_r = \tau(T). \quad (15)$$

Розбиття  $\Pi_A$  забезпечує виконання умови (7). Тому кожен блок *LUTer k* має  $NL_k$  елементи *LUT*:

$$NL_k = |\Phi^k| + |Z^k|. \quad (16)$$

Отже, для зменшення кількості елементів *LUT* в схемі  $U_2$  необхідно мінімізувати кількість функцій  $D_r \in \Phi^k$  і  $z_r \in Z^k$ . Однак це є вторинною проблемою, оскільки стани  $a_m \in A$  кодуються так, щоб оптимізувати схему блоку *LUTer tau*, а НМО  $Y_q \subseteq Y$  — блоку *LUTer Y*. Тому будемо шукати розбиття  $\Pi_A$ , що має мінімальну кількість блоків *K*.

Такий підхід запропонований у праці [21]. Ми використовуємо його для СМПА  $U_2$  і детально не розглядаємо. У цій статті пропонується метод синтезу СМПА  $U_2$ , що включає такі етапи:

1. Відмітку станів  $a_m \in A$  на початковій ГСА  $\Gamma$ .
2. Кодування станів  $a_m \in A$ , що оптимізує систему (2).

3. Знаходження розбиття  $\Pi_A$  з мінімальним числом блоків  $K$ .
4. Кодування станів кодами  $C(a_m)$ .
5. Кодування НМО  $Y_q \subseteq Y$ , оптимізує систему (14).
6. Формування систем функцій (10)–(11).
7. Формування систем функцій (12)–(13).
8. Формування систем функцій (2), (14) і (15).
9. Реалізація схеми СМПА в заданому базисі.

**Приклад синтезу автомата  $U_2$**

Розглянемо приклад синтезу СМПА  $U_2 (\Gamma_1)$ , де вираз  $U_i (\Gamma_j)$  означає, що автомат  $U_i$  синтезується по ГСА  $\Gamma_j$ . Відмічена станами автомата Мура ГСА  $\Gamma_1$ , приведена на рис. 4. Для реалізації схеми автомата використовуємо елементи  $LUT$ , що мають  $S = 4$  входи.

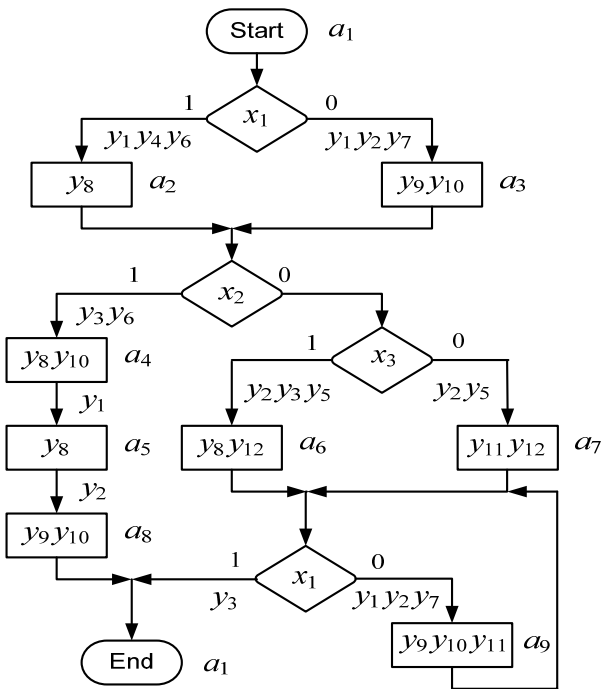


Рис. 4. Відмічена ГСА  $\Gamma_1$

Аналіз ГСА  $\Gamma_1$  дозволяє отримати такі множини:  $A = \{a_1, \dots, a_9\}$ ,  $X = \{x_1, x_2, x_3\}$ ,  $Y_{M1} = \{y_1, \dots, y_7\}$ ,  $Y_{M2} = \{y_8, \dots, y_{12}\}$ . Це дає такі параметри:  $M = 9$ ,  $L = 3$ ,  $N_1 = 7$ ,  $N_2 = 5$ ,  $N = 12$ . Відзначимо, що мікрооперації  $y_n \in Y_{M1}$  записані біля дуг ГСА  $\Gamma_1$ , а  $y_n \in Y_{M2}$  — в її операторних вершинах. Із формули (4) випливає, що  $R = 4$ ,  $|T| = |\Phi| = 4$ .

Мікрооперації  $y_n \in Y_{M1}$  створюють  $Q = 9$  наборів:  $Y_1 = \emptyset$ ,  $Y_2 = \{y_1, y_4, y_6\}$ ,  $Y_3 = \{y_1, y_2, y_7\}$ ,  $Y_4 = \{y_3, y_6\}$ ,  $Y_5 = \{y_1\}$ ,  $Y_6 = \{y_2, y_3, y_5\}$ ,  $Y_7 = \{y_2, y_5\}$ ,  $Y_8 = \{y_2\}$ ,  $Y_9 = \{y_3\}$ . Із виразу (10) випливає, що  $R_Q = 4$ ,  $|Z| = 4$ .

Для зменшення числа літералів в системі (2) використовуємо алгоритм кодування станів із праці [19]. Коди  $K(a_m)$  наведені на рис. 5.

Такий підхід дозволяє зменшити кількість елементів  $LUT$  і їх міжз'єднань у схемі блоку  $LUTer \tau$ .

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

Рис. 5. Коди станів автомата  $U_2(\Gamma_1)$

Максимальне число елементів в блоці  $A^k \in \Pi_A$  залежить від параметрів  $S_L$  і  $L_k$ . Нехай це число позначено як  $NS_k$ . Цей параметр визначається так:

$$NS_k = 2^{S-L_k} - 1 \quad (k = \overline{1, K}). \quad (17)$$

При  $S = 4$  маємо такі пари  $\langle L_k, NS_k \rangle$ :  $\langle 0, 15 \rangle$ ,  $\langle 1, 7 \rangle$ ,  $\langle 2, 3 \rangle$ ,  $\langle 3, 1 \rangle$ .

Використання методу [21] призводить до розбиття  $\Pi_A$  з  $K = 2$ , де  $A^1 = \{a_1, a_4, \dots, a_9\}$  та  $A^2 = \{a_2, a_3\}$ . Аналіз ГСА  $\Gamma_1$  показує, що  $X^1 = \{x_1\}$  і  $X^2 = \{x_2, x_3\}$ .

Із виразів (8)–(9) випливає, що  $R_1 = 3$ ,  $R_2 = 2$ ,  $R_A = 5$ .

Це дає множини  $\tau = \{\tau_1, \dots, \tau_5\}$ ,  $\tau^1 = \{\tau_1, \tau_2, \tau_3\}$ ,  $\tau^2 = \{\tau_4, \tau_5\}$ . Закодуємо стани  $a_m \in A^k$  таким чином (табл. 1).

Таблиця 1

Коди станів $a_m \in A^k$								
$a_m$	$\tau_1 \tau_2 \tau_3$	$\tau_4 \tau_5$	$a_m$	$\tau_1 \tau_2 \tau_3$	$\tau_4 \tau_5$	$a_m$	$\tau_1 \tau_2 \tau_3$	$\tau_4 \tau_5$
$a_1$	0 0 1	0 0	$a_4$	0 1 0	0 0	$a_7$	1 0 1	0 0
$a_2$	0 0 0	0 1	$a_5$	0 1 1	0 0	$a_8$	1 1 0	0 0
$a_3$	0 0 0	1 0	$a_6$	1 0 0	0 0	$a_9$	1 1 1	0 0

Використаємо алгоритм [19] для кодування НМО  $Y_q \subseteq Y$ . Коди  $K(Y_q)$  показано на карті Карно (рис. 6).

		$Z_1 Z_2$			
		00	01	11	10
$Z_3 Z_4$	00	$Y_1$	*	*	$Y_8$
	01	$Y_9$	$Y_4$	*	$Y_6$
	11	*	*	*	$Y_7$
	10	$Y_5$	$Y_2$	*	$Y_3$

Рис. 6. Коди наборів мікрооперацій для  $U_2(\Gamma_1)$

Для формування систем (10)–(11) необхідно побудувати таблиці блоків  $LUTer k$ .

Ці таблиці являють собою перетворені ПСТ і мають стовпці  $a_m, C(a_m), a_s, K(a_s), X_h^k, Z_h^k, \Phi_h^k, h$ .

У стовпці  $Z_h^k$  записуються змінні  $z_r \in Z$ , які дорівнюють 1 в коді  $K(Y_q)$  набору, який

формується на  $h$ -му переході. Таблиці формуються по ГСА. У нашому прикладі це табл. 2 і 3.

Таблиця 2

Таблиця блоку LUTer1

$a_m$	$C(a_m)$	$a_s$	$K(a_s)$	$X_h^1$	$Z_h^1$	$\Phi_h^1$	$h$
$a_1$	001	$a_2$	0100	$x_1$	$z_2 z_3$	$D_2$	1
		$a_3$	1000	$\overline{x_1}$	$z_1 z_3$	$D_1$	2
$a_4$	010	$a_5$	0111	1	$z_3$	$D_2 D_3 D_4$	3
$a_5$	011	$a_8$	1001	1	$z_1$	$D_1 D_4$	4
$a_6$	100	$a_1$	0000	$x_1$	$z_4$	—	5
		$a_9$	1011	$\overline{x_1}$	$z_1 z_3$	$D_1 D_3 D_4$	6
$a_7$	101	$a_1$	0000	$x_1$	$z_4$	—	7
		$a_9$	1011	$\overline{x_1}$	$z_1 z_3$	$D_1 D_3 D_4$	8
$a_8$	110	$a_1$	0000	1	—	—	9
$a_9$	111	$a_1$	0000	$x_1$	$z_4$	—	10
		$a_9$	1011	$\overline{x_1}$	$z_1 z_3$	$D_1 D_3 D_4$	11

Таблиця 3

Таблиця блоку LUTer2

$a_m$	$C(a_m)$	$a_s$	$K(a_s)$	$X_h^2$	$Z_h^2$	$\Phi_h^2$	$h$
$a_2$	01	$a_4$	0101	$x_2$	$z_2 z_4$	$D_2 D_4$	1
		$a_6$	0110	$\overline{x_2 x_3}$	$z_1 z_4$	$D_2 D_3$	2
		$a_7$	0010	$\overline{x_2 x_3}$	$z_1 z_3 z_4$	$D_3$	3
	10	$a_4$	0101	$x_2$	$z_2 z_4$	$D_2 D_4$	4
		$a_6$	0110	$\overline{x_2 x_3}$	$z_1 z_4$	$D_2 D_3$	5
		$a_7$	0010	$\overline{x_2 x_3}$	$z_1 z_3 z_4$	$D_3$	6

Функції (10)–(11) формуються тривіальним чином. Наприклад, із табл. 2 маємо (з урахуванням мінімізації):

$$D_4^1 = \overline{\tau_1} \tau_2 \vee \tau_1 \overline{\tau_2 x_1} \vee \tau_1 \tau_3 \overline{x_1};$$

$$z_3^1 = \overline{\tau_1} \tau_2 \tau_3 \vee \overline{\tau_1} \tau_2 \overline{\tau_3} \vee \tau_1 \overline{\tau_2 x_1} \vee \tau_1 \tau_3 \overline{x_1}. \quad (18)$$

Із табл. 3 маємо, наприклад, такі функції:

$$D_4^2 = \tau_2 x_2 \vee \tau_4 x_2;$$

$$z_3^2 = \tau_5 x_2 x_3 \vee \tau_4 x_2 x_3. \quad (19)$$

Функції (10)–(11) формуються так. Наприклад, функція  $D_4^1 \vee D_4^2$  визначає функцію  $T_4$  (вихід відповідного тригера блоку LUTer T).

Функція  $z_3$  утворюється, як диз'юнкція відповідних функцій із систем (18) і (19).

Система (2) формується на основі вмісту операторних вершин ГСА  $\Gamma_1$  і кодів з рис. 5. Це дає такий результат:

$$y_8 = A_2 \vee A_4 \vee A_5 \vee A_6 = T_2;$$

$$y_9 = A_3 \vee A_8 \vee A_9 = T_1;$$

$$y_{10} = A_3 \vee A_4 \vee A_8 \vee A_9 = T_1 \vee \overline{T_3} T_4 \quad (20)$$

$$y_{11} = A_7 \vee A_9 = \overline{T_2} T_3; \quad y_{12} = A_6 \vee A_7 = T_3 \overline{T_4}.$$

У функціях (18) символ  $A_m$  означає кон'юнкцію змінних  $T_r \in T$ , яка відповідає коду  $K(a_m)$  стану  $a_m \in A$ .

Система (14) формується на основі вмісту НМО  $Y_q \subseteq Y$  і кодів з рис. 6. У нашому прикладі маємо таку систему:

$$\begin{aligned} y_1 &= Y_2 \vee Y_3 \vee Y_5 = z_3 \overline{z_4}; \\ y_4 &= Y_2 = z_2 z_3; \\ y_2 &= Y_3 \vee Y_6 \vee Y_7 = z_1; \\ y_5 &= Y_6 \vee Y_7 = z_1 z_4; \\ y_3 &= Y_4 \vee Y_6 \vee Y_9 = \overline{z_3 z_4}; \\ y_6 &= Y_2 \vee Y_4 = z_2; \\ y_7 &= Y_3 = z_1 z_3 \overline{z_4}. \end{aligned} \quad (21)$$

Система (15) формується відповідно до табл. 1, при використанні кодів з рис. 5. Якщо  $\tau_r = 1$  для стану  $a_m$ , то кон'юнкція  $a_m$  входить в рівняння для  $\tau_r$ . У нашому випадку маємо таку систему:

$$\begin{aligned} \tau_1 &= A_6 \vee A_7 \vee A_8 \vee A_9 = T_1 T_4 \vee T_3 \overline{T_4}; \\ \tau_3 &= A_5 \vee A_7 \vee A_9 = T_3 T_4 \vee T_3 \overline{T_2}; \\ \tau_2 &= A_4 \vee A_5 \vee A_8 \vee A_9 = T_4; \\ \tau_4 &= A_2 = T_2 \overline{T_3 T_4}; \\ \tau_5 &= A_3 = T_1 \overline{T_4}. \end{aligned} \quad (22)$$

Останній етап запропонованого методу зводиться до застосування стандартних пакетів [10; 11].

### Висновки

Запропонований в роботі метод дає найкращі результати при виконанні таких умов:

$$K \leq S_L; \quad (23)$$

$$R \leq S_L; \quad (24)$$

$$R_Q \leq S_L. \quad (25)$$

При виконанні умови (23) блок *LUTer T* має один рівень логіки. Для реалізації його схеми потрібно не більше  $R + R_Q$  елементів *LUT*. При виконанні умови (24) блок *LUTer τ* має не більше  $N_2 + R_A$  елементів *LUT* і один рівень логіки. При виконанні умови (25) блок *LUTer Y* має не більше  $N_1$  елементів *LUT* і один рівень логіки.

Застосування моделі  $U_2$  призводить до трирівневих схем з регулярними міжз'єднаннями. Наприклад, змінні  $z_r \in Z$  пов'язані тільки з блоком *LUTer Y*, а змінні  $T_r \in T$  — з блоком *LUTer τ*.

Важливим є і той факт, що сигнали *Clock* і *Start* пов'язані тільки з одним блоком. Це дозволяє уникнути так званого перекошу синхронізації [22] без додаткових коштів.

Оптимізація системи (2) дозволяє зменшити число міжз'єднань в схемі блоку *LUTer τ*. Максимальне число зв'язків визначається добутком  $N_2 \times R$ , що дорівнює 20 в розглянутому прикладі. Однак аналіз системи (20) показує, що для реалізації МО  $y_n \in Y_{M_2}$  потрібно тільки 7 міжз'єднань.

Аналогічно, для реалізації системи (14) досить 11 міжз'єднань замість  $N_1 \times R_Q = 28$ .

При виконанні (24)–(25) можливе таке кодування станів і НМО, що кількість елементів *LUT*, що реалізують МО  $y_n \in Y$ , буде менше  $N$ . Так в нашому прикладі  $N = 12$ , а для реалізації мікрооперацій потрібно тільки 8 елементів *LUT*. Це пояснюється тим фактом, що не потрібно *LUT* елемент при  $y_n = T_r$  або  $y_n = z_r$ . Відповідні мікрооперації формуються на виходах блоку *LUTer T*.

Аналіз бібліотеки [11] показав, що умови (20)–(21) мають місце для 78 % усіх тестових прикладів. Дослідження проводили для мікросхем сімейства Virtex-6 ( $S = 6$ ). При цьому схеми СМПА  $U_2$  відрізнялися великою швидкістю, ніж їх аналоги, що мають структуру  $U_1$ . Для решти 22 % тестових прикладів виграш був значно меншим, оскільки блоки *LUTer T* і *LUTer τ* реалізовувалися у вигляді багаторівневих схем.

Подальші напрямки наших досліджень пов'язані з: 1) заміною деяких *LUT* блоками *EMB* і 2) використанням методів кодування логічних умов для зменшення параметра  $K$ .

### ЛІТЕРАТУРА

1. Grout I. Digital Systems Design with FPGAs and CPLDs. Amsterdam: Elsevier, 2008. 784 pp.
2. 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.
3. Грушницький Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем с использованием микросхем программируемой логики. СПб: БХВ. Петербург, 2002. 608 с.
4. Skliarova I., Sklyarov V., Sudnitson A. Design of FPGA-based circuits using Hierarchical Finite State Machines. Tallinn: TUT Press, 2012. 240 pp.
5. Jozwiak L., Chojnski A. Effective and efficient FPGA synthesis through general functional decomposition. *Journal of System Architecture*. 2003. №4. Pp. 247–265.
6. Баркалов А. А., Титаренко Л. А., Визор Я.Е., Матвиенко А.В. Уменьшение аппаратных затрат в совмещенных автоматах. *Управляющие системы и машины*. 2017. №4. С. 43–50.
7. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Реализация схемы совмещенного автомата в базисе FPGA. Комп'ютерні засоби, мережі та системи. К.: Ін-т кібернетики імені В. М. Глушкова НАН України, 2016. С. 10–19.
8. Varanov S. Logic Synthesis for Control Automata. Dordrecht: Kluwer Academic Publishers, 1994. 312 pp.
9. Intel. FPGAs and Programmable Devices. URL: [www.altera.com](http://www.altera.com) (date access 15.09.2019).

10. **Xilinx**. Adaptable & Real-Time AI Inference Acceleration. URL: [www.xilinx.com](http://www.xilinx.com) (date access 15.09.2019).
11. **Yang S.** Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina. 1991. 43 pp.
12. **Kubica M., Kania D., Kulisz J.** A technology mapping of FSMs based on a graph of excitations and outputs. *IEEE Access*. 2018. №6. Pp. 16123–16131.
13. **Rawski M., Tomaszewicz P., Borowski G., Łuba T.** Logic Synthesis Method of Digital Circuits Designed for Implementation with Embedded Memory Blocks on FPGAs. *Design of Digital Systems and Devices. LNEE 70*. Berlin: Springer, 2011. Pp. 121–144.
14. **Barkalov A., Titarenko L.** Logic Synthesis for FSM-based Control Units. Berlin: Springer, 2009. 233 pp.
15. **Barkalov A., Titarenko L., Mazurkiewicz M.** Foundations of embedded systems. Berlin: Springer, 2019. 196 pp.
16. **Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвієнко А.В.** Реализация схемы совмещенного микропрограммного автомата в базе FPGAs. *Проблеми інформатизації та управління*. 2015. Вип. 3(51). С. 5–13.
17. **Соловьев В.В.** Проектирование цифровых схем на основе программируемых логических интегральных схем. М. : Горячая линия ТЕЛЕКОМ, 2001. 636 с.
18. **Jozwiak L.** Using FPGAs in Cyber-Physical Synthesis. *Journal of System Architecture*. 2013. №2. Pp. 124–1365.
19. **Opara A., Kubica M., Kania D.** Method of improving time efficiency of decomposition dedicated at FPGA structures and using BDD in process of cyber-physical synthesis. *IEEE Access*. 2019. №1. Pp. 18101–18113.
20. **Barkalov A., Titarenko L., Mielcetek K.** Twofold state assignment for FPGA - based Mealy FSMs. *Proceedings of International Conference MOCAS-18* (Thessaloniki, Greece. – New York). IEEE Explore. 2018. Pp. 1–4.
21. **Barkalov A., Titarenko L., Mielcetek K.** Hardware Reduction for LUT-based Mealy FSMs. *International Journal of Applied Mathematics and Computer Science*. 2018. V. 28. № 3. Pp. 595–607. doi.org/10.2478/amcs-2018-0046.
22. **Sklyarov V., Skliarova I., Barkalov A., Titarenko L.** Synthesis and Optimization of FPGA-based Systems. Berlin: Springer, 2014. 432 pp.

**Баркалов О. О., Титаренко Л. О., Визор Я. Є., Матвієнко О. В.**  
**КОДУВАННЯ НАБОРІВ МІКРООПЕРАЦІЙ У ТРЬОХРІВНЕВОМУ СУМІЩЕНОМУ АВТОМАТІ**

В даний час моделі мікропрограмних автоматів (МПА) широко використовуються для завдання поведінки пристроїв керування (ПК). До однієї з таких моделей відноситься суміщений МПА (СМПА). Його характерною рисою є наявність двох типів вихідних сигналів. Вихідні сигнали автомата Милі існують при переходах між станами автомата. Вихідні сигнали автомата Мура визначаються тільки станами автомата і тривають майже цілий такт. У силу універсальності цієї моделі вона стала базовою для досліджень наведених в даній статті.

Найбільш популярним базисом, який використовується для реалізації цифрових систем, є мікросхеми FPGA (field-programmable logic arrays). Основними елементами FPGA, які використовуються для реалізації МПА, є елементи табличного типу LUT (look-up table), програмовані тригери і програмовані міжз'єднання.

В роботі запропоновано метод зменшення апаратних витрат у схемі суміщеного автомата, що реалізується в спільному базисі елементів LUT і блоків пам'яті ЕМВ. Метод заснований на заміні логічних умов і розбитті множини логічних станів на класи. Кожен клас відповідає окремому блоку схеми. Такий підхід призводить до схем з регулярною структурою і трьома логічними рівнями. Це спрощує завдання розміщення і трасування при реалізації схеми суміщеного мікропрограмного автомата (СМПА). Позитивною рисою запропонованої моделі є той факт, що сигнали синхронізації і запуску пов'язані тільки з одним блоком схеми. Це дозволяє уникнути проблем, пов'язаних з так званим перекосом синхронізації.

Аналіз спеціальної бібліотеки показав, що запропонований метод доцільно використовувати для 78% тестових прикладів. Дослідження проводилися для мікросхем сімейства Virtex-6 ( $S = 6$ ). При цьому розроблені моделі СМПА відрізнялися великою швидкодією, ніж їх аналоги, що мають попередню структуру. Для решти 22% тестових прикладів виграш був значно менше, так як спеціалізовані блоки LUT реалізовувалися у вигляді багаторівневих схем.

Подальший напрям досліджень пов'язаний з: 1) заміною деяких LUT блоками ЕМВ і 2) використанням методів кодування логічних умов для зменшення параметра  $K$ .

**Ключові слова:** суміщений мікропрограмний автомат; синтез; FPGA; LUT; розбиття; набори мікрооперацій.

**Barkalov A. A., Titarenko L. A., Vizer Y. E., Matvienko A. V.**  
**ENCODING OF COLLECTIONS OF MICROOPERATIONS FOR THREE-LEVEL  
COMBINED AUTOMATON**

*Presently the models of firmware automats (MPA) are widely used for the task of behavior of control (CU) unit. One of these models is a combined MPA (SMPA). Its characteristic feature is the presence of two types of output signals. The output signals of the Mile automaton exist at transitions between states of the automaton. The output of the Moore machine is determined only by the state of the machine and lasts almost a whole cycle. Due to the versatility of this model, it has become the basis for the research presented in this article. The most popular basis used to implement digital systems is FPGA (field-programmable logic arrays). The main elements of the FPGA used to implement MPAs are LUT (look-up table) elements, programmable triggers, and programmable interconnects.*

*The paper proposes a method of reducing hardware costs in the scheme of a combined machine, implemented in a common base of LUT elements and blocks of memory EMB. The method is based on replacement of logical conditions and partition of the set of states by classes. Each class corresponds to a single block of the circuit. This approach leads to circuits with regular structure and three levels of logic.*

*The proposed model leads to schemes with regular connections. This simplifies the placement and tracing tasks when implementing the SMPA scheme. A positive feature of the proposed model is the fact that the Clock and Start signals are associated with only one block of the circuit. This avoids the problems associated with the so-called distortion synchronization.*

*Analysis of the special library showed that the proposed method is appropriate to use for 78% of test cases. Studies were performed for chips of the Virtex-6 family ( $S = 6$ ). In this case, the developed SMPA models differed at high speed than their counterparts having the previous structure. For the remaining 22% of the test cases, the payoff was much smaller, as the specialized LUT units were implemented in the form of multilevel schemes.*

*A further direction of research is related to: 1) the replacement of some LUTs by EMBs and 2) the use of logical conditions encoding methods to reduce the parameter  $K$ .*

**Keywords:** combined microprogrammed automaton; synthesis; FPGA; LUT; partitioning; collections of microoperations.

**Баркалов А. А., Титаренко Л. А., Визор Я. Е., Матвиенко А. В.**  
**КОДИРОВАНИЕ НАБОРОВ МИКРООПЕРАЦИЙ В ТРЁХУРОВНЕВОМ  
СОВМЕЩЕННОМ АВТОМАТЕ**

*В настоящее время модели микропрограммных автоматов (МПА) широко используются для задания поведения устройства управления (УУ). К одной из таких моделей относится совмещенный МПА (СМПА). Его характерной чертой является наличие двух типов выходных сигналов. Выходные сигналы автомата Мили существуют при переходах между состояниями автомата. Выходные сигналы автомата Мура определяются только состояниями автомата и длятся практически целый такт. В силу универсальности этой модели мы используем ее в данной статье.*

*Наиболее популярным базисом, используемым для реализации цифровых систем, являются микросхемы FPGA (field-programmable logic arrays). Основными элементами FPGA, которые используются для реализации МПА, являются элементы табличного типа LUT (look-up table), программируемые триггера и программируемые межсоединения.*

*В работе предложен метод уменьшения аппаратных затрат в схеме совмещенного автомата, реализуемой в совместном базисе элементов LUT и блоков памяти EMB. Метод основан на замене логических условий и разбиении множества логических состояний на классы. Каждый класс соответствует отдельному блоку схемы. Такой подход приводит к схемам с регулярной структурой и тремя логическими уровнями.*

*Предложенная модель приводит к схемам с регулярными связями. Это упрощает задачи размещения и трассировки при реализации схемы СМПА. Положительной чертой предложенной модели является тот факт, что сигналы Clock и Start связаны только с одним блоком схемы. Это позволяет избежать проблем, связанных с так называемым перекосом синхронизации.*

*Анализ специальной библиотеки показал, что предложенный метод целесообразно использовать для 78 % тестовых примеров. Исследования проводились для микросхем семейства Virtex-6 ( $S = 6$ ). При этом разработанные схемы СМПА отличались большим быстродействием, чем их аналоги, имеющие предыдущую структуру. Для оставшихся 22% тестовых примеров выигрыш был значительно меньше, так как специализированные блоки LUT реализовывались в виде многоуровневых схем.*

*Дальнейшие направления исследований связано с: 1) заменой некоторых LUT блоками EMB и 2) использованием методов кодирования логических условий для уменьшения параметра  $K$ .*

**Ключевые слова:** совмещенный микропрограммный автомат; синтез; FPGA; LUT; разбиение; наборы микроопераций.

Стаття надійшла до редакції 04.11.2019 р.  
Прийнято до друку 09.12.2019 р.