

УДК 4.274

*С.О. ЦОЛОЛО*, канд. техн. наук, доц., ГВУЗ "ДонНТУ",  
Красноармійськ

### **ЗМЕНШЕННЯ ВИТРАТ АПАРАТУРИ ПРИ РЕАЛІЗАЦІЇ СХЕМИ АВТОМАТА МУРА НА ГІБРИДНИХ FPGA**

У роботі пропонується метод зменшення витрат апаратури у схемі мікропроцесорного автомату (МПА) Мура при реалізації в базисі гібридних FPGA. Метод використовує особливості МПА Мура і елементного базису FPGA та заснований на використанні двох джерел класів псевдоеквівалентних станів завдяки великій кількості входов вбудованих елементів PLA в сучасних гібридних FPGA. Лл.: 3. Бібліогр.: 17 назв.

**Ключові слова:** зменшення витрат апаратури, МПА Мура, гібридні FPGA, псевдоеквівалентні стани, вбудовані елементи PLA.

**Постановка проблеми та аналіз досліджень.** Одним з сучасних напрямків у технології FPGA є так називані гібридні FPGA [1, 2]. До складу подібних ПЛІС входять табличні елементи типу LUT (look-up table) і вбудовані програмувальні логічні матриці (PLA, programmable logic array). У літературі практично відсутні методи синтезу пристроїв керування, що орієнтовані на гібридні FPGA. Модель мікропрограмного автомата (МПА) Мура часто використовується при реалізації схем пристроїв керування [3, 4], які є важливою частиною цифрових схем [5]. У наш час програмувальні логічні інтегральні схеми (ПЛІС) типу FPGA (field programmable gate arrays) є популярним базисом для реалізації складних цифрових систем [6, 7]. При цьому велике значення має зменшення площі кристала, яку займає схема будь-якого блоку системи, у тому числі й МПА [8].

В якості одного із способів організації пристроїв керування часто використовуються автомати Мура і Мілі. У [9, 10] детально розглянуті алгоритми побудови керуючих та операційних автоматів, що об'єднуються для реалізації цифрових пристроїв. У цих роботах авторами пропонуються такі оригінальні підходи до синтезу логічних схем керуючих автоматів, як багаторівневі структури, принцип перетворення кодів об'єктів, можливості модифікації вихідних граф-схем алгоритмів, реалізації на лічильниках, використання блоків пам'яті.

Важливим для рівня витрат апаратури логічної схеми автомата Мура є етап кодування станів. В [11, 12] наведені ефективні алгоритми кодування станів автоматів і пристроїв керування на їх основі.

На даний момент ключовим моментом рішення задачі зменшення

витрат апаратури у схемі керуючого автомата є врахування особливостей елементного базису, в якому буде виконана логічна схема пристрою. Так, в роботах [13, 14] пропонуються способи оптимізації логічної схеми автомата Мура в базисах CPLD і FPGA, в основі яких лежить використання поняття псевдоеквівалентних станів, особливостей цільового базису, специфічні засоби кодування внутрішніх станів. В роботі [15] були запропоновані метод кодування наборів мікрооперацій і принцип розширення кодів станів переходу для ефективної реалізації автоматів Мура на схемах, що замовляються (ASICs).

Таким чином, можна зробити висновок, що тематика розробки методів зменшення апаратурних затрат в логічних схемах автомата Мура для різних елементних базисів є актуальною. Так, в даній роботі розглядається базис гібридних FPGA.

**Мета та завдання досліджень.** Мета – розробка методу зменшення витрат апаратури для схеми автомата Мура при її реалізації у базисі. Для досягнення мети були сформульовані і вирішені такі завдання:

- аналіз особливостей автомата (наявність класів псевдоеквівалентних станів) та елементного базису для їх ефективного врахування в методі, що розробляється;
- розробка способу зменшення схеми формування функцій збудження пам'яті автомата;
- вибір структурної схеми автомата Мура, що дозволяє зменшити витрати апаратури при збереженні рівня швидкодії пристрою.

**Результати розробки і досліджень.** Нехай алгоритм керування цифрової системи представлений ГСА (граф-схемою алгоритму)  $\Gamma = \Gamma(B, E)$ , де  $B = \{b_0, b_E\} \cup E_1 \cup E_2$  – множина вершин;  $b_0$  – початкова вершина ГСА;  $b_E$  – кінцева вершина ГСА;  $E_1$  – множина операторних вершин;  $E_2$  – множина умовних вершин,  $E = \{ \langle b_q, b_l \rangle \mid b_q, b_l \in B \}$  – множина дуг. У вершинах  $b_q \in E_1$  записуються набори мікрооперацій  $Y(b_q) \subseteq Y$ , де  $Y = \{y_1, \dots, y_N\}$  – множина мікрооперацій операційного автомата цифрової системи [1]. У вершинах  $b_q \in E_2$  записуються елементи множини логічних умов  $X = \{x_1, \dots, x_L\}$ . Початкова і кінцева вершини ГСА відповідають стану  $a_1 \in A = \{a_1, \dots, a_M\}$ , де  $A$  – множина станів автомата Мура, а кожна вершина  $b_q \in E_1$  відповідає одному з елементів множини  $A$  [3]. Логічна схема МПА Мура задається системою рівнянь

$$\Phi = \Phi(T, X), \quad (1)$$

$$Y = Y(T), \quad (2)$$

де  $\Phi = \{D_1, \dots, D_R\}$  – множина функцій порушення тригерів пам'яті станів;  $T = \{T_1, \dots, T_R\}$  – множина внутрішніх змінних, що кодує стани  $a_m \in A$ ,  $R = \lceil \log_2 M \rceil$ ;  $M$  – кількість станів. Система (1), (2) формується на основі прямої структурної таблиці (ПСТ) зі стовпцями:  $\Pi_A = \{B_1, \dots, B_I\}$  – поточний стан;  $K(a_m)$  – код стану  $a_m \in A$ ;  $a_s$  – стан переходу;  $K(a_s)$  – код стану  $a_s \in A$ ;  $X_h$  – кон'юнкція деяких елементів множини  $X$  (або їхніх заперечень), що визначає перехід  $\langle a_m, a_s \rangle$ ;  $\Phi_h$  – набір функцій збудження пам'яті МПА, що приймають одиничне значення для перемикавання пам'яті з  $K(a_m)$  в  $K(a_s)$ ;  $h = 1, \dots, H_1(\Gamma)$  – номер рядка таблиці. У стовпці  $a_m$  записується набір мікрооперацій  $Y(a_m) \subseteq Y$ , що сформовані у стані  $a_m \in A$ . Відмітимо, що  $Y(a_m) = Y(b_q)$ , де вершина  $b_q \in E_1$  відзначена станом  $a_m \in A$ .

Система (1), (2) визначає модель  $U_1$  автомата Мура (рис. 1), що включає блок переходів (БП), блок мікрооперацій (БМО) і реєстр станів Рг.

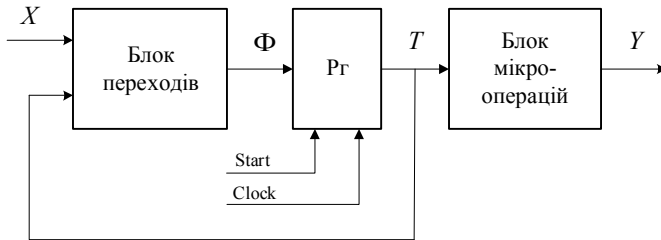


Рис. 1. Структурна схема МПА Мура  $U_1$

У схемі на рис. 1 блок БП реалізує функції (1), а блок БМО – функції (2). Коды станів  $a_m \in A$  зберігаються в реєстрі, що скидається сигналом Start і перемикається за сигналом Clock.

Як правило, число переходів  $H_1(\Gamma)$  більше числа переходів  $H_0(\Gamma)$  еквівалентного автомата Милі [1]. Це призводить до збільшення апаратних витрат у схемі МПА Мура в порівнянні із цим показником

еквівалентного автомата Милі. Параметр  $H_1(\Gamma)$  можна зменшити, завдяки наявності псевдоеквівалентних станів (ПЕС) МПА Мура [16]. Стани  $a_m, a_s \in A$  називаються ПЕС, якщо виходи відповідних їм вершин з'єднані із входом однієї й тієї ж вершини ГСА  $\Gamma$ . Нехай  $\Pi_A = \{B_1, \dots, B_I\}$  – розбивка множини  $A$  на класи ПЕС ( $I \leq M$ ). Побудуємо систему функцій

$$B_i = \bigvee_{m=1}^I C_{mi} A_m, \quad (i=1, \dots, I), \quad (3)$$

де  $C_{mi}$  – булева змінна, що дорівнює одиниці якщо і тільки якщо  $a_m \in B_i$ ;  $A_m$  – кон'юнкція внутрішніх змінних  $T_r \in T$ , що відповідають коду  $K(a_m)$  стану  $a_m \in A$ . Закодуємо стан  $a_m \in A$  так, щоб будь-яка функція системи (3) представлялася одним кон'юнктивним термом. Назвемо таке кодування оптимальним кодуванням станів.

Такий підхід веде до моделі  $U_2$ , структура якої збігається зі структурою моделі  $U_1$ , але число термів відповідає  $H_0(\Gamma)$ . Однак таке кодування не завжди можливо [4] через особливості ГСА. Наприклад, для  $R=2$ ,  $B_1 = \{a_1\}$ ,  $B_2 = \{a_2, a_3, a_4\}$  оптимальне кодування станів, що скорочує параметр  $H_2(\Gamma)$  до  $H_0(\Gamma)$ , неможливо. Тут ми вводимо позначення  $H_i(\Gamma_j)$ , що визначає число рядків у моделі  $U_i$  при інтерпретації ГСА  $\Gamma_j$ .

Число рядків ПСТ гарантоване дорівнює  $H_0(\Gamma)$ , якщо використати наступний підхід. Поставимо у відповідність класу  $B_i \in \Pi_A$  двійковий код  $K(B_i)$  розрядності  $R_B = \lceil \log_2 I \rceil$  і використаємо змінні  $\tau_r \in \tau$  для такого кодування, де  $|\tau| = R_B$ . У цьому випадку МПА Мура представляється у вигляді структури  $U_3$  (рис. 2).

У МПА  $U_3$  блок БП формує функції

$$\Phi = \Phi(\tau, X), \quad (4)$$

а блок перетворювача кодів (БПК) реалізує систему функцій

$$\tau = \tau(T). \quad (5)$$

При цьому код  $K(B_i)$  класу  $B_i \in \Pi_A$  формується на основі кодів станів  $a_m \in B_i$ . Блок БМО реалізує систему (2).

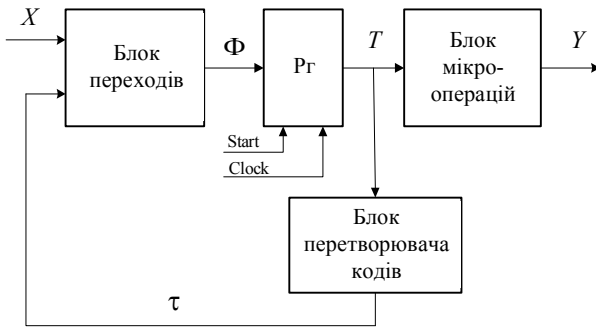


Рис. 2. Структурна схема МПА Мура  $U_3$

У роботі [16] показано, що  $H_3(\Gamma) = H_0(\Gamma)$ . Недоліком моделі  $U_3$  є наявність блоку БПК, що споживає деякі ресурси ПЛИС. У роботі пропонується метод синтезу МПА Мура, що дозволяє зберегти позитивні та усунути негативні якості моделі  $U_3$ .

Як вже було відзначено, метод орієнтований на технологію гібридних FPGA. У цьому випадку пропонується реалізувати схему БП на вбудованих PLA, а схеми PГ, БМО і БПК – на LUT-елементах. При цьому необхідно зменшувати число термів у функції (4). Число LUT елементів можна зменшити, якщо використати кілька джерел кодів класів ПЕС.

Закодуємо стани  $a_m \in A$  оптимальним чином. Нехай  $T(B_i)$  – число термів у функції  $B_i \in \Pi_A$ . Представимо множину  $\Pi_A$  у вигляді об'єднання множин  $\Pi_B$  і  $\Pi_C$ . При цьому розподіл класів виконується в такий спосіб:

$$\begin{aligned} (T(B_i) = 1) &\rightarrow B_i \in \Pi_B, \\ (T(B_i) > 1) &\rightarrow B_i \in \Pi_C. \end{aligned} \tag{6}$$

В цьому випадку перетворенню підлягають тільки коди станів  $a_m \in B_i$  для блоків  $B_i \in \Pi_C$ . Поставимо у відповідність кожному класу  $B_i \in \Pi_C$  двійковий код  $K(B_i)$  розрядності

$$R_c = \lceil \log_2(I_c + 1) \rceil, \tag{7}$$

де  $I_C = |\Pi_C|$ . Призначення одиниці у формулі (7) пояснимо пізніше.

Нехай наступні умови виконуються для PLA, що виходить до складу гібридної FPGA:

$$L + R + R_C \leq S, \quad (8)$$

$$H_0(I) \leq q. \quad (9)$$

В вирази (8), (9) входить число входів  $S$  і термів  $q$  блоку PLA. При виконанні (8), (9) блок БП реалізується у вигляді лише одного блоку PLA.

У роботі пропонується наступна модель  $U_4$  автомата Мура (рис. 3).

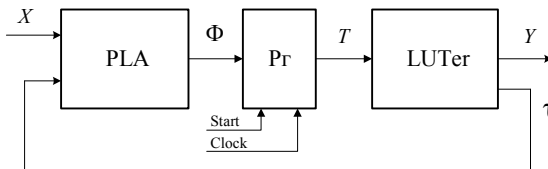


Рис. 3. Структурна схема МПА Мура  $U_4$

У МПА  $U_4$  блок PLA формує функції (8). Блок LUTer складається з LUT елементів і формує вихідні сигнали (2) і функції (5). У МПА  $U_4$  є два джерела кодів класів ПЕС. Коди класів  $B_i \in \Pi_B$  зберігаються в регістрі  $P_r$ , а коди класів  $B_i \in \Pi_C$  формуються блоком LUTer.

Для ідентифікації джерела коду ПЕС необхідний спеціальний код. Умовимося, що цей код визначається виразом

$$\tau_r = 0 \left( r = \overline{1, R_C} \right).$$

Наявністю цього коду й визначається необхідність додавання одиниці у виразі (7).

Таким чином, у роботі пропонується метод синтезу МПА Мура  $U_4$  за відзначеною ГСА, що включає наступні етапи:

1. Формування розбивки  $\Pi_A = \{B_1, \dots, B_I\}$ .
2. Оптимальне кодування станів  $a_m \in A$ .
3. Формування множин  $\Pi_B$  і  $\Pi_C$ .
4. Кодування класів  $B_i \in \Pi_C$ .
5. Формування вмісту блоку PLA.

6. Формування таблиці блоку LUTer.
7. Формування системи функцій, що задають схему МПА.
8. Реалізація схеми в заданому елементному базисі.

**Висновок.** Технологія гібридних FPGA вимагає адаптації методів синтезу МПА, орієнтованих на базис FPGA. Відсутність вбудованих блоків пам'яті викликає необхідність реалізації системи мікрооперацій на табличних елементах типу LUT. При виконанні умови (8) для реалізації кожної мікрооперації досить використати тільки один LUT. Система функцій збудження пам'яті реалізується на вбудованих блоках PLA. У роботі пропонується метод зменшення апаратурних витрат у схемі МПА Мура. Метод базується на використанні двох джерел класів псевдоеквівалентних станів. Це можливо, завдяки великій кількості входів PLA. Наприклад, у гібридних FPGA APEX20K фірми Altera цей параметр дорівнює 32. Застосування запропонованого методу доцільно при виконанні умов (8), (9), при цьому схема формування функцій збудження пам'яті реалізується на одному блоці PLA. Аналіз стандартних ГСА з бібліотеки [17] показав, що умови (8), (9) виконуються для 87% прикладів (для FPGA APEX 20K).

Наукова новизна запропонованого методу полягає в урахуванні особливостей автомата Мура і елементного базису гібридних FPGA для зменшення апаратурних витрат у схемі автомата. Практична значимість методу полягає в зменшенні вартості схеми автомата Мура на гібридних FPGA у порівнянні з відомими з літератури аналогами.

**Список літератури:** 1. *Kabiani A.* The Hybrid Field Programmable Architecture / *A. Kabiani, S. Brown.* – IEEE Design & Test of Computers. – 1999. – Vol. 16. – № 4. – P. 74-83. 2. Altera Corporation APEX20K PLD Family Data Sheet (2004) [електронний ресурс]. – Режим доступу: [www.altera.com](http://www.altera.com). 3. *Baranov S.* Logic Synthesis for Control Automata / *S. Baranov.* – Kluwer Academic Publishers, 1994. – 312 p. 4. *DeMicheli G.* Synthesis and Optimization of Digital Circuits / *G. DeMicheli.* – McGraw-Hill, 1994. – 636 p. 5. *Соловьев В.В.* Проектирование цифровых схем на основе программируемых логических интегральных схем / *В.В. Соловьев.* – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с. 6. *Грушиницкий П.И.* Проектирование систем с использованием микросхем программируемой логики / *П.И. Грушиницкий, А.Х. Мурцаев, Е.П. Угрюмов.* – СПб: БХВ. – Петербург, 2002. – 608 с. 7. *Skliarova I.* Design of FPGA-based circuits using Hierarchical Finite State Machines / *I. Skliarova, U. Sklyarov, A. Sudnitson.* – Tallinn: TUT Press, 2012. – 240 p. 8. *Czerwinski R.* Finite State Machine Logic Synthesis for complex programmable Logic Devices / *R. Czerwinski, D. Kania.* – Berlin: Springer, 2013. – 172 p. 9. *Barkalov A.A.* Synthesis of operational and control automata / *A.A. Barkalov, L.A. Titarenko.* – Donetsk: DonNTU, TechPark DonNTU UNITECH, 2009. – 256 p. 10. *Баркалов А.А.* Синтез микропрограммных автоматов на заказных и программируемых СБИС / *А.А. Баркалов, Л.А. Титаренко.* – Донецк: ДонНТУ, Технопарк ДонНТУ УНИТЕХ, 2009. – 336 с. 11. *El-Maleh A.* Finite state machine state assignment for area and power minimization / *A. El-Maleh, S.M. Sait, F.N. Khan* // Proceedings of IEEE International Symposium on Circuits and Systems. – 2006. – P. 5303-5306. 12. *Kubatova H.* FEL-Code: FSM internal state encoding method / *H. Kubatova, M. Becvar* // Proceedings of 5th International Workshop on

Boolean Problems. – Freiberg, 2002. – P. 109-114. **13.** *Баркалов А.А.* Оптимизация схемы МПА Мура на CPLD / *А.А. Баркалов, С.А. Ковалев, С.А. Цололо* // Материалы Восьмого международного научно-практического семинара "Практика и перспективы развития партнерства в сфере высшей школы". – Донецк – Таганрог, 2007. – Том 3. – С. 26-36. **14.** *Баркалов А.А.* Оптимизация логической схемы автомата Мура на FPGA / *А.А. Баркалов, А.А. Красичков, С.А. Цололо* // Наукowi праці ДонНТУ (Серия "Проблеми моделювання та автоматизації проектування динамічних систем"). – Донецк. – 2006. – № 5 (116). – С. 162-168. **15.** *Мальчева Р.В.* Матричная реализация автомата Мура с расширением кодов состояний перехода / *Р.В. Мальчева, К.А. Солдатов* // Наукowi праці ДонНТУ. (Серия "Информатика, кібернетика та обчислювальна техніка"). – Донецк. – 2010. – № 11 (164). – С. 79-83. **16.** *Баркалов А.А.* Принципы оптимизации логической схемы микропрограммного автомата Мура // Кибнетика и системный анализ. – 1998. – № 1. – С. 65-72. **17.** *Yang S.* Logic Synthesis and optimization bench-marks user guide / *S. Yan*. // Microelectronics Center of North Carolina, 1991. – 43 p.

**Bibliography (transliterated):** **1.** *Kabiani A.* The Hybrid Field Programmable Architecture / *A. Kabiani, S. Brown*. – IEEE Design & Test of Computers. – 1999. – Vol. 16. – № 4. – P. 74-83. **2.** Altera Corporation APEX20K PLD Family Data Sheet (2004) [elektronnij resurs]. – Rezhim dostupu: www.altera.com. **3.** *Baranov S.* Logic Synthesis for Control Automata / *S. Baranov*. – Kluwer Academic Publishers, 1994. – 312 p. **4.** *DeMicheli G.* Synthesis and Optimization of Digital Circuits / *G. DeMicheli*. – McGraw-Hill, 1994. – 636 p. **5.** *Solov'ev V.V.* Proektirovanie cifrovych shem na osnove programmirovemykh logicheskikh integral'nykh shem / *V.V. Solov'ev*. – M.: Gorjachaja liniya-TELEKOM, 2001. – 636 p. **6.** *Grushnickij R.I.* Proektirovanie sistem s ispol'zovaniem mikroshem programmirovemoj logiki / *R.I. Grushnickij, A.H. Mursaev, E.P. Ugrjumov*. – SPb: BHV. – Peterburg, 2002. – 608 p. **7.** *Skliarova I.* Design of FPGA-based circuits using Hierarchical Finite State Machines / *I. Skliarova, U. Sklyarov, A. Sudnitson*. – Tallinn: TUT Press, 2012. – 240 p. **8.** *Czerwinski R.* Finite State Machine Logic Synthesis for complex programmable Logic Devices / *R. Czerwinski, D. Kania*. – Berlin: Springer, 2013. – 172 p. **9.** *Barkalov A.A.* Synthesis of operational and control automata / *A.A. Barkalov, L.A. Titarenko*. – Donetsk: DonNTU, TechPark DonNTU UNITECH, 2009. – 256 p. **10.** *Barkalov A.A.* Sintez mikroprogrammnykh avtomatov na zakaznykh i programmirovemykh SBIS / *A.A. Barkalov, L.A. Titarenko*. – Doneck: DonNTU, Tehnopark DonNTU UNITEH, 2009. – 336 s. **11.** *El-Maleh A.* Finite state machine state assignment for area and power minimization / *A. El-Maleh, S.M. Sait, F.N. Khan* // Proceedings of IEEE International Symposium on Circuits and Systems. – 2006. – P. 5303-5306. **12.** *Kubatova H.* FEL-Code: FSM internal state encoding method / *H. Kubatova, M. Becvar* // Proceedings of 5th International Workshop on Boolean Problems. – Freiberg, 2002. – P. 109-114. **13.** *Barkalov A.A.* Optimizacija shemy MPA Mura na CPLD / *A.A. Barkalov, S.A. Kovalev, S.A. Cololo* // Materialy Vos'mogo mezhdunarodnogo nauchno-prakticheskogo seminaru "Praktika i perspektivy razvitiya partnerstva v sfere vsshej shkoly". – Doneck – Taganrog, 2007. – Tom 3. – P. 26-36. **14.** *Barkalov A.A.* Optimizacija logicheskoy shemy avtomata Mura na FPGA / *A.A. Barkalov, A.A. Krasichkov, S.A. Cololo* // Naukowi праці ДонНТУ. (Серия "Проблеми моделювання та автоматизації проектування динамічних систем"). – Doneck, 2006. – № 5 (116). – P. 162-168. **15.** *Mal'cheva R.V.* Matrichnaja realizacija avtomata Mura s rasshireniem kodov sostojanij perehoda / *R.V. Mal'cheva, K.A. Soldatov* // Naukowi праці ДонНТУ. (Серия "Информатика, кібернетика та обчислювальна техніка"). – Doneck. – 2010. – № 11 (164). – P. 79-83. **16.** *Barkalov A.A.* Principy optimizacii logicheskoy shemy mikroprogrammno avtomata Mura // Кибнетика i системный анализ. – 1998. – № 1. – P. 65-72. **17.** *Yang S.* Logic Synthesis and optimization bench-marks user guide / *S. Yan* // Microelectronics Center of North Carolina. – 1991. – 43 p.

Надійшла (received) 25.08.2015

*Статтю представив д.т.н., проф. ДонНТУ Святний В.А.*

Tsololo Sergii, PhD Tech  
Donetsk National Technical University  
Sq. Shibankova, 2, Krasnoarmiysk, Ukraine, 85300  
tel./phone: 093 883 21 28, e-mail: s.solos@gmail.com  
ORCID ID: 0000-0002-6425-4769