

О.О. БАРКАЛОВ

Університет Зеленогурський, Зелена Гура, Польща, Донецький національний університет імені Василя Стуса, Вінниця, Україна,
e-mail: *A.Barkalov@iie.uz.zgora.pl*.

Л.О. ТИТАРЕНКО

Університет Зеленогурський, Зелена Гура, Польща, Харківський національний університет радіоелектроніки, Харків, Україна,
e-mail: *L.Titarenko@iie.uz.zgora.pl*.

А.В. БАЄВ

Донецький національний університет імені Василя Стуса, Вінниця, Україна,
e-mail: *a.baev@donnu.edu.ua*.

О.В. МАТВІЄНКО

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, Україна,
e-mail: *avmatv@ukr.net*.

ПОДВІЙНЕ КОДУВАННЯ СТАНІВ У МІКРОПРОГРАМНИХ АВТОМАТАХ МУРА

Анотація. Запропоновано метод зменшення апаратних витрат у схемі автомата Мура, що реалізується в базисі ЕМВ і LUT. Метод ґрунтується на розбитті множини станів на класи, кожен з яких відповідає одному блоку логічних елементів. При цьому кожний стан має два коди. Такий підхід приводить до трирівневої схеми автомата Мура. Розглянуто приклад синтезу схеми МПА Мура з використанням запропонованого методу. Розглянуто умови його застосування. Дослідження на базі стандартних автоматів показали, що запропонований метод дає змогу зменшити апаратні витрати в порівнянні з іншими методами.

Ключові слова: автомат Мура, синтез, ЕМВ, LUT, структурна декомпозиція, розбиття.

ВСТУП

Модель мікропрограмного автомата (МПА) Мура широко використовують для синтезу різних послідовних блоків цифрових систем [1, 2]. Під час реалізації схем МПА потрібно розв'язати низку оптимізаційних задач, таких як мінімізація апаратних витрат, підвищення швидкодії, зменшення споживаної потужності [3, 4]. Методи розв'язання цих задач багато в чому залежать від елементного базису [5, 6]. У цій роботі пропонується метод зменшення апаратних витрат, орієнтований на НВІС типу FPGA (field-programmable logic array) [7, 8].

Базис FPGA став провідною платформою реалізації найрізноманітніших цифрових систем [8, 9]. Наприклад, в огляді [10] наведено понад 1700 різних прикладів систем у базисі FPGA. На підставі цього аналізу у цій роботі вибрано базис FPGA для реалізації схеми МПА Мура.

Зазвичай для реалізації схем МПА використовують такі компоненти мікросхем FPGA [6, 7]: елементи табличного типу LUT (look-up table), тригери, програмовані міжз'єднання і програмовані блоки входів–виходів [11]. До складу мікросхем FPGA також входять вбудовані блоки пам'яті ЕМВ (embedded memory blocks). Ці блоки мають властивість реконфігурації [12], що ефективно використовують під час реалізації схем автоматів [13–15]. Проте, блоки ЕМВ широко застосовують для реалізації операційних частин цифрових систем [10]. Тому розробник схем МПА рідко має можливість використовувати ЕМВ.

Елементи LUT мають вкрай обмежену кількість входів [7, 11]. Це призводить до збільшення кількості рівнів логіки у схемах МПА [16], зумовлене вико-

ристанням методів функціональної декомпозиції [5]. Відомо, що функціональна декомпозиція призводить до схем з нерегулярною системою міжз'єднань [16].

Альтернативою функціональній декомпозиції є структурна декомпозиція [17, 18]. Застосування структурної декомпозиції дає змогу підвищити регулярність міжз'єднань і зменшити кількість елементів LUT у схемах автоматів [18]. Одним із таких методів є подвійне кодування станів (ПКС) [19, 20]. Зазвичай ПКС застосовують для оптимізації показників МПА Мілі [19, 20].

У цій роботі пропонується використовувати ПКС для покращення характеристик схем МПА Мура. Розглядається випадок, коли деяка частина схеми реалізується на ЕМВ і схема МПА Мура є композицією блоків LUT, ЕМВ, тригерів і міжз'єднань. Для початкового завдання поведінки МПА використовується мова граф-схем алгоритму (ГСА) [21].

РЕАЛІЗАЦІЯ СХЕМ МПА В БАЗИСІ FPGA

Автомат Мура визначається вектором $S = \langle A, X, Y, \delta, \lambda, a_1 \rangle$ [21]. Тут $A = \{a_1, \dots, a_M\}$ — множина станів, $X_1 = \{x_1, \dots, x_L\}$ — множина входів, $Y = \{y_1, \dots, y_N\}$ — множина виходів, δ — функція переходів, λ — функція виходів, $a_1 \in A$ — початковий стан. У процесі функціонування автомат переходить із поточного стану $a_m \in A$ у стан переходу $a_s \in A$. При цьому формується вихідний сигнал $Y_q \subseteq Y$. Переходи залежать від вхідних сигналів. Функції δ і λ визначаються так:

$$\delta: A \times X \rightarrow A, \quad (1)$$

$$\lambda: A \rightarrow Y. \quad (2)$$

У формулі (1) знаком « \times » позначено добуток множин.

Початкову ГСА Γ позначають станами автомата Мура, використовуючи правила [21]. Для синтезу схеми ця ГСА перетворюється на пряму структурну таблицю (ПСТ). При цьому стани $a_m \in A$ кодують двійковими кодами $K(a_m)$, що мають R розрядів, $R = \lceil \log_2 M \rceil$.

Для кодування станів використовують внутрішні змінні $T = \{T_1, \dots, T_R\}$. Для зміни кодів станів використовують функції збудження пам'яті (ФЗП) з множини $\Phi = \{D_1, \dots, D_R\}$.

На основі ПСТ формують дві системи Булевих функцій (СБФ), які використовують для реалізації схеми МПА:

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

$$Y = Y(T). \quad (4)$$

Система (3) відповідає функції переходів (1), а система (4) — функції виходів (2). Системи (3), (4) представляють у вигляді диз'юнктивних нормальних форм (ДНФ), терми яких відповідають рядкам ПСТ [21, 22].

Елемент LUT має S_L входів, які призначені для вибірки біта інформації з одного з 2^{S_L} комірок пам'яті [6, 7]. Таким чином, LUT може реалізувати

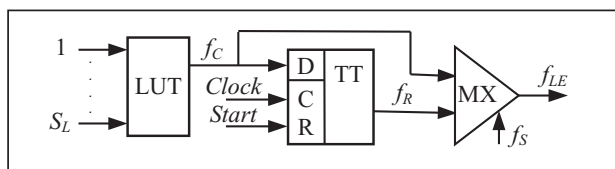


Рис. 1. Спрощена структура логічного елемента

довільну ДНФ, що має аргументи S_L . Вихід елемента LUT пов'язаний із входом тригера. Це з'єднання відбувається усередині логічного елемента (рис. 1).

Логічний елемент функціонує в такий спосіб. Набір вхідних змінних визначає функцію $f_C \in \{0,1\}$. За сигналом *Clock* значення f_C записується в тригер ($f_C := \{0,1\}$). За сигналом *Start* вміст тригера встановлюється в нуль (якщо це потрібно). Змінні f_C та f_R надходять на входи мультиплексора MX. Внутрішній сигнал $f_s \in \{0,1\}$ призначений для вибору змінної, що надходить на вихід f_{LE} .

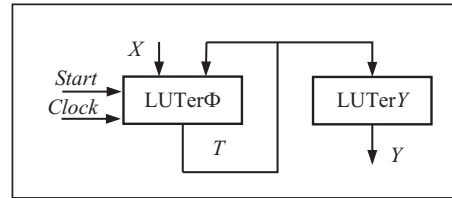


Рис. 2. Структурна схема PУ МПА Мура

Отже, логічний елемент (див. рис. 1) призначений для реалізації як комбінаційних схем, так і накопичувальних [16]. Відповідно до теореми В.М. Глушкова [23] логічні елементи є структурно повним базисом. Використовуючи їх, можна реалізувати PУ МПА Мура (рис. 2), де позначення LUTer означає схему, що складається з логічних елементів (див. рис. 1).

У автоматі PУ блок LUTerPhi реалізує СБФ (3). Тригери логічних елементів утворюють розподілений регістр кодів станів (RG). За сигналом *Start* у регістр заноситься код, що складається з нулів (зазвичай цей код відповідає початковому стану $a_1 \in A$). За сигналом *Clock* в RG записується код чергового стану. Блок LUTerY реалізує СБФ (4).

Нехай ДНФ функції $f_i \in \Phi \cup Y$, має $NA(f_i)$ літералів [22]. Якщо виконується умова

$$NA(f_i) \leq S_L, \quad (5)$$

то для реалізації відповідної схеми використовується лише один LUT. Якщо умова (5) виконується для кожної функції $f_i \in \Phi \cup Y$, то схема PУ МПА Мура складається з $R + N$ елементів.

Однак елементи LUT мають вкрай обмежену кількість входів. Так, для FPGA провідних виробників Xilinx [11] та Intel (Altera) [24] значення S_L не перевищує 6. При цьому навіть для автоматів середньої складності [21] цілком можливо, що $L + R \approx 40$. Така суперечність зумовлює застосування методів функціональної декомпозиції СБФ (3), (4) з усіма негативними наслідками.

Для поліпшення параметрів схем МПА можна використовувати блоки ЕМВ [13–15]. Характерною для блоків ЕМВ є можливість налаштування їхньої конфігурації. При цьому можна змінювати кількість входів (S_A) та виходів (t_F) за незмінної ємності блока (V_0): $V_0 = 2^{S_A} \times t_F$.

Пара $\langle S_A, t_F \rangle$ визначає конфігурацію ЕМВ. Наприклад, для мікросхем Virtex-7 [25] існують такі конфігурації: $\langle 15, 1 \rangle$, $\langle 14, 2 \rangle$, $\langle 13, 4 \rangle$, $\langle 12, 8 \rangle$, $\langle 11, 16 \rangle$, $\langle 10, 32 \rangle$ та $\langle 9, 64 \rangle$. Аналіз бібліотеки стандартних автоматів [26] показує, що 67 % автоматів можна реалізувати за допомогою лише одного блока ЕМВ. Для інших автоматів бібліотеки [26] потрібно представити схему як мережу блоків ЕМВ або мережу блоків ЕМВ і елементів LUT.

У цій роботі розглядається випадок, коли існує лише один «вільний» блок ЕМВ, порушується умова (5), не вистачає ресурсів ЕМВ для реалізації схеми МПА. Отже, схема МПА Мура реалізується у вигляді мережі елементів LUT і блока ЕМВ. Запропонований у роботі метод орієнтований на досить складні автомати, для яких умова (5) порушується практично для всіх функцій. Ця ситуація можлива, наприклад, якщо виконується умова $R > S_L$.

У цьому випадку кожен функцію потрібно перетворити, використовуючи різні методи функціональної декомпозиції [16, 27]. Остання зводиться до представлення функції f_i як композиції підфункцій, для яких виконується умова (5). При цьому

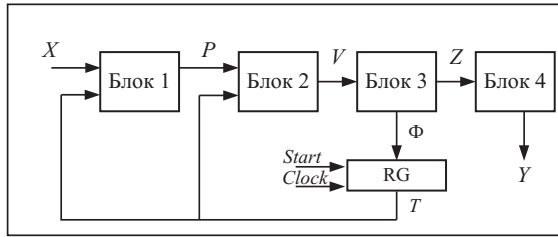


Рис. 3. Представлення схеми МПА у вигляді композиції структурних блоків

кожна підфункція реалізується схемою з одного елемента LUT. Декомпозицію потрібно виконати так, щоб мінімізувати кількість елементів LUT або мінімізувати кількість рівнів схеми, або мінімізувати узагальнену оцінку, що визначається добутком кількості елементів схеми на час затримки поширення сигналу [16].

Недоліком функціональної декомпозиції є велика кількість рівнів результатних схем і складна система міжз'єднань [16]. Відомо, що збільшення рівня інтеграції в мікросхемах FPGA призвело до того, що майже 70 % потужності споживається міжз'єднаннями, при цьому їхня затримка часу стає порівняною із затримкою на елементах LUT [28, 29]. Отже, у схемі МПА потрібно зменшувати кількість міжз'єднань. Це можна зробити, використовуючи різні методи структурної декомпозиції [17–20]. Зазначимо, що перші такі методи були запропоновані Майклом Уїлксом у 1951 р. [30]. Ці методи призначалися для зменшення ємності керівної пам'яті мікропрограмних пристроїв керування [30]. Під час використання структурної декомпозиції схема МПА представляється у вигляді композиції окремих структурних блоків (рис. 3).

Як видно з рис. 3, кожен блок має свою унікальну систему входів та виходів. Блок 1 виконує перетворення входів X у додаткові змінні з множини P , $|P| \ll L$. Блок 2 здійснює кодування термів ПСТ, що залежать від внутрішніх змінних $T_r \in T$ та елементів множини P . Блок 3 генерує ФЗП $D_r \in \Phi$ та змінні $z_r \in Z$, що кодують набори виходів МПА. Блок 4 перетворює змінні $z_r \in Z$ у виходи $y_n \in Y$.

Такий підхід спрощує систему з'єднань. При цьому кількість рівнів у схемі МПА може змінюватись від двох до чотирьох [17], що дає змогу зменшити кількість елементів схеми і збільшити її швидкодію [19]. Проте, цілком можливо, що схеми деяких блоків будуть багаторівневими [20], а це зменшує ефективність структурної декомпозиції, оскільки система міжз'єднань ускладнюється і швидкодія зменшується.

У цій роботі пропонується використовувати метод подвійного кодування станів [19, 20] для оптимізації характеристик схем МПА Мура. Раніше цей метод показав свою ефективність для оптимізації схем МПА Мілі [19, 20]. За виконання певних умов цей підхід дає змогу отримати схеми, що мають точно три рівні елементів. Далі розглядається випадок, коли частина схеми МПА реалізується на блоці ЕМВ. Однак лише один блок вбудованої пам'яті доступний розробнику схеми МПА.

ОСНОВНА ІДЕЯ ЗАПРОПОНОВАНОГО МЕТОДУ

Нехай для деякого МПА Мура S_0 отримано системи (3) та (4). Потрібно синтезувати логічну схему МПА з використанням одного блока ЕМВ та елементів LUT, що мають входи S_L . Нехай умова (5) порушується одночасно з виконанням умови $2^{R+L}(N+R) > V_0$. Це означає, що ресурсів блока ЕМВ замало для реалізації схеми МПА.

Нехай серед конфігурацій $\langle S_A, t_F \rangle$ є пара $\langle S_A^0, t_F^0 \rangle$, для якої виконується умова $(S_A^0 = L + R) \vee (S_A^0 - 1 < L + R) = 1$. Це означає, що t_F^0 функцій $f_i \in \Phi \cup Y$ мо-

жуть бути реалізовані як виходи ЕМВ. Функції (3) залежать від входів $x_l \in X$. Тому доцільно реалізувати ФЗП на ЕМВ.

Нехай виконується умова $n_E = t_F^0 - R > 0$.

У цьому випадку n_E функцій $y_n \in Y$ можуть бути реалізовані на ЕМВ. Для цього множини Y необхідно представити у вигляді об'єднання множин Y_E і Y_L , що не перетинаються (Y_E реалізуються на ЕМВ, Y_L реалізуються на елементах LUT). Такий підхід приведе до $P_E Y_{EL}$ МПА Мура (рис. 4).

Якщо $n_E = 0$, то $Y = Y_L$. При цьому LUTerY реалізує всі функції $y_n \in Y$. Якщо $n_E < 0$, то множина Φ розбивається на дві множини, що не перетинаються: Φ_E (реалізується на ЕМВ) і Φ_L (реалізується на елементах LUT). У результаті лише частина входів використовується елементами LUT. Це призводить до $P_{EL} Y_L$ автомата Мура (рис. 5).

В автоматі $P_E Y_{EL}$ множина $X = X_E \cup X_L$. Проте, можливе співвідношення $X_E \cap X_L \neq \emptyset$. Якщо $X_E \subset X$, то кількість виходів ЕМВ збільшується у $2 \times (L - |X_E|)$ рази. При цьому частина (чи всі) виходи можна реалізувати блоком ЕМВ. Це приводить до $P_{EL} Y_{EL}$ МПА (рис. 6).

Недолік моделей із блоком LUTer Φ очевидний: за порушення умови (5) для функцій $f_i \in \Phi_L$ блок LUTer Φ буде представлений багаторівневою схемою. У цій роботі пропонується альтернатива автоматам $P_E Y_{EL}$ у разі виконання умови $t_F^0 - R < 0$.

Знайдемо розбиття множини $A = \{A_E, A_L\}$. Клас A_E включає M_E станів $a_m \in A$, для яких ФВП записують у вигляді

$$\Phi_E = \Phi_E(\tau^E, X^E). \quad (6)$$

Стани $a_m \in \Phi_E$ кодується кодами $C(a_m)$, що мають R_E розрядів, $R_E = \lceil \log_2(M_E + 1) \rceil$. Тут одиниця додається обліку відношення $a_m \notin A_E$.

Функції (6) реалізуються блоком ЕМВ. Кількість елементів множини $A_E \subset A$ обмежена можливостями ЕМВ. Потрібно, щоб виконувалися такі умови:

$$2^{(L_E + R_E)} \times R \leq V_0, \quad (7)$$

$$2^{(L_E + R_E + 1)} \times R > V_0, \quad (8)$$

де $L_E = |X^E|$.

З умов (7), (8) випливає, що збільшення суми $L_E + R_E$ на одиницю унеможливує реалізацію СБФ (6) як одного блока ЕМВ. Умови (7), (8) обмежують потужності множин A_E та X^E .

Знайдемо розбиття Π_A множини $A_L \subset A$ на класи A^1, \dots, A^K . Клас A^k має M_k елементів, для кодування яких використовується R_k змінних:

$$R_k = \lceil \log_2(M_k + 1) \rceil \quad (k = \overline{1, K}). \quad (9)$$

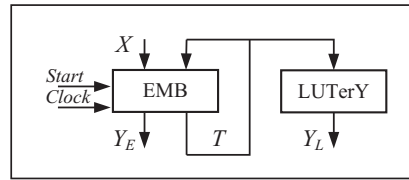


Рис. 4. Структурна схема $P_E Y_{EL}$ МПА Мура

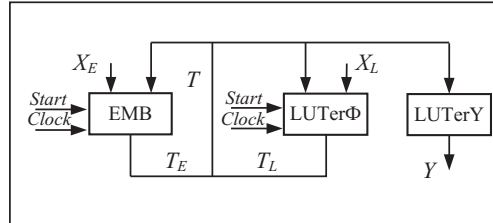


Рис. 5. Структурна схема $P_E Y_{EL}$ МПА Мура

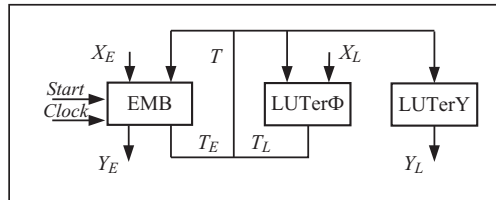


Рис. 6. Структурна схема $P_{EL} Y_{EL}$ МПА Мура

Переходи із станів $a_m \in A^k$ залежать від входів $x_l \in X^k$, де $|X^k|=L_k$. Розбиття Π_A повинне включати класи, що задовольняють умову

$$R_k + L_k \leq S_L. \quad (10)$$

При цьому кожному класу $A^k \in \Pi_A$ відповідає однорівневий блок LUTer k ($k \in \overline{1, K}$).

Для кодування станів використовуємо елементи множини $\tau = \{\tau_1, \dots, \tau_{R_A}\}$. Величина R_A складається із суми R_E і R_L , де R_L — кількість змінних, що кодують стани $a_m \in A^k$: $R_L = \sum_{k=1}^K R_k$.

Тепер кожен стан $a_m \in A^k$ має два коди. Код $K(a_m)$ визначає стан як елемент множини A . Для цих кодів використовуються внутрішні змінні $T_r \in T$. Код $C(a_m)$ визначає стан як елемент деякого класу розбиття множини A . Множина τ може бути представлена як $\tau = \tau^E \cup \tau^1 \cup \dots \cup \tau^K$. Елементи множини $\tau^E \subset \tau$ кодують стани $a_m \in A_E$. Елементи множини $\tau^k \subset \tau$ кодують стани $a_m \in A^k$, де $k = \{1, \dots, K\}$.

Кожен клас $A^k \in \Pi_A$ визначає множини $\tau^k \subset \tau$, $X^k \subseteq X$, $\Phi^k \subseteq \Phi$. Множина X^k включає входи, що визначають переходи зі станів $a_m \in A^k$. Множина Φ^k включає ФЗП, що дорівнюють 1 у випадку переходу зі станів $a_m \in A^k$. Клас $A^E \subset A$ визначає множини X^E , Φ^E і τ^E , що відповідають множинам X^k , Φ^k і τ^k .

Нехай для деякого автомата S знайдено множини A_E, A^1, \dots, A^K і виконано кодування станів кодами $K(a_m)$ і $C(a_m)$. Пропонується представити (рис. 7) автомат S як P_2Y МПА (2 означає, що автомат базується на подвійному кодуванні станів [20]).

У P_2Y МПА блок EMB реалізує СБФ (6), блок LUTer k реалізує СБФ $\Phi^k = \Phi^k(\tau^k, X^k)$.

$$\Phi^k = \Phi^k(\tau^k, X^k). \quad (11)$$

Блок LUTer T містить регістр RG, що зберігає коди $K(a_m)$. Цей блок реалізує СБФ

$$\Phi = \Phi(\Phi^E, \Phi^1, \dots, \Phi^K). \quad (12)$$

Блок LUTer Y формує функції системи (4). Блок LUTer τ призначений для перетворення кодів $K(a_m)$ на коди $C(a_m)$. Блок LUTer τ реалізує СБФ

$$\tau = \tau(T). \quad (13)$$

За виконання умов (9) та $K \leq S_L$,

$$R \leq S_L \quad (15)$$

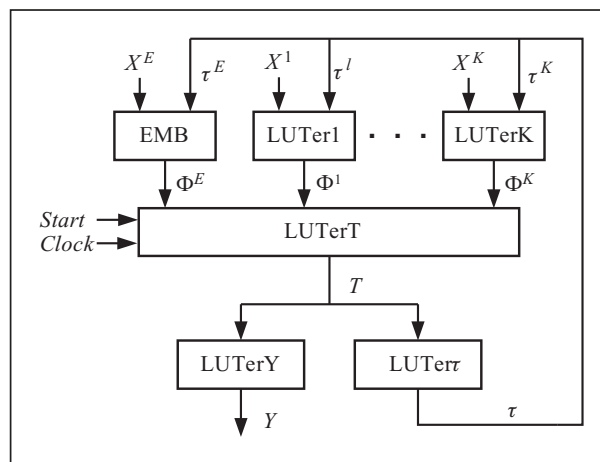


Рис. 7. Структурна схема P_2Y автомата Мура

схема автомата P_2Y має точно три рівні елементів LUT. Якщо умова (15) порушується, кількість елементів у схемі LUTer Y можна зменшити за рахунок кодування станів. Для цього можна використати, наприклад, метод [31].

У цій роботі пропонується метод синтезу схе-

ми P_2Y МПА на основі ГСА Γ . Запропонований метод включає такі етапи:

- позначення ГСА Γ та формування множини станів автомата Мура;
- розбиття множини станів на класи A_E та A_L ;
- розбиття множини A_L на класи A^k ($k=1, K$);
- кодування станів, що оптимізує СБФ (4);
- кодування станів як елементів класів A_E та A^k ;
- формування таблиці блока ЕМВ;
- формування таблиць блоків LUTer k та СБФ (11);
- формування таблиць блоків LUTer T та СБФ (12);
- формування таблиці блока LUTer τ та СБФ (13);
- реалізація схеми МПА у базисі FPGA.

Позначимо $P_2Y(\Gamma_j)$ те що, що схема автомата P_2Y Мура синтезується за ГСА Γ_j . Розглянемо приклад синтезу схеми МПА $P_2Y(\Gamma_1)$. Позначена ГСА Γ_1 показана на рис. 8. Для реалізації схеми МПА використовуються елементи LUT з $S_L = 5$ та один блок ЕМВ, який має такі конфігурації $\langle S_A, t_F \rangle$: $\langle 11, 1 \rangle$, $\langle 10, 2 \rangle$, $\langle 9, 4 \rangle$, $\langle 8, 8 \rangle$, $\langle 7, 16 \rangle$.

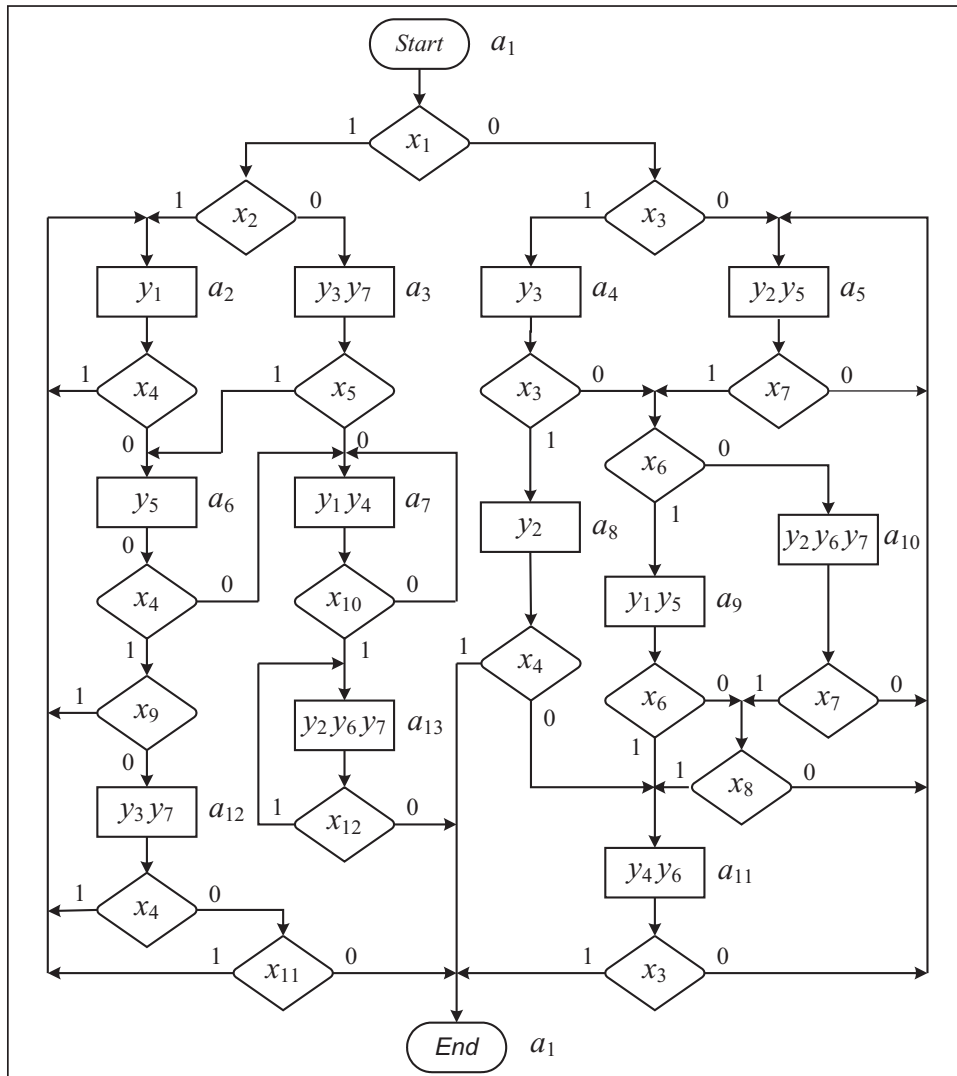


Рис. 8. Позначена ГСА Γ_1

Використовуючи підхід [20], отримуємо множину $A_E = \{a_1, a_4, a_5, a_8, a_9, a_{10}, a_{11}\}$ з $M_E = 7$. Використовуючи (13), отримуємо $R_E = 3$ і множину $\tau^E = \{\tau_1, \tau_2, \tau_3\}$. Аналіз ГСА Γ_1 дає змогу отримати множину $X^E = \{x_1, x_2, x_3, x_6, x_7, x_8\}$, для якої $L_E = 6$. Отже, маємо $R_E + L_E = 9$. Таким чином, до блока A_E не можна включити додаткові стани.

Операція A / A_E дає клас $A_L = \{a_2, a_3, a_6, a_7, a_{12}, a_{13}\}$. За умови $S_L = 5$ потрібно знайти класи $A^k \in \Pi_A$, які відповідають умові $R_k + L_k \leq 5$. Для цього використаємо метод [20] і отримаємо розбиття $\Pi_A = \{A^1, A^2\}$. Розбиття визначає такі множини: $A^1 = \{a_2, a_6, a_{12}\}$, $A^2 = \{a_3, a_7, a_{13}\}$, $X^1 = \{x_4, x_9, x_{11}\}$, $X^2 = \{x_5, x_{10}, x_{12}\}$. З формули (9) маємо $R_1 = R_2 = 2$. Оскільки $L_1 = L_2 = 2$, то умова (10) виконується для класів A^1 та A^2 . Очевидно, маємо множини $\tau^1 = \{\tau_4, \tau_5\}$, $\tau^2 = \{\tau_6, \tau_7\}$.

	$T_1 T_2$	00	01	11	10
$T_3 T_4$	00	a_1	a_2	a_7	a_8
	01	a_6	*	a_9	a_5
	11	a_{12}	*	*	a_{10}
	10	a_3	a_4	a_{11}	a_{13}

Рис. 9. Коды станів МПА Мура

Закодуємо стани $a_m \in A$ кодами $K(a_m)$, що мають чотири розряди, для мінімізації кількості аргументів у ДНФ функцій (4). У розглядуваному випадку $R = 4 < S_L = 5$, тобто умова (15) виконується. Однак оптимальне кодування станів дає змогу зменшити кількість міжз'єднань між блоками LUTerT і LUTerY, що приведе до зменшення споживаної потужності [19].

Для кодування використовуємо підхід [31]. Один із можливих варіантів кодування наведено на рис. 9.

Використовуючи коди $K(a_m)$, отримуємо таку СБФ:

$$\begin{aligned}
 y_1 &= A_2 \vee A_7 \vee A_9 = T_2 \overline{T_3}; & y_4 &= A_7 \vee A_{11} = T_1 T_2 \overline{T_4}; \\
 y_2 &= A_5 \vee A_8 \vee A_{10} \vee A_{13} = T_1 \overline{T_2}; & y_5 &= A_5 \vee A_6 \vee A_9 = \overline{T_3} T_4; \\
 y_3 &= A_3 \vee A_4 \vee A_{12} = \overline{T_1} \overline{T_3}; & y_6 &= A_{10} \vee A_{11} \vee A_{13} = T_1 T_3; \\
 & & y_7 &= A_3 \vee A_{10} \vee A_{12} \vee A_{13} = \overline{T_2} T_3.
 \end{aligned} \quad (16)$$

У загальному випадку система (4) має максимально $R \times N = 4 \times 7 = 28$ літералів. Система (16) має 15 літералів. Таким чином, кількість міжз'єднань зменшено у 1.87 рази.

Як показано в [19, 20], коди $C(a_m)$ не впливають на апаратні витрати. Тому стани можна закодувати довільно. Для кожного класу будемо використовувати код із усіма нулями для відображення того, що стан не належить цьому класу. Один із варіантів кодування наведено у табл. 1.

Таблиця блока ЕМВ містить такі стовпчики: $a_m, C(a_m), X^E, \Phi^E, h$. Код $C(a_m)$ і вектор вхідних сигналів $\langle x_1, \dots, x_l \rangle$, де $x_l \in X^E$, визначають адресу комірки ЕМВ. Стовпчик Φ^E містить інформацію, що зберігається в комірках пам'яті. Таблиця ЕМВ має H_E рядків, $H_E = 2^{R_E + L_E}$. Переходи зі стану $a_m \in A_E$ задаються частиною таблиці блока ЕМВ, що має $H(a_m)$ рядків, $H(a_m) = 2^{L_E}$.

У розглядуваному прикладі маємо $H_E = 512$ і $H(a_m) = 64$. Перші $H(a_m)$ рядків містять нулі, оскільки код 00...0 не використовується для кодування станів $a_m \in A_E$. Перші 10 рядків таблиці переходів із стану $a_1 \in A_E$ показані у табл. 2.

У табл. 2 додано стовчик a_s , щоб показати стан переходу для вектора входів. Як відомо, для D тригерів набір ФЗП збігається з кодом стану переходу [5]. Наразі коди беруться з карти Карно (рис. 9).

Розглянемо рядок 65 (див. табл. 2). Як впливає із ГСА Γ_1 , якщо $x_1 = x_3 = 0$, стан переходу $a_s = a_5$. З рис. 9 знаходимо код $K(a_5) = 1001$. Цей код записаний у стовпчиках $D_1 - D_4$ рядка 65. Аналогічно заповнюються інші рядки таблиці.

Таблиця блока LUTerK має стовпчики: $a_m, a_s, K(a_s), X_h, \Phi_h, h$. Наприклад, блок LUTer1 представляється табл. 3, а блок LUTer2 — табл. 4.

Система (11) формується з урахуванням кожної з цих таблиць. З табл. 3 та табл. 4 маємо:

$$D_1^1 = \tau_4 \tau_5 x_4; \quad D_2^1 = \tau_4 \tau_5 x_4 \vee \tau_4 \tau_5 x_4 x_{11};$$

$$D_3^1 = \tau_4 \tau_5 x_4 x_9; \quad D_4^1 = \tau_4 \tau_5 x_4 \vee \tau_4 \tau_5 x_4 x_9;$$

$$D_1^2 = \tau_6 \tau_7 x_5 \vee \tau_6 \tau_7 \vee \tau_6 \tau_7 x_{12}; \quad D_2^2 = \tau_6 \tau_7 x_5 \vee \tau_6 \tau_7 x_{10};$$

$$D_3^2 = \tau_6 \tau_7 x_{10} \vee \tau_6 \tau_7 x_{12}; \quad D_4^2 = \tau_6 \tau_7 x_5.$$

Таблиця блока LUTerT має $K + 3$ стовпчиків: $\Phi, A_E, A^1, \dots, A^K, r$. Якщо $D_r \in \Phi^E$, то на перетині рядка D_r та стовпчика A_E записується 1, інакше 0. Якщо $D_r \in A^k$, то 1 записується на перетині рядка D_r та стовпчика A^k . Для цього прикладу блок LUTerT представлено у табл. 5. Ця таблиця призначена для

Таблиця 2. Частина таблиці блока ЕМВ автомата $P_2Y(\Gamma_1)$

a_m	$C(a_m)$			X^E						Φ^E				h	a_s
	τ_1	τ_2	τ_3	x_8	x_7	x_6	x_3	x_2	x_1	D_1	D_2	D_3	D_4		
a_1	0	0	1	0	0	0	0	0	0	1	0	0	1	65	a_5
a_1	0	0	1	0	0	0	0	0	1	0	0	1	0	66	a_3
a_1	0	0	1	0	0	0	0	1	0	1	0	0	1	67	a_5
a_1	0	0	1	0	0	0	0	1	1	0	1	0	0	68	a_2
a_1	0	0	1	0	0	0	1	0	0	0	1	1	0	69	a_4
a_1	0	0	1	0	0	0	1	0	1	0	0	1	0	70	a_3
a_1	0	0	1	0	0	0	1	1	0	0	1	1	0	71	a_4
a_1	0	0	1	0	0	0	1	1	1	0	1	0	0	72	a_2
a_1	0	0	1	0	0	1	0	0	0	1	0	0	1	73	a_5
a_1	0	0	1	0	0	1	0	0	1	0	0	1	0	74	a_3

Таблиця 3. Таблиця LUTer1 МПА $P_2Y(\Gamma_1)$

a_m	$C(a_m)$	a_s	$K(a_s)$	X_h	Φ_h	h
a_2	01	a_2	0100	x_4	D_2	1
		a_6	1001	\bar{x}_4	D_4	2
a_6	10	a_2	1000	x_4x_9	D_2	3
		a_{12}	0101	$x_4\bar{x}_9$	D_3D_4	4
		a_7	1100	\bar{x}_4	D_1D_2	5
a_{12}	11	a_2	0000	x_4	D_2	6
		a_2	1010	$x_4\bar{x}_{11}$	D_2	7
		a_1	0000	$\bar{x}_4\bar{x}_{11}$	—	8

Таблиця 4. Таблиця LUTer2 МПА $P_2Y(\Gamma_1)$

a_m	$C(a_m)$	a_s	$K(a_s)$	X_h	Φ_h	h
a_3	01	a_6	0001	x_5	D_4	1
		a_7	1100	\bar{x}_5	D_1D_2	2
a_1	10	a_{13}	1010	x_{10}	D_1D_3	3
		a_7	1100	\bar{x}_{10}	D_1D_2	4
a_{13}	11	a_{13}	1010	x_{12}	D_1D_3	5
		a_1	0000	\bar{x}_{12}	—	6

Таблиця 5. Таблиця блока LUTerT МПА $P_2Y(\Gamma_1)$

Φ_h	A_E	A^1	A^2	r
D_1	1	1	1	1
D_2	1	1	1	2
D_3	1	1	1	3
D_4	1	1	1	4

Таблиця 6. Таблиця блока LUTerT МПА $P_2Y(\Gamma_1)$

a_m	$K(a_m)$	$C(a_m)$	τ_m	m
a_1	0000	0010000	τ_3	1
a_2	0100	0000100	τ_5	2
a_3	0010	0000001	τ_7	3
a_4	0110	0100000	τ_2	4
a_5	1001	0110000	$\tau_2\tau_3$	5
a_6	0001	0001000	τ_4	6
a_7	1100	0000010	τ_6	7
a_8	1000	1000000	τ_1	8
a_9	1101	1010000	$\tau_1\tau_3$	9
a_{10}	1011	1100000	$\tau_1\tau_2$	10
a_{11}	1110	1110000	$\tau_1\tau_2\tau_3$	11
a_{12}	0011	0001100	$\tau_4\tau_5$	12
a_{13}	1010	0000011	$\tau_6\tau_7$	13

формування СБФ (12). Для розглядуваного прикладу всі рівняння СБФ (12) мають однаковий вигляд: $D_r = D_r^E \vee D_r^1 \vee D_r^2$ ($r=1, R$). Тут верхній індекс показує, яким блоком формується значення відповідної ФЗП D .

Таблиця блока LUTerT має стовпчики: $a_m, K(a_m), C(a_m), \tau_m, m$. У стовпчику τ_m записуються змінні, що відповідають 1 у коді. Для розглядуваного прикладу блок LUTerT представлено у табл. 6.

Для заповнення табл. 6 використано коди з рис. 9 та табл. 1. Система (13) може бути оптимізована з урахуванням несуттєвих наборів, позначених знаком «*» у карті Карно (рис. 9). Наприклад, такі рівняння можна отримати з табл. 6:

$$\begin{aligned} \tau_1 &= A_8 \vee A_9 \vee A_{10} \vee A_{11} = \\ &= \overline{T_1 T_2 T_3 T_4} \vee \dots \vee T_1 T_2 T_3; \end{aligned}$$

$$\begin{aligned} \tau_2 &= A_4 \vee A_5 \vee A_{10} \vee A_{11} = \\ &= T_2 T_3 \vee T_1 \overline{T_2 T_4}. \end{aligned}$$

Решта рівнянь формуються аналогічно. Значимо, що у разі використання елементів LUT мінімізація ДНФ має сенс, якщо деяка змінна виключається з усіх термів функції [32].

Останній крок запропонованого методу синтезу призначений для розв'язання низки складних задач, таких як розміщення та трасування, формування конфігураційних таблиць для кожного елемента схеми та між'єднань, моделювання та визначення фізичних характеристик схеми МПА. Для розв'язання цих задач використовують стандартні САПР, такі як Vivado [33] або Quartus [34]. Цей етап у статті не розглядається.

ВИСНОВКИ

Для реалізації схем МПА у базисі FPGA необхідно розв'язувати низку оптимізаційних задач. Ідеальна схема має мінімально можливу кількість еле-

ментів, максимальну частоту та мінімальне споживання електричної потужності. Однак насправді практично неможливо досягти оптимумів для цих трьох характеристик одночасно. Наприклад, зменшення апаратних витрат зазвичай зумовлює зменшення частоти функціонування (збільшення часу такту) схеми МПА [16, 32].

Під час використання блоків пам'яті ЕМВ можна отримати схеми МПА з параметрами, близькими до оптимальних [5]. Проте, блоки ЕМВ широко використовують для реалізації операційних блоків цифрових систем [10, 14]. Це призводить до ситуації, коли ресурсів «вільних» блоків пам'яті не вистачає для реалізації схеми МПА. Тоді схема МПА представляється як мережа блоків ЕМВ і елементів LUT [35]. Подібна ситуація розглянута у цій статті.

Відомо, що зменшення апаратних витрат можна досягти, використовуючи методи структурної декомпозиції схем МПА [17, 32]. У цій роботі пропонується використовувати подвійне кодування станів [19, 20] для зменшення кількості елементів LUT у схемі автомата Мура. При цьому частина схеми реалізується з допомогою блока ЕМВ.

За виконання умов (10), (14) та (15) схема МПА гарантовано має три рівні логічних елементів. У першому рівні реалізуються часткові ДНФ функцій збудження пам'яті. На другому рівні ці часткові ДНФ перетворюються на внутрішні змінні, що кодують стан МПА. На третьому рівні формуються виходи МПА і додаткові змінні, які кодують стани МПА і відіграють роль зворотного зв'язку з елементами першого рівня.

У роботі розглядається чотири підходи до реалізації МПА Мура у вигляді композиції ЕМВ та елементів LUT. Дослідження, проведені з використанням пакету Vivado [33] та бібліотеки [26], дали змогу порівняти характеристики схем для цих чотирьох підходів.

Як показали дослідження, запропонований підхід (автомат Мура P_2Y) дає змогу в середньому на 18 % покращити апаратні витрати. При цьому час такту для всіх варіантів схеми майже однаковий. Виграш від застосування цього підходу збільшується зі зростанням показника $L + R$. Зазначимо, що для 67 % стандартних прикладів із бібліотеки [26] достатньо одного блока ЕМВ. Тому дослідження проводилися на 33 % автоматів [26].

Як показав аналіз результатів досліджень, лише 30 % переходів МПА реалізуються блоком ЕМВ. Це призводить до складних схем блока LUTerФ, який реалізує 70 % переходів, що залишилися. У результаті ці схеми мають більше елементів та їхніх рівнів, ніж схеми еквівалентних автоматів P_2Y .

Подальші напрямки досліджень пов'язані із застосуванням запропонованого підходу для реалізації автоматів на лічильниках [36, 37] та для синтезу нейронподібних мереж [38]. Крім того, планується розробити методи оптимізації автоматів Мура з подвійним кодуванням станів, які враховують наявність класів псевдоеквівалентних станів [32].

СПИСОК ЛІТЕРАТУРИ

1. Skliarova I., Sklyarov V., Sudnitson A. Design of FPGA-based circuits using hierarchical finite state machines. Tallinn: TUT Press, 2012. 240 p.
2. Czerwinski R., Kania D. Finite state machines logic synthesis for complex programmable logic devices. Berlin: Springer, 2013. 172 p.
3. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. Москва: Горячая линия — ТЕЛЕКОМ, 2001. 636 с.

4. Tiwari A., Tomko K. Saving power by mapping finite state machines into embedded memory blocks in FPGAs. *Proc. Design, Automation and Test in Europe Conference and Exhibition (Paris, France, 6–20 Feb., 2004)*. 2004. Vol. 2. P. 916–921.
5. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and optimization of FPGA-based systems. Berlin: Springer, 2014. 432 p.
6. Grout I. Digital systems design with FPGAs and CPLDs. Amsterdam: Elsevier, 2008. 784 p.
7. Maxfield C. The design warrior's guide to FPGAs. Orlando: Academic Press, 2004. 542 p.
8. Kuon I., Tessier R., Rose J. FPGA architecture: Survey and challenges. *Foundations and Trends in Electronic Design Automation*. 2008. Vol. 2, N 2. P. 135–253.
9. Sass R., Schmidt A. Embedded system design with platform FPGAs: Principles and practices. Amsterdam: Morgan Kaufmann Publishers, 2010. 409 p.
10. Ruiz-Rosero J., Ramirez-Gonzalez G., Khanna R. Field programmable gate array applications — a scientometric review. *Computation*. 2019. Vol. 7(3). P. 63.
11. UG473 (v1.14) July 3, 2019. URL: www.xilinx.com.
12. Rafla N.I., Gauba I. A reconfigurable pattern matching hardware implementation using on-chip ram-based FSM. *53rd IEEE International Midwest Symposium on Circuits and Systems*. 2010. P. 49–52.
13. Sklyarov V. Synthesis and implementation of RAM-based finite states machines in FPGAs. *Proc. of Field-Programmable Logic and Applications: The Roadmap to Reconfigurable Computing*. Villach: Springer-Verlag, 2000. P. 718–727.
14. Senhaji-Navarro R., Garcia-Vargas I., Jimenes-Moreno G., Civit-Balcells A., Guerra-Gutierrez P. ROM-based FSM implementation using input multiplexing in FPGA devices. *Electronics Letters*. 2004. Vol. 40, N 20. P. 1249–1251.
15. Senhaji-Navarro R., Garcia-Vargas I., Guisado L.J. Performance evaluation of RAM-based implementation of finite states machines in FPGAs. *19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012)*. (Seville, Spain, Dec., 2012). 2012. P. 225–228.
16. Kubica M., Opara A., Kania D. Technology mapping for LUT- based. FPGA. Berlin: Springer, 2021. P. 216.
17. Barkalov O., Titarenko L., Barkalov Jr. A structural decomposition as a tool for the optimization of an FPGA-based implementation of a Mealy FSM. *Cybernetics and Systems Analysis*. 2012. Vol. 48, N 2. P. 313–322.
18. Barkalov A.A., Titarenko L.A., Baiev A.V., Matvienko A.V. Joint use of methods of structural decomposition for optimizing the circuit of moore FSM. *Cybernetics and Systems Analysis*. 2021. Vol. 57, N 2. P. 173–184.
19. Barkalov A., Titarenko L., Mielcarek K. Hardware reduction for LUT-based Mealy FSMs. *International Journal of Applied Mathematics and Computer Science*. 2018. Vol. 28, N 3. P. 595–607.
20. Barkalov A., Titarenko L., Mielcarek K. Improving characteristics of LUT-based Mealy FSMs. *International Journal of Applied Mathematics and Computer Science*. 2020. Vol. 30, N 3. P. 745–759.
21. Baranov S. Logic synthesis for control automata. Dordrecht: Kluwer Academic Publishers, 1994. 312 p.
22. DeMicheli G. Synthesis and optimization of digital circuits. New York: McGraw-Hill, 1994. 576 p.
23. Глушков В.М. Синтез цифровых автоматов. Москва: Физматгиз, 1962. 476 с.
24. Intel® FPGAs and Programmable Devices. URL: <https://www.intel.com/content/www/us/en/products/programmable.html>.
25. VC709 Evaluation Board for the Virtex-7 FPGA. User Guide; UG887 (v1.6); Xilinx, Inc.: San Jose, CA, USA, 2019.

26. Yang S. Logic synthesis and optimization benchmarks user guide. Version 3.0. Techn. Rep. Microelectronics Center of North Carolina, 1991. 43 p.
27. Rawski M., Selvaraj H., Luba T. An application of functional decomposition in ROM-based FSM implementation in FPGA devices. *Journal of System Architecture*. 2005. Vol. 51, N 6–7. P. 424–434.
28. Kubica M., Kania D. Area-oriented technology mapping for LUT-based logic blocks. *International Journal of Applied Mathematics and Computer Science*. 2017. Vol. 27, N 1. P. 207–222.
29. Machado L., Cortadella J. Support-reducing decomposition for FPGA mapping. *IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2020. Vol. 39, N 1. P. 213–224.
30. Wilkes M., Stringer J. Microprogramming and the design of the control circuits in electronic digital computer. *Proc. of Cambridge Philosophical Society*. 1953. Vol. 49. P. 230–238.
31. Ачасова С.М. Алгоритмы синтеза автоматов на программируемых матрицах. Москва: Радио и связь, 1987. 136 с.
32. Barkalov A., Titarenko L., Mielcarek K., Chmielewski S. Logic synthesis for FPGA-based control units. Structural Decomposition in Logic Design. *Lecture Notes in Electrical Engineering*. Springer, 2020. Vol. 636. P. 247.
33. Vivado Design Suite. URL: <https://www.xilinx.com/products/design-tools/vivado.html>, 2020, accessed: January, 2020.
34. Quartus II. URL: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html>, accessed: January, 2020.
35. Rawski M., Tomaszewicz P., Borowski G., Luba T. Logic synthesis method of digital circuits designed for implementation with embedded memory blocks on FPGAs. *Design of Digital Systems and Devices. Lecture Notes in Electrical Engineering*. Berlin: Springer, 2011. Vol. 79. P. 121–144.
36. Баркалов А.А., Титаренко Л.А. Преобразование кодов в композиционных микропрограммных устройствах управления. *Кибернетика и системный анализ*. 2011. № 5. С. 107–118.
37. Баркалов А.А., Титаренко Л.А., Ефименко К.Н. Оптимизация схем композиционных микропрограммных устройств управления. *Кибернетика и системный анализ*. 2011. № 1. С. 179–188.
38. Opanasenko V.N., Kryvyi S.L. Synthesis of neural-like networks on the basis of conversion of cyclic Hamming codes. *Cybernetics and Systems Analysis*. 2017. Vol. 53, N 4. P. 627–635. <https://doi.org/10.1007/s10559-017-9965-z>.

A.A. Barkalov, L.A. Titarenko, A.V. Baev, A.V. Matviienko
TWOFOLD STATE ASSIGNMENT FOR MOORE FINITE STATE MACHINES

Abstract. A method is proposed for reducing hardware in circuits of Moore finite-state machines (FSMs) implemented with EMB and LUTs. The method divides a set of states into classes, each corresponding to one block of logical elements. Moreover, each state has two codes. This approach leads to the three-level circuit of the Moore FSM. An example of the Moore FSM synthesis using the proposed method and the application conditions for this method are considered. Studies based on standard benchmark FSMs have shown that the proposed method reduces the hardware amount compared to other known solutions.

Keywords: Moore FSM, synthesis, EMB, LUT, structural decomposition, partition.

Надійшла до редакції 11.01.2022