



УДК 004.274

О.О. БАРКАЛОВ

Університет Зеленогурський, Зелена Гура, Польща, 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*.

О.М. ГОЛОВІН

Інститут кібернетики ім. В.М. Глушкова НАН України, Київ, Україна, e-mail: *o.m.golovin.1@gmail.com*.

О.В. МАТВІЄНКО

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

ОПТИМІЗАЦІЯ СХЕМИ АВТОМАТА МІЛІ У ЗМІШАНОМУ ЕЛЕМЕНТНОМУ БАЗИСІ

Анотація. Запропоновано метод зменшення кількості елементів LUT у схемі автомата Мілі, яка реалізується в базисі FPGA. При цьому частина схеми реалізується за допомогою блока пам'яті ЕМВ. Метод базується на подвійному кодуванні станів і кодуванні наборів мікрооперацій. Наведено приклад синтезу схеми МПА з використанням запропонованого методу. За виконання певних умов схема МПА має точно три рівні логічних елементів. Розглянуто методи поліпшення характеристик схеми, які базуються на оптимальному кодуванні станів і наборів мікрооперацій.

Ключові слова: мікропрограмний автомат Мілі, синтез, FPGA, ЕМВ, LUT, кодування.

ВСТУП

Одним з основних послідовних блоків цифрових систем є пристрій керування (ПК) [1]. Часто закон функціонування ПК визначають за допомогою моделі мікропрограмного автомата (МПА) Мілі [2, 3]. При цьому синтез схеми ПК зводиться до синтезу схеми МПА [4, 5]. Синтезуючи МПА, розробник прагне отримати схему з оптимальними характеристиками: мінімальні витрати апаратури, максимальна операційна частота, мінімальна споживана потужність [6, 7]. Ці показники є взаємозалежними. Наприклад, зменшення апаратних витрат зумовлює зменшення швидкодії та споживаної потужності [1, 8]. У цій роботі запропоновано спосіб зменшення кількості конфігурованих логічних блоків CLB (configurable logic blocks) у схемі МПА Мілі, що реалізується в базисі FPGA (field-programmable logic array) [9, 10].

Упродовж останніх тридцяти років FPGA використовувалися як ефективна платформа для реалізації різних цифрових систем [11]. Наприклад, у [12] розглянуто майже 1700 систем, реалізованих у базисі FPGA. За прогнозами [11, 12], цей базис домінуватиме ще кілька десятиріч у цифровій техніці. Цим і обґрунтовано вибір базису FPGA в описаних далі дослідженнях.

© О.О. Баркалов, Л.О. Титаренко, О.М. Головін, О.В. Матвієнко, 2024

Для реалізації схем МПА можна використовувати два типи CLB: елементи табличного типу LUT (look-up-table) та вбудовані блоки пам'яті EMB (embedded memory blocks) [1, 6]. Блоки CLB реалізують функції алгебри логіки [5], що задають схему МПА. Ці блоки зв'язуються в схему за допомогою програмованих між'єднань. Зазвичай ПК — це синхронні блоки [4], які потребують використання схем синхронізації. Для оптимізації швидкодії та потужності необхідно зменшувати кількість між'єднань та локалізувати синхронну частину схеми МПА [13].

У [13, 14] запропоновано метод подвійного кодування станів МПА Мілі. Цей підхід дає змогу отримати схеми з меншою кількістю елементів LUT, ніж у схем, що базуються на функціональній декомпозиції (ФД) [3]. Потреба у ФД виникає, якщо кількість входів блоків CLB є недостатньою для однорівневої реалізації схеми МПА [9, 10].

Для оптимізації апаратних витрат у цій роботі запропоновано два підходи. Перший — подвійне кодування станів для реалізації схеми МПА у базисі EMB та LUT. Такий базис називається змішаним [1]. Орієнтація на змішаний елементний базис — основна відмінність цього підходу від методів, наведених у [13, 14]. Другий — використання методу подвійного кодування наборів мікрооперацій (НМО) [15].

Обидва підходи є методами структурної декомпозиції (СД) [6] схем МПА. Якщо ФД орієнтована на представлення систем булевих функцій (СБФ) як композиції підфункцій, то СД базується на представленні схеми як композиції блоків з унікальними системами входів і виходів. Для визначення закону функціонування МПА Мілі використовується мова граф-схем алгоритму (ГСА) [4].

РЕАЛІЗАЦІЯ МПА МІЛІ У БАЗИСІ FPGA

Автомат Мілі є шестикомпонентним вектором $S = \langle A, X, Y, \delta, \lambda, a_1 \rangle$, де $A = \{a_1, \dots, a_M\}$ — скінченна множина внутрішніх станів; $X = \{x_1, \dots, x_L\}$ — скінченна множина входів; $Y = \{y_1, \dots, y_N\}$ — скінченна множина виходів; $\delta: A \times X \rightarrow A$ — функція переходів; $\lambda: A \times X \rightarrow Y$ — функція виходів; $a_1 \in A$ — початковий стан [4]. Далі називатимемо входи логічними умовами, а виходи — мікроопераціями (МО).

Для синтезу МПА Мілі за ГСА Г потрібно [4]:

- позначити ГСА Г станами автомата Мілі;
- закодувати стани двійковими кодами $K(a_m)$, що мають R розрядів;
- побудувати пряму структурну таблицю (ПСТ) МПА Мілі;
- визначити системи булевих функцій, що задають схему МПА;
- реалізувати схему МПА у заданому елементному базисі.

Існує чимало методів кодування станів [16]. При цьому число R внутрішніх змінних, що кодують стан, варіюється від $R = M$ (унітарне кодування) до

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

У цій статті розглянуто максимальне кодування станів, яке визначає вираз (1).

Для кодування станів використовують внутрішні $T_r \in T$ змінні, де $|T| = R$. Коди станів зберігаються у регістрі RG, що складається з R тригерів. У разі синтезу МПА у базисі FPGA регістр RG складається з D тригерів [1]. Для запису інформації в RG використовують функції збудження пам'яті, що утворюють множину $\Phi = \{D_1, \dots, D_R\}$. Для установки RG у початковий стан використовують сигнал *Start*, а для запису інформації в RG — імпульс синхронізації *Clock* [4, 5].

Пряма структурна таблиця МПА Мілі має такі стовпці [5]: a_m — початковий 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 — вхідний сигнал, що визначає перехід з a_m в a_s ; Y_h — НМО, які формуються на переході $\langle a_m, a_s \rangle$; Φ_h — набір функцій збудження пам'яті (ФЗП) для запису коду $K(a_s)$ у RG; h — номер переходу. Вхідний сигнал X_h — це кон'юнкція логічних умов (або їхніх інверсій), що визначають перехід $\langle a_m, a_s \rangle$. Отже, ПСТ містить у собі множини A, X, Y і стан $a_1 \in A$. Рядок h визначає функції δ і λ : $a_s = \delta(a_m, X_h)$ і $Y_h = \lambda(a_m, X_h)$. Таким чином, ПСТ визначає всі компоненти вектора $S = \langle A, X, Y, \delta, \lambda, a_1 \rangle$.

Використовуючи ПСТ, можна знайти СБФ [4]:

$$\Phi = \Phi(T, X), \quad (2)$$

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

Системи (2), (3) є диз'юнктивними нормальними формами (ДНФ), що залежать від термів $F_h = A_m X_h$ ($h=1, H$). Тут A_m — кон'юнкція внутрішніх змінних, що відповідає коду вихідного стану $a_m \in A$ рядка $h \in \{1, \dots, H\}$ ПСТ.

Як впливає з аналізу функцій (2), (3), вони можуть залежати від R до $L + R$ змінних. Для автоматів середньої складності [4] маємо $L \approx 30$, $N \approx 50$, $R \approx 8$, $H \approx 2000$. Таким чином, ДНФ будь-якої з функцій (2), (3) може мати до 40 літералів [4].

У цій роботі розглянуто реалізацію СБФ (2), (3) у базисі FPGA. Нині провідним виробником мікросхем FPGA є фірма Xilinx [17]. Тому запропонований метод зорієнтовано на мікросхеми цієї фірми та використано термінологію з документації Xilinx.

Основним блоком CLB є логічний елемент, що складається з елемента LUT та тригера, тип входів якого можна програмувати [18]. Елемент LUT складається з комірок статичної пам'яті; кожна комірка зберігає 1 біт інформації. Конкретну комірку вибирають, використовуючи вхідну комбінацію, що має S_L змінних. Таким чином, елемент LUT може реалізувати довільну логічну функцію, яка залежить не більше ніж від S_L аргументів.

Тригери логічного елемента мають входи синхронізації, скидання та установки в одиницю [9, 10]. Отже, логічні елементи являють собою структурно повний базис [19]. На виході логічного елемента маємо або вихід елемента LUT, або вихід тригера.

Позначимо терміном LUTer блок, що складається із логічних елементів мікросхеми FPGA. Для реалізації СБФ (2), (3) потрібно два блоки (модель U_1 , рис. 1).

У МПА U_1 регістр RG складається з R тригерів, розподілених між елементами блока LUTerФ. Сигнал *Start* задає нульовий код в RG, завдяки чому автомат встановлюється в початковий стан $a_1 \in A$. Імпульси *Clock* надходять на входи синхронізації тригерів та визначають хід автоматного часу [4]. Отже, в МПА U_1 блок LUTer реалізує СБФ (2), а LUTerY — СБФ (3).

Системи (2), (3) можна представити у вигляді таблиці істинності [5], що має H_0 рядків, $H_0 = 2^{R+L}$, та має $L + R$ вхідних та $N + R$ вихідних стовпців. Для реалізації таких таблиць доцільно використовувати блоки ЕМВ.

Блок ЕМВ має S_A входів та t_F виходів, що визначає ємність блока V_0 ($V_0 = 2^{S_A \times t_F}$) [9, 10]. Пара $\langle S_A, t_F \rangle$ визначає конфігурацію блока ЕМВ. Наприклад, для ЕМВ сім'ї Virtex-7 [20] існують такі конфігурації: $\langle 15, 1 \rangle$, $\langle 14, 2 \rangle$, ..., $\langle 9, 64 \rangle$.

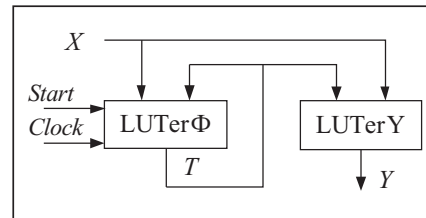


Рис. 1. Структурна схема МПА Мілі U_1

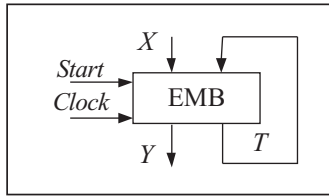


Рис. 2. Структурна схема МПА U_2

Таким чином, ЕМВ можуть «налаштовуватись» за характеристиками МПА.

Нехай серед пар $\langle S_A, t_F \rangle$ існує пара $\langle S_A^0, t_F^0 \rangle$, для якої виконуються такі умови:

$$S_A^0 \geq L + R, \quad (4)$$

$$t_F^0 \geq N + R. \quad (5)$$

У цьому випадку схема МПА представляється моделлю U_2 (рис. 2).

Схема U_2 складається із одного блока ЕМВ. Для FPGA фірми Xilinx ЕМВ є синхронним блоком з можливістю скидання вихідного регістра [21]. Цим пояснюється відсутність регістра RG у схемі (див. рис. 2). Імпульси *Start* і *Clock* надходять безпосередньо до ЕМВ. Цей підхід можна використовувати лише для простих автоматів.

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

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

то схема МПА U_1 складається із $N + R$ елементів LUT. В іншому випадку схема є багаторівневою. Якщо умови (4), (5) порушуються, то МПА U_2 є мережею блоків ЕМВ. Якщо умови (4)–(6) порушуються, виникає проблема оптимізації схем МПА U_1 і U_2 .

Розглянемо деякі методи оптимізації.

ОПТИМІЗАЦІЯ СХЕМ АВТОМАТІВ МПА МЛІ

Якщо схема МПА U_1 є багаторівневою, зменшення кількості елементів LUT може бути досягнуто за рахунок:

- оптимального кодування станів [1, 4, 5];
- оптимальної ФД [3];
- реалізації частини схеми автомата блоками ЕМВ [8, 22];
- СД схеми [6, 15].

Кодування називається оптимальним, якщо воно зменшує кількість літералів у ДНФ функцій (2), (3). Наприклад, за унітарного кодування станів ($R = M$) кон'юнкції A_m мають по одному літералу [5]. Іншим прикладом є алгоритм JEDI [23], у якому величина R визначається виразом (1). Існує чимало методів кодування станів [1, 4, 5]. Проте ступінь зменшення кількості літералів багато в чому залежить від особливостей ГСА [6]. Тож кодування станів є допоміжним засобом оптимізації схем.

Функціональна декомпозиція зводиться до розбиття функції f_i на підфункції з меншою кількістю літералів. Цей процес припиняється, якщо для всіх підфункцій кількість літералів не перевищує S_L . Функціональна декомпозиція вважається оптимальною, якщо за її застосування маємо схему з кращими характеристиками, ніж у схем, отриманих за допомогою інших методів ФД [3]. Методи ФД використовуються у всіх САПР, орієнтованих на FPGA. Прикладом таких САПР є Vivado [23] та Quartus [24].

Природним недоліком ФД є непередбачувана кількість рівнів логіки у схемі та складна система міжз'єднань [3, 6]. Схеми, що базуються на ФД, є достатньо повільними і споживають багато додаткової енергії через складну систему міжз'єднань [26].

Зменшення площі кристала, що займає схема МПА, можна досягти за рахунок заміни частини елементів LUT блоками ЕМВ. У МПА U_3 блок ЕМВ реалізує лише частину функцій (2), (3). Позначення ЕМВ означає, що схема складається із блоків ЕМВ.

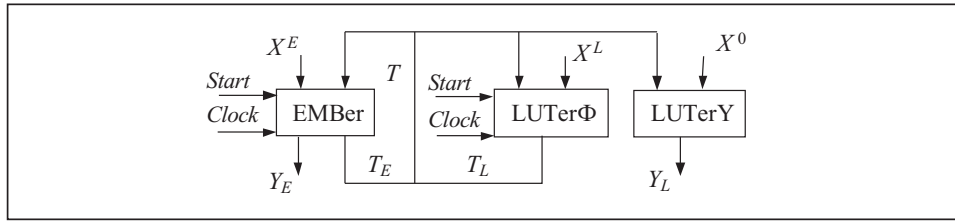


Рис. 3. Структурна схема МПА U_3

На рис. 3 індекс E означає, що множина є входом або виходом блока EMBer. Аналогічно індекс L показує належність множини одному з блоків LUTer. Для множин на рис. 3 виконуються такі умови: $X = X^E \cup X^L \cup X^0$, $Y = Y_E \cup Y_L$, $Y_E \cap Y_L \neq \emptyset$, $T = T_E \cup T_L$, $T_E \cap T_L \neq \emptyset$. Зазначимо, що множини X^E , X^L , X^0 можуть мати спільні частини. Вочевидь, кожен із блоків (див. рис. 3) реалізує деяку підсистему функцій (2), (3).

Для зменшення апаратних витрат потрібно якомога більшу частину схеми U_3 реалізувати на блоках ЕМВ. Проте це не завжди можливо, оскільки блоки ЕМВ широко використовуються для реалізації операційних блоків цифрових систем [16]. Тому розробник схеми ПК може мати недостатню кількість таких блоків. У цій роботі розглянуто крайній випадок, коли для реалізації частини схеми МПА можна використовувати лише один блок ЕМВ.

Очевидно, блоки LUTer і LUTerY можуть бути багаторівневими. Якщо умова (6) порушується для функцій $f_i \in \Phi_L \cup Y_L$, то для синтезу схеми потрібно використовувати методи ФД.

Структурна декомпозиція зводиться до представлення схеми МПА у вигляді композиції кількох логічних блоків [6]. Кожен блок має унікальну систему входів та виходів і реалізує системи функцій, відмінні від (2), (3).

Одним із методів СД є метод кодування НМО [1]. При цьому кожному набору $Y_q \subseteq Y$ ставиться у відповідність двійковий код $K(Y_q)$. Якщо в операторних вершинах ГСА Γ записані Q наборів, то розрядність $K(Y_q)$ визначають як

$$R_Q = \lceil \log_2 Q \rceil. \quad (7)$$

Для кодування НМО використовують елементи множини $Z = \{z_1, \dots, z_{R_Q}\}$. У цьому випадку систему (3) можна замінити такими СБФ:

$$Z = Z(T, X), \quad (8)$$

$$Y = Y(Z). \quad (9)$$

Цей підхід зумовлює побудову МПА U_4 , схема якого може бути реалізована на елементах LUT, блоках ЕМВ або у змішаному базисі. Узагальнену структурну схему МПА U_4 показано на рис. 4.

У МПА U_4 блок ФЗП реалізує СБФ (2), блок Z — систему (8), блок Y — систему (9). Недоліком такого підходу є відсутність гарантії, що всі блоки представлені однорівневими схемами. Проте система (9) залежить лише від R_Q змінних. Тому схема блока LUTerY U_4 набагато простіша цієї схеми в еквівалентному МПА U_1 .

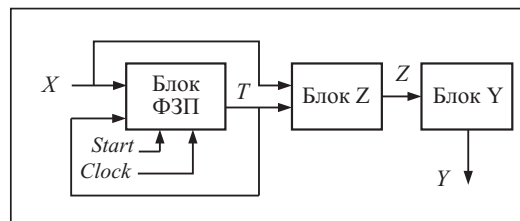


Рис. 4. Структурна схема МПА U_4

У цій роботі запропоновано метод оптимізації МПА Мілі, що реалізується в базисі блоків ЕМВ та елементів LUT. Розглянуто крайній випадок, коли лише один блок ЕМВ може бути використаний. Запропонований метод базується на спільному використанні методів подвійного кодування станів [13, 14] та кодування НМО $Y_q \subseteq Y$. Позначимо МПА, що базується на цих методах, U_5 .

ОСНОВНА ІДЕЯ ЗАПРОПОНОВАНОГО МЕТОДУ

Нехай для деякої ГСА Γ знайдено множини A і $Y_q \subseteq Y$. Закодуємо НМО $Y_q \subseteq Y$ кодами, що мають R_Q розрядів, а стани $a_m \in A$ — двійковими кодами $K(a_m)$. Побудуємо ПСТ МПА U_5 зі стовпцями a_m , $K(a_m)$, a_s , $K(a_s)$, X_h , Z_h , Φ_h , h . Стовпець Z_h містить змінні $z_r \in Z$, що відповідають одиницям у коді набору, записаного в стовпці Y_h ПСТ еквівалентного МПА U_1 .

Знайдемо розбиття $\Pi_A = \{A_E, A_L\}$ множини станів A . Кожен клас Π_A визначає множини X^i , Z^i , Φ^i , де $i \in \{E, L\}$. Наприклад, множина $X^E \subseteq X$ складається з логічних умов, що визначають переходи зі станів $a_m \in A^E$. Аналогічно визначається решта множин $(X^E, Z^E, \Phi^E, X^L, Z^L, \Phi^L)$.

Клас A_E визначає СБФ

$$\Phi^E = \Phi^E(X^E, \tau^E), \quad (10)$$

$$Z^E = Z^E(X^E, \tau^E). \quad (11)$$

Множина τ^E складається з R_E внутрішніх змінних, що кодують стани $a_m \in A_E$. Якщо $|A_E| = M_E$, то

$$R_E = \lceil \log_2(M_E + 1) \rceil. \quad (12)$$

Одиниця в (12) додається для врахування співвідношення $a_m \notin A_E$.

Переходи зі станів $a_m \in A_E$ реалізуються на блоці ЕМВ. Нехай $|L^E| = L^0$, $|\Phi^E| = R^0$ і $|Z^E| = R_Q^0$. У цьому випадку для реалізації систем (10), (11) потрібна конфігурація $\langle S_A^0, t_F^0 \rangle$, для якої виконуються умови

$$S_A^0 = L^0 + R_E, \quad t_F^0 \geq R^0 + R_Q^0. \quad (13)$$

Знайдемо розбиття $\Pi_B = \{A^1, \dots, A^K\}$ множини $A_L \in \Pi_A$. Нехай переходи зі станів $a_m \in A^k$ залежать від L_k логічних умов, що утворюють множину $X^k \subseteq X$. Нехай під час переходів зі стану $a_m \in A$ формуються змінні $z_r \in Z^k$, де $Z^k \subseteq Z$, і ФЗП $D_r \in \Phi^k$, де $\Phi^k \subseteq \Phi$. Нехай стани $a_m \in A^k$ закодовані кодами $C(a_m)$, що мають R_k біт:

$$R_k = \lceil \log_2 |A^k| + 1 \rceil \quad (k = \overline{1, K}). \quad (14)$$

Розбиття Π_B формуються так, що виконується умова

$$L_k + R_k \leq S_L \quad (k = \overline{1, K}). \quad (15)$$

Тоді для реалізації кожної функції з множин Z^k та Φ^k достатньо одного елемента LUT. Ці елементи утворюють блок LUTerk, що реалізує системи

$$\Phi^k = \Phi^k(X^k, \tau^k), \quad (16)$$

$$Z^k = Z^k(X^k, \tau^k). \quad (17)$$

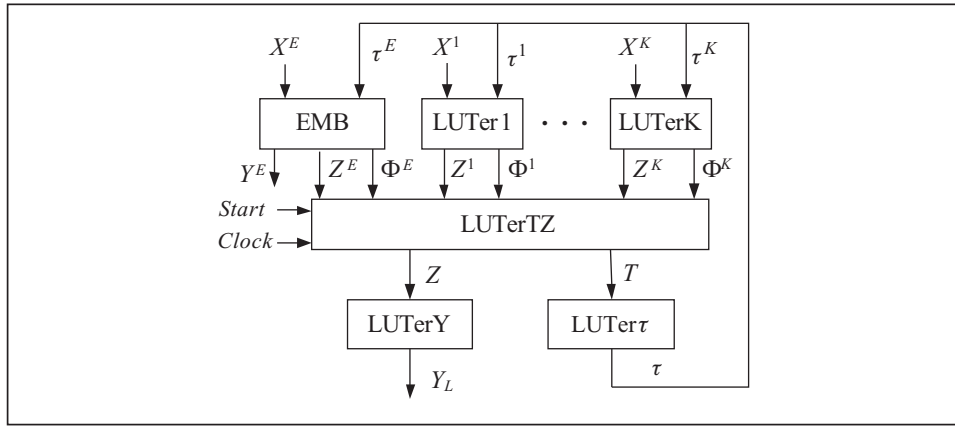


Рис. 5. Структурна схема МПА Мілі U_5

Нехай ЕМВ має Δ_t вільних виходів, $\Delta_t = t_F^0 - (R^0 + R_Q^0)$. Якщо серед МО $y_n \in Y$ є такі, що входять до складу лише НМО, які формуються тільки на переходах зі станів $a_m \in A_E$, то на виході ЕМВ можна сформувати не більше ніж Δ_t таких мікрооперацій. Ці МО утворюють множину $Y_E \subseteq Y$. Тоді залишається реалізувати схеми для інших МО, що утворюють множину $Y_L = Y \setminus Y_E$.

На базі цих положень пропонуємо модель МПА Мілі U_5 (рис. 5). Схема складається з блока ЕМВ та трьох рівнів елементів LUT.

У МПА U_5 блок ЕМВ реалізує СБФ (10), (11) та $Y_E = Y_E(\tau^E, X^E)$. Блоки LUTer1–LUTerK реалізують СБФ (16), (17). Блок LUTerZ реалізує СБФ

$$Z = Z(Z^E, Z^1, \dots, Z^K), \quad (18)$$

$$\Phi = \Phi(\Phi^E, \Phi^1, \dots, \Phi^K). \quad (19)$$

Блок LUTerY реалізує СБФ

$$Y_L = Y_L(Z). \quad (20)$$

Блок LUTer τ виконує перетворення $K(a_m) \rightarrow C(a_m)$, та реалізує СБФ

$$\tau = \tau(T). \quad (21)$$

У цій роботі запропоновано метод синтезу МПА U_5 за ГСА Г, який включає такі кроки:

- 1) формування множини A за ГСА Г;
- 2) кодування станів $a_m \in A$ кодами $K(a_m)$;
- 3) формування НМО $Y_E \in Y$ та кодування їх кодами $K(Y_q)$;
- 4) формування ПСТ автомата U_5 ;
- 5) формування класу A_E та кодування станів $a_m \in A_E$ кодами $C(a_m)$;
- 6) формування множин мікрооперацій Y_E та Y_L ;
- 7) формування таблиці блока ЕМВ;
- 8) формування таблиць блоків LUTer k та систем (16), (17);
- 9) формування таблиці блока LUTerZ та СБФ (18), (19);
- 10) формування СБФ (20);
- 11) формування таблиці блока LUTer τ та СБФ (21);
- 12) реалізація МПА у заданому елементному базисі.

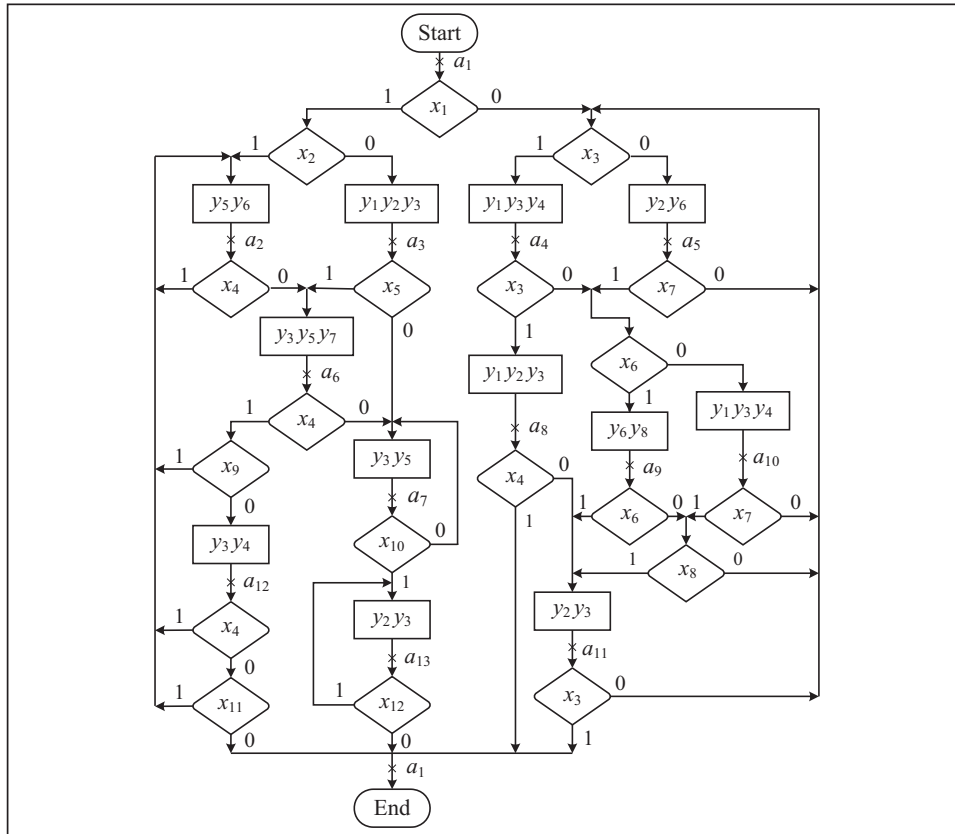


Рис. 6. ГСА Γ_1 , позначена станами автомата Мілі

Вираз $U_i(\Gamma_j)$ означає, що МПА U_i синтезується за ГСА Γ_j . Розглянемо приклад синтезу МПА $U_5(\Gamma_1)$. На рис. 6 ГСА Γ_1 позначена станами автомата Мілі за правилами, наведеними в [4].

ПРИКЛАД СИНТЕЗУ СХЕМИ МПА З ВИКОРИСТАННЯМ ЗАПРОПОНОВАНОГО МЕТОДУ

Аналізуючи ГСА Γ_1 , можна знайти множини $A = \{a_1, \dots, a_{13}\}$, $X = \{x_1, \dots, x_{12}\}$, $Y = \{y_1, \dots, y_8\}$, що дає параметри $M=14$, $L=12$, $N=8$. Використовуючи (1), отримуємо $R=4$, $T = \{T_1, \dots, T_4\}$ і $\Phi = \{D_1, \dots, D_4\}$. Закодуємо стани кодами тривіально: $K(a_1) = 0000$, $K(a_2) = 0001$, ..., $K(a_{13}) = 1100$.

Аналіз операторних вершин ГСА Γ_1 дає змогу знайти такі НМО: $Y_1 = \{y_1, y_2, y_3\}$, $Y_2 = \{y_1, y_3, y_4\}$, $Y_3 = \{y_5, y_6\}$, $Y_4 = \{y_2, y_6\}$, $Y_5 = \{y_3, y_5, y_7\}$, $Y_6 = \{y_3, y_5\}$, $Y_7 = \{y_6, y_8\}$, $Y_8 = \{y_3, y_4\}$, $Y_9 = \{y_2, y_3\}$ і $Y_{10} = \emptyset$. Отже, $Q=10$. Використовуючи (7), отримуємо $R_Q=4$, $Z = \{z_1, \dots, z_4\}$. Закодуємо НМО тривіально: $K(Y_1) = 0001$, $K(Y_2) = 0010$, ..., $K(Y_9) = 1001$, $K(Y_{10}) = 0000$.

Побудуємо ПСТ автомата $U_5(\Gamma_1)$. Для формування вмісту стовпця Z_h знаходимо набір $Y_q \subseteq Y$ із відповідної операторної вершини. Потім у рядку h цього стовпця записуються змінні $z_r \in Z$, що дорівнюють 1 у коді $K(Y_q)$. У такий спосіб формується табл. 1.

Нехай є блок ЕМВ з конфігураціями $\langle 12, 1 \rangle$, $\langle 11, 2 \rangle$, $\langle 10, 4 \rangle$, $\langle 9, 8 \rangle$, $\langle 8, 16 \rangle$. Виберемо конфігурацію, достатню для реалізації $R + R_Q = 8$ функцій. Вибираємо конфігурацію $\langle 9, 8 \rangle$. Маємо $S_A^0 = 9$, $t_F^0 = 8$. Отже, для класу A_E має виконуватись умова $L_0 + R_E = 9$.

Таблиця 1. ПСТ автомата $U_5(\Gamma_1)$

a_m	$C(a_m)$	a_s	$K(a_s)$	X_h	Z_h	Φ_h	h
a_1	0000	a_2	0001	x_1x_2	z_3z_4	D_4	1
		a_3	0010	$\overline{x_1x_2}$	z_4	D_3	2
		a_4	0011	$\overline{x_1x_3}$	z_3	D_3D_4	3
		a_5	0100	$\overline{x_1x_3}$	z_2	D_2	4
a_2	0001	a_2	0001	x_4	z_3z_4	D_4	5
		a_6	0101	$\overline{x_4}$	z_2z_4	D_2D_4	6
a_3	0010	a_6	0101	x_5	z_2z_4	D_2D_4	7
		a_7	0110	$\overline{x_5}$	z_2z_3	D_2D_3	8
a_4	0011	a_8	0111	x_3	z_4	$D_2D_3D_4$	9
		a_9	1000	$\overline{x_3x_6}$	$z_2z_3z_4$	D_1	10
		a_5	0100	$\overline{x_3x_6}$	z_2	D_2	11
a_5	0100	a_9	1000	x_7x_6	$z_2z_3z_4$	D_1	12
		a_{10}	1001	$\overline{x_7x_6}$	z_3	D_1D_4	13
		a_5	0100	$\overline{x_7}$	z_2	D_2	14
a_6	0101	a_2	0001	x_4x_9	z_3z_4	D_4	15
		a_{12}	1011	$\overline{x_4x_9}$	z_1	$D_1D_3D_4$	16
		a_7	0110	$\overline{x_4}$	z_2z_3	D_2D_3	17
a_7	0110	a_{13}	1100	x_{10}	z_1z_4	D_1D_2	18
		a_7	0110	$\overline{x_{10}}$	z_2z_3	D_2D_3	19
a_8	0111	a_1	0000	x_4	—	—	20
		a_{11}	1010	$\overline{x_4}$	z_2z_3	D_1D_3	21
a_9	1000	a_{11}	1010	x_6	z_2z_3	D_1D_3	22
		a_{11}	1010	$\overline{x_6x_8}$	z_2z_3	D_1D_3	23
		a_5	0100	$\overline{x_6x_8}$	z_2	D_2	24
a_{10}	1001	a_{11}	1010	x_7x_8	z_2z_3	D_1D_3	25
		a_5	0100	$\overline{x_7x_8}$	z_2	D_2	26
		a_5	0100	$\overline{x_7}$	z_2	D_2	27
a_{11}	1010	a_1	0000	x_3	—	—	28
		a_5	0100	$\overline{x_3}$	z_2	D_2	29
a_{12}	1011	a_2	0001	x_4	z_3z_4	D_4	30
		a_2	0001	$\overline{x_4x_{11}}$	z_3z_4	D_4	31
		a_1	0000	$\overline{x_4x_{11}}$	—	—	32
a_{13}	1100	a_{13}	1100	x_{12}	z_1z_4	D_1D_2	33
		a_1	0000	$\overline{x_{12}}$	—	—	34

Знайдемо клас A_E , що містить максимально можливу кількість станів. У розглядуваному випадку це клас з $M_E = 7$ і $R_E = 3$. Аналіз ПСТ (див. табл. 1) дає змогу знайти множину $X_E = \{x_1, x_3, x_6, x_7, x_8\}$ з $L^0 = 6$. Отже, $L^0 + R_E =$

		$z_1 z_2$			
		00	01	11	10
z_3	0	Y_8	Y_6	Y_4	Y_2
	1	Y_1	Y_7	Y_3	Y_5

Рис. 7. Коды НМО автомата $U_5(\Gamma_1)$

$= 3 + 6 = 9$ і умова (13) виконується. Тепер можна знайти клас $A_L = A \setminus A_E = \{a_2, a_3, a_6, a_7, a_{12}, a_{13}\}$.

З (12) маємо $R_E = 3$ та $\tau^E = \{\tau_1, \tau_2, \tau_3\}$.

Закодуємо стани $a_m \in A_E$ у такий спосіб: $C(a_1) = 001$, $C(a_4) = 010$, $C(a_5) = 011$, $C(a_8) = 100$, $C(a_9) = 101$, $C(a_{10}) = 110$, $C(a_{11}) = 111$. Зазначимо, що коди $C(a_m)$ не

впливають на складність схеми МПА.

Аналіз ГСА Γ_1 та класів A_E і A_L показує, що МО $y_1 \in Y$ формується тільки під час переходів із станів $a_m \in A_E$. Якщо виключити y_1 з наборів $Y_1 - Y_9$, то отримаємо такі рівняння: $Y_1 \setminus \{y_1\} = Y_9$ та $Y_2 \setminus \{y_1\} = Y_8$. Отже, число НМО зменшується до $Q = 8$, що дає $R_Q = 3$. Таким чином, формування множин $Y_E = \{y_1\}$ та $Y_L = Y \setminus \{y_1\}$ доцільне.

Перерахуємо нові НМО: $Y_1 = \{y_2, y_3\}$, $Y_2 = \{y_3, y_4\}$, $Y_3 = \{y_5, y_6\}$, $Y_4 = \{y_2, y_6\}$, $Y_5 = \{y_3, y_5, y_7\}$, $Y_6 = \{y_3, y_5\}$, $Y_7 = \{y_6, y_8\}$, $Y_8 = \emptyset$.

Закодуємо НМО так, щоб зменшити кількість літералів у СБФ (20). Такі методи використовують результати із [27, 28]. Один із можливих варіантів кодування НМО показано на рис. 7.

Система $Y_L = Y_L(Z)$ у цьому прикладі має такий вигляд:

$$\begin{aligned} y_2 &= Y_1, \quad y_3 = Y_1 \vee Y_2 \vee Y_5 \vee Y_6, \quad y_4 = Y_2 \vee Y_4 = \overline{z_1 z_3}, \\ y_5 &= Y_3 \vee Y_5 = z_1 z_3, \quad y_6 = Y_3 \vee Y_4 \vee Y_6 \vee Y_7 = z_2, \quad y_7 = Y_5, \quad y_8 = Y_7. \end{aligned} \quad (22)$$

Як впливає з цього етапу, виключення y_1 з $Y_1 - Y_9$ дало змогу зменшити число елементів LUT у блоках LUTerK (оскільки $h = 2$, а не 4). Крім того, в блоці LUTerY відсутній елемент LUT, що формує мікрооперацію $y_1 \in Y$.

Із СБФ (22) впливає, що МО y_6 формується як вихід блока LUTerTZ. При цьому блок LUTerY містить $N - 2 = 6$ елементів LUT. Очевидно, початкова ПСТ (див. табл. 1) має бути змінена: з'явиться стовпець Y_E , а в стовпці Z_h мають бути тільки $R_Q = 3$ змінні $z_r \in Z$.

Для формування таблиць блоків LUTerK потрібно знайти розбиття Π_B на множини A_L , що відповідає умові (15). Використовуючи методи з [13, 14], отримуємо $\Pi_B = \{A^1, A^2\}$, де $A^1 = \{a_2, a_6, a_{12}\}$ та $A^2 = \{a_3, a_7, a_{13}\}$. Множини X_1 , X_2 формуються на підставі табл. 1: $X^1 = \{x_4, x_9, x_{11}\}$ і $X^2 = \{x_5, x_{10}, x_{12}\}$. Для формування множин Z^E, Z^1, Z^2 потрібно перетворити ПСТ автомата $U_5(\Gamma_1)$.

У табл. 2 наведено вміст стовпців Y_E та Z_h перетвореної ПСТ. Пояснимо принцип формування цієї таблиці. Наприклад, у разі $h = 2$ НМО $Y_1 = \{y_1, y_2, y_3\}$ розбивається на y_1 і $Y_1 = \{y_2, y_3\}$. З рис. 7 маємо $K(Y_1) = 001$. Отже, для рядка 2 перетвореної ПСТ маємо $Y_E = y_1$, $Z_h = z_3$. Аналогічно заповнені всі стовпці табл. 2.

Таблиця блока ЕМВ має H_E рядків, $H_E = 2^{L^0 + R_E}$. Кожному стану $a_m \in A_E$ відповідає $H(a_m)$ рядків цієї таблиці, $H(a_m) = 2^{L^0}$. Таблиця має такі стовпці: a_m , $C(a_m)$, X^E (адреса комірки ЕМВ), Y_E , Z^E , Φ^E , h . У цьому прикладі маємо $R_E = 3$, $L^0 = 6$, що дає $H_E = 512$, $H(a_m) = 64$. Перші вісім рядків таблиці блока ЕМВ для переходів зі стану $a_1 \in A_E$ показано у табл. 3.

Таблиця 2. Вміст стовпців Y_E , Z_h

h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Y_E	—	y_1	y_1	—	—	—	—	—	y_1	—	—	—	y_1	—	—	—	—
Z_h	$z_1 z_2 z_3$	z_3	z_1	$z_1 z_2$	$z_1 z_2 z_3$	$z_1 z_3$	$z_1 z_3$	z_2	z_3	$z_2 z_3$	$z_1 z_2$	$z_2 z_3$	z_1	$z_1 z_2$	$z_1 z_2 z_3$	z_1	z_2
h	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Y_E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Z_h	z_3	z_2	—	z_3	z_3	z_3	$z_1 z_2$	z_3	$z_1 z_2$	$z_1 z_2$	—	$z_1 z_2$	$z_1 z_2 z_3$	$z_1 z_2 z_3$	—	z_3	—

Таблиця 3. Частина таблиці блока ЕМВ автомата $U_5(\Gamma_1)$

a_m	$C(a_m)$			X^E						Y^E	Z^E			Φ^E				h
	τ_1	τ_2	τ_3	x_8	x_7	x_6	x_3	x_2	x_1	y_1	z_1	z_2	z_3	D_1	D_2	D_3	D_4	
a_1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	65
a_1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	1	0	66
a_1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	1	0	0	67
a_1	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	1	68
a_1	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0	69
a_1	0	0	1	0	0	0	1	0	1	1	0	0	1	0	0	1	0	70
a_1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	1	71
a_1	0	0	1	0	0	0	1	1	1	0	1	1	1	0	0	0	1	72

Якщо $C(a_m) = 000$, то 64 рядки таблиці мають лише нулі. Тому переходи для $C(a_1) = 000$ починаються з рядка 65. Наприклад, для рядка 65 маємо $x_1 = x_3 = 0$, що відповідає переходу $\langle a_1, a_5 \rangle$. Цей перехід представлено рядком 4 ПСТ (див. табл. 1). Як випливає з табл. 1, $D_1 = D_3 = D_4 = 0$, $D_2 = 1$, що записано в стовпцях D_1 – D_4 табл. 3. У рядку 4 табл. 2 записані $y_1 = 0$, $z_1 = z_2 = 1$, $z_3 = 0$. Ця інформація міститься в стовпцях Y_E та Z^E табл. 3. Інші рядки табл. 3 заповнені аналогічно. Зазначимо, що рядки 65, 67, 69 відповідають рядку 4 ПСТ, рядки 66, 70 — рядку 2, рядки 68, 72 — рядку 1, рядок 71 — рядку 3.

Для формування блоків LUTer1–LUTerK потрібно знайти розбиття Π_B множини A_L . Використовуючи методи з [6], можна знайти $\Pi_B = \{A^1, A^2\}$, де $A^1 = \{a_2, a_6, a_{12}\}$ та $A^2 = \{a_3, a_7, a_{13}\}$. З ПСТ (див. табл. 1) знайдемо множини $X^1 = \{x_4, x_9, x_{11}\}$ і $X^2 = \{x_5, x_{10}, x_{12}\}$. З табл. 2 маємо $Z^1 = Z^2 = \{z_1, z_2, z_3\}$.

Отже, кожен клас $A^k \in \Pi_B$ має три елементи. З (14) маємо $R_1 = R_2 = 2$. Нехай $\tau^1 = \{\tau_4, \tau_5\}$, $\tau^2 = \{\tau_6, \tau_7\}$. Закодуємо стани $a_m \in A^k$ тривіально: $C(a_2) = C(a_3) = 01$, $C(a_6) = C(a_7) = 10$ і $C(a_{12}) = C(a_{13}) = 11$. Використовуючи коди $C(a_m)$, ПСТ (див. табл. 1) та вміст табл. 2, будуємо таблиці блоків LUTer1 (табл. 4) та LUTer2 (табл. 5).

Системи (16), (17) формуються тривіально. Наприклад, з табл. 4 маємо $D_1 = \tau_4 \tau_5 x_4 x_9$. Аналогічно з табл. 5 отримаємо $z_1 = \tau_6 \tau_7 x_5$.

Таблиця 4. Таблиця блока LUTer1 автомата $U_5(\Gamma_1)$

a_m	$C(a_m)$	a_s	$K(a_s)$	X_h	Z_h	Φ_h	h
a_2	01	a_2	0010	x_4	$z_1 z_2 z_3$	D_3	1
		a_6	0101	$\overline{x_4}$	$z_1 z_3$	$D_2 D_4$	2
a_6	10	a_2	0010	$x_4 x_9$	$z_1 z_2 z_3$	D_3	3
		a_{12}	1011	$\overline{x_4 x_9}$	z_1	$D_1 D_3 D_4$	4
		a_7	0110	$\overline{x_4}$	z_2	$D_2 D_3$	5
a_8	11	a_2	0001	x_4	$z_1 z_2 z_3$	D_4	6
		a_2	0001	$\overline{x_4 x_{11}}$	$z_1 z_2 z_3$	D_4	7
		a_1	0000	$\overline{x_4 x_{11}}$	—	—	8

Таблиця 5. Таблиця блока LUTer2 автомата $U_5(\Gamma_1)$

a_m	$C(a_m)$	a_s	$K(a_s)$	X_h	Z_h	Φ_h	h
a_3	01	a_6	0101	x_5	$z_1 z_3$	$D_2 D_4$	1
		a_7	0110	$\overline{x_5}$	z_2	$D_2 D_3$	2
a_7	10	a_{13}	1100	x_{10}	z_3	$D_1 D_2$	3
		a_7	0110	$\overline{x_{10}}$	z_2	$D_2 D_3$	4
a_{13}	11	a_{13}	1100	x_{12}	z_3	$D_1 D_2$	5
		a_1	0000	$\overline{x_{12}}$	—	—	6

Таблиця 6. Таблиця блока LUTer τ $U_5(\Gamma_1)$

a_m	$K(a_m)$	$C(a_m)$	τ_m	m	a_m	$K(a_m)$	$C(a_m)$	τ_m	m
a_1	0000	0010000	$\tau_6 \tau_7$	1	a_8	0111	1000000	τ_1	8
a_2	0001	0000100	τ_5	2	a_9	1000	1010000	$\tau_1 \tau_3$	9
a_3	0010	0000001	τ_7	3	a_{10}	1001	1100000	$\tau_1 \tau_2$	10
a_4	0011	0100000	τ_2	4	a_{11}	1010	1110000	$\tau_1 \tau_2 \tau_3$	11
a_5	0100	0110000	$\tau_2 \tau_3$	5	a_{12}	1011	0001100	$\tau_4 \tau_5$	12
a_6	0101	0001000	τ_4	6	a_{13}	1100	0000011	$\tau_6 \tau_7$	13
a_7	0110	0000010	τ_6	7	—	—	—	—	—

Таблиця блока LUTerTZ формується з урахуванням аналізу таблиць блоків ЕМВ, LUTer1 і LUTer2. У наведеному прикладі кожен із цих блоків формує часткові функції. (Ці таблиці не розглядаються.) Системи (18), (19) формуються тривіально: $D_1 = D_1^1 \vee D_1^2 \vee D_1^E, \dots, Z_3 = z_3^1 \vee z_3^2 \vee z_3^E$. Оскільки $K < S_L = 5$, кожна з функцій (18), (19) реалізується схемою одного елемента LUT.

Система $Y_L = Y_L(Z)$ вже сформована. У наведеному прикладі це СБФ (22).

Таблиця блока LUTer має стовпці a_m , $K(a_m)$, $C(a_m)$, τ_m , m . Таблиця будується на базі кодів $K(a_m)$ та $C(a_m)$. Цей блок представлено табл. 6.

З табл. 6 маємо СБФ (21). Наприклад, можна отримати таку ДНФ: $\tau_3 = [A_1 \vee A_5] \vee [A_5 \vee A_{11}] = \overline{T_1} T_3 T_4 \vee T_1 \overline{T_2} T_4$. Аналогічно формуються інші функції СБФ (21).

На останньому етапі запропонованого методу застосовано стандартні САПР для технологічного відображення схеми МПА у базисі FPGA [9, 10]. Наприклад, для FPGA Xilinx використано пакет Vivado [24]. (У наведеному прикладі цей етап не розглядається.)

АНАЛІЗ ЗАПРОПОНОВАНОГО МЕТОДУ

Наведений метод використовують для розв'язання кількох оптимізаційних задач.

Почнемо з кодування станів $K(a_m)$. Тут можливі два підходи. У першому підході стани кодуються так, щоб зменшити кількість елементів LUT у блоках LUTer1–LUTerK. Це може бути зроблено після формування розбиття Π_B .

Позначимо $A(A^k)$ множину станів переходу зі станів $a_m \in A^k$. Маємо $A(A^1) = \{a_1, a_2, a_6, a_7, a_{12}\}$ і $A(A^2) = \{a_1, a_6, a_7, a_{13}\}$. Закодуємо стани так: для станів $a_m \in A(A^k)$ максимально можливе число однакових розрядів кодів дорівнює нулю. Перший підхід до кодування показано на рис. 8.

Як випливає з рис. 8, $T_1 = 0$ для кодів станів з множини $A(A^1)$ і $T_1 = T_2 = 0$ — з множини $A(A^2)$. Це дає $D_1^1 = D_1^2 = 0$, $D_2^2 = 0$. У цьому підході є змога виключити два елементи LUT із LUTer2 і один — із LUTer1. Крім того, для D_1 маємо рівняння $D_1 = D_1^E$, що виключає один LUT із LUTerZ. Таким чином, цей підхід уможливує зменшення складності схеми на чотири елементи LUT.

У другому підході оптимізується кількість міжз'єднань у схемі LUTerT. У загальному випадку є $R^0 \times R$ таких міжз'єднань (у розглядуваному прикладі $R^0 \times R = 28$). Закодуємо стани, як показано на рис. 9.

Використовуючи табл. 6 та коди станів, наведені на рис. 9, можна отримати таку СБФ:

$$\begin{aligned} \tau_1 &= A_8 \vee A_9 \vee A_{10} \vee A_{11} = \overline{T_1} T_3, \quad \tau_2 = A_4 \vee A_5 \vee A_{10} \vee A_{11} = \overline{T_1} T_4, \\ \tau_3 &= A_1 \vee A_5 \vee A_9 \vee A_{11} = \overline{T_1} \overline{T_2}, \quad \tau_4 = A_6 \vee A_{12} = T_1 \overline{T_2} \overline{T_3}, \\ \tau_5 &= A_2 \vee A_{12} = T_1 \overline{T_3} \overline{T_4}, \quad \tau_6 = A_7 \vee A_{13} = T_1 \overline{T_3} T_4, \quad \tau_7 = A_3 \vee A_{13} = T_1 T_3 \overline{T_4}. \end{aligned}$$

Ця система має 18 літералів, що в 1.56 раза менше за максимальне значення. Як випливає з [26], зменшення кількості міжз'єднань зменшує споживану потужність та збільшує швидкодію схеми.

Для синтезу МПА U_5 доцільно використовувати обидва підходи до кодування станів, а потім вибрати той, за яким можна отримати схему з найкращими характеристиками.

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

Рис. 8. Перший підхід до оптимального кодування станів

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

Рис. 9. Другий підхід до оптимального кодування станів

Аналогічна ситуація і для НМО $Y_q \subseteq Y_L$. Кодування у першому підході можна виконати так, щоб зменшити кількість елементів LUT у блоках LUTer1–LUTerK та LUTerZ, а у другому підході — оптимізувати блок LUTerY.

У розглядуваному випадку перший підхід не може бути використаний через малу кількість НМО. Загалом, класи розбиття Π_A та Π_B треба формувати так, щоб зменшити кількість НМО, які формуються на переходах зі станів цих класів.

Якщо число НМО для цього класу перевищує 2^{R_Q-1} , всі змінні $z_r \in Z$ формуються відповідним блоком LUTerK (або EMB). Якщо це число належить інтервалу $[2^{R_Q-2}, 2^{R_Q-1}]$, то формується лише $R_Q - 1$ змінних, а якщо інтервалу $[2^{R_Q-3}, 2^{R_Q-2}]$, то кількість змінних $z_r \in Z$ для відповідного блока дорівнює $R_Q - 2$. Цей ряд можна продовжити.

Отже, якість МПА $U_5(\Gamma_j)$ можна покращити завдяки оптимальному кодуванню станів. При цьому може бути зменшена кількість елементів LUT або на першому та другому рівнях схеми, або лише на третьому рівні. Аналогічно на характеристики схеми впливає результат кодування НМО. Розроблення подібних методів кодування є одним із напрямів подальших досліджень авторів.

ВИСНОВКИ

Методи структурної декомпозиції дають змогу зменшити площу кристала, яку займає схема МПА [1]. Ці методи базуються на виокремленні деяких блоків з унікальними системами входів та виходів. Методи формування цих блоків значною мірою залежать від особливостей елементного базису, з допомогою якого реалізується схема МПА.

У цій роботі розглянуто реалізацію МПА у базисі FPGA, коли блоки пам'яті EMB та елементи LUT застосовуються разом. Запропоновано використовувати метод подвійного кодування станів [13, 14] та кодування мікрооперацій [1] для оптимізації схеми. За виконання певних умов такий підхід гарантує отримання тривірневої схеми МПА Мілі.

Існує чимало інших підходів до оптимізації схем МПА у базисі FPGA [3, 6]. У роботі порівняно підхід (МПА U_5) з підходом, що базується на функціональній декомпозиції (МПА U_3). Дослідження проводилися з використанням бібліотеки [29] та САПР Vivado [24]. Як елементний базис використовували мікросхеми сім'ї Virtex-7 [20]. Проведені дослідження показали, що запропонований підхід дає змогу в середньому на 20 % зменшити кількість елементів LUT у схемі автомата (порівняно з МПА U_3). При цьому виграш збільшується зі збільшенням кількості входів і станів МПА.

У статті показано, що схема U_5 може бути покращена за рахунок певного кодування станів та наборів мікрооперацій. Розроблення подібних методів є одним із напрямів досліджень авторів. Крім того, планується адаптувати запропонований метод до особливостей суміщених МПА [30], автоматів на лічильниках [27, 28] та автоматів для виконання логічних операцій над нечіткими множинами [31].

СПИСОК ЛІТЕРАТУРИ

1. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and optimization of FPGA-based systems. Berlin: Springer, 2014. 432 p.

2. Czerwinski R., Kania D. Finite state machines logic synthesis for complex programmable logic devices. Berlin: Springer, 2013. 172 p.
3. Kubica M., Opara A., Kania D. Technology mapping for LUT-based FPGA. Cham: Springer, 2021. 207 p. <https://doi.org/10.1007/978-3-030-60488-2>.
4. Baranov S. Logic synthesis for control automata. Dordrecht: Kluwer Academic Publishers, 1994. 312 p.
5. DeMicheli G. Synthesis and optimization of digital circuits. New York: McGraw-Hill, 1994. 576 p.
6. Barkalov A., Titarenko L., Mielcarek K., Chmielewski S. Logic synthesis for FPGA-based control units. *Lecture Notes in Electrical Engineering*. 2020. Vol. 636. 247 p.
7. 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* (6–20 Feb, 2004, Paris, France). 2004. Vol. 2. P. 916–921.
8. 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.
9. Grout I. Digital systems design with FPGAs and CPLDs. Amsterdam: Elsevier, 2008. 784 p.
10. Maxfield C. The design warrior's guide to FPGAs. Orlando: Academic Press, 2004. 542 p.
11. Sass R., Schmidt A. Embedded system design with platform FPGAs: Principles and practices. Amsterdam: Morgan Kaufmann Publishers, 2010. 409 p.
12. Ruiz-Rosero J., Ramirez-Gonzalez G., Khanna R. Field programmable gate array applications — as centometric review. *Computation*. 2019. Vol. 7, Iss. 4. P. 63. <https://doi.org/10.3390/computation7040063>.
13. 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 4. P. 745–759.
14. 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.
15. Barkalov A.A., Titarenko L.A., Barkalov A.A., 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. <https://doi.org/10.1007/s10559-012-9410-2>.
16. Skliarova I., Sklyarov V., Sudnitson A. Design of FPGA-based circuits using hierarchical finite state machines. Tallinn: TUT Press, 2012. 240 p.
17. UG473 (v1.14) July 3, 2019. URL: www.xilinx.com.
18. 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.
19. Глушков В.М. Синтез цифровых автоматов. Москва: Физматгиз, 1962. 476 с.
20. VC709 Evaluation Board for the Virtex-7 FPGA. User Guide; UG887 (v1.6); Xilinx, Inc.: San Jose, CA, USA, 2019.
21. 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.
22. 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.

23. Kubica M., Kania D. Area-oriented technology mapping for LUT-based logic blocks. *International Journal of Applied Mathematics and Computer Science*. 2017. Vol. 27(1). P. 207–222.
24. Vivado design suite. 2020. URL: <https://www.xilinx.com/products/design-tools/vivado.html> (Acceed: January, 2020).
25. Quartus II. URL: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html> (Acceed: January, 2020).
26. 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.
27. Баркалов А.А., Титаренко Л.А. Преобразование кодов в композиционных микропрограммных устройствах управления. *Кибернетика и системный анализ*. 2011. № 5. С. 107–118.
28. Баркалов А.А., Титаренко Л.А., Ефименко К.Н. Оптимизация схем композиционных микропрограммных устройств управления. *Кибернетика и системный анализ*. 2011. № 1. С. 179–188.
29. Yang S. Logic synthesis and optimization benchmarks user guide. Version 3.0. Techn. Rep. Microelectronics Center of North Carolina, 1991. 43 p.
30. Barkalov A.A., Titarenko L.A., Baiev A.V., Matviienko A.V. Optimizing the combined automation scheme in the asis basis. *Cybernetics and Systems Analysis*. 2020. Vol. 56, N 6. P. 863–871. <https://doi.org/10.1007/s10559-020-00306-w>.
31. Kryvyi S.L., Opanasenko V.N., Zavyalov S.B. Logical operations over fuzzy sets and relations in automaton interpretation. *Cybernetics and Systems Analysis*. 2020. Vol. 56, N 6. P. 1012–1020. <https://doi.org/10.1007/s10559-020-00321-x>.

A.A. Barkalov, L.A. Titarenko, O.M. Golovin, A.V. Matviienko

OPTIMIZATION OF MEALY FSM CIRCUIT IN MIXED ELEMENT BASIS

Abstract. A method of reducing LUT count in the FPGA-based circuit of Mealy finite state machine (FSM). A part of the circuit is implemented using embedded memory block (EMB). The method is based on the twofold state assignment and encoding collections of microoperations. An example of the synthesis of FSM circuit using the proposed method is given. When certain conditions are met, there are exactly three levels of logic elements in the FSM circuit. Methods for improving the characteristics of a circuit based on optimal coding of states and collections of microoperations are considered.

Keywords: Mealy FSM, synthesis, FPGA, EMB, LUT, encoding.

Надійшла до редакції 16.01.2024