Кібернетика та комп'ютерні технології

УДК 004.274 DOI:10.34229/2707-451X.24.2.9

О.О. БАРКАЛОВ, Л.О. ТІТАРЕНКО, О.М. ГОЛОВІН, О.В. МАТВІЄНКО

ОПТИМІЗАЦІЯ СХЕМИ МІКРОПРОГРАМНОГО АВТОМАТА МІЛІ У БАЗИСІ LUT І EMB

Запропоновано метод зменшення кількості елементів у схемі мікропрограмного автомата (МПА) Мілі, що реалізується у базисі FPGA. Метод заснований на розширеному кодуванні наборів мікрооперацій, при якому код набору включає код стану переходу. Для реалізації частини схеми МПА використовують блок вбудованої пам'яті EMB. Якщо можливостей EMB не вистачає для реалізації схеми, то частина схеми реалізується на елементах LUT. Наведено приклад синтезу схеми МПА з використанням запропонованого методу.

Ключові слова: *МПА Мілі, синтез FPGA, EMB, LUT, розширені коди наборів мікрооперацій.*

Вступ. Цифрова система є сукупністю комбінаційних і послідовних блоків [1, 2]. Послідовні блоки можна розділити на бібліотечні та нестандартні. До першого класу належать, наприклад, лічильники або регістри зсуву [3]. Для реалізації схем таких блоків використовують стандартні програми САПР [4]. До нестандартних блоків належать, наприклад, пристрої управління (ПУ), для яких немає стандартних бібліотечних рішень [4]. Цим пояснюється постійний інтерес до методів синтезу та оптимізації схем нестандартних послідовних блоків (НБП).

Поведінка НБП може бути представлена моделлю мікропрограмного автомата (МПА) Мілі [1]. При синтезі схеми МПА виникає низка оптимізаційних завдань [5]: 1) зменшення площі мікросхеми, яку займає МПА; 2) підвищення швидкодії; 3) зменшення споживаної потужності. Вважається, що рішення першого з цих завдань дозволяє поліпшити інші характеристики схеми [2]. Методи вирішення цього завдання залежить від особливостей елементної бази [6]. У нашій роботі ми розглядаємо завдання зменшення площі при реалізації схеми МПА в базисі FPGA (field-programmable logic array).

Базис FPGA широко використовується для реалізації різноманітних цифрових систем [7, 8]. Для реалізації схеми МПА використовуються елементи LUT (look-up table), блоки вбудованої пам'яті EMB (embedded memory blocks), тригера, що програмуються, міжз'єднання, і програмовані входи-виходи [9, 10]. Площу прийнято оцінювати числом елементів LUT та блоків EMB [7, 10]. Тому для вирішення першого завдання необхідно зменшувати кількість елементів у схемі МПА.

При проектуванні МПА найкращі результати дає спільне використання LUT та EMB [12]. Проте блоки EMB часто використовуються при реалізації різних операційних блоків цифрових систем [13, 14]. Тому розробник може мати дуже обмежену кількість таких блоків. Ми розглядаємо випадок, коли у розпорядженні розробника є лише один "вільний" блок EMB.

Реалізація схеми МПА Мілі у базисі FPGA. У кожний момент часу МПА Мілі знаходиться в одному із станів $a_m \in \{a_1, \dots, a_M\} = A$, де a_1 – початковий стан. Кожен із H переходів МПА ініціюється вхідними сигналами X_h ($k \in \{1,\dots, H\}$). Вхідний сигнал є кон'юнкцією логічних умов (ЛУ) $x_l \in X = \{x_1, \dots, x_L\}$. На h-му переході формується набір мікрооперацій (НМО) $Y_h \subseteq Y$, де $Y = \{y_1, \dots, y_N\}$ – множина мікрооперацій (МО). Ми розглядаємо практичний випадок, коли кількості станів, ЛУ та МО кінцеві.

Переходи МПА є списком, який задає таблиця переходів [1, 2]. Таблиця переходів має рядки H і стовпці a_m, a_s, X_h, Y_h, h. Тут a_m – вихідний стан, a_s – стан переходу. Для синтезу схеми МПА необхідно розширити таблицю переходів наступними стовпцями: $K(a_m)$ код стану $a_m \in A$, $K(a_s)$ – код стану переходу, Φ_h – набір функцій збудження пам'яті, що змінює $K(a_m)$ на $K(a_s)$ [1, 2].

Коди станів це двійкові вектори розрядності R. Ми розглядаємо максимальне кодування станів [15], при якому

$$R = \lceil \log_2 M \rceil. \tag{1}$$

Для кодування станів використовуються елементи множини $T = \{T_1, \dots, T_R\}$.

У схемі МПА виділяють комбінаційну частину і пам'ять станів [16]. Пам'ять це R-розрядний регістр RG, що складається з тригерів. Тригер номер *r* відповідає *r*-му розряду коду $K(a_m)$, тобто змінної $T_r \in T$. Як правило [16], ці тригери мають тип D. Тому множина функцій збудження пам'яті (ФЗП) складається із змінних D_r: $\Phi = \{D_1, \dots, D_R\}$. Для запису в RG коду початкового стану використовується одиночний імпульс Start. Зміна інформації в RG відбувається за певного фронту імпульсу синхронізації Clock [17].

Елемент LUT має S_L входів та один вихід. Один LUT реалізує довільну функцію алгебри логіки (ФАЛ), що залежить від 1 до S_L змінних. Функціональні можливості LUT обмежені функціями шести змінних ($S_L = 6$) [18]. Для визначення будемо розглядати FPGA Virtex-7 фірми AMD Xilinx [19]. У цьому випадку 4 елементи LUT $S_L = 6$ входять в конфігурований логічний блок (КЛБ). Вихід LUT пов'язаний із входом тригера. Використовуючи вбудований мультиплексор, вихід КЛБ передається або вихід LUT (комбінаційна схема), або вихід тригера (біт коду станів).

Використовуючи метод структурного синтезу [1], автомат можна подати у вигляді двох систем булевих функцій:

$$\Phi = \Phi(T, X); \tag{2}$$

$$Y = Y(T, X). \tag{3}$$

При використанні КЛБ, заснованих на LUT, системи (2), (3) реалізуються за допомогою блоків LUTer і LUTerY, відповідно [20]. Під словом LUTer розуміється блок, що складається з елементів LUT. Блок LUTerΦ включає регістр RG, розподілений між блоками КЛБ, що формують ФЗП. У загальному випадку МПА Мілі представляється у вигляді моделі U_1 (рис.1).

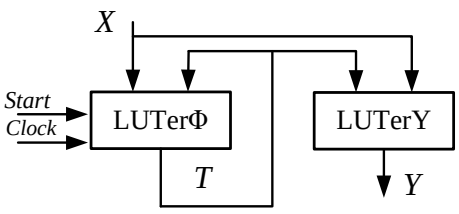

РИС. 1. Структурна схема МПА U_1

Сигнали Start та Clock керують регістром RG, розподіленим між тригерами блоку LUTerФ. Виходи RG це змінні $T_r \in T$, що кодують стани.

Блок EMB – конфігурований блок пам'яті, що складається з осередків. Число осередків визначається числом адресних розрядів S_A, розрядність яких t_F змінюється від 1 до $t_{\max}$. Блок має постійну ємність V_0:

$$V_0 = 2^{S_A} \bullet t_F . \tag{4}$$

Параметр $t_{\max}$ визначається співвідношенням параметрів V_0 та мінімальним значенням числа адресних входів. Пара $< S_A,\ t_F >$ визначає конфігурацію блоку EMB. Наприклад, для EMB сімейства Virtex-7 [19] $V_0 = 32$ Кбіт і є такі пари <15, 1>, <14, 2>,..., <9, 64>. Отже, $t_{\max} = 64$ та мінімум $S_A = 9$.

При реалізації у базисі EMB системи (2), (3) представляються як таблиці істинності. Якщо є пара $< S_0,\ t_0 >$, для якої

$$S_0 \geq L + R ; \tag{5}$$

$$t_0 \geq N + R , \tag{6}$$

то МПА можна реалізувати за допомогою одного блоку EMB. Таке рішення є оптимальним за площею, швидкодією та споживаною потужністю [22].

Якщо умови (5), (6) не виконуються, то схема це мережа EMB. У найкращому разі, кожен EMB реалізує унікальні функції систем (2), (3). Блок EMB_i має входи T та $X^i \in X$ і виходи Y^i та T^i. Це відображено на моделі U_2 (рис. 2).

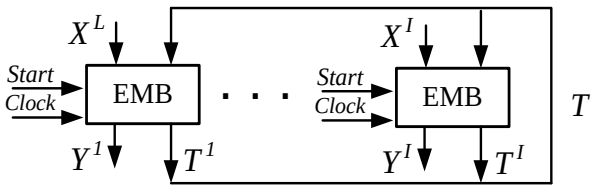

РИС. 2. Структурна схема МПА U_2

Блоки EMB мають внутрішній регістр, керований сигналами Start та Clock. Число блоків I залежить від характеристик МПА та блоків EMB [5]. Існують способи зменшення значення I.

Позначимо символом $NA(f_i)$ число аргументів функції $f_i \in \Phi \cup Y$. При виконанні умови

$$NA(f_i) \leq S_L \tag{7}$$

блоки LUTerФ і LUTerY мають один рівень. Це найкраще рішення для схем, заснованих на LUT [4]. Якщо (7) порушується, схема є багаторівневою. Це призводить до погіршення характеристик схеми порівняно з однорівневою [17].

З цього аналізу випливає наступний підхід реалізації схем МПА в базисі FPGA. При виконанні умов (5), (6) для схеми достатньо одного блоку EMB. При порушенні (5), (6) необхідно перевірити умову (7). За умови (7) схема реалізується на елементах LUT. Далі характеристики цієї схеми порівнюються із характеристиками МПА U_2. Якщо розробник не має I блоків EMB, необхідно використовувати змішаний базис LUT і EMB [10, 12]. Саме цьому підходу присвячується наша стаття.

Реалізація схем МПА у змішаному елементному базисі. Спільне використання елементів LUT та блоків EMB засноване на заміні ЛУ та/або кодуванні наборів мікрооперацій [17]. В цьому разі вводяться додаткові системи булевих функцій (СБФ).

При заміні ЛУ [17] множина X замінюється множиною $P = \{p_1, ..., p_G\}$, де $G << L$. Для заміни формується СБФ

$$P = P(T, X), \tag{8}$$

кожна з функцій якої відповідає мультиплексору [22]. Функції (2), (3) перетворюються на СБФ

$$\Phi = \Phi(T, P); \tag{9}$$

$$Y = Y(T, P). \tag{10}$$

Цей підхід доцільно використовувати, якщо для EMB виконуються умови

$$S_0 \geq L + R; \tag{11}$$

$$t_0 \geq N + R. \tag{12}$$

Якщо умова (12) не виконуються, то для реалізації систем (9), (10) необхідно

$$n_1 = \lceil (N + R) / t_0 \rceil \tag{13}$$

блоків EMB. Це веде до МПА U3 (рис. 3).

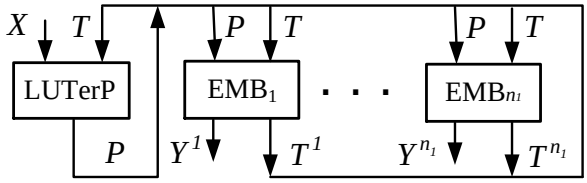

РИС. 3. Структурна схема МПА U_3

При кодуванні НМО $Y_q \subseteq Y$ кожен набір кодується кодом $K(Yq)$ [17] розрядності

$$R_Q = \lceil \log_2 Q \rceil. \tag{14}$$

Для кодування НМО використовуються змінні $z_r \in Z$, де $Z = \{z_1, ..., z_{RQ}\}$. Формування кодів пов'язане з реалізацією СБФ

$$Z = Z(T, X). \tag{15}$$

В результаті МО $y_n \in Y$ представляються СБФ

$$Y = Y(Z). \tag{16}$$

яка замінює СБФ(3).

У нашій статті розглядається граничний випадок, коли розробник має лише один "вільний" блок EMB. Цей блок доцільно використовувати для реалізації СБФ (16) [17]. Такий підхід призводить до МПА U4 (рис. 4).

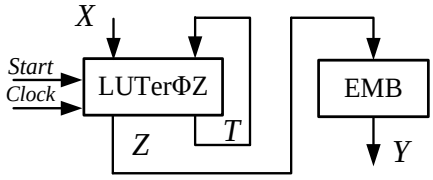

РИС. 4. Структурна схема МПА U_4

У МПА U4 блок LUTerФZ реалізує СБФ (2) та (15). Блок EMB реалізує СБФ (16). Такий підхід можливий, якщо є конфігурація $< S_0,\ t_0 >$ така, що при $S_0 = R_Q$ виконується умова $t_0 \geq N$ [22].

Ці два підходи можна поєднати. У цьому випадку схема МПА заснована на СБФ (8), (9), (16) та

$$Z = Z\ (T,\ P). \tag{17}$$

Дослідження [17] показали, що найкращі характеристики має МПА U_5 (рис. 5).

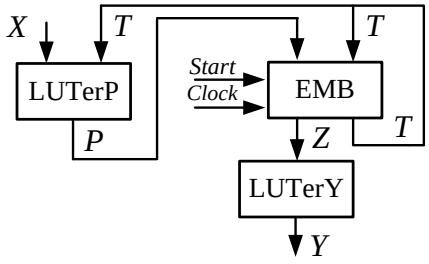

РИС. 5. Структурна схема МПА U_5

У МПА U_5 LUTerP реалізує СБФ (8), а LUTerY – СБФ (16). При цьому Блок EMB реалізує СБФ (9) та (17). Цей підхід можливий при виконанні умов

$$S_0 = G + R; \tag{18}$$

$$t_0 \geq R_Q + R. \tag{19}$$

Аналіз схеми рис. 5 показує низку недоліків цього підходу. По-перше, необхідно формувати дві множини додаткових змінних ($P,\ Z$). Це збільшує кількість потрібних ресурсів. По-друге, схема має три рівні логіки, що призводить до зменшення швидкодії порівняно, наприклад, з МПА U_4. Для оптимізації характеристик схеми МПА необхідно зменшити принаймні кількість рівнів логіки. Такий підхід пропонується у цій роботі.

Основна ідея пропонованого методу. Нехай у розпорядженні розробника МПА є лише один блок EMB і виконується умова

$$2^{R+L} \bullet (N + R)\ > V_0. \tag{20}$$

У цьому випадку функціональних можливостей EMB не вистачає для реалізації схеми МПА. Однак частину схеми за допомогою EMB реалізувати можна.

Нехай НМО $Y_q \subseteq Y$ формується разом із $H\left(Y_q\right)$ станів переходу $a_s \in A$. Знайдемо максимальне число M_{max} станів, при переході до яких формуються однакові НМО:

$$H_{\max} = \max\left(H\left(Y_1\right),\ldots,\ H\left(Y_Q\right)\right). \tag{21}$$

Знайдемо число додаткових змінних R_V, що дозволяє відрізнити стан переходу:

$$R_V = \lceil \log_2 H_{\max} \rceil. \tag{22}$$

Закодуємо НМО $Y_q \subseteq Y$ кодами $K\left(Y_q\right)$ та сформуємо множину Z. Закодуємо стани $a_m \in A$ кодами $K\left(a_m\right)$, розрядність R яких визначається формулою (1). Отже, нам відомі величини параметрів $R,\ R_V,\ R_Q$.

Нехай серед конфігурацій ЕМВ є пара $< S_0,\ t_0 >$, для якої

$$S_0 = R_V + R_Q; \tag{23}$$

$$t_0 \geq R. \tag{24}$$

Нехай $A(Y_q) \subseteq A$ – множина станів, при переходах у які формується НМО $Y_q \subseteq Y$. Закодуємо стани двійковими кодами $C(a_m)$ розрядності R_V. Будемо використовувати для кодування змінні, $v_r \in V$ де $V = \{v_1, \ldots, v_{R_V}\}$.

Назвемо код $EC(Y_q) = C\ (a_m) * K\ (Y_q)$ розширеним кодом НМО $Y_q \subseteq Y$. Якщо стан $a_m \in A$ входить у K_m до множини $A(Yq)$, код $K(a_m)$ є диз'юнкцією K_m розширених кодів. Таким чином, змінні $T_r \in T$ залежать від змінних $v_r \in V,\ z_r \in Z$:

$$K(a1) = 000,\ K(a2) = 001. \tag{25}$$

Такий підхід визначає МПА U_6 (рис. 6).

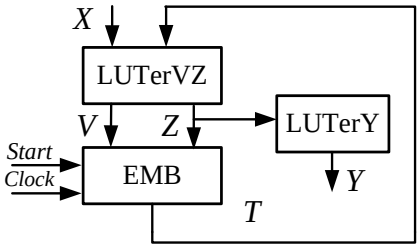

РИС. 6. Структурна схема МПА U_6

У МПА U_6 блок LUTerVZ реалізує СБФ (15) та

$$V=V(T, X). \tag{26}$$

Блок LUTerY реалізує СБФ (16), а блок ЕМВ – СБФ (25). Фактично, блок ЕМВ перетворює коди НМО $EC(Y_q)$ у коди станів і формує змінні $T_r \in T$. Блок LUTerVZ формує необхідні розширені коди. Зазначимо, що структурна схема U_6 відповідає $t_0 = R$.

Якщо виконується умова (23) і при цьому $t_0 > R$, то частину МО $y_n \in Y$ можна реалізувати на блоці ЕМВ. При цьому множина Y може бути представленою як об'єднання множин $Y = Y_E \cup Y_L$, які не перетинаються. Мікрооперації $y_n \in Y_E$ генеруються блоком ЕМВ, а мікрооперації $y_n \in Y_L$ – блоком LUTerY. Це призводить до МПА U_7 (рис. 7).

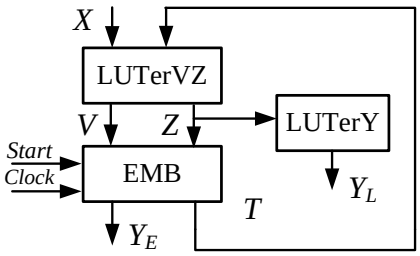

РИС. 7. Структурна схема МПА U_7

У МПА U_7 блок EMB реалізує СБФ (25) та

$$Y_E = Y_E(Z). \tag{27}$$

Число МО (n_E) у множині $Y_E \subseteq Y$ визначається виразом

$$n_E = t_0 - R. \tag{28}$$

Блок LUTerY реалізує СБФ

$$Y_L = Y_L(Z). \tag{29}$$

число елементів, у якій дорівнює $n_L = N - n_E$.

У цій роботі пропонується наступний метод реалізації МПА U_7:

– формування множини A та кодування його елементів;

– формування множини наборів мікрооперацій;

– визначення параметрів R_V, R_Q, n_E, n_L;

– кодування НМО, що оптимізує СБФ (29);

– кодування НМО розширеними кодами ЕС(Y_q);

– формування таблиці блоку LUTerVZ та СБФ (15) та (26);

– формування таблиці блоку EMB;

– реалізація схеми МПА у заданому елементному базисі.

Приклад синтезу МПА. Нехай для реалізації схеми МПА розробник може використовувати один блок EMB і кількість елементів LUT, що мають $S_L = 5$ входів. Нехай EMB має такі конфігурації: <10, 1>, <9, 2>, <8, 4>, <7, 8>. Розглянемо застосування запропонованого методу синтезу автомата S_1 (табл.1).

Аналіз табл.1 дозволяє знайти множини $A = \{a_1, \dots, a_6\}$, $X = \{x_1, \dots, x_8\}$. $Y = \{y_1, \dots, y_8\}$, що дає $M = 6$, $L = 8$, $N = 8$. Отже, $M = 6$, що дає $R = 3$ та множини $T = \{T_1, T_2, T_3\}$, $\Phi = \{D_1, D_2, D_3\}$. Закодуємо стани $a_m \in A$ тривіальним чином: $K(a_1) = 000$, $K(a_2) = 001$, ..., $K(a_6) = 101$.

Ми додали в табл. 1 стовпець q, який є індексом НМО. Таким чином, з табл.1 можна знайти $Q = 7$ НМО: $Y_1 = \varnothing$, $Y_2 = \{y_1, y_2, y_3\}$, $Y_3 = \{y_2, y_7, y_8\}$, $Y_4 = \{y_4, y_6, y_8\}$, $Y_5 = \{y_1, y_3\}$, $Y_6 = \{y_2, y_8\}$, $Y_7 = \{y_5, y_6.\}$.

Знайдемо параметри R_Q, R_V, n_E, n_L, як випливає з (14), $R_Q = 3$, що дає множину $Z_1 = \{z_1, z_2, z_3\}$. Побудуємо множину $A(Y_q)$. Наприклад, НМО Y_2 формується при переходах у стани a_2 (рядки 1 і 12) та a_6 (рядок 8). Маємо. Продовжуючи подібний аналіз, знайдемо такі множини: $A(Y_2) = \{a_2, a_6\}$, $A(Y_2) = \{a_2, a_6\}$, $A(Y_2) = \{a_2, a_6\}$, $A(Y_1) = \{a_1\}$, $A(Y_2) = \{a_2, a_6\}$, $A(Y_3) = \{a_3, a_5\}$, $A(Y_4) = \{a_4, a_6\}$, $A(Y_5) = \{a_3\}$, $A(Y_6) = \{a_2, a_4\}$, $A(Y_7) = \{a_4\}$. Отже, маємо $M(Y_1) = M(Y_5) = M(Y_7) = 1$ і $M(Y_2) = M(Y_3) = M(Y_4) = 2$. З виразу (21) маємо $Max = 2$, і (22) отримуємо $R_V = 1$, $V = \{v_1\}$.

Отже, $R_V + R_Q = 4$ і з конфігурацій необхідно вибрати пару <7, 8>, що дає $t_0 = 8$. Оскільки $R = 3$, $n_E = 8 - 3 = 5$. Отже 5 мікрооперацій реалізується на EMB та $N - n_E = 3$ – на елементах LUT.

ТАБЛИЦЯ 1.Таблиця переходів МПА S_1

a_m	a_s	X_h	Y_h	q	h
a_1	a_2	x_1	$y_1y_2y_7$	2	1
	a_3	$\overline{x_1}x_2$	$y_2y_7y_8$	3	2
	a_4	$\overline{x_1}\,\overline{x_2}$	$y_4y_6y_8$	4	3
a_2	a_3	x_3	y_1y_3	5	4
	a_4	$\overline{x_3}$	y_3y_8	6	5
a_3	a_5	x_6	$y_2y_7y_8$	3	6
	a_2	$\overline{x_6}x_4$	y_3y_8	6	7
	a_6	$\overline{x_6}\,\overline{x_4}$	$y_1y_2y_7$	2	8
a_4	a_1	1	–	1	9
a_5	a_5	x_5	$y_2y_7y_8$	3	10
	a_6	$\overline{x_5}$	$y_4y_6y_8$	4	11
a_6	a_2	x_7	$y_1y_2y_7$	2	12
	a_4	$\overline{x_7}x_8$	y_5y_6	7	13
	a_1	$\overline{x_7}\,\overline{x_8}$	–	1	14

Закодуємо НМО так, щоб для трьох МО кількість аргументів була мінімальною. Для цього знайдемо залежність між функціями $y_n \in Y$ та змінними Z_q, $K(Y_q)$. Поки що ми не знаємо формул для Z_q, як кон'юнкцій змінних . Ця система має такий вигляд:

$$y_1 = Z_2 \vee Z_5;\ y_2 = Z_2 \vee Z_3 \vee Z_5;\ y_3 = Z_5 \vee Z_6;\ y_4 = Z_4;$$
$$y_5 = Z_7;\ y_6 = Z_4 \vee Z_7;\ y_7 = Z_2 \vee Z_3;\ y_8 = Z_3 \vee Z_4 \vee Z_6. \quad (30)$$

Використовуємо для кодування НМО $y_q \subseteq Y$ метод [17]. Один із варіантів кодування показано на рис.8.

z_3 \ z_1z_2	00	01	11	10
0	Y_1	Y_2	Y_5	Y_7
1	Y_6	Y_3	*	Y_4

РИС. 8. Коди НМО автомата S_1

З карти Карно (рис. 8) маємо СБФ (16). Вона має вигляд:

$$y_1 = z_2\overline{z_3};\ y_2 = z_2;\ y_3 = \overline{z_1}\,\overline{z_2}z_3 \vee z_1z_2;\ y_4 = z_1z_3;$$
$$y_5 = z_1\overline{z_2}\,\overline{z_3};\ y_6 = z_1\overline{z_2};\ y_7 = \overline{z_1}z_2;\ y_8 = z_3. \quad (31)$$

Виберемо 3 формули, що мають мінімум літералів. Якщо якісь дві формули мають однакове число літералів, то виберемо формулу з меншим індексом.

Застосувавши цей метод, отримаємо множини $Y_L = \{y_2, y_8, y_1\}$, $Y_E = \{y_3, y_4, y_5, y_6, y_7\}$. Очевидно, щоб реалізувати схеми для y_2, y_8 елементи LUT не потрібні. Ці МО представляються як виходи блоку LUTerVZ. Отже, СБФ (29) виглядають так:

$$y_1 = z_2\overline{z_3}; y_2 = z_2; y_8 = z_3. \quad (32)$$

Для формування кодів $EC(Y_q)$ необхідно визначити коди $C(a_1)$. Вочевидь, для кожної пари станів один код дорівнює 0, а другий – 1. Умовимося код 0 присвоювати стану з меншим індексом. В результаті отримаємо наступні коди: $C(a_2) = C(a_3) = C(a_4) = 0$, $C(a_4) = C(a_5) = C(a_6) = 1$. Зазначимо, що стан $a_4 \in A$ має два коди. Ці коди залежать від того, який НМО формується при переході в цей стан.

Маючи коди $K(Y_q)$, $C(a_m)$, можна отримати розширені коди НМО. Ці коди для нашого прикладу наведені в табл. 2.

У табл. 2 показано відповідність між НМО і станами переходу з одного боку і кодами $K(a_s)$. Стовпець Y_q включає НМО, а стовпець $K(Y_q)$ – їх коди; стовпець a_s включає стани переходу відповідні даним НМО, а стовпець $C(a_s)$ – код цього стану, як елемента множини $A(Y_q)$; стовпець $EC(Yq)$ містить розширений код НМО, а стовпець $K(a_s)$ – відповідний код стану переходу, як елемента множини A; стовпець h показує, яким рядкам вихідної таблиці переходів відповідає цей рядок таблиці розширених НМО. У відповідних стовпцях показані змінні, що утворюють коди $K(Y_q)$, $C(a_s)$, $EC(Yq)$ та $K(a_s)$.

Інформація табл. 2 використовується для формування таблиці блоку LUTerVZ. Ця таблиця має стовпці a_m, X_h, V_h, Z_h, h. Стовпці a_m, X_h беруться з таблиці переходів (табл.1), а стовпці V_h, Z_h – з табл. 2. При цьому кожна пара $< a_s, Y_q >$ з табл.1 перетворюється на набір $< v_1, z_1, z_2, z_3 >$. Перший компонент набору записаний у стовпці V_h, а три чергових – у стовпці V_h. У прикладі блок LUTerVZ наведений у табл. 3.

ТАБЛИЦЯ 2. Таблиця розширених НМО

Y_q	$K(Y_q)$	a_S	$C(a_s)$	$EC(Yq)$	$K(a_S)$	h
	$z_1 z_2 z_3$		v_1	$v_1 z_1 z_2 z_3$	$T_1 T_2 T$	
Y_1	0 0 0	a_1	0	0 0 0 0	0 0 0	14, 9
Y_2	0 1 0	a_2	0	0 0 1 0	0 0 1	1, 12
Y_2	0 1 0	a_6	1	1 0 1 0	1 0 1	8
Y_3	0 1 1	a_3	0	0 0 1 1	0 1 0	2
Y_3	0 1 1	a_5	1	1 0 1 1	1 0 0	6, 10
Y_4	1 0 1	a_4	0	0 1 0 1	0 1 1	3
Y_4	1 0 1	a_6	1	1 1 0 1	1 0 1	11
Y_5	1 1 0	a_3	0	0 1 1 0	0 1 0	4
Y_6	0 0 1	a_2	0	0 0 0 1	0 0 1	7
Y_6	0 0 1	a_4	1	1 0 0 1	0 1 1	5
Y_7	1 0 0	a_4	0	0 1 0 0	0 1 1	13

ТАБЛИЦЯ 3. Таблиця блока LUTerVZ

a_m	$K(a_m)$	V_h	Z_h	h	X_h
a_1	000	—	z_2	1	x_1
		—	z_2z_3	2	$\overline{x_1}x_2$
		—	z_1z_3	3	$\overline{x_1}\,\overline{x_2}$
a_2	001	—	z_1z_2	4	x_3
		v_1	z_3	5	$\overline{x_3}$
a_3	010	v_1	z_2z_3	6	x_6
		—	z_3	7	$\overline{x_6}x_4$
		v_1	z_2	8	$\overline{x_6}\,\overline{x_4}$
a_4	011	—	—	9	1
a_5	100	v_1	z_2z_3	10	x_5
		v_1	z_1z_3	11	$\overline{x_5}$
a_6	101	—	z_2	12	1
		—	z_1	13	$\overline{x_7}x_8$
		—	—	14	$\overline{x_7}\,\overline{x_8}$

З табл. 3 можна отримати СБФ (15) і (26), наприклад, такі СБФ:

$$z_1 = \overline{T_1}\,\overline{T_2}\,\overline{T_3}\,\overline{x_1}\,\overline{x_2} \vee \overline{T_1}\,\overline{T_2}T_3x_3 \vee T_1\overline{T_2}\,\overline{T_3}\,\overline{x_5} \vee T_1\overline{T_2}T_3\overline{x_7}x_8;$$

$$v_1 = \overline{T_1}\,\overline{T_2}T_3\overline{x_3} \vee \overline{T_1}T_2\overline{T_3}x_6 \vee \overline{T_1}T_2\overline{T_3}\,\overline{x_6}\,\overline{x_4} \vee T_1\overline{T_2}\,\overline{T_3}. \qquad (32)$$

Таблиця блоку ЕМВ (табл. 4) включає стовпці V, Z (адреса комірки пам'яті), T, Y_E (вміст комірки), b, h.

Таблиця має $B = 2^{S_0}$ рядків; у нашому випадку $B = 128$. Однак $R_V + R_Q = 4$. Тому три старші розряди семи розрядної адреси є несуттєвими. Ми можемо визначити адресу як $<***v_1z_1z_2z_3>$. Отже, кожен розширений код НМО відповідає 8 рядкам таблиці ЕМВ. Цей факт відбито у стовпці b табл. 4.

У стовпці h табл. 4 показано, яким рядкам таблиці переходів (табл.1) відповідає кожен рядок табл. 4. Зазначимо, що інформація зі стовпців b і h показана лише для "внутрішнього використання".

Останній крок реалізації пов'язаний з використанням САПР, які виконують технічне відображення [7, 8]. Для FPGA фірми Xilinx із цією метою використовується САПР Vivado [23]. У нашій статті ми не розглядаємо цей крок для нашого прикладу.

ТАБЛИЦЯ 4. Таблиця EMB для МПА U_7

Адреса	Вміст	b	h
*** v_1 z_1 z_2 z_3	T_1 T_2 T_3 y_3 y_4 y_5 y_6 y_7		
*** 0 0 0 0	0 0 0 0 0 0 0 0	0 — 7	9, 14
*** 0 0 0 1	0 0 1 1 0 0 0 1	8 — 15	7
*** 0 0 1 0	0 0 1 0 0 0 0 1	16 — 23	1, 12
*** 0 0 1 1	0 1 0 0 0 0 0 1	24 — 31	2
*** 0 1 0 0	0 1 1 0 0 1 1 0	32 — 39	13
*** 0 1 0 1	0 1 1 0 1 0 1 0	40 — 47	3
*** 0 1 1 0	0 1 0 1 0 0 0 0	48 — 55	4
*** 0 1 1 1	0 0 0 0 0 0 0 0	56 — 63	—
*** 1 0 0 0	0 0 0 0 0 0 0 0	64 — 71	—
*** 1 0 0 1	0 1 1 1 0 0 0 0	72 — 79	5
*** 1 0 1 0	1 0 1 0 0 0 0 1	80 — 87	8
*** 1 0 1 1	1 0 0 0 0 0 0 1	88 — 95	6, 10
*** 1 1 0 0	0 0 0 0 0 0 0 0	96 — 103	—
*** 1 1 0 1	1 0 1 0 1 0 1 0	104 — 111	11
*** 1 1 1 0	0 0 0 0 0 0 0 0	112 — 119	—
*** 1 1 1 1	0 0 0 0 0 0 0 0	120 — 127	—

Результати. Метод запропонований у цій статті є альтернативою МПА U_5 із трьома рівнями логіки. У МПА U_5 блок EMB використовується для формування кодів станів та НМО. Ми провели порівняння ефективності запропонованого МПА U_7 з U_5. Для дослідження використовувалися стандартні бенчмарки із бібліотеки [23]. Дослідження показали, що наш підхід дозволяє зменшити кількість LUT для 28 % усіх бенчмарків, а U_5 – лише для 9 %. Зазначимо, що з 64 % стандартних МПА [17] достатньо лише одного блоку EMB для реалізації всієї схеми. Таким чином, апріорі не можна сказати, яка модель (U_5 або U_7) дає найкращі результати для конкретного МПА.

Висновки. Проблема оптимізації характеристик ПУ не втрачає актуальності, починаючи з робіт М. Вілкса у 1951р. При кожній зміні елементного базису змінюється лише підхід до вирішення цієї проблеми. Поява мікросхем FPGA не змінила цієї ситуації.

У цій статті пропонується спосіб зменшення кількості елементів LUT у схемі МПА автомата Мілі. При цьому частина схеми реалізується за допомогою блоку EMB. Для вирішення цього завдання ми пропонуємо новий підхід до кодування наборів мікрооперацій. Кожен набір це розширений код, що має у своєму складі коди станів, при переході в які генеруються мікрооперації цього набору. Для перетворення розширених кодів НМО на коди станів використовується EMB. Решта схеми реалізується на елементах LUT.

У наших подальших дослідженнях ми сподіваємося адаптувати цей метод для врахування особливостей автомата Мура [17] та змішаних МПА [24, 25]. Крім того, ми плануємо дослідити застосування цього методу для реалізації схем з використанням FPGA фірми Intel (Altera) [26].

Авторські внески.

Баркалов О.О.: концептуалізація, методологія, формальний аналіз, написання – оригінальна чернетка. Тітаренко Л.О.: узагальнення, концептуалізація, методологія. Головін О.М.: формальний аналіз, ресурси, написання – рецензування та редагування, програмне забезпечення. Матвієнко О.В.: формальний аналіз, дослідження, ресурси, візуалізація, написання – рецензування та редагування.

Список літератури

1. 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. https://doi.org/10.1109/DATE.2004.1269007
2. 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. https://doi.org/10.1007/978-3-030-38295-7
3. Baranov S. Logic synthesis for control automata. Dordrecht: Kluwer Academic Publishers, 1994. 312 p. https://doi.org/10.1007/978-1-4615-2692-6
4. DeMicheli G. Synthesis and optimization of digital circuits. New York: McGraw-Hill, 1994. 576 p.
5. Barkalov A., Titarenko L., Kolopienczyk M., Mielcarek K., Bazydlo G. Logic synthesis for FPGA-based Finite State Machine. *Studies in Systems, Decision and Control*. Vol. 38. Springer International Publishing, Cham Heidelberg, 2015. https://doi.org/10.1007/978-3-319-24202-6
6. Maxfield C. The design warrior's guide to FPGAs. Orlando: Academic Press, 2004. 542 p.
7. Ruiz-Rosero J., Ramirez-Gonzalez G., Khanna R. Field Programmable Gate Array Applications – A Scientometric Review. Computation 2019. **7** (4). 63. https://doi.org/10.3390/computation7040063
8. Barkalov A., Titarenko L., Mielcarek K. Improving characteristics of LUT–based Mealy FSMs. *International Journal of Applied Mathematics and Computer Science*. 2020. **30** (4). P. 745–759. https://doi.org/10.3390/electronics10080901
9. Bacchetta P., Daldos L., Sciuto D., Silvano C. Low-power state assignment techniques for finite state machines. In Proceedings of the 2000 IEEE International Symposium on Circuits and Systems (ISCAS'2000) (Geneva, 2000), vol. 2. IEEE. P. 641–644. https://doi.org/10.1109/ISCAS.2000.856410
10. Kubica M., Opara A., Kania D. Technology Mapping for LUT-based. FPGA. Berlin: Springer, 2021. https://doi.org/10.1007/978-3-030-60488-2
11. Grout I. Digital systems design with FPGAs and CPLDs. Amsterdam: Elsevier, 2008. 784 p. https://doi.org/10.1016/B978-0-7506-8397-5.X0001-3
12. Sklyarov V. Synthesis and Implementation of RAM-based Finite States Machines in FPGAs. In Proceeding of Field-Programmable Logic and Applications: The Roadmap to Reconfigurable Computing. Villach: Springer-Verlag, 2000. P. 718–727. https://doi.org/10.1007/3-540-44614-1_76
13. Barkalov A. Microprogramcontrol unit as composition of automate with programmable and hardwired logic. *Automatics and Computer Science*. 1983. **17** (4). P. 36–41.
14. Amann R., Baitinger U. Optimal state chains and states codes in finite state machines. IEEE Transactions on Computer-Aided Design. 1989. **8** (2). P. 153–170. https://doi.org/10.1109/43.21834
15. Barkalov A., Shwec A. Synthesis of compositional microprogram control unit with modified microinstruction addressing. *Automatic Control and Computer Sciences*. 1994. **28** (5). P. 22–30.
16. Barkalov A.A. Multilevel programmable logic array schemes for microprogrammed automata. *Cybern Syst Anal*. 1994. **30**. P. 489–495. https://doi.org/10.1007/BF02366558
17. 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,Berlin, 2020. Vol. 636. https://doi.org/10.1007/978-3-030-38295-7
18. Xilinx. http://www.xilinx.com (https://www.amd.com/en.html) (звернення: 01.01.2024)
19. Virtex-7 T and XT FPGAs Data Sheet. https://docs.xilinx.com/v/u/en-US/ds183_Virtex_7_Data_Sheet (звернення: 01.01.2024)
20. Barkalov O., Titarenko L., Mielcarek K. Hardware reduction for LUT based Mealy FSMs. *International Journal of Applied Mathematics and Computer Science*. 2018. Vol. 28, No. 3. P. 595–607. https://doi.org/10.2478/amcs-2018-0046
21. Barkalov, O., Titarenko, L., and Mielcarek, K. Improving characteristics of LUT-basedMealy FSMs. International Journal of Applied Mathematics and Computer Science. 2020. Vol. 30, No. 4. P. 745–759. https://doi.org/10.34768/amcs-2020-0055
22. Vivado Design Suite. https://www.xilinx.com/products/design-tools/vivado.html (звернення: 01.01.2024)
23. McElvain, Kenneth. IWLS'93 Benchmark Set: Version 4.0. https://www.researchgate.net/publication/238195666_Benchmark_set_Version_40 (звернення: 01.01.2024)
24. Machado L., Cortadella J. Support-Reducing Decomposition for FPGA Mapping. IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems. 2020. **39** (1). P. 213–224. https://doi.org/10.1109/TCAD.2018.2878187

25. Баркалов А.А., Титаренко Л.А. Преобразование кодов в композиционных микропрограммных устройствах управления. *Кибернетика и системный анализ*. 2011. № 5. С. 107–118. http://nbuv.gov.ua/UJRN/KSA_2011_47_5_11.
26. www.altera.com (звернення: 01.01.2024)

Одержано 10.04.2024

Баркалов Олександр Олександрович,
доктор технічних наук, професор Університету Зеленогурського (Польща),
https://orcid.org/0000-0002-4941-3979
A.Barkalov@iie.uz.zgora.pl

Тітаренко Лариса Олександрівна,
доктор технічних наук, професор Університету Зеленогурського (Польща),
професор Харківського національного університету радіоелектроніки,
https://orcid.org/0000-0001-9558-3322
L.Titarenko@iie.uz.zgora.pl

Головін Олександр Миколайович,
кандидат технічних наук, старший науковий співробітник
Інституту кібернетики імені В.М. Глушкова НАН України, Київ,
https://orcid.org/0000-0002-0279-812X
o.m.golovin.1@gmail.com

Матвієнко Олександр Володимирович,
науковий співробітник Інституту кібернетики імені В.М. Глушкова НАН України, Київ.
https://orcid.org/0000-0003-1838-1422
avmatv@ukr.net

УДК 004.274

О.О. Баркалов [1], Л.О. Тітаренко [1, 2], О.М. Головін [3], О.В. Матвієнко [3 *]

Оптимізація схеми мікропрограмного автомата Мілі у базисі LUT і EMB

[1] Університет Зеленогурський, Зелена Гура, Польща
[2] Харківський національний університет радіоелектроніки, Україна
[3] Інститут кібернетики імені В.М. Глушкова НАН України, Київ
[*] Листування: avmatv@ukr.net

Вступ. Цифрова система – це сукупність комбінаційних і послідовних блоків. Послідовні блоки можна розділити на бібліотечні та нестандартні класи. До першого класу належать, наприклад, лічильники або регістри зсуву. Для реалізації схем таких блоків використовуються стандартні програми САПР. А для другого класу, яким є блок керування (БК), стандартних бібліотечних рішень не існує. Цим пояснюється актуальність методів синтезу та оптимізації схем нестандартних послідовних блоків, наприклад КУ.

При синтезі схеми мікропрограмного автомата (МПА) виникає низка оптимізаційних задач, які спрямовані на покращення характеристик БК. Способи вирішення цих завдань залежать від характеристик елементної бази. У цьому документі розглядається реалізація схеми МПА на основі FPGA.

Основними блоками FPGA, які використовуються для реалізації схеми МПА, є елементи LUT (таблиці перегляду) та елементи EMB (блоки вбудованої пам'яті). Тому для вирішення задач оптимізації при розробці схеми автомата необхідно зменшити кількість цих елементів.

Мета роботи. Ця робота представляє підхід до зниження апаратурних витрат при реалізації МПА Мілі в базисі FPGA.

Метод заснований на розширеному кодуванні множин мікрооперацій, в якому код множини включає також код перехідного стану. Код стану є частковим, оскільки він визначається для множини станів при переході, з яких утворюється ця множина. Для реалізації частини схеми МПА використовується

вбудований блок пам'яті EMB. Якщо можливостей EMB недостатньо для реалізації схеми, то частина схеми реалізується на елементах LUT. Частину вихідних сигналів (мікрооперацій) пропонується реалізувати на EMB. Наведено приклад синтезу схеми МПА за запропонованим методом.

Результати. Для дослідження ефективності запропонованого методу проведено порівняння ПУ відомої структури (U_5) з ПУ, яке отримане із застосуванням запропонованого методу (U_7). При цьому використовувалися стандартні бенчмарки із відомої бібліотеки. Дослідження показали, що U_7 дозволяє зменшити число LUT на 28 % всіх бенчмарків, а U_5 – лише на 9 %. Важливо зазначити, що для реалізації всієї схеми 64 % стандартних МПА досить одного блоку EMB.

Висновки. Запропонований спосіб дозволяє знизити апаратурні витрати (кількість елементів LUT). У статті наведено умови застосування запропонованого методу та результати експериментів з перевірки ефективності запропонованого підходу до реалізації автоматів на мікросхемах сімейства Virtex-7 і промисловому пакеті Vivado.

Ключові слова: МПА Мілі, синтез, FPGA, EMB, LUT, розширені коди наборів мікрооперацій.

UDC 004.274

Alexandr Barkalov [1], Larysa Titarenko [1, 2], Oleksandr Golovin [3], Oleksandr Matvienko [3 *]

Optimization of the Microprogram Mealy Machine Circuit Based on LUT and EMB

[1] *University of Zielona Gora, Poland*
[2] *Kharkiv National University of Radio Electronics, Ukraine*
[3] *V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv*
[*] *Correspondence: avmatv@ukr.net*

Introduction. A digital system is a collection of combinational and sequential blocks. Sequential blocks can be divided into library and non-standard classes. The first class includes, for example, counters or shift registers. To implement the circuits of such blocks, standard CAD programs are used. And for the second class, which is the control unit (CU), there are no standard library solutions. This explains the relevance of methods for synthesis and optimization of circuits of non-standard sequential blocks, such as CU.

When synthesizing a finite state machine (FSM) circuit, a number of optimization problems arise that are aimed at improving CU characteristics. Methods for solving these problems depend on elemental base characteristics. This paper discusses the implementation of the FSM circuit on a FPGA (field-programmable logic array) basis.

The main FPGA blocks that are used for FSM circuit implementation are LUT (look-up table) elements and EMB (embedded memory blocks) elements. Therefore, to solve optimization problems while developing an FSM circuit, it is necessary to reduce the number of these elements.

The purpose of the article. This work presents an approach to lower hardware costs in the FSM Mealy technique, which uses FPGA for implementation.

The method is based on the extended coding of micro-operation sets, in which the set code also includes the transition state code. The state code is partial since it is determined for a set of states upon transition, from which this set is formed. To implement part of the FSM circuit, the built-in memory block EMB is used. If EMB capabilities are not enough to implement the circuit, then part of the circuit is implemented on LUT elements. It is proposed to implement part of the output signals (micro-operations) on EMB. An example of the synthesis of an FSM circuit using the proposed method is given.

Results. To study the effectiveness of the proposed method, a comparison was made between the control unit of the known structure (U5) and the control unit obtained using the proposed method (U7). In this case, standard benchmarks from a well-known library were used. Research has shown that U7 can reduce the number of LUTs by 28 % of all benchmarks, and U5 only by 9 %. It is important to note that when implementing the entire 64 % standard MPA circuit, one EMB block is sufficient.

Conclusions. The proposed method allows for a reduction in hardware costs (the number of LUT elements). The article shows the conditions for applying the proposed method. Results of experiments examining the effectiveness of the suggested approach to automata implementation with Virtex-7 family chips and the Vivado industrial package are given

Keywords: Mealy FSM, synthesis, FPGA, EMB, LUT, extended codes of micro-operation sets.