

ОПТИМІЗАЦІЯ СХЕМИ АВТОМАТА МУРА У ЗМІШАНОМУ ЕЛЕМЕНТНОМУ БАЗИСІ

Вступ. Одним з найважливіших блоків цифрових систем є пристрій управління [1], що координує взаємодію інших блоків системи. Пристрій управління (ПУ) перебуває у активному стані у кожному такті роботи системи. Тому характеристики схеми ПУ мають значний вплив на якість системи в цілому [2]. Для поліпшення, наприклад, швидкодії системи, необхідно підвищувати операційну частоту ПУ [3]. Цей ефект може бути досягнутий завдяки зменшенню кількості рівнів логічних елементів у схемі ПУ. У цій роботі ми розглядаємо метод розв'язання цієї задачі під час реалізації схеми ПУ у базисі FPGA [4]. Для представлення закону функціонування ПУ використовується модель мікропрограмного автомата (МПА) Мура [5]. Наш вибір обґрунтований широким використанням цієї моделі під час синтезу схем ПУ [2, 6].

При синтезі схем МПА виникає низка оптимізаційних задач [4]: зменшення апаратних витрат, підвищення швидкодії, мінімізація споживаної потужності, спільна оптимізація апаратурно-часових характеристик. Методи вирішення цих задач залежать від особливостей алгоритму управління, моделі автомата та елементного базису. Наприклад, лінійні алгоритми управління [7] реалізуються автоматами на лічильниках [8]. У роботі ми розглядаємо проблему підвищення швидкодії схеми автомата Мура, реалізованої в базисі мікросхем FPGA (field-programmable logic array) [9].

Базис FPGA – один із основних базисів реалізації сучасних цифрових систем. За прогнозами фахівців, така ситуація утримається ще кілька десятиліть [10]. Враховуючи цей факт, ми вибрали FPGA як платформу, що реалізує схеми МПА.

Одним із провідних виробників мікросхем FPGA є фірма Xilinx [11]. Мікросхеми цієї фірми мають так звану "острівну" (island-style) архітектуру. "Острівками" є різні конфігуровані логічні блоки CLB (configurable logic blocks), оточені "морем" програмованих міжз'єднань. Схема МПА може бути або мережею CLB типу LUT (look-up table), або мережею

Запропоновано метод оптимізації схеми автомата Мура у базисі FPGA. Метод заснований на використанні класів псевдоеквівалентних станів автомата Мура. Стани кодуються таким чином, що код стану складається з коду класу псевдоеквівалентних станів і коду набору мікрооперацій. Запропонований метод синтезу дозволяє оптимізувати число елементів LUT (look-up table), коли в схемі мікропрограмного автомата можна використовувати тільки один блок ЕМВ (embedded memory block).

Ключові слова: Автомат Мура, FPGA, LUT, ЕМВ, псевдоеквівалентні стани, синтез.

CLB типу EMB (embedded memory block), або змішаною мережею, що складається з елементів LUT та блоків EMB. У даній роботі ми розглядаємо саме змішану мережу.

Зазвичай, блоки EMB реалізують різні операційні пристрої цифрових систем [12]. Тому розробник ПУ може використовувати обмежену кількість таких блоків пам'яті. Ми розглядаємо крайній випадок, коли є тільки один "вільний" блок EMB. Це найбільш важка проблема. Для завдання алгоритму управління використовуємо мову граф-схем алгоритму (ГСА) [6].

Особливості реалізації МПА Мура та базису FPGA. Особливістю автомата Мура є відсутність прямої залежності виходів $y_n \in Y = \{y_1, \dots, y_N\}$ від входів $x_l \in X = \{x_1, \dots, x_L\}$ [5, 6]. Для реалізації схеми МПА по ГСА Г необхідно [6]:

- 1) побудувати множину внутрішніх станів $A = \{a_1, \dots, a_M\}$;
- 2) виконати кодування станів $a_m \in A$ кодами $K(a_m)$, що мають R розрядів;
- 3) сформувані множини внутрішніх змінних $T = \{T_1, \dots, T_R\}$ та функцій збудження пам'яті (ФЗП) $\Phi = \{D_1, \dots, D_R\}$;
- 4) побудувати пряму структурну таблицю (ПСТ), що визначає переходи $\langle a_m, a_s \rangle$ між станами МПА;
- 5) побудувати на основі ПСТ системи булевих функцій (СБФ)

$$\Phi = \Phi(T, X); \tag{1}$$

$$Y = Y(T); \tag{2}$$

- 6) побудувати логічну схему МПА у заданому елементному базисі.

Розглянемо ГСА Г1 (рис.1).

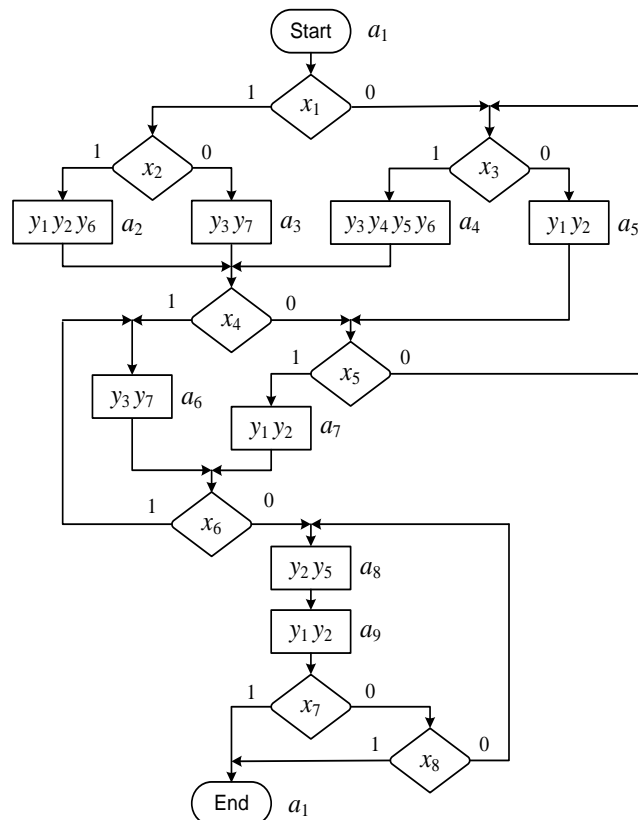


РИС. 1. Позначена ГСА Г1

Ця ГСА позначена станами автомата Мура, використовуючи правила [6]. Початкова вершина Start та кінцева вершина End позначені початковим станом $a_1 \in A$. Кожна операторна вершина, що містить набір мікрооперацій (НМО) $Y_q \subseteq Y$, позначається унікальним станом. Зазначимо, що НМО може бути порожнім. Переходи між станами a_m та a_s позначаються логічними умовами (ЛУ), записаними в умовних вершинах ГСА.

Переходи визначаються кон'юнкціями X_h ($h = \overline{1, N}$), які складаються з ЛУ (чи їх інверсій). Наприклад, якщо $x_1 = 1$ та $x_2 = 0$, то кон'юнкція визначає перехід $\langle a_1, a_3 \rangle$.

Як впливає з ГСА Г1, відповідний автомат Мура має $M = 9$ станів, $L = 8$ логічних умов, $N = 7$ мікрооперацій. Для визначення кількості переходів необхідно знайти всі шляхи, що ведуть з вихідних станів $a_m \in A$ у стани переходу $a_s \in A$ [6]. У нашому прикладі є $H = 23$ таких шляхів. Отже, ПСТ автомата Мура, заданого ГСА Г1, має $H = 23$ рядки.

Виберемо для кодування станів метод максимального кодування [12], за якого

$$R = \lceil \log_2 M \rceil. \tag{3}$$

Для нашого прикладу $R = 4$, $T = \{T_1, \dots, T_4\}$, $\Phi = \{D_1, \dots, D_4\}$. Закодуємо стани тривіальним чином: $K(a_1) = 0000$, $K(a_2) = 0001, \dots, K(a_9) = 1000$. Використовуючи ці коди, побудуємо ПСТ автомата Мура (табл.1).

ТАБЛИЦЯ 1. ПСТ автомата Мура P(Г1)

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Φ_h	h
1	2	3	4	5	6	7
a_1 (-)	0000	a_2	0001	$x_1 x_2$	D_4	1
$x_1 x_2$		a_3	0010	$\overline{x_1 x_2}$	D_2	2
		a_4	0011	$\overline{x_1 x_3}$	$D_3 D_4$	3
		a_5	0100	$\overline{x_1 x_3}$	D_4	4
a_2 ($y_1 y_2 y_6$)	0001	a_6	0101	x_4	$D_2 D_4$	5
		a_7	0110	$\overline{x_4 x_5}$	$D_2 D_3$	6
		a_5	0100	$\overline{x_4 x_5}$	D_2	7
a_3 ($y_3 y_7$)	0010	a_6	0101	x_4	$D_2 D_4$	8
		a_7	0110	$\overline{x_4 x_5}$	$D_2 D_3$	9
		a_5	0100	$\overline{x_4 x_5}$	D_2	10
a_4 ($y_3 y_4 y_5 y_6$)	0011	a_6	0101	x_4	$D_2 D_4$	11
		a_7	0110	$\overline{x_4 x_5}$	$D_2 D_3$	12
		a_5	0100	$\overline{x_4 x_5}$	D_2	13
a_5 ($y_1 y_2$)	0100	a_7	0110	x_5	$D_2 D_3$	14
		a_5	0100	$\overline{x_5}$	D_2	15
a_6 ($y_3 y_7$)	0101	a_6	0101	x_6	$D_2 D_4$	16
		a_8	0111	$\overline{x_6}$	$D_2 D_3 D_4$	17

1	2	3	4	5	6	7
a ₇ (y ₁ y ₂)	0110	a ₆	0101	x ₆	D ₂ D ₄	18
		a ₈	0111	$\overline{x_6}$	D ₂ D ₃ D ₄	19
a ₈ (y ₂ y ₅)	0111	a ₉	1000	1	D ₁	20
a ₉ (y ₁ y ₂)	1000	a ₁	0000	x ₇	—	21
		a ₁	0000	$\overline{x_7 x_8}$	—	22
		a ₈	0111	$\overline{x_7 x_8}$	D ₂ D ₃ D ₄	23

У даній роботі символом "Тип МПА (Гj)" ми позначаємо те, що МПА певного типу синтезується по ГСА Гj. Якщо схема автомата синтезується з урахуванням функцій (1), (2), такий МПА прийнято називати автоматом типу Р [6]. З табл. 1 однозначно визначаються функції (1), (2), тому дана таблиця задає схему автомата Р(Г1), що і знайшло свій відбиток у назві цієї таблиці. Структурну схему Р МПА Мура показано на рис. 2.

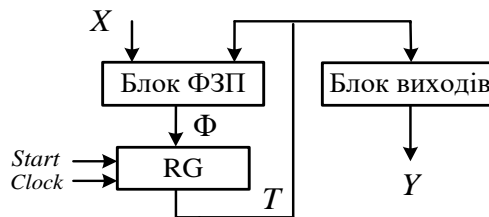


РИС. 2. Структурна схема Р МПА Мура

У Р МПА (рис. 2) блок ФЗП реалізує СБФ (1). Функції (1) формують коди станів переходу, які записуються за сигналом Clock у реєстр коду станів RG. Виходи реєстру RG надходять як зворотний зв'язок у блок ФЗП, а також у блок виходів. Блок виходів формує функції системи (2).

Сучасні мікросхеми FPGA включають до свого складу такі CLB як LUT та EMB. Розглянемо особливості цих блоків [4, 9].

Елемент LUT має S_L входів та один вихід. Вхідна комбінація вибирає із осередків пам'яті LUT значення функції f_C ∈ {0, 1}. Таким чином, LUT реалізує довільну логічну функцію, що залежить не більше, ніж від S_L аргументів. Вихід f_C пов'язаний безпосередньо з входом синхронного тригера. По сигналу Clock виконується операція f_R := f_C, де f_R – вихід тригера. Тригер та LUT утворюють логічний елемент [9]. На вихід логічного елемента можна передати інформацію з виходу LUT, або з виходу тригера. Таким чином, логічні елементи на базі LUT можуть використовуватися для реалізації як комбінаційних, так і накопичувальних схем [9, 12].

Реєстр RG розподіляється між елементами блоку CLB ФЗП. Зазначимо, що за сигналом Start у RG записується код початкового стану a₁ ∈ A [6]. Як правило, цей код складається з одних нулів [6]. Позначатимемо блок, що складається з CLB на основі LUT, символом LUTer.

Основний недолік елементів LUT – вкрай обмежена кількість входів [10]. Для FPGA фірми Xilinx число входів S_L = 6 [11]. Це значення вважається оптимальним для дотримання балансу між характеристиками елемента LUT [10]. Проте навіть у автоматів середньої складності функції (1), (2) можуть залежати від 30 – 40 аргументів [6].

Така невідповідність між величинами S_L та $R + L$ веде до необхідності функціональної декомпозиції (ФД) систем (1), (2). Це призводить до багаторівневих схем блоків ФЗП [7, 8]. Якщо виконується умова

$$R > S_L, \tag{4}$$

то схема блоку виходів також багаторівнева.

Кількість елементів LUT у схемі МПА можна зменшити, якщо частина схеми реалізується за допомогою блоків ЕМВ [13]. Блоки ЕМВ мають можливість конфігурації [9 – 13], тобто зміни числа адресних входів (S_A) і виходів (t_F) при постійній ємності (V_0): $V_0 = 2^{S_A} \times t_F$.

Для ЕМВ фірми Xilinx $V_0 = 32$ Кбіт [11]. При цьому є такі пари : <15, 1>, <14, 2>, ..., <9, 64>. Умовимося називати пару конфігурацією ЕМВ.

Вочевидь, один ЕМВ може реалізувати до t_F функцій, які залежать не більше ніж S_A аргументів. Якщо існує конфігурація $\langle S_A^0, t_F^0 \rangle$, для якої $S_A^0 \geq L + R$ і $t_F^0 \geq R + N$, то відповідний МПА реалізується одним блоком ЕМВ (рис. 3).

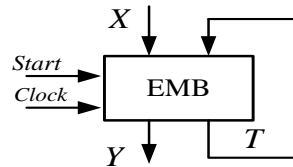


РИС. 3. Тривіальна реалізація МПА на ЕМВ

Як видно з рис. 3, регістр RG – частина блоку ЕМВ. Тривіальна реалізація МПА (рис. 3) призводить до схеми з оптимальними характеристиками [6]. Наші дослідження бібліотеки стандартних автоматів [14] показали, що для 67% автоматів така реалізація можлива. Для решти 33% необхідно більше одного блоку ЕМВ. Однак, дуже часто у розпорядженні розробника схеми ПУ є вкрай обмежена кількість блоків ЕМВ [13]. І тоді схема МПА представляється як композиції блоків ЕМВ і елементів LUT [12].

У цій роботі ми розглядаємо крайній випадок, коли для реалізації схеми ПУ можна використовувати лише один блок ЕМВ. Для реалізації систем (1), (2) у загальному випадку можна використовувати, наприклад, модель U_1 (рис. 4).

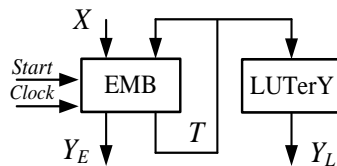


РИС. 4. Структурна схема МПА U_1

У МПА U_1 СБФ (1) реалізується на ЕМВ. Множина Φ розбивається на підмножини Y_E та Y_L . Такий підхід можливий, якщо виконується умова $(S_A^0 = L + R) \wedge (t_F^0 > R) = 1$.

Якщо виконується умова

$$(S_A^0 = L + R) \wedge (t_F^0 < R) = 1, \tag{5}$$

то множина Φ розбивається на підмножини Φ_E та Φ_L . Функції $D_r \in \Phi_L$ можуть залежати від ЛУ $x_1 \in X_L \subseteq X$. Такий підхід призводить до МПА U_2 (рис. 5).

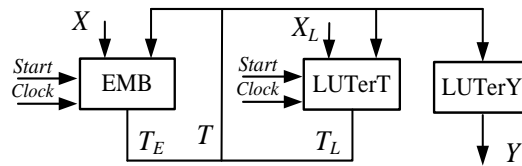


Рис. 5. Структурна схема МПА U_2

У МПА U_2 EMB генерує частину $K(a_m)$, представлену змінними $T_r \in T_E \subset T$. Інші змінні формуються як виходи блоку LUTerT. Очевидно, блок LUTerT включає частину RG ($R - t_F^0$ розрядів). Об'єднання T_E і T_L дає шину T , яка використовується як зворотний зв'язок між блоками МПА.

Недоліком МПА U_1 та U_2 є необхідність багаторівневої реалізації блоків LUTerY при виконанні умови (5). У цій роботі ми пропонуємо підхід, що дозволяє зменшити вплив величини R на блок LUTerY.

Основна ідея пропонованого методу. Нехай в операторних вершинах ГСА $\Gamma \in Q$ різних наборів. Закодуємо набори кодами $K(Y_q)$ розрядності $R_Q = \lceil \log_2 Q \rceil$.

Нехай виконуються умови (4) та

$$R_Q \leq S_L. \tag{6}$$

У цьому випадку ми пропонуємо представити коди стану як конкатенації кодів НМО і класів псевдоеквівалентних станів (ПЕС) [12]. Нагадаємо, що стани входять до одного класу ПЕС, якщо переходи з цих станів ідентичні. Це визначення дозволяє знайти розбиття Π_A множини A на класи ПЕС.

Наприклад, для ГСА Γ_1 можна знайти розбиття $\Pi_A = \{B_1, \dots, B_6\}$, де $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5\}$, $B_4 = \{a_6, a_7\}$, $B_5 = \{a_8\}$, $B_6 = \{a_9\}$. Отже, маємо $I = 6$ класів ПЕС. Для їх кодування необхідно $R_I = \lceil \log_2 I \rceil$ змінних, що утворюють множину $T_I \subseteq T$.

В операторних вершинах ГСА Γ записані такі НМО: $Y_1 = \emptyset$, $Y_2 = \{y_1, y_2, y_6\}$, $Y_3 = \{y_3, y_7\}$, $Y_4 = \{y_3, y_4, y_5, y_6\}$, $Y_5 = \{y_1, y_2\}$, $Y_6 = \{y_2, y_5\}$. Таким чином, $Q = 6$ та $R_Q = 3$. У запальному випадку змінні, що кодують НМО, утворюють множину $T_Y \subseteq T$.

Отже, $T = T_I \cup T_Y$, де $|T| = R_I + R_Q$. Якщо $*$ – знак конкатенації, то код $K(a_m)$ представляється в наступному вигляді:

$$K(a_m) = K(Y_q) * K(B_i). \tag{7}$$

Формула (7) визначає МПА Мура U_3 (рис. 6).

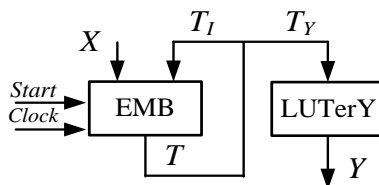


Рис. 6. Структурна схема МПА U_3

Як видно з рис. 6, конфігурація блоку пам'яті повинна задовольняти такій умові: $2^{L+R_I} \times (R_I + R_Y) \leq V_0$. Саме ця ситуація розглядається у даній статті.

Пропонований метод синтезу МПА $U_3(\Gamma_1)$ включає наступні етапи:

- 1) формування множини станів A ;
- 2) розбиття множини A на класи ПЕС;
- 3) кодування класів $B_i \in \Pi_A$;
- 4) формування наборів $Y_q \subseteq Y$;
- 5) кодування НМО $Y_q \subseteq Y$ кодами $K(Y_q)$;
- 6) формування прямої структурної таблиці МПА U_3 ;
- 7) формування системи функцій

$$Y = Y(T_Y); \tag{8}$$

- 8) формування таблиці блоку ЕМВ;
- 9) реалізація схеми МПА у заданому елементному базисі.

Приклад синтезу схеми МПА Мура $U_3(\Gamma_1)$

Перші два етапи синтезу для цього прикладу вже виконані. Отже, $I = 6$, що дає $RI = 3$ та $T_1 = \{T_1, T_2, T_3\}$. Закодуємо класи ПЕМ тривіальним чином: $K(B_1) = 000, \dots, K(B_6) = 101$.

Для цього прикладу маємо $Q = 6$, що дає $R_Q = 3$ і множину $T_Y = \{T_4, T_5, T_6\}$. Нехай для реалізації схеми МПА застосовуються елементи LUT, у яких $S_L = 3$. З формули (3) маємо $R = 4$. Таким чином, умова (4) виконується і блок LUTerY МПА, $U_2(\Gamma_1)$ має 2 рівня елементів.

Очевидно, умова (6) виконується, тому застосування запропонованого методу можливе. Нехай для реалізації схеми МПА можна використовувати один блок ЕМВ з конфігурацією $\langle 11, 6 \rangle$. Для цього прикладу $R_1 + R_Y = 6$ і $R_1 = 3$. Оскільки $L = 8$, то $R_1 + L = 11 = S_A^0$. При цьому $t_F^0 = 6$. Отже, блок ЕМВ точно підходить для реалізації СБФ (1). У цьому разі СБФ (1) може бути знайдено з ПСТ автомата $U_3(\Gamma_1)$.

Закодуємо НМО, використовуючи метод [15]. Це дозволяє зменшити кількість між'єднань між блоками ЕМВ та LUTerY. Один із варіантів кодування показаний картою Карно на рис.7.

		$T_4 T_5$			
		00	01	11	10
T_6	0	Y_1	Y_3	*	Y_5
	1	*	Y_4	Y_6	Y_2

РИС. 7. Коды наборів мікрооперацій автомата $U_3(\Gamma_1)$

Для формування ПСТ автомата U_3 необхідно:

- 1) побудувати систему узагальнених формул переходів (УФП) для класів $B_i \in \Pi_A$ [12];
- 2) подати коди станів $a_m \in A$ у вигляді (7);
- 3) побудувати ПСТ зі стовпцями $B_i, K(B_i), a_s, K(a_s), X_h, \Phi_h, h$.

Використовуючи правила [12], можна побудувати таку систему УФП для ДСА Γ_1 :

$$\begin{aligned}
 B_1 &\rightarrow \overline{x_1 x_2 a_2} \vee \overline{x_1 x_2 a_3} \vee \overline{x_1 x_3 a_4} \vee \overline{x_1 x_3 a_5}; \\
 B_2 &\rightarrow \overline{x_4 x_5 a_7} \vee \overline{x_4 x_5 a_5} \vee \overline{x_4 a_6}; & B_3 &\rightarrow \overline{x_5 a_7} \vee \overline{x_5 a_5}; \\
 B_4 &\rightarrow \overline{x_6 a_6} \vee \overline{x_6 a_8} & B_5 &\rightarrow a_9; & B_6 &\rightarrow \overline{x_7 a_1} \vee \overline{x_7 x_8 a_1} \vee \overline{x_7 x_8 a_8}.
 \end{aligned}
 \tag{9}$$

Коди станів формуються очевидним чином. Для нашого прикладу ці коди наведені в табл. 2.

ТАБЛИЦЯ 2. Коди станів МПА $U_3(\Gamma_1)$

a_m	$K(a_m)$		a_m	$K(a_m)$		a_m	$K(a_m)$	
	$K(B_i)$	$K(Y_q)$		$K(B_i)$	$K(Y_q)$		$K(B_i)$	$K(Y_q)$
	$T_1T_2T_3$	$T_4T_5T_6$		$T_1T_2T_3$	$T_4T_5T_6$		$T_1T_2T_3$	$T_4T_5T_6$
a_1	000	000	a_4	001	011	a_7	011	100
a_2	001	101	a_5	010	100	a_8	100	111
a_3	001	010	a_6	011	010	a_9	101	100

У табл. 2 коди $K(Y_q)$ взято з рис. 7. З цієї таблиці маємо, наприклад, $K(a_1) = 000000$, $K(a_2) = 0011001$, $K(a_3) = 001010$ тощо.

Для формування ПСТ автомата $U_3(\Gamma_1)$ використовуємо систему (9). Коди станів переходу беруться із табл. 2. У результаті маємо табл. 3.

ТАБЛИЦЯ 3. ПСТ автомата Мура $U_3(\Gamma_1)$

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Φ_h	h
B_1	000	a_2	001101	x_1x_2	$D_3 D_4 D_6$	1
		a_3	001010	$x_1\overline{x_2}$	$D_3 D_5$	2
		a_4	001011	$\overline{x_1}x_3$	$D_3 D_5 D_6$	3
		a_5	010100	$\overline{x_1}\overline{x_3}$	$D_2 D_4$	4
B_2	001	a_7	011100	$\overline{x_4}x_5$	$D_2 D_3 D_4$	5
		a_5	010100	x_4x_5	$D_2 D_4$	6
		a_6	011010	x_4	$D_2 D_3 D_5$	7
B_3	010	a_7	000000	x_5	$D_2 D_3 D_4$	8
		a_5	011100	$\overline{x_5}$	$D_2 D_4$	9
B_4	011	a_6	011010	x_6	$D_2 D_3 D_4$	10
		a_8	100111	$\overline{x_6}$	$D_1 D_4 D_5 D_6$	11
B_5	100	a_9	101100	1	$D_1 D_3 D_4$	12
B_6	101	a_1	000000	x_7	—	13
		a_1	000000	$\overline{x_7}x_8$	—	14
		a_8	100111	$\overline{x_7}\overline{x_8}$	$D_1 D_4 D_5 D_6$	15

Система (8) формується на основі вмісту наборів мікрооперацій. Для формування термів системи (8) використовуються коди $K(Y_q)$. У нашому прикладі маємо таку систему функцій:

$$\begin{aligned}
 y_1 &= Y_2 \vee Y_5 = T_4 \overline{T_5}; & y_2 &= Y_2 \vee Y_5 \vee Y_6 = T_4; & y_3 &= Y_3 \vee Y_4 = \overline{T_4} T_5; \\
 y_4 &= Y_4 = \overline{T_4} T_6; & y_5 &= Y_4 \vee Y_6 = T_5 T_6; & y_6 &= Y_2 \vee Y_4; & y_7 &= Y_3 = T_5 \overline{T_6}.
 \end{aligned}
 \tag{10}$$

Як випливає з СБФ (10) блок LUTerY включає 6 елементів LUT. Функція y_2 реалізується як вихід тригера T_4 . Тільки функція y_6 залежить від усіх змінних. Число аргументів у термах (10) дорівнює 13, що відповідає 13 міжз'єднанням між блоками ЕМВ та LUTerY. У загальному випадку число міжз'єднань визначається як $N \times R_Q = 21$. Таким чином, оптимальне кодування НМО дозволило зменшити число міжз'єднань в 1,62 рази.

Таблиця блоку ЕМВ формується на основі табл. 3 і має наступні стовпці: B_i , $K(B_i)$, X (адреса осередку пам'яті), Φ (вміст осередку пам'яті), h . Таблиця має H_0 рядків, де $H_0 = 2^{R_1+L}$.

Таблиця блоку ЕМВ складається з I підтаблиць, при цьому підтаблиця i відповідає блоку $B_i \in \Pi_A$ і має H_i рядків, де $H_i = 2^L$ ($i \in \{1, \dots, I\}$).

Для нашого прикладу маємо $L + R_1 = 11$, що дає 2048 рядків. Кожен рядок відповідає одному осередку блоку ЕМВ. Кожна підтаблиця має $H_i = 2^8 = 256$ рядків. Оскільки $I = 6$, ПСТ (табл. 3) містить $I \times H_i = 6 \times 256 = 1536$ рядків. Таким чином, $(H_0 - I \times H_i)$ рядків не містять корисної інформації.

Частина таблиці блоку ЕМВ для прикладу наведена в табл. 4. Тут наведено 10 перших рядків підтаблиці для класу $B_6 \in \Pi_A$. Попередні підтаблиці займають $256 \times 5 = 1280$ рядків. Тому табл. 4 починається з $h = 1281$.

ТАБЛИЦЯ 4. Фрагмент таблиці блоку ЕМВ

B_i	$K(B_i)$	X $x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8$	Φ $D_1 D_2 D_3 D_4 D_5 D_6$	h
	$T_1 T_2 T_3$			
B_6	1 0 1	0 0 0 0 0 0 0 0	1 0 0 1 1 1	1281
B_6	1 0 1	0 0 0 0 0 0 0 1	0 0 0 0 0 0	1282
B_6	1 0 1	0 0 0 0 0 0 1 0	0 0 0 0 0 0	1283
B_6	1 0 1	0 0 0 0 0 0 1 1	0 0 0 0 0 0	1284
B_6	1 0 1	0 0 0 0 0 1 0 0	0 0 0 0 0 0	1285
B_6	1 0 1	0 0 0 0 0 1 0 1	0 0 0 0 0 0	1286
B_6	1 0 1	0 0 0 0 0 1 1 0	0 0 0 0 0 0	1287
B_6	1 0 1	0 0 0 0 0 1 1 1	0 0 0 0 0 0	1288
B_6	1 0 1	0 0 0 0 1 0 0 0	1 0 0 1 1 1	1289
B_6	1 0 1	0 0 0 0 1 0 0 1	0 0 0 0 0 0	1290

Існує очевидний зв'язок між ПСТ (табл. 3) та таблицею блоку ЕМВ (табл. 4). У табл. 4 маємо $K(B_i) = 101$, що відповідає класу $B_6 \in \Pi_A$. Таким чином, табл. 4 відповідає рядкам 13 – 15 ПСТ (табл. 3). Якщо $x_7 = 1$, то відповідні 128 рядків аналогічні рядку 13 ПСТ (табл. 3). Якщо $x_7 = 0$ і

$x_8 = 1$, то 64 рядки відповідають рядку 14. Якщо $x_7 = x_8 = 0$, це відповідає рядку 15. Тільки рядок 15 містить інформацію, відмінну від нуля. Ця інформація записується в 64 рядках, наприклад, 1281, 1289, 1297 і так далі. Якщо $x_7 = x_8 = 1$, то в табл. 3 немає відповідних переходів. Однак у табл. 4 ця ситуація відповідає рядкам з $x_7 = 1$.

Останній етап синтезу пов'язаний із використанням промислових САПР. У разі використання FPGA фірми Xilinx потрібна САПР Vivado [16]. На цьому етапі формуються таблиці істинності кожного елемента LUT і блоку ЕМВ. Необхідно виконати розміщення та трасування, а також сформулювати двійковий потік (bit stream) для елементів та з'єднань. Ми не розглядаємо цей етап для нашого прикладу.

Висновок. Найкращі характеристики мають схеми автоматів, реалізовані за допомогою блоків вбудованої пам'яті ЕМВ [4, 9, 13]. Проте цілком можлива ситуація, коли необхідної кількості блоків немає. У цьому випадку схема МПА є мережею, що складається з блоків ЕМВ та елементів LUT. При цьому виникає проблема оптимізації кількості елементів LUT у схемі МПА. У цій роботі запропоновано метод вирішення цієї проблеми, якщо кількість внутрішніх змінних $T_r \in T$ перевищує кількість входів елементів LUT.

Пропонований метод заснований на використанні класів псевдоеквівалентних станів автомата Мура. Стан кодується таким чином, що код стану складається з коду класу ПЕС і коду набору мікрооперацій. Такий підхід збільшує розрядність коду стану порівняно з мінімальним значенням, що визначається формулою (3).

Пропонований метод дозволяє оптимізувати схему блоку виходів, якщо виконується умова (6). У цьому випадку блок виходів складається з не більше ніж N елементів, де N – число мікрооперацій. Значимо, що за умови (4) блок виходів має як мінімум $2N$ елементів LUT.

Ми досліджували ефективність запропонованого методу для стандартних автоматів із бібліотеки [14] на FPGA сімейства Virtex-7 фірми Xilinx, що мають $S_L = 6$ [17]. Для вирішення задач, пов'язаних з технологічним відображенням схем автоматів, використано САПР Vivado [16]. Результати досліджень показали, що запропонований метод дозволяє зменшити число LUT у схемах МПА на 12% – 19% порівняно з відомими методами. У всіх випадках використовувався лише один блок ЕМВ.

Надалі ми плануємо розробити метод оптимізації схеми МПА для випадку, коли ресурсів ЕМВ не вистачає для реалізації системи функцій збудження пам'яті. Ця ситуація відповідає МПА U_2 (рис. 5). В цьому випадку ми плануємо використовувати метод заміни логічних умов [12]. Цей підхід може бути використаний для оптимізації схем сполучених МПА [18, 19].

Список літератури

1. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. М.: Горячая линия – ТЕЛЕКОМ, 2001. 636 с.
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. Berlin: Springer, 2021.
4. Grout I. Digital systems design with FPGAs and CPLDs. Amsterdam: Elsevier, 2008. 784 p.
5. Глушков В.М. Синтез цифровых автоматов. М.: Физматгиз, 1962. 476 с.
6. Baranov S. Logic synthesis for control automata. Dordrecht: Kluwer Academic Publishers, 1994. 312 p.

7. Barkalov A., Titarenko L., Mielcarek K., and Chmielewski S.. Logic Synthesis for FPGA–Based Control Units. – Structural Decomposition in Logic Design. Vol. 636 of Lecture Notes in Electrical Engineering. Springer, 2020. <https://doi.org/10.1007/978-3-030-38295-7>
8. Barkalov A.A., Titarenko L.A. Code transformation in compositional microprogram control units. *Cybernetics and Systems Analysis*. 2011. **47** (5). P. 763–772. <https://doi.org/10.1007/s10559-011-9355-x>
9. Maxfield C. The design warrior’s guide to FPGAs. Orlando: Academic Press. 2004. 542 p.
10. 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>
11. UG473 (v1.14. July 3. 2019). <https://www.xilinx.com/> (звернення: 14.08.2022)
12. Sklyarov V., Skliarova I., Barkalov A., Titarenko L. Synthesis and optimization of FPGA-based systems. Berlin: Springer. 2014. 432 p. https://doi.org/10.1007/978-3-319-04708-9_6
13. 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.
14. Yang S. Logic synthesis and optimization benchmarks user guide. Version 3.0. Techn. Rep. Microelectronics Center of North Carolina, 1991. 43 p.
15. Ачасова С.М. Алгоритмы синтеза автоматов на программируемых матрицах. М.: Радио и связь, 1987. 136 с.
16. Vivado Design Suite. <https://www.xilinx.com/products/design-tools/vivado.html> (звернення: 01.01.2020)
17. VC709 Evaluation Board for the Virtex-7 FPGA. User Guide; UG887 (v1.6); Xilinx, Inc.: San Jose, CA, USA, 2019.
18. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Реализация схемы совмещенного автомата в базе се FPGA. *Комп’ютерні засоби, мережі та системи*. 2016. С. 10–19. <http://dspace.nbuu.gov.ua/handle/123456789/122858>
19. Баркалов А.А., Титаренко Л.А., Визор Я.Е., Матвиенко А.В. Кодирование выходных переменных в совмещенном автомате. *Комп’ютерні засоби, мережі та системи*. 2018. С. 73–80. <http://dspace.nbuu.gov.ua/handle/123456789/150610>

Одержано 14.08.2022

Баркалов Александр Александрович,

доктор технічних наук, професор Університету Зеленогурського (Польща),

<https://orcid.org/0000-0002-4941-3979>A.Barkalov@iee.uz.zgora.pl**Титаренко Лариса Олександрівна,**

доктор технічних наук, професор Університету Зеленогурського (Польща),

професор Харківського національного університету радіоелектроніки,

<https://orcid.org/0000-0001-9558-3322>L.Titarenko@iee.uz.zgora.pl**Головін Александр 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**Матвиенко Александр Володимирович,**

науковий співробітник Інституту кібернетики імені В.М. Глушкова НАН України, Київ.

<https://orcid.org/0000-0003-1838-1422>avmatv@ukr.net

UDC 004.274

Alexander Barkalov¹, Larysa Titarenko^{1,2}, Oleksandr Golovin³, Oleksandr Matvienko^{3*}

Optimization of a Moore Automaton Circuit in a Mixed Element Basis

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. The control unit is one of the most important building blocks of any digital system. The main function of the control unit is to coordinate the interaction between all system blocks. Therefore, the characteristics of a control unit circuit have a significant impact on the quality of the system as a whole.

To represent the law of functioning of a control unit, the models of the Moore and Mealy finite state machines (FSM) are used. When synthesizing circuits of FSMs, it is necessary to solve a number of optimization problems, such as the reducing hardware amount, increasing performance, minimizing power consumption, joint optimization of hardware-temporal characteristics. Methods for solving these problems largely depend on the used logical elements.

Currently, FPGA microchips are one of the main platforms in which modern digital systems are implemented. The main blocks in the FPGA, which are used in the implementation of FSM circuits, are embedded memory blocks (EMBs), logical blocks LUT (look-up table) and a system of programmable interconnections. The best characteristics are possessed by FSM circuits implemented with EMBs. However, EMBs are widely used to implement various operational blocks of digital systems. Therefore, the designer of a control unit circuitry can use a rather limited number of EMBs.

Purpose of the article. The article deals with the extreme case when there is only a single "free" EMB available. In this case, the FSM circuit is represented by a network consisting of this EMB and LUTs. There is proposed a method for the synthesis of an FSM with the optimization of the number of LUTs, when only one EMB block is available for implementing some part of the circuit.

The proposed method is based on the using classes of pseudoequivalent states of Moore FSMs. The states are coded in such a way that the state code consists of the code of the class of pseudoequivalent states and the code of a collection of microoperations.

Results. Studies of the effectiveness of the proposed method were carried out on standard FSMs. FPGAs of the Virtex-7 family from Xilinx were used as an implementation platform. The research results showed that the proposed method allows reducing the number of LUTs in FSM circuits by 12 % – 19 % in comparison with the known methods. In all cases, only a single EMB was used.

Conclusions. The effectiveness of the proposed method allows us to recommend it for use in the synthesis of FSMs if there is of an extreme shortage of EMBs.

Keywords: Moore FSM, FPGA, LUT, EMB, pseudoequivalent states, synthesis.