Кібернетика та комп'ютерні технології

УДК 004.274 DOI:10.34229/2707-451X.24.4.11

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

ОПТИМІЗАЦІЯ СХЕМИ КОМПОЗИЦІЙНОГО МІКРОПРОГРАМНОГО ПРИСТРОЮ УПРАВЛІННЯ З БАЗОВОЮ АРХІТЕКТУРОЮ

Запропоновано метод зменшення кількості елементів LUT у схемі композиційного пристрою управління, що реалізується у базисі FPGA. Оптимізація досягається за рахунок перетворення адрес виходів операторних лінійних ланцюгів у часткові коди. Цей підхід призводить до схеми адресації мікрокоманд, з трьома рівнями і регулярною структурою. Подібні схеми мають кращі характеристики, ніж їх аналоги, засновані на функціональній декомпозиції. Розглянуто приклад синтезу пристрою управління із запропонованою структурою. Показано умови, за виконанням яких запропонований метод дає найкращі результати.

Ключові слова: *композиційні мікропрограмні пристрої управління, LUT, EMB, синтез.*

Вступ. Пристрій управління (ПУ) один із найважливіших блоків цифрових систем [1]. На відміну від інших блоків, ПУ генерує сигнали у кожному такті роботи системи. У процесі генерації споживається значна кількість електричної потужності [2]. Наразі проблема зменшення споживаної потужності набуває особливого значення [3, 4]. Одним із способів вирішення цієї проблеми є регуляризація схеми ПУ та зменшення числа міжз'єднань між її елементами [3, 5]. У цій статті пропонується метод вирішення цього завдання при реалізації схем композиційних мікропрограмних пристроїв управління (КМПУ) [6, 7] у базисі FPGA (field-programmable logic array) [8].

Мікросхеми FPGA широко застосовуються під час реалізації різних цифрових систем [9]. На думку експертів, ці мікросхеми домінуватимуть у цифровому проектуванні ще кілька десятиліть [8]. Цим фактором обґрунтовано вибір саме цього елементного базису. Пропонований метод орієнтований на FPGA, вироблені фірмою AMD Xilinx [10]. Цей вибір зумовлений провідною позицією фірми на ринку мікросхем FPGA [11].

Основна ідея запропонованого методу зводиться до адаптації методу подвійного кодування станів до особливостей КМПУ з базовою архітектурою. Аналогами стану є мікрокоманди КМПУ. Тому оптимізація досягається за рахунок подвійної адресації мікрокоманд.

Пропонований метод покращує характеристики схеми адресації мікрокоманд. За певних умов схема адресації є дворівневою. Схема включає мінімально можливе число елементів і має регулярну систему міжз'єднань.

Особливості КМПУ та базису FPGA.

КМПУ ґрунтуються на наявності операторних лінійних ланцюгів (ОЛЛ) алгоритму управління [12]. У цій роботі для представлення алгоритмів управління використовується мова граф-схем алгоритмів (ГСА) [13].

Множину операторних вершин ГСА позначимо $B=\{b_1,..., b_M\}$. Кожна ОЛЛ α_g включає максимально можливе число вершин. Внаслідок цього формування ОЛЛ множина B розбивається на G класів, які утворюють множину ОЛЛ.

Кожна операторна вершина відповідає унікальному осередку управляючої пам'яті (УП). Якщо $|B|=M$, розрядність адреси УП визначається наступним чином:

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

Характерною властивістю ОЛЛ є наявність безумовних переходів між її сусідніми елементами. Якщо ОЛЛ α_g включає пару $< b_i, b_j >$, існує наступна залежність між адресами $A(b_i)$ і $A(b_j)$ [6]:

$$A(b_j)=A(b_i)+1. \quad (2)$$

У рівнянні (2) символ $A(b_k)$ відповідає адресі мікрокоманди (МК), що визначається вершиною $b_k \in B$.

Для виконання природної адресації МК (2) схема КМПУ включає лічильник адреси мікрокоманди (ЛА). ЛА складається з R тригерів, виходи яких утворюють множину $T=\{T_1, ..., T_R\}$. Зазвичай [5], це синхронні тригери типу D. Для зміни адреси в ЛА використовуються два підходи. При переході між сусідніми вершинами ОЛЛ до вмісту ЛА додається одиниця. При переході між різними ОЛЛ адреса переходу формується за допомогою функцій збудження пам'яті (ФЗП) $D_r \in D =\{D_1, ..., D_R\}$.

Мікрокоманди складаються з мікрооперацій $y_n \in Y$, де $Y=\{y_1, ..., y_N\}$. В результаті мікрооперації є функціями адресних змінних $T_r \in T$. Функції $D_r \in D$ залежать від змінних $T_r \in T$ і логічних умов (ЛУ) $x_l \in X=\{x_1, ..., x_L\}$. Логічні умови записані в умовних вершинах ГСА, і служать для вибору чергової гілки алгоритму управління.

Схема адресації мікрокоманд (САМ) представляється системою булевих функцій (СБФ):

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

Управляюча пам'ять реалізує: СБФ $Y=Y(T)$.

Для керування режимом роботи лічильника використовується додаткова змінна y_0. Якщо $y_0=1$, до вмісту ЛА додається 1. Якщо $y_0=0$, вміст ЛА визначається ФЗП (3). Зміна вмісту ЛА можлива лише за певним фронтом сигналу синхронізації Clock. Для запису початкової адреси ЛА використовується сигнал Start. Як правило, початкова адреса є нульовою [7].

КМПУ з базовою архітектурою U1 має структурну схему, показану на рис. 1.

УП реалізується як поєднання стандартних блоків пам'яті [7]. Для оптимізації УП можна використовувати метод кодування наборів мікрооперацій [13]. Схема блоку САМ реалізується на логічних елементах. Для зменшення числа цих елементів необхідно враховувати особливості елементного базису.

У статті розглядається випадок реалізації КМПУ у базисі FPGA. Для реалізації схеми використовуються такі ресурси мікросхеми: елементи табличного типу LUT (look-up table), вбудовані блоки пам'яті EMB (embedded memory blocks) та програмовані міжз'єднання [8].

Елементи LUT є блоками однорозрядної пам'яті, що мають S_L адресних входів. Елемент LUT зберігає таблицю істинності булевської функції, яка залежить від не більше, ніж S_L аргументів. Ці елементи використовуються для реалізації САМ.

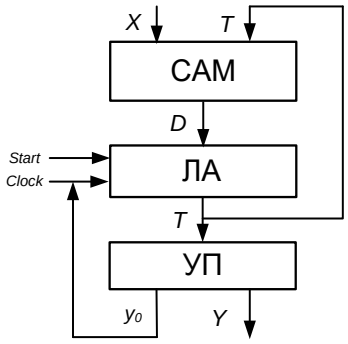

РИС. 1. Структурна схема КМПУ U_1

Блоки ЕМВ можуть змінювати свою конфігурацію (число адресних входів S_A та виходів t_F пам'яті). При цьому зберігається загальна кількість біт пам'яті

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

Питання побудови пам'яті з використанням ЕМВ є досить дослідженим [3]. Ми його не розглядаємо.

Вихід елемента LUT може бути з'єднаний із входом тригера. Це дозволяє організовувати регістри та лічильники. Такі пристрої називаються «прихованими» [14]. Елементи LUT об'єднуються у блок LUTerT.

З цього аналізу можна отримати структурну схему F-КМПУ U_1 (рис. 2). Літера "F" означає, що схема реалізована на базі FPGA.

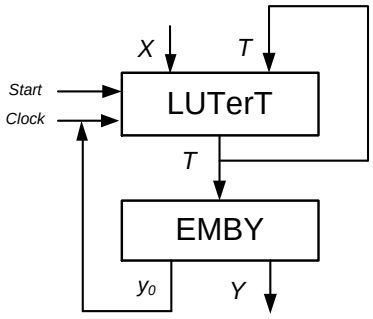

РИС. 2. Структурна схема F-КМПУ U_1

Основна проблема під час реалізації схеми блоку LUTerT пов'язана з вкрай малим числом входів елементів LUT. Нехай функція $D_r \in D$ є диз'юнктивною нормальною формою (ДНФ), що має $NA(D_r)$ аргументів. Нехай виконується умова

$$NA(D_r) > S_L . \tag{4}$$

У цьому випадку схема, що генерує $D_r \in D$, буде багаторівневою та багатоелементною.

Для реалізації таких схем використовується метод функціональної декомпозиції [15]. Подібні методи призводять до схем з нерегулярною структурою та складною системою міжз'єднань. Такі схеми є повільними та споживають багато енергії.

У цій роботі ми розглядаємо випадок, коли умова (4) виконується. Для оптимізації характеристик схеми блоку LUTerT ми пропонуємо використовувати метод, аналогічний до подвійного кодування станів [16]. Цей метод відноситься до групи методів структурної декомпозиції [17].

Основна ідея запропонованого методу

Нехай для деякої ГСА Г знайдено множину ОЛЛ $C = \{\alpha_1, ..., \alpha_G\}$. Кожна ОЛЛ має хоча б один вхід і лише один вихід $O_g \in O$ (Г). Тут $O(\Gamma)$ – множина виходів ОЛЛ, що має точно G елементів. Формальне визначення ОЛЛ можна знайти, наприклад, у роботі [7]. Там же розглядаються методи формування множини C. Зазначимо, що вихід є останнім компонентом вектора $\alpha_g \in C$.

Виконаємо природну адресацію мікрокоманд у межах кожної ОЛЛ. При цьому мікрокоманда Y_m відповідає операторній вершині $b_m \in B$. Тепер кожному виходу $O_g \in O$ (Г) відповідає адреса, що має R розрядів.

Знайдемо розбиття $\Pi_O = \{O^1, ..., O^K\}$ множини $O(\Gamma)$ на класи сумісних ОЛЛ. Кожен клас $O^k \in \Pi_O$ визначає множини X^k і D^k. Множина $X^k \subseteq X$ включає логічні умови, що визначають переходи з виходів ОЛЛ $O_g \in O^k$. Множина $D^k \subseteq D$ включає ФЗП, що формуються на переходах з виходів $O_g \in O^k$.

Нехай клас O^k містить M_k елементів. Закодуємо ці виходи частковими кодами $K(O_g)$, що мають R_k розрядів, де $R_k = \lceil \log_2(M_k + 1) \rceil$.

Тут одиниця додана для врахування співвідношення $O_g \notin O^k$. Будемо використовувати елементи множини τ^k для кодування виходів, де $|\tau^k| = R_k$.

Нехай $|X^k| = L_k$. Для кожного класу $O^k \in \Pi_O$ виконується умова

$$R_k + L_k \le S_L. \tag{5}$$

У цьому випадку будь-яка часткова ФЗП $D_r^k \in D^k$ представляється схемою, що складається з одного елемента LUT. Очевидно, що клас $O^k \in \Pi_O$ відповідає блоку LUTerk, що складається з елементів LUT.

Зазначимо, що виходи $O_i, O_j \in O$ (Г) є сумісними, якщо їх включення до класу O^k не порушує умови (5). Таким чином, виходи ОЛЛ можуть включатися до різних класів $O^k \in \Pi_O$ на підставі аналізу умови (5). Для формування розбиття Π_O ми пропонуємо використовувати метод з роботи [16].

Кожен блок LUTerk ($k \in \{1, ..., K\}$) визначається наступною СБФ:

$$D^k = D^k(\tau^k, x^k). \tag{6}$$

Змінні, що кодують виходи ОЛЛ, утворюють множину $\tau = \tau^1 \cup \tau^2 \cup ... \cup \tau^k$. Множина τ складається з R_O елементів:

$$R_O = R_1 + R_2 + ... + R_K. \tag{7}$$

Функції (6) є частковими. Для формування остаточних значень функцій $D_r \in D$ необхідно використовувати функціональний асемблер, що реалізує СБФ:

$$D = D(D^1,\ D^2,...,\ D^K)\ .$$

Кожна з функцій $D_r \in D$ є диз'юнкцією часткових функцій:

$$D_r = D_r^1 \vee D_r^2 \vee ... \vee D_r^K\ . \tag{8}$$

Для формування змінних $\tau_r \in \tau$ необхідно використовувати перетворювач адрес мікрокоманд. Цей блок реалізує СБФ

$$\tau\ =\ \tau\ (T)\ . \tag{9}$$

З цих міркувань, ми пропонуємо архітектуру F-КМПУ U_2 (рис. 3).

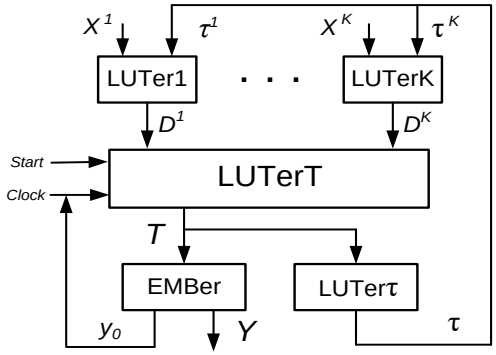

РИС. 3. Структурна схема F-КМПУ U_2

У КМПУ U_2 блоки LUTer1-LUTerK генерують функції (6). Це перший рівень логіки. Блок LUTerT виконує функції асемблера. Цей блок включає R тригерів, що утворюють лічильник адреси мікрокоманд. Тому сигнали Start, Clock є входами цього блоку. Виходи тригерів $T_r \in T$ утворюють адреси мікрокоманд. Щоб показати, що пам'ять може складатися із кількох блоків EMB, позначимо УП символом EMBer. Блок LUTerτ реалізує СБФ (9).

При виконанні умов (5) та

$$R \le S_L \tag{10}$$

схема КМПУ U_2 включає три рівні елементів LUT. У нашій статті ми розглядаємо цю ситуацію.

Для реалізації схеми КМПУ U_2 пропонується наступний метод:

- Формування розбиття $C\ =\{\alpha_1,\ ...,\ \alpha_G\}$ множини операторних вершин.
- Природна адресація мікрокоманд у межах кожної ОЛЛ.
- Формування розбиття Π_O множини виходів $O(\Gamma)$ на класи сумісних виходів.
- Кодування виходів усередині кожного класу $O^k \in \Pi_O$.
- Формування таблиць переходів для класів $O^k \in \Pi_O$.
- Формування СБФ (6).
- Формування таблиці блоку LUTerT та СБФ (8).
- Формування таблиці блоку LUTer і СБФ (9).
- Формування таблиці блоку EMBerY.
- Реалізація схеми КМПУ із використанням внутрішніх ресурсів схеми FPGA.

Приклад синтезу КМПУ U_2

Нехай символ $U_i(\Gamma_j)$ означає, що схема КМПУ U_i реалізується за ГСА Γ_j. Розглянемо приклад реалізації схеми F-КМПУ $U_2(\Gamma_1)$.

Граф схема алгоритму Γ_1 показано на рис. 4. Для реалізації схеми КМПУ використовуємо елементи LUT, що мають $S_L = 4$ входи.

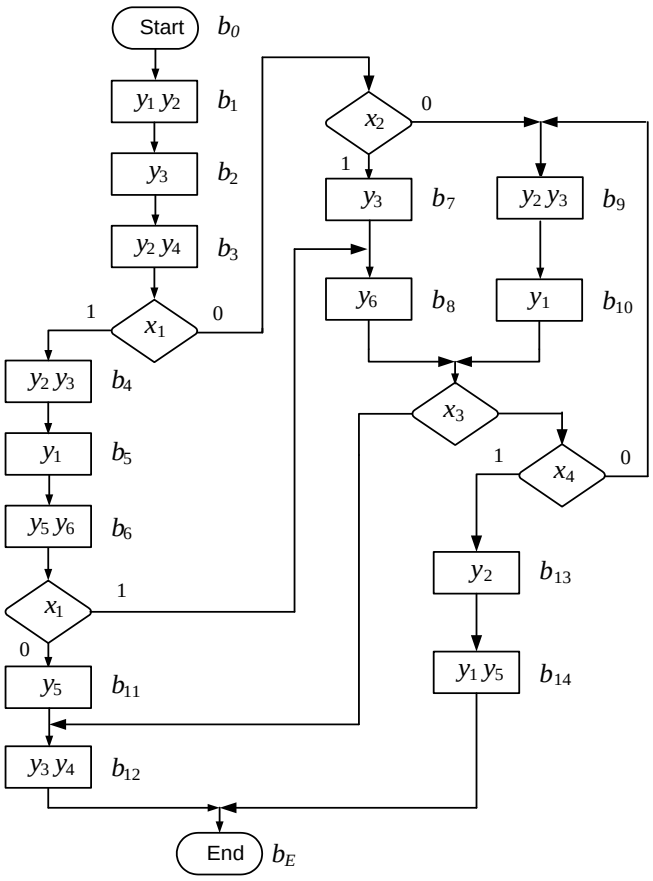

РИС. 4. Граф схема алгоритму Γ_1

Аналіз Γ_1 дає множини, $B=\{b_1,..., b_{14}\}$, $X=\{x_1, \dots, x_4\}$ і $Y=\{y_1, \dots, y_6\}$. Це визначає параметри $M=14$, $L=4$ и $N=6$. Використовуючи (1), знаходимо $R=4$. Це дає множини $T=\{T_1, \dots, T_4\}$ і $D=\{D_1, \dots, D_4\}$.

Кожна ОЛЛ це послідовність операторних вершин, між якими немає умовних вершин [7]. Кожна вершина може бути елементом лише однієї ОЛЛ. Використовуючи метод з роботи [7], сформуємо таку множину для ГСА Γ_1: $C=\{\alpha_1, ..., \alpha_6\}$. Множина включає наступні ОЛЛ: $\alpha_1 = < b_1, b_2, b_3 >$, $\alpha_2 = < b_4, b_5, b_6 >$, $\alpha_3 = < b_7, b_8 >$, $\alpha_4 = < b_9, b_{10} >$, $\alpha_5 = < b_{11}, b_{12} >$ і $\alpha_6 = < b_{13}, b_{14} >$. Останні компоненти ОЛЛ є їх виходами. Це дає множину $O(\Gamma_1) = \{O_1, ..., O_6\}$, де $O_1 = b_3$, $O_2 = b_6$, $O_3 = b_8$, $O_4 = b_{10}$, $O_5 = b_{12}$ і $O_6 = b_{14}$.

Природна адресація МК здійснюється за алгоритмом, запропонованому у [7]. Тоді адреси будуть такими: $A(Y_1)=0000$, $A(Y_2)=0001,...,$ $A(Y_{14})=1101$.

Для формування множини Π_O використовується метод, розглянутий у [16]. Цей спосіб орієнтований на автомат Мілі. Однак, ми модифікували даний метод з урахуванням особливостей КМПУ U_2. Оскільки $R = 4$ і $S_L = 4$, неможливо представити ФЗП для МК Y_3, Y_6, Y_8, Y_{10} у вигляді формули, що реалізується одним елементом LUT. Тому в даному випадку має бути застосовано наш метод.

Застосовуючи цей метод отримаємо розбиття $\Pi_O = \{O^1, O^2\}$, де $O^1 = \{O_1, O_2, O_5\}$ і $O^2 = \{O_3, O_4, O_6\}$. Ці класи визначають множини $X^1 = \{x_1, x_2\}$ і $X^2 = \{x_3, x_4\}$ з $L1 = L2 = 2$. Кожен клас має три елементи $(M_1 = M_2 = 3)$. Використання [3] дає $R_1 = R_2 = 2$. Отже, для кожного класу виконується умова $2 + 2 = 4 = S_L$. Очевидно, згідно з (7), маємо $R_O = 4$.

Оскільки $R_1 = 2$ і $R_2 = 2$, отримаємо $\tau^1 = \{\tau_1, \tau_2\}$ і $\tau^2 = \{\tau_3, \tau_4\}$. В результаті ми маємо множину $\tau = \{\tau_1, ..., \tau_4\}$. Використовуємо код 00 для співвідношення $O_m \notin O^k$. Закодуємо виходи у такий спосіб: $K(O_1) = K(O_3) = 01$, $K(O_2) = K(O_4) = 10$ і $K(O_5) = K(O_6) = 11$. Ці часткові коди використовуються у таблицях переходів для класів O^1 і O^2.

Таблиця блоку LUTerk містить такі стовпці: O_m, $K(O_m)$, Y_S, $A(Y_S)$, X_h, D_h, h. Стовпці таблиці мають наступний зміст: O_m – виходи ОЛЛ, що входять до класу $O^k \in \Pi_O$; $\mathrm{K}(\mathrm{O_m})$ – часткові коди цих виходів; Y_S – МК, що відповідають вершинам переходу з вершини-виходу ОЛЛ; $A(Y_S)$ – адреси переходів; X_h – кон'юнкції $< O_m, Y_S >$; D_h – функції $D_r \in D$, що дорівнюють значенню бітів у адресі $A(Y_S)$; h – номер переходу.

Для побудови таблиць блоків LUTerk необхідно отримати формули переходів із вершин-виходів ОЛЛ [7]. У цьому прикладі це така система:

$$O_1 \to x_1 b_4 \vee \overline{x_1} x_2 b_7 \vee \overline{x_1}\,\overline{x_2} b_9; \qquad O_3, O_4 \to x_3 b_{12} \vee \overline{x_3} x_4 b_{13} \vee \overline{x_3}\,\overline{x_4} b_9;$$
$$O_2 \to x_1 b_8 \vee \overline{x_1} b_{11}; \qquad O_5, O_6 \to b_E.$$

Для побудови таблиці блоку LUTer1 використовуються формули для O_1, O_2 і O_5, для побудови таблиці блоку LUTer2 – формули для виходів O_3, O_4 і O_6. В результаті маємо табл.1 та табл. 2.

ТАБЛИЦЯ 1. Таблиця блоку LUTer1

O_m	$K(O_m)$	Y_S	$A(Y_S)$	X_h	D_h	
O_1	01	Y_4	0011	x_1	$D_3 D_4$	
		Y_7	0110	$\overline{x_1} x_2$	$D_2 D_3$	
		Y_9	1000	$\overline{x_1}\,\overline{x_2}$	D_1	
O_2	10	Y_8	0111	x_1	$D_2 D_3 D_4$	
		Y_{11}	1010	$\overline{x_1}$	$D_1 D_3$	
O_5	11	Y_E	0000	1	—	

З табл. 1 формується СБФ для LUTer1:

$$D_1^1 = \overline{\tau}_1\tau_2\overline{x}_1\overline{x}_2 \vee \tau_1\overline{\tau}_2\overline{x}_1; \qquad D_3^1 = \overline{\tau}_1\tau_2 x_1 \vee \overline{\tau}_1\tau_2\overline{x}_1 x_2 \vee \tau_1\overline{\tau}_2;$$

$$D_2^1 = \overline{\tau}_1\tau_2\overline{x}_1 x_2 \vee \tau_1\overline{\tau}_2 x_1; \qquad D_4^1 = \overline{\tau}_1\tau_2 x_1 \vee \tau_1\overline{\tau}_2 x_1.$$

З табл. 2 формується СБФ для LUTer2:

$$D_1^2 = \overline{\tau}_3\tau_4 \vee \tau_3\overline{\tau}_4; \qquad D_3^2 = \overline{\tau}_3\tau_4 x_3 \vee \tau_3\overline{\tau}_4 x_3;$$

$$D_2^2 = \overline{\tau}_3\tau_4\overline{x}_3 x_4 \vee \tau_3\overline{\tau}_4\overline{x}_3 x_4; \qquad D_4^2 = D_3^2.$$

ТАБЛИЦЯ 2. Таблиця блоку LUTer2

O_m	$K(O_m)$	Y_S	$A(Y_S)$	X_h	D_h	h
O_3	01	Y_{12}	1011	x_3	$D_1D_3D_4$	1
		Y_{13}	1100	$\overline{x}_3x_4$	D_1D_2	2
		Y_9	1000	$\overline{x}_3\overline{x}_4$	D_1	3
O_4	10	Y_{12}	1011	x_3	$D_1D_3D_4$	4
		Y_{13}	1100	$\overline{x}_3x_4$	D_1D_2	5
		Y_9	1000	$\overline{x}_3\overline{x}_4$	D_1	6
O_6	11	Y_E	0000	1	—	4

Блок LUTerT – це таблиця, що має стовпці D_r, $B1$ і $B2$. Стовпець Bk відповідає блоку LUTerk. Якщо функція $D_r^k \neq 0$, то на перетині рядка D_r і стовпця Bk записується "1", інакше – 0. Для нашого прикладу, блок LUTerT наведений у табл. 3.

Таблиця блоку LUTerT дозволяє побудувати систему диз'юнкцій (8). У нашому прикладі кожна функція – це диз'юнкція з двома літералами.

ТАБЛИЦЯ 3. Таблиця блоку LUTerT

D_r	$B1$	$B2$	D_r	$B1$	$B2$
D_1	1	1	D_3	1	1
D_2	1	1	D_4	1	1

Блок LUTerτ будується з урахуванням залежності (9). Для знаходження СБФ (9) необхідно побудувати таблицю зі стовпцями: O_g – вихід ОЛЛ $\alpha_g \in C$; $A(O_g)$ – адреса МК, що відповідає вершині-виходу ОЛЛ $\alpha_g \in C$; τ_1, ..., τ_{R_O} – змінні, що кодують виходи ОЛЛ; g – номер ОЛЛ. Блок LUTerτ описує табл. 4.

З табл. 4 можна отримати ДНФ:

$$\tau_1 = O_2 \vee O_5 = \overline{T}_1T_2\overline{T}_3\overline{T}_4 \vee T_1\overline{T}_2\overline{T}_3T_4; \qquad \tau_3 = O_4 \vee O_6 = T_1\overline{T}_3T_4;$$

$$\tau_2 = O_1 \vee O_5 = \overline{T}_1\overline{T}_2T_3\overline{T}_4 \vee T_1\overline{T}_2T_3T_4; \qquad \tau_4 = O_3 \vee O_6 = \overline{T}_1T_2T_3T_4 \vee T_1T_2\overline{T}_3T_4.$$

Для реалізації системи мікрооперацій достатньо використовувати блок EMB, що має $S_A = R = 4$. Кожна МК включає до $N+2=8$ розрядів. Шість розрядів необхідні для зберігання кодів мікрооперацій, один розряд відображає сигнал y_O. Останній розряд зберігає змінну y_E (вона показана на структурних схемах). Змінна y_E зупиняє доступ синхронізації до лічильника. Тому змінна y_E записується у вершинах, з яких є перехід у кінцеву вершину ГСА (у вершину b_E).

ТАБЛИЦЯ 4. Таблиця блоку LUTerτ

O_g	$A(O_g)$	τ_1	τ_2	τ_3	τ_4	g
O_1	0010	0	1	0	0	1
O_2	0100	1	0	0	0	2
O_3	0111	0	0	0	1	3
O_4	1001	0	0	1	0	4
O_5	1011	1	1	0	0	5
O_6	1101	0	0	1	1	6

Змінна y_O є у всіх операторних вершинах, які не є вихідними ОЛЛ. Використовуючи y_O, реалізуємо режим природної адресації (2). Для нашого прикладу управляюча пам'ять наводиться у табл. 5.

ТАБЛИЦЯ 5. Таблиця блоку LUTerY

Адреса	b_m	y_0	y_E	y_1	y_2	y_3	y_4	y_5	y_6
0000	b_1	1	0	1	1	0	0	0	0
0001	b_2	1	0	0	0	1	0	0	0
0010	b_3	0	0	0	1	0	1	0	0
0011	b_4	1	0	0	1	1	0	0	0
0100	b_5	1	0	1	0	0	0	0	0
0101	b_6	0	0	0	0	0	0	1	1
0110	b_7	1	0	0	0	1	0	0	0
0111	b_8	0	0	0	0	0	0	0	1
1000	b_9	1	0	0	1	1	0	0	0
1001	b_{10}	0	0	1	0	0	0	0	0
1010	b_{11}	1	0	0	0	0	0	1	0
1011	b_{12}	0	1	0	0	1	1	0	0
1100	b_{13}	1	0	0	1	0	0	0	0
1101	b_{14}	0	1	1	0	0	0	1	0

Виконання останнього етапу (реалізація схеми) пов'язане із застосуванням САПР типу Vivado [18]. Ми не розглядаємо цей етап, оскільки вся необхідна інформація для реалізації схеми отримана. Це системи функцій, що представляють елементи LUT, та таблиці блоку EMBerY.

Результати

Запропонований метод дозволяє отримати схему КМПУ із регулярною структурою. Регулярність полягає у тому, що: 1) логічні умови пов'язані лише з елементами першого рівня; 2) сигнали синхронізації пов'язані лише з другим рівнем схеми; 3) кожна часткова функція може бути реалізована схемою, що складається з одного елемента LUT. Аналіз схем мікропрограмних автоматів з подвійним кодуванням станів [16] показав, що регулярні схеми мають ряд переваг у порівнянні зі схемами, що базуються на функціональній декомпозиції. Ці переваги такі: менша кількість елементів LUT та міжз'єднань; більш висока частота імпульсів синхронізації (вища швидкодія); менше значення споживаної потужності.

Очевидно, що всі ці позитивні якості мають бути притаманні і КМПУ із регулярною схемою адресації мікрокоманд. Запропонований метод доцільно застосовувати лише у разі виконання умови (4). В іншому випадку оптимізації не потрібно, тому що кожна функція $D_r \in D$ є одноелементною схемою.

Висновки

Найкращі умови для застосування запропонованого методу такі: виконується умова (4), тобто потрібна декомпозиція схеми адресації; виконується умова (10), тобто схема блоку LUTerT має один рівень елементів LUT; кожна ФЗП $D_r \in D$ блоку LUTerT генерується одним елементом LUT. Це можливо при виконанні умови $K \le S_L$.

Подальший напрямок наших досліджень пов'язаний із використанням запропонованого методу для автоматів Мілі, які реалізуються з використанням лічильників. Крім того, ми плануємо адаптувати методи комплексного кодування станів до особливостей КМПУ.

Авторські внески

Баркалов О.О.: концептуалізація, методологія, формальний аналіз, написання – оригінальна чернетка.

Тітаренко Л.О.: узагальнення, концептуалізація, методологія.

Сабурова С.О.: формальний аналіз, ресурси – рецензування та редагування.

Головін О.М.: формальний аналіз, ресурси, написання – рецензування та редагування, програмне забезпечення.

Матвієнко О.В.: формальний аналіз, дослідження, ресурси, візуалізація, написання – рецензування та редагування.

Фінансування. Автори не отримували фінансування для проведення досліджень та написання статті.

Список літератури

1. DeMicheli G. Synthesis and optimization of digital circuits. New York: McGraw-Hill, 1994. 576 p.
2. 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.
3. 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
4. Kubica M., Kania D. Area-oriented technology mapping for LUT-based logic blocks. *International Journal of Applied Mathematics and Computer Science*. 2017. **27** (1). P. 207–222.
5. Skliarova I., Sklyarov V., Sudnitson A. Design of FPGA-based circuits using hierarchical finite state machines. Tallinn: TUT Press, 2012. 240 p.

6. Баркалов А.А., Титаренко Л.А. Преобразование кодов в композиционных микропрограммных устройств управления. *Кибернетика и системный анализ*. 2011. № 5. С. 107–118. http://nbuv.gov.ua/UJRN/KSA_2011_47_5_11.
7. Barkalov A. Microprogramcontrol unit as composition of automate with programmable and hardwired logic. *Automatics and Computer Science*. 1983. **17** (4). P. 36–41.
8. Kuon I., Tessier R., Rose J. FPGA Architecture: Survey and Challenges. *Foundations and Trends in Electronic Design Automation*. 2008. Vol. 2, No. 2. P. 135–253.
9. Ruiz-Rosero J., Ramirez-Gonzalez G., Khanna R. Field Programmable Gate Array Applications – A Scientometric Review. *Computation*. 2019. **7** (4). P. 63. https://doi.org/10.3390/computation7040063
10. Xilinx. http://www.xilinx.com (https://www.amd.com/en.html) (звернення: 01.01.2024)
11. Marwedel P. Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things, 3rd ed. Springer International Publishing, 2018.
12. Barkalov A., Titarenko L., Mielcarek K., Chmielewski S. Logic Synthesis for FPGA-Based Control Units – Structural Decomposition in Logic Design. Vol. 636 of Lecture Notes in Electrical Engineering. Springer,Berlin, 2020. https://doi.org/10.1007/978-3-030-38295-7
13. Baranov S. Logic synthesis for control automata. Dordrecht: Kluwer Academic Publishers, 1994. 312 p. https://link.springer.com/book/10.1007/978-1-4615-2692-6
14. Sass R., Schmidt A. Embedded System Design with platform FPGAs: Principles and Practices. Amsterdam: Morgan Kaufmann Publishers, 2010. 409 p.
15. 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
16. Barkalov O., Titarenko L., Mielcarek K. Improving characteristics of LUT-based Mealy 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
17. Barkalov A., Titarenko L., Bieganowski J., Krzywicki K. Basic approaches for reducing power consumption in finite state machine circuits – Review. *Applied Sciences*. 2024. 14, 2693. P. 1–32.
18. Vivado Design Suite. 2020. https://www.xilinx.com/products/design-tools/vivado.html (звернення: 01.01.2024)

Одержано 03.10.2024

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.

Сабурова Світлана Олександрівна,
кандидат технічних наук, доцент Харківського національного університету радіоелектроніки,
https://orcid.org/0000-0001-6286-1648
sabsvet@gmail.com

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], С.О. Сабурова [2], О.М. Головін [3], О.В. Матвієнко [3]

Оптимізація схеми композиційного мікропрограмного пристрою управління з базовою архітектурою

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

Вступ. Пристрій управління є найважливішим блоком цифрових систем. На відміну від інших блоків, пристрій управління генерує сигнали у кожному такті роботи системи, і тому споживає значну кількість електричної потужності. В даний час проблема зменшення споживаної потужності набуває особливого значення.

Під час реалізації різних цифрових систем широко застосовуються мікросхеми FPGA (field-programmable logic array). На думку експертів, ці мікросхеми матимуть широке застосування у проектуванні цифрових пристроїв ще кілька десятиліть. Цим фактором обумовлений вибір саме цього елементного базису. Пропонований метод орієнтований на FPGA, які виробляються фірмою AMD Xilinx. Цей вибір зумовлений провідною позицією фірми на ринку мікросхем FPGA.

Мета роботи. Одним із способів вирішення проблеми зниження споживання потужності є збільшення регулярності схеми пристрою управління та зменшення числа з'єднань між її елементами. У цій статті пропонується метод вирішення цього завдання при реалізації схем композиційних мікропрограмних пристроїв управління (КМПУ) у базисі FPGA.

Для реалізації схеми КМПУ використовуються такі ресурси мікросхеми FPGA: елементи табличного типу LUT (look-up table), вбудовані блоки пам'яті EMB (embedded memory blocks) та програмовані міжз'єднання [8].

Основна ідея запропонованого методу зводиться до адаптації методу подвійного кодування станів до особливостей КМПУ з базовою архітектурою. Аналогами станів є мікрокоманди КМПУ. Тому оптимізація досягається за рахунок подвійної адресації мікрокоманд.

Результати. Запропонований метод дозволяє отримати схему КМПУ із регулярною структурою. Регулярність полягає у тому, що: логічні умови пов'язані лише з елементами першого рівня, сигнали синхронізації пов'язані лише з другим рівнем схеми, кожна часткова функція є схемою, що складається з одного елемента LUT. Аналіз схем мікропрограмних автоматів з подвійним кодуванням станів показує, що регулярні схеми у порівнянні зі схемами, що базуються на функціональній декомпозиції, мають ряд переваг: менша кількість елементів LUT та міжз'єднань, більш висока частота імпульсів синхронізації (вища швидкодія), менша кількість споживаної потужності.

Висновки. Запропонований метод доцільно застосовувати у тих випадках, коли через малу кількість входів елементів LUT схеми FPGA відомі методи вимагають застосування функціональної декомпозиції, яка призводить до схем з нерегулярною структурою та складною системою міжз'єднань. Такі схеми є повільними та споживають багато енергії.

Ключові слова: композиційні мікропрограмні пристрої управління, LUT, EMB, синтез.

UDC 004.274

Alexandr Barkalov [1], Larysa Titarenko [1, 2], Svitlana Saburova [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, Kharkiv, Ukraine*
[3] *V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv*
[*] *Correspondence: avmatv@ukr.net*

Introduction. The control unit is the most important block of digital systems. Unlike other blocks, the control unit generates signals in each cycle of the system and therefore consumes a significant amount of elec-

trical power. Currently, the problem of reducing power consumption is of particular importance. FPGA (field-programmable logic array) chips are widely used in the implementation of various digital systems. According to experts, these chips will be widely used in the design of digital devices for several decades to come. This factor determines the choice of this particular element basis. The proposed method is focused on FPGA, which is manufactured by AMD Xilinx. This choice is due to the company's leading position in the FPGA chip market.

The purpose of the article. One of the ways to reduce power consumption is to regularize the control device circuit and reduce the number of connections between its elements. This article proposes a solution to this problem when implementing composite microprogrammed control device (CMCD) circuits in the FPGA basis.

The following FPGA chip resources are used to implement the CMCD circuit: elements of the LUT (look-up table) type, embedded memory blocks (EMB) and programmable interconnections.

The main idea of the proposed method is to adapt the method of double coding of states to the features of the CMCD with the basic architecture. The analogs of the states are the CMCD microinstructions. Therefore, optimization is achieved due to double addressing of microinstructions.

Results. The proposed method allows to obtain a CMCU circuit with a regular structure. The regularity consists in the fact that: logical conditions are associated only with the elements of the first level, synchronization signals are associated only with the second level of the circuit; any partial function is a circuit consisting of one LUT element. Analysis of the circuits of microprogrammed machines with double coding of states shows that regular circuits have a number of advantages over circuits based on functional decomposition: a smaller number of LUT elements and interconnections, a higher frequency of synchronization pulses (high speed), a lower value of consumed power.

Conclusions. The proposed method is appropriate to use in cases where, due to the small number of inputs of the LUT elements of FPGA circuits, known methods require the use of functional decomposition, which leads to circuits with an irregular structure and a complex interconnection system. Such circuits have low performance and consume a lot of energy.

Keywords: composite microprogrammed control device, LUT, EMB, synthesis.