

УДК 681.3

О.О. Баркалов, О.С. Лаврік, О.М. Мірошкін

University of Zielona Gora, Zielona Gora, Poland

Донецький національний технічний університет, м. Донецьк

E-mail: a.barkalov@iie.uz.zgora.pl, alexandrlavrik@cs.donntu.edu.ua, miroshkinan@gmail.com**ЗАСТОСУВАННЯ ПЕРЕТВОРЮВАЧА КОДІВ ДЛЯ ЗМЕНШЕННЯ АПАРАТУРНИХ ВИТРАТ В СХЕМІ ПРИСТРОЮ КЕРУВАННЯ****Анотація**

Баркалов О.О., Лаврік О.С., Мірошкін О.М. Застосування перетворювача кодів для зменшення апаратних витрат пристрою керування. В роботі запропоновано метод зменшення апаратних витрат, орієнтований на композиційні мікропрограмні пристрої керування з розділенням кодів, що реалізовані на мікросхемах типу CPLD. Метод базується на великому коефіцієнті об'єднання за входом у макроосередків ПМЛ, що дозволяє використовувати більше ніж одне джерело для кодів класів ОЛЛ. Наведено приклад використання запропонованого методу.

Ключові слова: композиційний мікропрограмний пристрій керування, псевдоеквівалентний операторний лінійний ланцюг, розділення кодів.

Вступ. Одним з важливих блоків цифрових систем є пристрій керування (ПК) [1].

Якщо алгоритм, що реалізовується, має лінійний характер, для його надання можна використовувати модель композиційного мікропрограмного пристрою керування (КМПК) з розділенням кодів [4]. В даний час для реалізації логічних схем пристроїв керування широко використовуються програмовані логічні інтегральні схеми (ПЛІС) типу CPLD (Complex Programmable Logic Devices) [6]. Такі ПЛІС включають макрокомірки програмованої матричної логіки (ПМЛ), що мають великий коефіцієнт об'єднання по входу (декілька десятків) [2,3]. Висока вартість цього базису вимагає рішення актуальної задачі зменшення числа корпусів ПЛІС в схемі. При її рішенні слід мати на увазі не тільки специфіку алгоритму керування, що інтерпретується, але і особливості елементного базису. У такому разі, завдяки великому коефіцієнту об'єднання по входу макрокомірок ПМЛ, коди класів псевдоеквівалентних операторних лінійних ланцюгів (ОЛЛ) можуть бути представлені більш ніж одним джерелом [7,8]. У даній роботі пропонується метод зменшення апаратних витрат в схемі КМПК, що базується на використанні трьох джерел кодів.

Особливості КМПК з розділенням кодів. Нехай ГСА Γ представлена множиною вершин V та дуг E . Нехай $B = \{b_0, b_E\} \cup E_1 \cup E_2$, де b_0 – початкова вершина, b_E – кінцева вершина, E_1 – множина операторних вершин, де $|E_1| = M$, E_2 – множина умовних вершин. Вершина $b_q \in E_1$ містить набір мікрооперацій (МО) $Y(b_q) \subseteq Y$, де $Y = \{y_1, \dots, y_N\}$ – множина мікрооперацій [1]. Кожна вершина $b_q \in E_2$ містить елементи множини логічних умов $X = \{x_1, \dots, x_L\}$.

Сформуємо набір операторних лінійних ланцюгів $C = \{\alpha_1, \dots, \alpha_G\}$ для ГСА Γ , де кожна ОЛЛ $\alpha_g \in C$ – це послідовність операторних вершин, причому кожній парі її сусідніх вершин відповідає одна з дуг ГСА. Кожна ОЛЛ $\alpha_g \in C$ має лише один вихід O_g і довільну кількість входів. Визначення ОЛЛ, їх входів та виходів можуть бути знайдені в [4].

Назвемо ГСА Γ лінійною ГСА (ЛГСА), якщо виконується наступна умова:

$$\frac{M}{G} \geq 2. \quad (1)$$

Кожна вершина $b_q \in E_1$ відповідає мікрооперації MI_q , що зберігається в керуючій пам'яті (КП) КМПК по адресу A_q . Для адресації мікрокоманд достатньо

$$R = \lceil \log_2 M \rceil \quad (2)$$

біт. Нехай ОЛЛ $\alpha_g \in C$ включає F_g компонент и нехай $Q = \max(F_1, \dots, F_G)$. Закодуємо ОЛЛ $\alpha_g \in C$ двійковими кодами $K(\alpha_g)$ розрядності R_G , де:

$$R_G = \lceil \log_2 G \rceil. \quad (3)$$

Закодуємо кожний компонент ОЛЛ $\alpha_g \in C$ двійковими кодами $K(b_q)$ розрядності R_Q , де:

$$R_Q = \lceil \log_2 Q \rceil. \quad (4)$$

Нехай компоненти закодівані таким чином, що

$$K(b_{g(i+1)}) = K(b_{g(i)}) + 1, \quad (5)$$

де $i = 1, \dots, F_g - 1; g = 1, \dots, G$. Нехай для кодування компонент ОЛЛ використовуються змінні $T_r \in T$, а для кодування ОЛЛ – змінні $\tau_r \in \tau$, де $|\tau| = R_G, |T| = R_Q$. При виконанні умови

$$R_G + R_Q = R, \quad (6)$$

ЛГСА Γ може бути представлена моделлю КМПК с розділенням кодів U_1 (див.Рис.1).

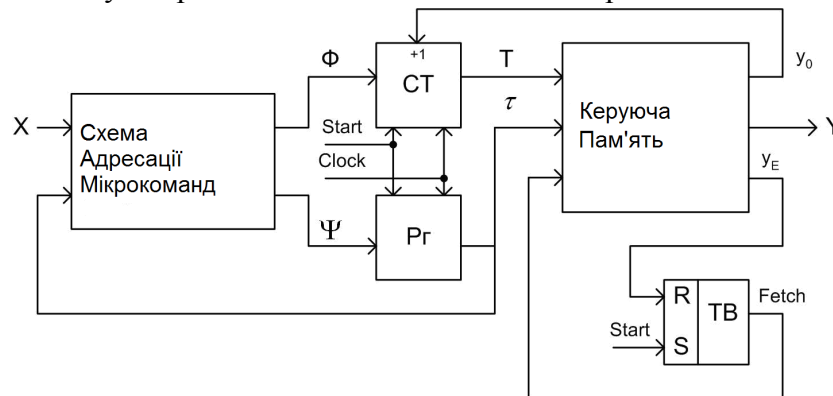


Рисунок 1 – Структурна схема КМПК U_1

В КМПК U_1 схема адресації мікрокоманд (САМ) реалізує систему функцій збудження лічильника СТ и регістра Рг:

$$\Phi = \Phi(\tau, X), \quad (7)$$

$$\Psi = \Psi(\tau, X).$$

При цьому, адрес мікрокоманди MI_q надається у вигляді:

$$A(b_q) = K(\alpha_g) * K(b_q), \quad (8)$$

де вершина b_q – це компонент ОЛЛ $\alpha_g \in C$ и “*” – знак операції конкатенації.

КМПК U_1 працює наступним чином. По сигналу $Start = 1$ початкова адреса (усі нулі) заноситься в Рг и СТ. У той же час, тригер вибірки ТВ встановлюється в одиничний стан. При цьому $Fetch = 1$, що дозволяє вибірку мікрокоманд з КП. Кожна комірка КП містить мікрооперації $y_n \in Y$ та спеціальні змінні y_0 и y_E . Якщо $y_0 = 1$, то к змісту СТ додається одиниця и адресується наступний компонент поточного ОЛЛ. Інакше – значення СТ и Рг завантажуються з САМ. Перший випадок відповідає переходу з будь якого компоненту ОЛЛ, окрім виходу. Другий випадок відповідає переходу з виходу ОЛЛ. Якщо $y_E = 1$, тригер вибірки скидається, сигнал $Fetch = 0$ и робота КМПК зупиняється. Це відповідає переходу з вер-

шини $b_q \in E_1$, де $\langle b_q, b_E \rangle \in E$.

Зазначимо, що ОЛЛ $\alpha_i, \alpha_j \in C$ є псевдоеквівалентними ОЛЛ (ПОЛЛ) [4], якщо їх виходи з'єднані з однією ж самою вершиною ГСА Г. Апаратурні витрати в логічній схемі САМ можуть бути зменшені введенням спеціального блоку перетворювача кодів (ПК), що перетворює коди ОЛЛ в коди класів ПОЛЛ. В цьому випадку САМ реалізує H_0 термів, де H_0 – число переходів еквівалентного автомата Милі, що є мінімально можливим числом переходів [4]. Проте, цей блок споживає деякі ресурси ПЛІС або ППЗП, з яких будується КП.

У даній роботі пропонується метод синтезу КМПК, в якому використовується три джерела кодів класів ПОЛЛ, що дозволяє знизити апаратурні витрати, споживані блоком ПК, а в певних випадках цей блок взагалі може не використовуватися.

Основна ідея пропонованого метода. Нехай $C_1 \subseteq C$ буде множиною ОЛЛ, де $\alpha_g \in C_1$, якщо її вихід не з'єднаний з кінцевою вершиною b_E . Нехай $\Pi_C = \{B_1, \dots, B_I\}$ – розбиття множини C_1 на класи ПОЛЛ. Закодуємо ОЛЛ $\alpha_g \in C_1$ таким чином, щоб більшість класів $B_i \in \Pi_C$ було представлено одним інтервалом R_G -мірного булева простору. Для цього може бути використаний широко відомий алгоритм ESPRESSO [5]. Нехай $\Pi_C = \Pi_A \cup \Pi_B$, де $B_i \in \Pi_A$, якщо класу відповідає один інтервал, інакше $B_i \in \Pi_B$. Якщо виконується умова:

$$\Pi_B = \emptyset, \quad (9)$$

то блок ПК відсутній, і регістр Рг є джерелом усіх кодів $K(B_i)$ для класів $B_i \in \Pi_C$. Інакше перетворенню підлягають лише адреси виходів ОЛЛ, що входять у класи $B_i \in \Pi_B$. Для кодування цих класів достатньо:

$$R_B = \lceil \log_2(|\Pi_B| + 1) \rceil \quad (10)$$

Одиниця додається до $|\Pi_B|$ для зазначення ситуації $B_i \notin \Pi_B$. Відзначимо, що частина кодів може бути реалізована на ППЗП, а схеми блоків САМ, СТ, Рг і ТБ реалізуються на макроосередках ПМЛ. Для реалізації КП потрібні зовнішні ППЗП, такі, що мають t виходів, де $t \in \{1, 2, 4, 8, 16\}$ [2, 3].

При використанні унітарного кодування мікрооперацій [6], кожне слово в КП складається з $N+2$ біт. Число 2 додається к N для реалізації додаткових змінних y_0 и y_E (див. Рис. 1).

Якщо кожне ППЗП має t виходів и не менш ніж M слів, то достатньо K_0 мікросхем для реалізації КП, де:

$$K_0 = \left\lceil \frac{N+2}{t} \right\rceil. \quad (11)$$

У останній мікросхемі залишається R_0 вільних виходів, де:

$$R_0 = K_0 t - N - 2. \quad (12)$$

Якщо виконується умова:

$$R_0 \geq R_B, \quad (13)$$

то все класи $B_i \in \Pi_B$ можуть бути представлені за допомогою КП и блок ПК відсутній. Інакше множина Π_B надається у вигляді $\Pi_E \cup \Pi_D$, де $\Pi_D = \Pi_B \setminus \Pi_E$. При цьому:

$$|\Pi_E| = 2^{R_0} - 1 \quad (14)$$

Одиниця в (14) віднімається, щоб відобразити ситуацію $B_i \notin \Pi_E$. Розрядність блоку ПК визначається як:

$$R_D = \lceil \log_2(|\Pi_D| + 1) \rceil. \quad (15)$$

Одиниця додається у (15), для відображення ситуації $B_i \notin \Pi_D$. У цілому, перетворенню

підлягають лише адреси виходів ОЛЛ $\alpha_g \in B_i$, де $B_i \in \Pi_D$. Якщо все множини Π_A, Π_E и Π_D , не є пустими, то для інтерпретації ЛГСА Γ пропонується КМПК U_2 (див.Рис. 2).

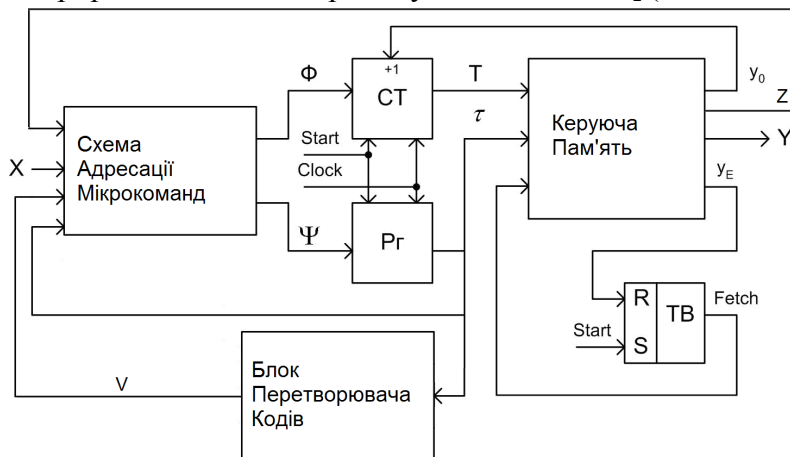


Рисунок 2 – Структурна схема КМПК U_2

В КМПК U_2 , блок САМ реалізує функції:

$$\Phi = \Phi(\tau, V, Z, X), \tag{16}$$

$$\Psi = \Psi(\tau, V, Z, X),$$

а блок ПК реалізує функцію:

$$V = V(\tau). \tag{17}$$

Змінні $v_r \in V$ використовуються для кодування класів $B_i \in \Pi_D$, $|V| = R_D$. Змінні $z_r \in Z$ використовуються для кодування класів $B_i \in \Pi_E$, $|Z| = R_0$. В залежності від змінних τ і T керуюча пам'ять реалізує функції Z, Y, y_0 та y_E . Принцип дії КМПК U_1 и U_2 практично ідентичні.

В роботі пропонується наступний алгоритм синтезу КМПК U_2 :

1. Побудова множин C, C_1 и Π_C для ЛГСА Γ .
2. Кодування ОЛЛ $\alpha_g \in C$ и їх компонентів.
3. Побудова Π_A, Π_E и Π_D .
4. Кодування класів $B_i \in \Pi_E \cup \Pi_D$.
5. Побудова таблиці переходів КМПК.
6. Специфікація керуючої пам'яті.
7. Специфікація блоку ПК.
8. Реалізація схеми КМПК в даному елементному базисі.

Приклад застосування пропонованого методу.

Нехай для деякої ГСА Γ_1 побудовані множини $C = \{\alpha_1, \dots, \alpha_8\}$, $C_1 = \{\alpha_1, \dots, \alpha_7\}$ і $\Pi_C = \{B_1, B_2, B_3\}$, де $\alpha_1 = \langle b_1, \dots, b_4 \rangle$, $\alpha_2 = \langle b_5, \dots, b_8 \rangle$, $\alpha_3 = \langle b_9, \dots, b_{11} \rangle$, $\alpha_4 = \langle b_{12}, \dots, b_{14} \rangle$, $\alpha_5 = \langle b_{15}, \dots, b_{17} \rangle$, $\alpha_6 = \langle b_{18}, \dots, b_{21} \rangle$, $\alpha_7 = \langle b_{22}, \dots, b_{24} \rangle$, $\alpha_8 = \langle b_{25}, \dots, b_{28} \rangle$, $B_1 = \{\alpha_1\}$, $B_2 = \{\alpha_2, \alpha_3, \alpha_4\}$, $B_3 = \{\alpha_5, \alpha_6, \alpha_7\}$. Отже кількість ОЛЛ $G=8$, для їх кодування достатньо $R_G=3$, максимальна кількість компонент у ОЛЛ $Q=4$, для їх кодування достатньо $R_Q=2$, загальна кількість операторних вершин $M=28$, для їх кодування достатньо $R=5$. Таким чином, умова (6) виконується і доцільно застосувати метод розділення кодів. Умова (1) також виконується, тому ГСА Γ_1 є лінійною.

Закодуємо ОЛЛ $\alpha_g \in C$, як показано на рис. 3, використовуючи змінні $\tau_r \in \tau$.

Компоненти ОЛЛ $\alpha_g \in C$ закодовані тривіально [4]: перший компонент має код 00, другий – 01 и так далі, для того щоб відповідати умові (5). Для кодування компонентів використовуються змінні $T_r \in T = \{T_1, T_2\}$.

		$\tau_2 \tau_3$			
τ_1	00	01	11	10	
0	α_1	α_2	α_3	α_4	
1	α_5	α_6	α_7	α_8	

Рисунок 3 – Коды ОЛЛ для ГСА Γ_1 .

З рис. 3 видно, що клас B_1 наданий одним інтервалом булева простору и відповідає коду $K(B_1)=000$. Класу B_2 відповідають інтервали 001 и 01*, а класу B_3 інтервали 10* и 111. Таким чином, $\Pi_A = \{B_1\}$, $\Pi_B = \{B_2, B_3\}$ і згідно з (10) $R_B = 2$.

Нехай для ГСА Γ_1 кількість мікрооперацій $N=13$ и нехай для реалізації КП використовуються мікросхеми ППЗП з кількістю виходів $t = 4$. У такому разі, з (11) маємо $K_0 = 4$ мікросхеми необхідно, а з умови (12) – $R_0 = 1$. Умова (13) не виконується і в КМПК $U_2(\Gamma_1)$ використовуються блок ПК. Нехай $\Pi_E = \{B_2\}$, тоді $\Pi_D = \{B_3\}$. Отже, множини Π_A, Π_E, Π_D сформовані. Змінна $z_1 \in Z$ використовується для кодування класу B_2 . Нехай $K(B_2)=1$, де $z_1 = 0$ вказує на ситуацію $B_i \notin \Pi_E$. Нехай $K(B_3)=1$, де $v_1 = 0$ вказує на ситуацію $B_i \notin \Pi_D$.

Нехай переходи для класів $B_i \in \Pi_C$ задаються наступною системою узагальнених формул переходу (ОФП)[4]:

$$\begin{aligned}
 B_1 &\rightarrow x_1 \overline{b_5} \vee x_1 x_2 \overline{b_9} \vee x_1 x_2 x_3 \overline{b_{11}} \vee x_1 x_2 x_3 b_{18}; \\
 B_2 &\rightarrow x_4 \overline{b_{15}} \vee x_4 x_5 \overline{b_{19}} \vee x_4 x_5 x_6 \overline{b_{22}} \vee x_4 x_5 x_6 b_{13}; \\
 B_3 &\rightarrow x_7 \overline{b_{11}} \vee x_7 x_8 \overline{b_9} \vee x_7 x_8 b_{25}.
 \end{aligned}
 \tag{18}$$

Ця система є основою для будови таблиці переходів, що має наступні стовбці: $B_i, K_A(B_i), K_E(B_i), K_D(B_i), b_q, A(b_q), X_h, \Psi_h, \Phi_h, h$. Тут $K_A(B_i)$ – це код $K(B_i)$, де $B_i \in \Pi_A$; $K_E(B_i)$ – код для $B_i \in \Pi_E$; $K_D(B_i)$ – код для $B_i \in \Pi_D$; X_h – вхідний сигнал з ОФП; Ψ_h – набір вихідних функцій що завантажують код $K(\alpha_g)$ в Рг; Φ_h – набір вихідних функцій що завантажують код $K(b_q)$ в СТ; $h=1, \dots, H_0$ номер переходу. В цьому випадку $\Psi = \{D_1, D_2, D_3\}$, $\Phi = \{D_4, D_5\}$. Визначимо декілька адрес мікрооперацій. Наприклад, вершина b_{22} є першим компонентом ОЛЛ $\alpha_7 \in C$, тому з (8) $A(b_{22})=11100$. Аналогічно знайдемо $A(b_{12})=01000$, $A(b_4)=00011$ и $A(b_{19})=10101$.

Нехай символ $U_i(\Gamma_j)$ використовується для означення інтерпретації ГСА Γ_j за допомогою моделі КМПК $U_i(i=1,2)$.

Частина таблиці переходів для класів $B_1, \dots, B_3 \in \Pi_C$ показана в табл. 1.

Ця таблиця використовується для побудови системи (16). Наприклад, наступні вираження отримані з табл. 1:

$$\begin{aligned}
 D_2 &= \overline{\tau_1 \tau_2 \tau_3 z_1 x_1 x_2} \vee \overline{\tau_1 \tau_2 \tau_3 z_1 x_1 x_3} \vee \overline{z_1 v_1 x_4 x_5} \vee \overline{z_1 v_1} \quad (\text{після мінімізації}); \\
 D_4 &= z_1 v_1 x_7.
 \end{aligned}$$

Керуюча пам'ять КМПК $U_2(\Gamma_1)$ специфікується з використанням широко відомих методів [4].

Таблиця 1 – Частина таблиці переходів для КМПК $U_2(\Gamma_1)$

B_i	$K_A(B_i)$	$K_E(B_i)$	$K_D(B_i)$	b_q	$A(b_q)$	X_h	Ψ_h	Φ_h	h	
	$\tau_1\tau_2\tau_3$	z_1	v_1							
B_1	000	0	0	b_5	001 00	$\overline{x_1}$	D_3	-	1	
				b_9	010 00	$x_1\overline{x_2}$	D_2	-	2	
									
B_2	***	1	0	b_{15}	100 00	$\overline{x_4}$	D_1	-	5	
									
				b_{13}	011 01	$x_4x_5x_6$	D_2D_3	D_5	8	
B_3	***	0	1	b_{11}	010 10	$\overline{x_7}$	D_2	D_4	9	
									

Таблиця блока перетворення адреси будується для класів $B_i \in \Pi_D$. У цьому прикладі вона має 3 строки (Табл. 2).

Таблиця 2 – Таблиця блока ПК пристрою $U_2(\Gamma_1)$

α_g	$A(O_g)$	B_i	$K_D(B_i)$	V_i	j
α_5	100 10	B_3	1	v_1	1
α_6	101 11				2
α_7	111 10				3

З цієї таблиці маємо систему (17) котра в нашому випадку надається наступною ДНФ:

$$v_1 = \tau_1\tau_2 \vee \tau_1\tau_3.$$

Звернемо увагу на те, що $H_0 = 11$, що дорівнює загальній кількості термів в системі (18). У випадку КМПК $U_1(\Gamma_1)$ маємо $H = 22$. Таким чином, застосування запропонованого методу дозволяє зменшити це значення в два рази. Очікується, що зменшення кількості макрокомірок ПМЛ в логічній схемі блока САМ буде мати той же порядок [4].

Висновки.

Запропонований метод спрямований на зменшення числа макрокомірок ПМЛ в схемі адресації мікрокоманд КМПК з розділенням кодів. Це можливо завдяки таким головним чинникам, як великий коефіцієнт об'єднання по входу сучасних макрокомірок ПМЛ, а також природна надмірність мікросхем ППЗП, обумовлена тим, що кількість їх виходів обмежене визначеною множиною чисел; існування класів псевдоеквівалентних ОЛЛ.

Дослідження показали, що загальна кількість макрокомірок зменшується до 12% для КМПК $U_2(\Gamma_1)$ у порівнянні з еквівалентним КМПК $U_1(\Gamma_1)$. Слід відзначити, що даний метод може бути застосований лише для інтерпретації лінійних ГСА при умові виконання (6).

Подальші наші дослідження пов'язані з визначенням можливості застосування запропонованого підходу для реалізації КМПК на мікросхемах FPGA [5,10].

Література

1. Baranov S. Logic Synthesis for Control Automata. Kluwer Academic Publishers, 1994.–312 pp.
2. Электронный ресурс. Xilinx CPLDs
http://www.xilinx.com/products/silicon_solutions/cplds/index.htm.
3. Электронный ресурс. Altera devices overview.
http://www.altera.com/products/devices/common/devfamily_overview.html.
4. Barkalov A., Titarenko L. Logic Synthesis for Compositional Microprogram Control Units. – Berlin: Springer, 2008. – 272 pp.
5. Maxfield C. The Design Warrior's Guide to FPGAs. – Amsterdam: Elsevier, 2004. – 541 pp.
6. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.
7. Баркалов А.А., Зеленёва И.Я., Лаврик А.С. Использование особенностей ПЛИС для оптимизации схемы устройства КПрвления. / Наукові праці Донецького національного технічного університету. Серія «Інформатика, кібернетика і обчислювальна техніка» (ІКОТ-2008). Випуск 9 (132) – Донецьк: ДонНТУ. – 2008.С. 178-182.
8. Баркалов А.А., Ковалёв С.А., Красичков А.А., Лаврик А.С. Оптимизация устройства КПрвления с преобразователем адреса микрокоманд. /Материалы Девятого Международного научно-практического семинара. В 3-х кн. – Таганрог. Кн. 3. 2008. С. 12-20.
9. Электронный ресурс. CoolRunner CPLD Datasheet.
<http://www.xilinx.com/support/documentation/coolrunner-ii.htm>
10. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608 с.

Аннотация

Баркалов А.А., Лаврик А.С., Мирошкин А.Н. Применение преобразователя кодов для уменьшения аппаратных затрат в схеме устройства управления с разделением кодов. В работе предложен метод уменьшения аппаратных затрат, ориентированный на композиционные микропрограммные устройства управления с разделением кодов, реализованные на микросхемах типа CPLD. Метод основан на большом коэффициенте объединения по входу макроячеек ПМЛ, что позволяет использовать более одного источника для кодов классов ОЛЦ. Приведен пример реализации предложенного метода.

Ключевые слова: Композиционное микропрограммное устройство управления, псевдоэквивалентные операторные линейные цепи, разделение кодов.

Abstract

Barkalov A., Lavrik A., Miroshkin A. Application of code converter for control unit hardware amount reduction. The method of hardware reduction is proposed which is oriented on compositional microprogram control units with code sharing and PAL-based CPLD chips. The method is based on a wide fan-in of PAL macrocells allowing using more than one source for codes of operational linear chains. An example of proposed method application is given.

Keywords: compositional microprogram control unit, pseudoequivalent operational linear chain, code sharing.

Здано в редакцію:
02.04.2010р.

Рекомендовано до друку:
к.т.н, доц. Маренич К.М.