Alexander BARKALOV, Larysa TITARENKO, **Łukasz SMOLIŃSKI**
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI
ul. Licealna 9, 65-417 Zielona Góra

Optimization of control unit based on construction of CPLD

Prof. Alexander BARKALOV

Was born in Russia in 1954. Since 2003 he has been a professor of computer engineering at the Institute of Computer Engineering and Electronics, University of Zielona Góra, Poland, and he is also a professor at the Institute of Computers, Donetsk National Technical University, Ukraine. His current research interests include the theory of digital automata, especially methods for synthesis and optimization of control units implemented with field programmable logic devices.

e-mail: A.Barkalov@iie.uz.zgora

Prof. Larysa TITARENKO

Was born in Ukraine in 1971. Since 2007 she has been a professor of telecommunications at the Institute of Computer Engineering and Electronics, University of Zielona Góra, Poland, and she is also a professor at the Institute of Telecommunications Systems, Kharkov National University of Radioelectronics, Ukraine. Her current research interests include telecommunication systems, antennas and digital automata, and their of applications.

e-mail: L.Titarenko@iie.uz.zgora

MA Łukasz SMOLIŃSKI

Born in Poland in 1984. In 2009 obtained a master's degree in computer science. Currently a second year student in the doctoral studies at the University of Zielona Gora.

e-mail: L.Smolinski@weit.uz.zgora.pl

Abstract

The method of hardware reduction dedicated for a compositional microprogram control unit implemented in CPLD is proposed. The method is based on using more than one source of microinstruction address. Such an approach enables decreasing the number of logic blocks used for implementation of the controller in the target CPLD. The paper presents the conditions required to use the method and a calculation example of its application.

Keywords: CPLD, PAL, mikroprogram control unit.

Optymalizacja jednostki kontrolnej bazująca na budowie układów CPLD

Streszczenie

W artykule przedstawiona została metoda zmniejszenia powierzchni sterowników sprzętowych realizowanych w układach typu CPLD. Wprowadzono modyfikacje w strukturze sterownika, modyfikacje których głównym zadaniem jest redukcja liczby wykorzystanych elementów logicznych podczas implementacji sterownika w układach CPLD. Zaprezentowana została bazowa metodologia projektowa, dla której wprowadzono odpowiednie modyfikacje. Modyfikacje, które pozwalają zmniejszyć liczbę potrzebnych elementów logicznych wykorzystanych przy implementacji realizowanego sterownika. Przedstawione modyfikacje bazują na wykorzystaniu więcej niż jednego źródła danych przy wyznaczaniu kolejnego adresu mikroinstrukcji. W artykule przedstawiony został schemat logiczny dla zmodyfikowanej struktury sterownika. Zaprezentowano i omówiono warunki potrzebne do zastosowania zaprezentowanej metody oraz podano odpowiednie przykłady obliczeniowe. W artykule przedstawione zostały wyniki oraz wnioski z badań przeprowadzonych przez autorów.

Słowa kluczowe: CPLD, PAL, mikroprogram control unit.

1. Introduction

A control unit is one of the very important parts of any digital system [5]. If a control algorithm to be interpreted is a linear one, then it can be implemented using the model of compositional microprogram control unit (CMCU) [4]. The programmable logic devices with programmable array logic (PAL) macrocells are widely used for implementation of logic circuits of control units [1, 7]. The high cost of such devices requires optimization of hardware amount in the circuit of CMCU. One of the ways for this task solution is decrease in the number of conjunctive terms in the sum-of-product (SOP) forms of the functions to be implemented [6, 2]. This paper proposes an approach for this problem solution based on a wide fan-in of PAL macrocells [1, 7]. This approach aims at CMCU with address transformer [4] and a control algorithm is represented as a graph-scheme of the algorithm (GSA) [3].

2. Peculiarities of CMCU with address transformer

Let GSA be represented by sets of vertices B and arcs E. Let, where b0 is an initial vertex, bE is a final vertex, E1 is a set of operator vertices, where and E2 is a set of conditional vertices. A vertex contains a microinstruction in which there is a set of data-path micro operations [3]. Each vertex contains a single element of a set of logical conditions. Let GSA be a linear GSA, that is a GSA with more than 75% of operator vertices.

Let us form a set of operational linear chains (OLC) for GSA, where each OLC is a sequence of operator vertices and each pair of its adjacent components corresponds to some arc of the GSA. Each OLC has only one output and arbitrary number of inputs. Formal definitions of OLC, its input and output can be found in [4]. Each vertex corresponds to microinstruction MIq kept in a control memory (CM) of CMCU and it has an address. The microinstructions can be addressed using

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

bits represented by variables $T_r \in T = \{T_1,...,T_R\}$. Let OLC $\alpha_g \in C$ include F_g components and the following condition takes place:

$$A(b_{gi+1}) = A(b_{gi}) + 1, \tag{2}$$

In equation (2) b_{gi} is the i-th component of OLC $\alpha_g \in C$, where $i = 1,..., F_g - 1$.

If outputs O_i, O_j are connected with an input of the same vertex, then OLC $\alpha_i, \alpha_j \in C$ are pseudoequivalent OLC (POLC) [4]. Let us construct the partition $\Pi_C = \{B_1,..., B_I\}$ of the set $C_1 \subseteq C$ on the classes of POLC. Let us point out that $\alpha_g \in C_1$ if $\langle O_g, B_E \rangle \notin E$. Let us encode the classes $B_i \in \Pi_C$ by binary codes $K(B_i)$ with

$$R_1 = \lceil \log_2 I \rceil \tag{3}$$

bits and use the variables $\tau_r \in \tau = \{\tau_1,...,\tau_{R_1}\}$ for the encoding. In this case a GSA Γ can be interpreted using the model of CMCU U_1 with the address transformer (Fig. 1).

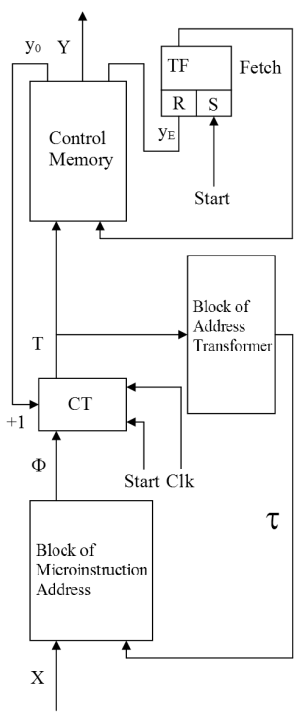
The pulse Start causes loading of the first microinstruction address into a counter CT and set up of a fetch flip-flop TF. If Fetch = 1, then microinstructions can be read out the control memory CM. If a current microinstruction does not correspond to an OLC output, then a special variable y_0 is formed together with microoperations $Y_q \subseteq Y$. If y_0 = 1, then content of the CT is incremented according to the addressing mode (2). Otherwise, the block of microinstruction address BMA generates functions

$$\Phi = \Phi(\tau, X) \tag{4}$$

to load the next microinstruction address into the CT. At the same time, the block of address transformer BAT generates functions

$$\tau = \tau(T). \tag{5}$$

If the output of OLC $\alpha_g \notin C_1$ is reached, then $y_E = 1$. It causes cleaning the TF and operation of CMCU U_1 is terminated.



Rys. 1. Diagram strukturalny CMCU U_1
Fig. 1. Structural diagram of CMCU U_1

Such organization of CMCU allows decrease in the number of terms in functions Φ from H1 till H0, where H1, H0 is the number of terms for equivalent finite state machines (FSM) Moore and Mealy, respectively. But the block BAT consumes some macrocells or cells of PROM used for implementation of CM. In this paper we propose some CMCU U2, where H2 = H0 and block BAT consumes less hardware than its counterpart in U1. Here H2 means the number of terms in functions Φ for CMCU U2.

3. The main idea of the proposed method

Let us point out that logic circuits for BMA, CT, TF and BAT are implemented as the parts of CPLD. To implement the CM one should use PROM chips with t outputs, where t = 1, 2, 4, 8, 16 [1, 7]. Let us address the components of OLC $\alpha_g \in C_1$ in such a manner that condition (2) takes place and maximal possible amount of classes $B_i \in \Pi_C$ is represented by a single generalized interval of R-dimensional Boolean space. Such addressing needs a special algorithm which should be developed.

Let $\Pi_C = \Pi_A \cup \Pi_B$, where $B_i \in \Pi_A$ if this class is represented by one interval, and $B_i \in \Pi_B$ otherwise. The counter CT is a source of the codes for $B_i \in \Pi_A$. If condition

$$\Pi_B = \emptyset \tag{6}$$

takes place, then the block BAT is absent. Otherwise, only output addresses for OLC from classes $B_i \in \Pi_B$ should be transformed. It is enough

$$R_2 = \lceil \log_2(I_B + 1) \rceil \tag{7}$$

bits for such encoding, where $I_B = |\Pi_B|$ and 1 is added to take into account the case when $B_i \in \Pi_A$. Let us point out that some part of these codes can be implemented using free outputs of PROM. Let us use one-hot encoding of microoperations [4] when CM word has N+2 bits. In this case CM can be implemented using

$$R_0 = \left\lceil \frac{N+2}{t} \right\rceil \tag{8}$$

chips with enough amount of cells (not less than M). Obviously, that R_3 outputs of PROM are free, where

$$R_3 = R_0 * t - N - 2 \tag{9}$$

If the condition

$$R_3 \geq R_2, \tag{10}$$

takes place, then CM is a source of the codes for $B_i \in \Pi_B$ and the block BAT is absent. Otherwise, we can represent Π_B as $\Pi_E \cup \Pi_D$, where $I_E = |\Pi_E|$, $I_D = |\Pi_D|$. In this case

$$I_E = 2^{R_3} - 1, \tag{11}$$

$$R_4 = \lceil \log_2(I_D + 1) \rceil. \tag{12}$$

The value I_E is decremented to represent the situation that $B_i \in \Pi_E$. The value of I_D is incremented to show the case that $B_i \in \Pi_D$. Thus, only the outputs of OLC $\alpha_g \in B_i$ should be transformed, where $B_i \in \Pi_D$.

In the common case, when $\Pi_i \neq \varnothing (i = A, E, D)$, the following CMCU U_2 is proposed for interpretation of GSA Γ (Fig. 2).

In CMCU U2, codes $K_A(B_i)$ of the classes $B_i \in \Pi_A$ are represented by variables $T_r \in T$; codes $K_E(B_i)$ of the classes $B_i \in \Pi_E$ are represented by variables $v_r \in V$, where $|V| = R_3$; codes $K_D = (B_i)$ of the classes $B_i \in \Pi_D$ are represented by variables $z_r \in Z$, where $|Z| = R_4$. In CMCU U2, the block BMA implements functions

$$\Phi = \Phi(T, Z, V, X), \tag{13}$$

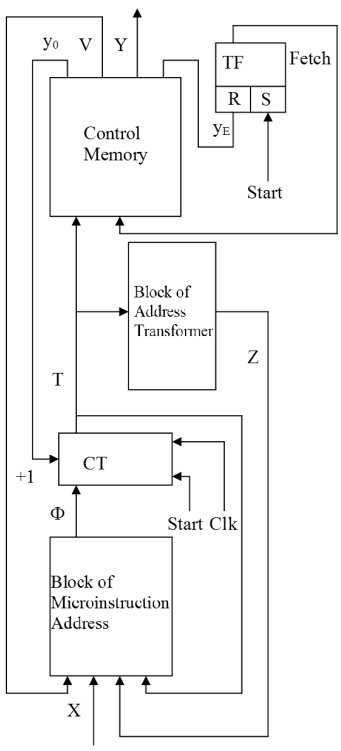
and the block BAT implements functions

$$Z = Z(T). \tag{14}$$

Let symbol $U_i(\Gamma_j)$ stand for implementation of GSA Γ_j by CMCU U_i, and symbol $Q_i(\Gamma_j)$ for the number of macrocells in the logic circuit of the block BAT for CMCU $U_i(\Gamma_j)$, where i = 1, 2. Let inputs of q-th macrocell of BAT receive L_q logical conditions and let each macrocell has S inputs. Application of the proposed method has sense if the condition

$$L_q + R + R_3 + R_4 \le S \tag{15}$$

takes place, where q = 1, … , $Q_2(\Gamma_j)$.



Rys. 2. Diagram strukturalny CMCU U_2
Fig. 2. Structural diagram of CMCU U_2

In this paper there is proposed the method of CMCU U_2 synthesis with the following steps:
1. Construction of the sets C, C_1 and Π_C for GSA Γ.
2. Microinstruction addressing.
3. Construction of the sets Π_A, Π_E and Π_D.
4. Encoding of the classes $B_i \in \Pi_E \cup \Pi_D$.
5. Construction of the control memory content.
6. Construction of the transition table for CMCU.
7. Construction of the table for the address transformer.
8. Synthesis of the logic circuit of CMCU.

4. Example of application of proposed method

Let the sets $C=\{\alpha_1,...,\alpha_9\}$, $C_1=\{\alpha_1,...,\alpha_8\}$ and $\Pi_C=\{B_1,...,B_5\}$ are formed for GSA Γ_1, where $\alpha_1=\langle b_1,b_2\rangle$, $\alpha_2=\langle b_3,...,b_6\rangle$, $\alpha_3=\langle b_7,b_8\rangle$, $\alpha_4=\langle b_5,...b_{13}\rangle$, $\alpha_5=\langle b_4,...,b_{17}\rangle$, $\alpha_6=\langle b_{18},...,b_{21}\rangle$, $\alpha_7=\langle b_{22},...b_{25}\rangle$, $\alpha_8=\langle b_{26},...,b_{28}\rangle$, $\alpha_9=\langle b_{29},...,b_{31}\rangle$, $B_1=\{\alpha_1\}$, $B_2=\{\alpha_2,\alpha_3\}$, $B_3=\{\alpha_4,\alpha_5\}$, $B_4=\{\alpha_6,\alpha_7\}$, $B_5=\{\alpha_8\}$. Thus, I = 5, R_1 = 3, $\tau=\{\tau_1,\tau_2,\tau_3\}$, M = 31, R = 5.

Let us address the microinstructions using some modification of the algorithm from [4]. Now we have

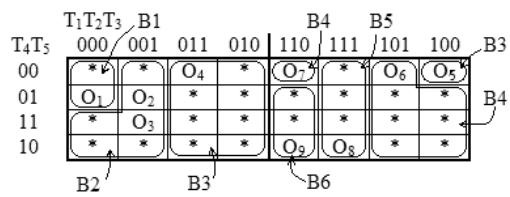
$$A(b_1)=00000,\ldots,A(b_{25})=110000,$$

$$A(b_{26})=11100,\ldots,A(b_{28})=11110,$$

$$A(b_{29})=11001,\ldots,A(b_{31})=11011.$$

Let us construct the Karnaugh map marked by the variables $T_r \in T=\{T_1,...,T_5\}$ (Fig. 3). This map contains outputs of OLC $\alpha_g \in C$ and code space intervals corresponding to the classes $B_i \in \Pi_C$.

The sign * in this map stands for the case when a vertex $b_q \in E_1$ with address A(b_q) is not the output of OLC $\alpha_g \in C_1$. The following code intervals can be derived from Fig. 3: class B_1 corresponds to interval 0000*, class B_2 to 001**, class B_3 to 01*** and 10000, B_4 to 101** and 11000, class B_5 to 111**. Let us point out that $\alpha_9 \notin C_1$ and class $B_6=\{\alpha_9\}$ is not considered here.



Rys. 3. Siatka Karnaugh dla wyjść OLC
Fig. 3. Karnaugh map for outputs of OLC

The obtained intervals determine the sets $\Pi_A=\{B_1,B_2,B_5\}$ and $\Pi_B=\{B_3,B_4\}$. Let N=13 for GSA Γ1 and t=4 for PROM chips in use. In this case we can get R3 = 1 (from (9)) and R2 = 2 (from (7)). Thus, condition (10) is violated and block BAT should be used in CMCU U2(Γ1). Let $\Pi_E=\{B_3\}$, then $\Pi_D=\{B_4\}$. Thus, the sets Π_A, Π_E and Π_D are constructed. Obviously, $V=\{v_i\}$, $Z=\{z_i\}$, let. $K_E(B_3)$=1, $K_D(B_4)$=1. As it was found, $K_A(B_1)$=0000*, $K_A(B_2)$=001**, $K_A(B_5)$=111**. The content of the control memory is constructed in a trivial way [4] and this step is here omitted. Let us point out, that the cells with addresses 10100 and 11000 include the variable v1 = 1. Let transitions for classes B2, B3, B4 be described by the following system of generalized transition formulae [3]:

$$\begin{aligned} B_2 &\to x_3 b_9 \vee \overline{x_3} b_{26};\\ B_3 &\to x_1 b_{18} \vee \overline{x_1} x_2 b_{20} \vee \overline{x_1}\,\overline{x_2} b_{26};\\ B_4 &\to x_5 b_{27} \vee \overline{x_5} b_5. \end{aligned} \tag{16}$$

The system (16) determines the fragment of the transition table with 7 lines (Table 1).

Tab. 1. Tabela tranzycji dla CMCU U2(Γ1)
Tab. 1. Part of the transition table for CMCU U2(Γ1)

B_i	$K_A(B_i)$ $T_1T_2T_3T_4T_5$	$K_B(B_i)$ V_1	$K_D(B_i)$ Z_1	b_q	$A(b_q)$	X_k	Φ_k	h
B_2	001**	0	0	b_9	01000	x_3	D_2	1
				b_{26}	11100	$\overline{x_3}$	$D_1D_2D_3$	2
B_3	*****	1	0	b_{18}	10001	x_1	D_1D_5	3
				b_{20}	10011	$\overline{x_1}x_2$	$D_1D_4D_5$	4
				b_{26}	11100	$\overline{x_1}\,\overline{x_2}$	$D_1D_2D_3$	5
B_4	*****	0	1	b_{27}	11101	x_5	$D_1D_2D_3D_5$	6
				b_5	00100	$\overline{x_5}$	D_3	7

Connection between Table 1 and system (16) is a transparent one. Let us point out that the case v1 = z1 = 0 corresponds to classes $B_i \in \Pi_A$. Otherwise, $B_i \in \Pi_E \cup \Pi_D$ and content of the column $K_A(B_i)$ is ignored. Table 1 is a base for construction of the system (13). For example, the following parts of SOP can be derived from Table 1:

$$D_1 = \overline{T}_1\overline{T}_2T_3\overline{v}_1\overline{z}_1\overline{x}_3 \vee v_1\overline{z}_1 \vee \overline{v}_1z_1x_5;$$
$$D_2 = \overline{T}_1\overline{T}_2T_3\overline{v}_1\overline{z}_1 \vee v_1\overline{z}_1x_1\overline{x}_2 \vee \overline{v}_1z_1x_5.$$

The table of BAT is constructed for the classes $B_i \in \Pi_D$. In our particular case this table includes 2 lines (Table 2).

Tab. 2. Tabela bloków BAT dla CMCU U2(Γ1)
Tab. 2. Table of the block BAT for CMCU $U_2(\Gamma_1)$

α_g	$A(O_g)$	B_i	$K_D(B_i)$	Z_j	j
α_6	10100	B_4	1	z_1	1
a_7	1000				2

Table 2 is the basis to construct the system (14). In our case this system is represented as the following one:

$$z_1 = T_1\overline{T}_2T_3\overline{T}_4\overline{T}_5 \vee T_1T_2\overline{T}_3\overline{T}_4\overline{T}_5.$$

Synthesis of a logic circuit of CMCU $U_2(\Gamma_j)$ is reduced to implementation of systems (13)-(14) using PAL macrocells and control memory using PROM chips. These problems are well-known [4,2] and they are not discussed in our paper.

Let us point out that table of block BAT for CMCU $U_1(\Gamma_1)$ includes 8 lines and $R_1 = 3$. Let macrocell PAL have q=3 terms, then the block BAT of CMCU $U_2(\Gamma_1)$ is implemented using $Q_2(\Gamma_1) = 1$ macrocells. If the classes $B_i \in \Pi_C$ of CMCU $U_1(\Gamma_1)$ are encoded in the following way: $K(B_1) = 000,\ldots, K(B_5) = 100$, then $Q_1(\Gamma_1) = 4$ (if q = 3). In both cases the block BMA is implemented with the same amount of macrocells.

5. Conclusion

The proposed method aims at decrease in the hardware amount (the number of macrocells) in the circuit of an address transformer. In this case the number of macrocells in the block of microinstruction address as well as the number of PROM chips in the block of control memory do not change. The method is based on use of three sources of the codes of pseudoequivalent OLC classes. It is possible due to a wide fan-in of industrial PAL macrocells. Let us point out that the block BAT can be eliminated if one of the conditions (6) or (10) takes place.

Our experiments show that the number of macrocells in the block BAT is decreased up to 60-70% in comparison with the known methods for CMCU design. The total decrease in the hardware amount is up to 10% in comparison with CMCU $U_1(\Gamma_1)$.

6. References

[1] Altera devices, http://www.altera.com/products/devices/ common/dev-family_overview.html.
[2] Baranov S.: Logic Synthesis for Control Automata, New York: Kluwer Academic Publishers, 1994.
[3] Baranov S.: Logic and system design of digital systems, Tallinn: TUT Press, 2008.
[4] Barkalov A., Titarenko L.: Logic Synthesis for FSM - based Control Units, Berlin: Springer, 2009.
[5] DeMicheli G.: Synthesis and Optimization of Digital Circuits, McGraw-Hill, 1994. – 636 pp.
[6] Maxfield C.: The Design Warrior's Guide to FPGAs, Amsterdam: Elseveir, 2004.
[7] Xilinx CPLDs, http://www.xilinx.com/products/silicon_solutions/ cplds/index.htm.

otrzymano / received: 18.10.2011
przyjęto do druku / accepted: 01.12.2011 *artykuł recenzowany / revised paper*

INFORMACJE

Nowa inicjatywa PAK

Na stronie internetowej Wydawnictwa PAK został utworzony dział: **Niepewność wyników pomiarów** w którym są zamieszczane aktualne informacje dotyczące problemów teoretycznych i praktycznych związanych z szacowaniem niepewności wyników pomiarów. W dziale znajdują się:

- aktualne informacje o publikacjach dotyczących niepewności wyników,
- informacje o przedsięwzięciach naukowo–technicznych i edukacyjnych, o tematyce związanej z niepewnością,
- dokumenty dotyczące niepewności,
- pytania do ekspertów (FAQs).

Zapraszamy:

- autorów opublikowanych prac dotyczących niepewności o nadsyłanie tekstów do zamieszczenia w tym dziale,
- organizatorów przedsięwzięć naukowo – technicznych lub edukacyjnych do nadsyłania informacji o imprezach planowanych lub odbytych,
- zainteresowanych zagadnieniami szczegółowymi do nadsyłania pytań do ekspertów.

Materiały mogą mieć formę plików lub linków do źródeł. Warunkiem zamieszczenia w tym dziale strony internetowej PAK materiałów lub linków jest przysłanie do redakcji PAK pocztą zwykłą zgody właściciela praw autorskich na takie rozpowszechnienie. Zamieszczanie i pobieranie materiałów i informacji w tym dziale strony internetowej jest bezpłatne. Redakcja PAK będzie nadzorować zawartość działu, ale za szczegółowe treści merytoryczne odpowiadają autorzy nadsyłanych materiałów.

Tadeusz SKUBIS
Redaktor naczelny Wydawnictwa PAK