Alexander A. BARKALOV, Larysa TITARENKO, Jacek BIEGANOWSKI
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI

Synthesis of control unit with modified operational linear chains

Prof. dr hab. inż. Alexander A. BARKALOV

W latach 1976-1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki Donieckiej. Współpracował aktywnie z Instytutem Cybernetyki im. V.M. Glushkova w Kijowie, gdzie uzyskał tytuł doktora habilitowanego ze specjalnością informatyka. W latach 1996-2003 pracował jako profesor w Instytucie Informatyki Narodowej Politechniki Donieckiej. Od 2004 pracuje jako profesor na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: a.barkalov@iie.uz.zgora.pl

Dr hab. Larysa TITARENKO

W 2004 roku obroniła rozprawę habilitacyjną i uzyskała tytuł doktora habilitowanego ze specjalnością telekomunikacja. W latach 2004-2005 pracowała jako profesor w Narodowym Uniwersytecie Radioelektroniki w Charkowie. Od 2005 pracuje jako adiunkt na Wydziale Elektrotechniki, Informatyki Telekomunikacji Uniwersytetu Zielonogórskiego.

e-mail: l.titarenko@iie.uz.zgora.pl

Mgr inż. Jacek BIEGANOWSKI

Ukończył w roku 2003 studia na kierunku informatyka o specjalności inżynieria komputerowa. Od 2004 roku pracuje w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego na stanowisku asystenta. Jego zainteresowania związane są z programowaniem, systemami operacyjnymi, techniką cyfrową oraz algorytmami ewolucyjnymi.

e-mail: j.bieganowski@iie.uz.zgora.pl

Abstract

The method that considers optimization of the amount of PAL macrocells in the circuit of compositional microprogram control unit is proposed. The method is based on the introduction of additional microinstructions codes of the classes of pseudoequivalent operational linear chains. The proposed method is based on usage of the natural redundancy of embedded memory blocks which are used to implement the control memory. An example of application of proposed method is given.

Keywords: microinstruction, control memory, compositional microprogram control unit, CPLD, PAL.

Synteza jednostki sterującej z wykorzystaniem zmodyfikowanych liniowych łańcuchów operacyjnych

Streszczenie

W artykule przedstawiono metodę optymalizacji liczby makrokomórek PAL mikroprogramowalnego układu sterującego. Proponowana metoda wykorzystuje dodatkowe mikroinstrukcje zawierające kody pseudo-równoważnych liniowych łańcuchów operacyjnych. Rozwiązanie wykorzystuje osadzone bloki pamięci, które często pozostają niezagospodarowane, do implementacji pamięci sterownika. W artykule przedstawiono także przykład zastosowania omawianej metody.

Słowa kluczowe: mikroinstrukcja, mikroprogramowy układ sterujący, CPLD, PAL.

1. Introduction

One of the most important blocks of a digital system is a control unit (CU) coordinating the cooperation of other system blocks [2]. Now programmable logic devices such as CPLD are widely used for implementation of the circuits of CU [3, 4]. The problem of the optimization of hardware amount in the circuit of CU is still actual task in computer science [1, 4] and its solution permits to decrease the cost of the system [9]. One of the ways of optimization is a choice of the structure of CU that perfectly fits to the peculiarities of control algorithm to be interpreted [8]. For example, the linear control algorithm can be interpreted by compositional microprogram control unit (CMCU) [7, 8]. It is well known that optimization of the number of PAL (Programmable Array Logic) elements in the circuit based on CPLD architecture can be reduced due to the minimization of amount of terms in disjunctional normal forms (DNF) of implemented Boolean functions [3]. In this work we propose the method of minimization of the amount of PAL macrocells in the logic circuit of CMCU. This method is based on introduction of additional microinstructions into interpreted control algorithm.

2. Background of CMCU

Let control algorithm be represented by flow-chart of algorithm (FCA) Γ [9] with set of the nodes $B=\{b_0,b_E\}\cup E_1\cup E_2$ and set of edges $E=\{\langle b_q,b_t\rangle \mid b_q,b_t\in B\}$ Here b_0 is an initial node, b_E is a final node, E_1 is a set of operational nodes, E_2 is a set of conditional nodes. An operational node $b_1\in E_1$ contains a collection of microoperations $Y(b_q)\subseteq Y$, where $Y=\{y_1,...,y_N\}$ is a set of microoperations of a digital system. A conditional node $b_q\in E_2$ contains element of the set of logic conditions $X=\{x_1,...,x_L\}$A FCA Γ is named as linear FCA (LFCA), when number $M=|E_1|$ of its operational nodes exceeds the 75% of total number of the nodes [5].

Let set $C=\{\alpha_1,...,\alpha_G\}$ to be formed for LFCA Γ, where $\alpha_g\in C$ is operational linear chain (OLC). An OLC α_g such a sequence of operational nodes $\langle b_{g1},...,b_{gFg}\rangle$, that an edge $\langle b_{gi},...,b_{gi+1}\rangle\in E$ exists for each pair of its adjacent components (i=1,...,F_g–1). Each OLC α_g has only one output O_g and arbitrary number of inputs. The formal definitions of OLC, input of OLC and output of OLC can be found, for example, in [5, 7, 8]. Let each node $b_q\in E_1$ corresponds to microinstruction MI_q with address $A(b_q)$ and let this address has

$$R=\left]log_2 M\right[\tag{1}$$

bits.

Let us to carry out the natural addressing [7, 8] of microinstructions MI_q, where $b_q\in E_1$. In this case condition

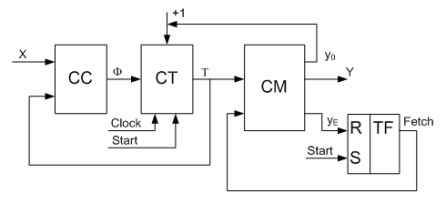
$$A\left(b_{gi+1}\right)=A\left(b_{gi}\right)+1\ \left(g=1,...,G;i=1,...,F_g\right) \tag{2}$$

is hold. For such addressing of microinstructions an algorithm form [5] can be used.

In this case CMCU U_1 (Fig. 1) can be used for interpretation of LFCA Γ [2]. The CMCU U_1 operates in the following manner. If pulse *Start*=1, then the address of the first microinstruction (MI) of a control algorithm is loaded into counter CT. In the same time the flip-flop TF is set up and *Fetch*=1. It initiates the fetching of MIs from control memory CM. The current MI is read out from the CM. If it corresponds to the node $b_q\neq O_g$, then a variable $y_0=1$ is formed together with microoperations $Y(b_q)$. If y_0=1, then pulse *Clock* causes an increment of content of the counter for

addressing of the next microinstruction MI_j, where $\langle b_q, b_j \rangle \in E$. If $b_q=O_g$, then $y_0=0$ and address of the next microinstruction to be executed (address of transition) is formed by circuit CC that forms excitation functions of the flip-flops of the counter CT

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



Rys. 1. Struktura mikroprogramowego układu sterującego U_1
Fig. 1. The diagram of CMCU U_1 structure

Here $T=\{T_1,...,T_R\}$ is the set of internal variables corresponding to address bits of microinstruction address. When microinstruction MI_q is readout and $\langle b_q, b_E \rangle \in E$, then $y_E=1$, TF is reset *Fetch*=0 and operation of CMCU U_1 is terminated.

If logic circuit of CMCU U_1 is implemented as a part of the SoPC (system-on-a-programmable-chip) [4, 6], then circuit CC can be implemented using PAL macrocells of CPLD and control memory can be implemented using EMB (embedded memory block). The main feature of CMCU U_1 is a minimal number of outputs of the circuit CC when comparing with other organizations of CMCU [5, 7, 8]. It gives a potential possibility to minimize the number of macrocells in the circuit CC. But circuit CC and counter CT form Moore finite-state-machine (FSM) [5], where number of the terms in the DNF of functions (3) can be much more, than in case of equivalent Mealy FSM [9]. To minimize the number of macrocells in the circuit CC, the numbers of the terms in the system (3) of CMCU U_1 should be decreased. It can be executed due to existence of pseudoequivalent OLC (POLC) of FCA Γ [8], which correspond to the pseudoequivalent states of Moore FSM [10].

The OLC $\alpha_i, \alpha_j \in C$ are named as POLC, if their outputs are connected with the input of the same node of FCA Γ [5]. Let $\Pi_C=\{B_1,...,B_I\}$ be a partition of the set $C' \subset C$ on the classes of POLC, where, $\alpha_j \notin C'$, if its output is connected with input of final node b_E. Let us encode each class $B_i \in \Pi_C$ by binary code $K(B_i)$ with R_1 bits, where

$$R_1 = \left\rceil \log_2 I \right\lceil. \tag{4}$$

The number of terms in the system (3) can be diminished up to the corresponding value of equivalent Mealy FSM due to usage of address transformer AT [1]. AT transforms the addresses of the outputs of POLC $\alpha_i \in B_i$ into the code $K(B_i)$, where $i=1,...,I$. Let us point out that such approach leads to usage of some resources of SoPC for implementation of AT. It has sense only, if common price of CC and AT in resulting circuit will be less, than the price of CC in CMCU U_1.

3. The main idea of proposed method

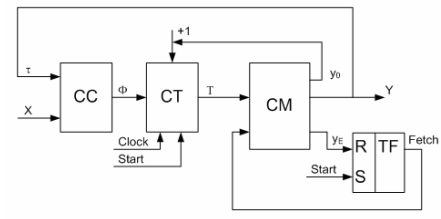
Control memory of CMCU U_1 contains 2^R words and M of them contain the microinstructions of interpreted algorithm. If condition

$$2^R - M \geq |C'| \tag{5}$$

holds, then each of OLC $\alpha_j \in C'$ can be modified by inserting an additional node O_g. In this case the microprogram includes two types of microinstructions (2).

The first bit of any format corresponds to variable y_0, field FY (Fig. 2a) contains information about the microoperations to be formed, field FB contains code $K(B_i)$. All additional microinstructions have format that is shown in Fig. 2b. The node O_g contains the code $K(B_i)$, where $\alpha_g \in B_i$. Such approach leads to CMCU U_2 with modified OLCs (Fig. 3).

Rys. 2. Formaty mikroinstrukcji
Fig. 2. Formats of microinstructions



Rys. 3. Struktura mikroprogramowego układu sterującego U_2
Fig. 3. The diagram of CMCU U_2 structure

The principles of operation of both U_1 and U_2 are the same, but excitation functions of CT in CMCU U_2 depend on variables $\tau_r \in \tau = \{\tau_1,...,\tau_{R1}\}$ that are used to encode the classes $B_i \in \Pi_C$:

$$\Phi = \Phi(\tau, X). \tag{6}$$

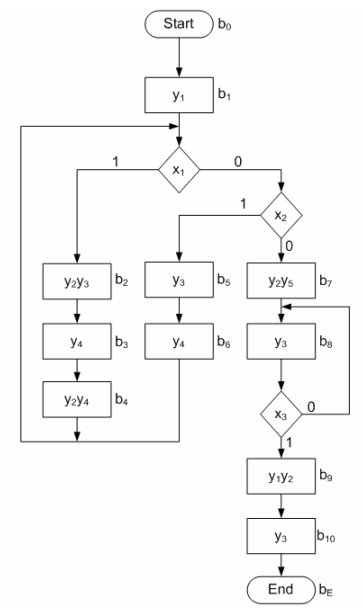
The field FB is the source of the variables $\tau_r \in \tau$. Let us point out that operational unit [2, 7] of the digital system is in idle state, if counter CT contains an address of an additional microinstruction.

The method of design of CMCU U_2 is proposed in this work. It includes the following steps:

1. Construction of the set C of LFCA Γ.
2. Modification of the OLC $\alpha_g \in C'$.
3. Natural addressing of microinstructions.
4. Encoding of the classes of POLC $B_i \in \Pi_C$.
5. Construction of the content of control memory.
6. Construction of the table of transitions of CMCU.
7. Implementation of the circuit CC using the macrocells PAL and implementation of CM using the EMBs.

4. Example of application of proposed method

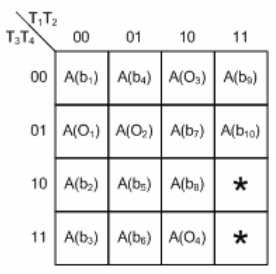
Let the symbols $U_i(\Gamma_j)$ mean that CMCU U_i interprets the LFCA Γ_j. Let us discuss an example of design of CMCU $U_2(\Gamma_1)$, where LFCA Γ_1 is shown in Fig. 4.



Rys. 4. Początkowa sieć działań Γ_1
Fig. 4. Initial flow-chart of algorithm Γ_1

In case of LFCA Γ_1 we have set C = {$\alpha_{1,...,}\alpha_5$}, where $\alpha_1 = \langle b_1\rangle$, $I_1^1 = b_1$; $\alpha_2 = \langle b_2, b_3, b_4\rangle$, $I_2^1 = b_2$; $\alpha_3 = \langle b_5, b_6\rangle$, $I_3 = b_5$; $\alpha_4 = \langle b_7, b_8\rangle$, $I_4^1 = b_7$, $I_4^2 = b_8$; $\alpha_5 = \langle b_9, b_{10}\rangle$, $I_5^1 = b_9$. It is clear that $\alpha_5 \notin C'$, $|G'| = 4$. In this example $M = 10$, $2^R = 16$, condition (5) holds and application of proposed method has sense.

Let us insert the components O_g into OLC $\alpha_g \in C'$. It leads to modified OLCs $\alpha_1 = \langle b_1, O_1\rangle$, $\alpha_2 = \langle b_2, b_3, b_4, O_2\rangle$, $\alpha_3 = \langle b_5, b_6, O_3\rangle$, $\alpha_4 = \langle b_7, b_8, O_4\rangle$. The natural addressing (2) is executed when there are used methods from [5] and in our case it leads to following addresses of microinstructions (Fig. 5).



Rys. 5. Adresy mikroinstrukcji dla układu $U_2(\Gamma_1)$
Fig. 5. Addresses of microinstructions of CMCU $U_2(\Gamma_1)$

Let us divide LFCA Γ_1 into partitions $\Pi_C = \{B_1, B_2\}$, where $B_1 = \{\alpha_1, \alpha_2, \alpha_3\}$, $B_2 = \{\alpha_4\}$. It means that $I = 2$, $R_1 = 1$, $\tau = \{\tau_1\}$. Let $K(B_1) = 0$, $K(B_2) = 1$. In case of CMCU $U_2(\Gamma_1)$ the microprogram occupies 14 cells of EMB. The first 4 lines of control memory of CMCU $U_2(\Gamma_1)$ are shown in the Table 1.

Tab. 1. Fragment zawartości pamięci sterującej CMCU $U_2(\Gamma_1)$
Tab. 1. A piece of the content of control memory of CMCU $U_2(\Gamma_1)$

Address	y_0	FY						Symbol	
$T_1T_2T_3T_4$		FB						b_q	B_i
		y_1/τ_1	y_2	y_3	y_4	y_5	y_E		
0000	1	1	0	0	0	0	0	b_1	B_1
0001	0	0	*	*	*	*	*	O_1	B_1
0010	1	0	1	1	0	0	0	b_2	B_1
0011	1	0	0	0	1	0	0	b_2	B_1

The second bit of the word of CM corresponds to y_1 (*if* $y_0 = 1$) or to τ_1 (if $y_0 = 0$). The idle state of operational unit can be organized by termination of its timing pulses, if $y_0 = 0$.

To form a table of transitions of CMCU U_2 the system of the formulae of transition [9] should be used. In our case it is the following one:

$$B_1 \to x_1 b_2 \vee \overline{x}_1 x_2 b_5 \vee \overline{x}_1 \overline{x}_2 b_7;$$
$$B_2 \to x_3 b_9 \vee \overline{x}_3 b_8. \qquad (7)$$

The table of transitions consists the columns: B_i, $K(B_i)$, b_q, $A(b_q)$, X_h, Φ_h, h. The system (7) is the base on which Table 2 is to be formed

Tab. 2. Tabela przejść mikropogramowego układu sterującego $U_2(\Gamma_1)$
Tab. 2. Table of transitions of CMCU $U_2(\Gamma_1)$

B_i	$K(B_i)$	b_q	$A(b_q)$	X_h	Φ_h	h
B_1	0	b_2	0010	x_1	D_3	1
		b_5	0110	$/x_1x_2$	D_2D_3	2
		b_7	1001	$/x_1/x_2$	D_1D_4	3
B_2	1	b_9	1100	x_3	D_1D_2	4
		b_8	1010	x_3	D_1D_3	5

It is clear from Table 2, that $\Phi = \{D_1, \ldots, D_4\}$ Table of transitions is the base on which the system (6) is formed. For example, from the Table 2 we can form: $D_3 = \overline{\tau}_1 x_1 \vee \overline{\tau}_1 \overline{x}_1 x_2 \vee \tau_1 \overline{x}_3$.

The implementation of the circuit of CMCU U_2 is reduced to the one of the system (6) that use macrocells of CPLD and implementation of the CM on EMBS use the table of its content. These questions are out of the scope of this article.

Let us point out that number of terms in the DNF of system (6) is equal to number of the lines $H_2(\Gamma)$ of the table of transitions. In our case $H_2(\Gamma_1) = 5$, but table of transitions of CMCU $U_1(\Gamma_1)$ consists $H_1(\Gamma_1) = 11$ lines. It gives us possibility to decrease amount of hardware in the circuit CC of CMCU $U_2(\Gamma_1)$ in comparison with hardware amount in the circuit CC of CMCU $U_1(\Gamma_1)$.

5. Conclusion

Proposed method of modification of OLC permits to decrease the amount of microcells PLA in the combinational part of CMCU. Optimization is based on natural redundancy of the embedded memory blocks implementing the control memory of CMCU. Such approach permits to eliminate any additional blocks in the structure of CMCU, that means optimization is not connected with introduction of the address transformer AT.

Our research show that decrease of hardware amount is proportional to coefficient

$$\eta = \frac{H_1(\Gamma)}{H_2(\Gamma)}. \qquad (8)$$

The value of η depends on characteristics of particular LFCA Γ. It is equal to ratio of the lengths of direct structural tables of equivalent Moore and Mealy FSM used for implementation of FCA Γ.

The main disadvantage of proposed method is an increase in time of digital system operation due to idle cycles of operational unit (data path of the system). Therefore, it can be applied, when project time characteristics correspond to initial restrictions. Our researches show that application of proposed method permits up to 18–22% decrease in hardware amount in comparison with CMCU without modification of operational linear chains.

6. References

[1] De Micheli G. Synthesis and Optimization od Digital Circuits. McGraw Hill, NY, 1994, 578 pp.
[2] Gajski D. Principles of Digital Design. Prentice Hall, NY, 1997, 418 pp.
[3] V. V. Solovjev. Design of digital systems using the programmable logic integrated circuits. Hot Line-Telecom, Moscow, 2001, (in Russian), 636 pp.
[4] D. Kania. Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL. Zeszyty Naukowe Politechniki Śląskiej, Gliwice, 2004, 240 str.
[5] A. A. Barkalov. Synthesis of control units on programmable logic devices. Donetsk National Technical University, Donetsk, 2002, (in Russian), 262 pp.
[6] C. Maxfield. The Design Warrior's Guide to FPGAs. Academic Press, Inc., Orlando, FL, USA, 2004, 542 pp.
[7] M. Adamski and A. Barkalov. Architectural and sequential synthesis of digital devices, ISBN: 83-7481-039-4. University of Zielona Góra, Zielona Góra, 2006, 199 pp.
[8] A. Barkalov and M. Węgrzyn. Design of control units with programmable logic, ISBN: 83-7481-042-4. University of Zielona Góra Press, Zielona Góra, 2006, 150 pp.
[9] S. I. Baranov. Logic synthesis of Control Automata. Kluwer Academic Publishers, Boston, 1994, 312 pp.
[10] A. A. Barkalov. Principles of optimization of logic circuit of Moore FSM, pages 65–72. Cybernetics and system analysis, 1998, (in Russian).

Artykuł recenzowany