

Małgorzata KOŁOPIEŃCZYK, Larysa TITARENKO, Aleksander BARKALOV
UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Design of CMCU with expanded microinstruction and elementary OLC's

Mgr inż. Małgorzata KOŁOPIEŃCZYK

Mgr inż. Małgorzata Kołopieńczyk jest absolwentem Uniwersytetu Zielonogórskiego. Ukończyła studia o specjalności Inżynieria Komputerowa. Od roku 1999 pracuje jako asystent na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.



e-mail: m.kolopienczyk@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 i Telekomunikacji Uniwersytetu Zielonogórskiego.



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

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

Abstract

The method of design of compositional microprogram control unit with codes sharing and expanded microinstruction is proposed. The proposed method is based on application of special address transformer to form an address of microinstruction on the base of its representation as pair <code of operational linear chain, code of component>. The control algorithm is described using flow-chart with elementary operational linear chains. Such approach permits to use all positive features of codes sharing independently on characteristics of interpreted flow-chart of algorithm. The proposed method permits to decrease the size of control memory in comparison with all known methods of such control units design. An example of proposed method application is given.

Keywords: Compositional Microprogram Control Unit, code sharing, elementary operational linear chains.

Projektowanie CMCU z rozszerzonymi mikroinstrukcjami i elementarnymi OLC

Streszczenie

W artykule zaprezentowano sposób projektowania mikroprogramowego układu sterującego z wykorzystaniem metody współdzielenia kodów z rozszerzonymi mikroinstrukcjami. W metodzie tej wprowadzono specjalny transformator adresów, który generuje adresy mikroinstrukcji reprezentowane poprzez pary <kod łańcucha, kod elementu łańcucha>. Do opisu algorytmu sterowania wykorzystano sieć działań z elementarnymi łańcuchami. Takie podejście umożliwiło wykorzystanie wszystkich zalet metody współdzielenia kodów niezależnie od charakterystyki interpretowanej sieci działań. Zaproponowana metoda umożliwia zmniejszenie rozmiaru pamięci w stosunku do wszystkich innych znanych metod projektowania jednostek sterujących. W artykule przedstawiono także przykład zastosowania proponowanej metody.

Słowa kluczowe: Mikroprogramowany układ sterujący, współdzielenie kodów, elementarne łańcuchy.

1. Introduction

A control unit (CU) is one of the most important parts of any digital system [8]. It can be implemented as compositional

microprogram control unit (CMCU) [3]. Modern technology level permits to implement sufficiently complex digital system using single integrated chip of the type "system-on-a-chip" (SoC) [4, 5]. As a rule, SoC includes the field-programmable gate arrays (FPGA), which consist from LUT-elements (look-up table) and dedicated memory blocks (DMB) [4]. If model of CMCU is used to implement the control unit, then FPGA-blocks implement the system of microinstruction addressing and DMB implement the control memory with these microinstructions.

The limited amount of inputs of LUT-elements leads to necessity of decreasing of the amount of arguments in the addressing circuit of CMCU. One of the methods to solve this problem is application of code sharing with expanded microinstruction and it permits to use the optimization methods of Moore FSM [1] in case of CMCU. The method of CMCU design is proposed in this article, such that it permits to use all advantages of code sharing and to preserve a minimal value of control memory. Moreover, there is a potential possibility to decrease the amount of DMB in comparison with known design methods.

2. Main definitions and background of CMCU

Let a control algorithm of a digital system is represented as a flow-chart of algorithm (FCA) Γ [1] with set of nodes $B = \{b_0, b_E\} \cup B_1 \cup B_2$ and set of edges $E = \{(b_i, b_j) \mid b_i, b_j \in B\}$. Here b_0 is a start node of FCA, b_E is a finish node of FCA, B_1 is a set of operational nodes with the collections of microoperations (microinstructions) $Y_{q_i} \subseteq Y$, where $Y = \{y_1, \dots, y_N\}$, B_2 is a set of conditional nodes with elements of the set of logic conditions $X = \{x_1, \dots, x_L\}$.

Let us form a set of elementary operational linear chains (EOLC) $C = \{\alpha_1, \dots, \alpha_G\}$ for FCA Γ and let this set satisfies to the following condition:

$$\begin{aligned} D^i \cap D^j &= 0 (i \neq j; i, j \in \{1, \dots, G\}); \\ D^1 \cup D^2 \cup \dots \cup D^G &= B^1 \\ G &\Rightarrow \min. \end{aligned} \quad (1)$$

Let F_g is an amount of components of EOLC and let $F_{max} = \max(F_1, \dots, F_G)$. Let us encode the EOLC by the binary code $K(\alpha_g)$ with R_1 bits, where $R_1 = \lceil \log_2 G \rceil$, and let us encode the components of EOLC by binary codes $K(b_i)$ with R_2 bits, where $R_2 = \lceil \log_2 F_{max} \rceil$. Let us execute this encoding in such manner that condition

$$K(b_{g_{i+1}}) = K(b_{g_i}) + 1 \quad (g = 1, \dots, G) \quad (2)$$

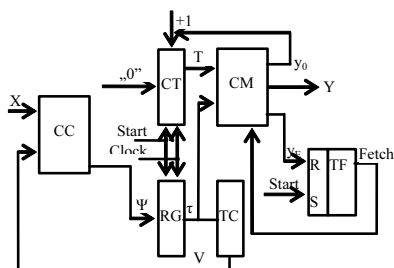
holds for any pair of adjacent components $b_{g_i}, b_{g_{i+1}}$ $i = (1, \dots, F_g - 1)$ of any EOLC.

In this case an address $A(b_i)$ of microinstruction $Y(b_i)$ from the node $b_i \in B_I$, such that $b_i \in D^g$, can be represented as

$$A(b_i) = K(\alpha_g) * K(b_i) \quad (3)$$

where $*$ is a sign of concatenation. The representation (3) is named codes sharing [3]. Let us find partition $\Pi_C = \{B_1, \dots, B_I\}$ of the set C on the classes of pseudoequivalent *OLCs*. Let us encode each class $B_i \in \Pi_C$ by binary code $K(B_i)$ with R_0 bits, where $R_0 = \lceil \log_2 I \rceil$, and let us use the variables from the set $V = \{v_1, \dots, v_{R_0}\}$ for such encoding.

The Fig. 1 shows the structural diagram of *CMCU* U_1 , which is based on address representation (3) and transformation of the codes of *EOLC* into the codes of the classes of pseudoequivalent *OLC*.



Rys. 1. Struktura mikroprogramowanego układu sterującego U_1
Fig. 1. Structural diagram of *CMCU* U_1

Here combinational circuit *CC* implements the system of excitation functions of the flip-flops of register *RG*

$$\Psi = (V, X). \quad (4)$$

It means that the circuit *CC* represents a circuit of microinstruction addressing of *CMCU*. Counter *CT* keeps a code $K(b_i)$ of component of *EOLC*, which is represented by internal variables $T_r \in T$, where $|T| = R_2$. Register *RG* keeps a code $K(\alpha_g)$ of *EOLC*, which is represented by internal variables $\tau_r \in \tau$, where $|\tau| = R_1$. The code transformer *TC* implements the system of functions

$$V = V(\tau) \quad (5)$$

and it forms the codes of the classes $B_i \in \Pi_C$. The control memory *CM* keeps microoperations $y_n \in Y$, the internal variable of synchronization control y_0 and internal variable of microinstruction fetch control y_E . The flip-flop *TF* forms signal *Fetch* to write out microinstructions from *CM*.

Such organization of a control unit permits to use the minimal number of feedback signals in comparison with all known structures of *CMCU* [3]. But if condition

$$R_1 + R_2 > R \quad (6)$$

holds, where parameter R is determined using amount of operational nodes $M = |B_I|$ as

$$R = \lceil \log_2 M \rceil, \quad (7)$$

then the size of control memory *CM* is increased in comparison with its value

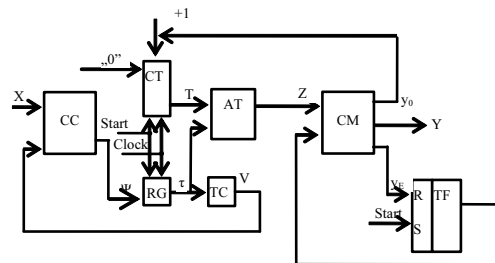
$$V_0 = 2^R * N_0, \quad (8)$$

where N_0 is amount of bits needed to represent the microoperations $y_n \in Y$ and internal variables y_0 and y_E . This size of *CM* characterizes all known structures of *CMCU* [3]. If condition (6) holds, then amount of *DMBs* to implement the circuit of *CM* is increased. The method of organization of *CMCU* with expanded microinstruction and elementary operational linear chains is

proposed here and this method permits to save and even to decrease the size of *CM* in comparison with (8) for arbitrary *FCA* Γ .

3. Design of *CMCU* with *EMI* and *EOLC*

The proposed method is based on introducing of the block of address transformer (*AT*) in the *CMCU* U_1 . This block transforms an address (3) into address with minimal bits capacity R and it leads to *CMCU* U_2 (Fig. 2).



Rys. 2. Struktura mikroprogramowanego układu sterującego U_2
Fig. 2. Structural diagram of *CMCU* U_2

Here the address transformer *AT* forms functions

$$Z = Z(T, \tau), \quad (9)$$

which are used for microinstruction addressing, $Z = \{z_1, \dots, z_R\}$. The principles of operation of both *CMCU* U_1 and U_2 are the same.

Control memory of *CMCU* U_2 keeps the elements of the set $Y_0 = Y \cup \{y_0, y_E\}$. Let *CM* keeps M_I different collections $Y_m \subseteq Y_0$ and let us name them as expanded microinstructions (*EMI*). Let us encode each *EMI* $Y_m \subseteq Y_0$ by binary code $K(Y_m)$ with

$$R_3 = \lceil \log_2 M_I \rceil \quad (10)$$

bits and let us use the variables $z_r \in Z$ for such encoding. The code $K(Y_m)$ can be used as address of *EMI* Y_m ($m = 1, \dots, M_I$) in the control memory. If condition

$$R_3 < R \quad (11)$$

holds, then size of *CM* is less than V_0 .

Here we are proposing the method of design of *CMCU* U_2 and this method includes the following steps: 1. Construction of a transformed flow-chart Γ ; 2. Construction of the set *EOLC* of the transformed *FCA*; 3. Encoding of *EOLCs*; 3. Encoding of components of *EOLCs*; 4. Addressing of *EMI*; 5. Construction of content of control memory; 6. Encoding of the classes of pseudoequivalent *EOLCs*; 7. Construction of the table of transitions of *CMCU*; 8. Construction of the table of address transformer; 9. Construction of the table of code transformer; 10. Construction of the functions Ψ, V, Z ; 10. Implementation of the circuit of *CMCU*.

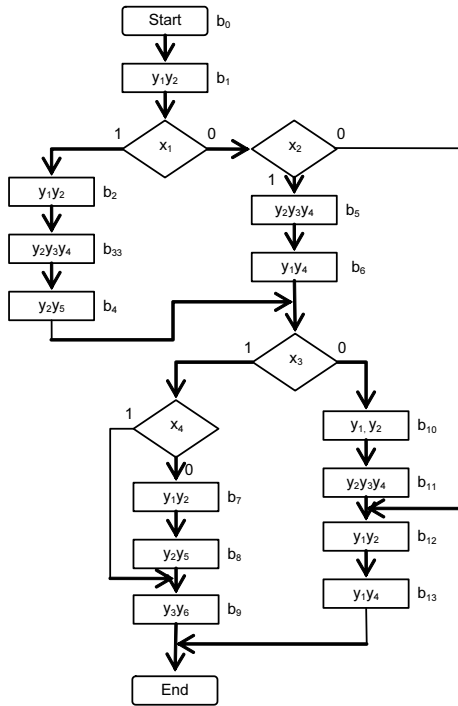
Let us discuss an application of this method using as example the *CMCU* $U_2(\Gamma)$, that is *CMCU* U_2 that interprets *FCA* Γ_1 (Fig. 3).

Transformation of initial *FCA*

This transformation is reduced to introduction of additional operational nodes and to inserting of additional internal variables into the nodes of initial *FCA*. In case of *FCA* Γ_1 the internal variable y_E is inserted into the nodes b_9 and b_{13} , the structure of *FCA* is unchangeable.

Construction of the set *EOLC*

Using the method from the work [3] leads to the set $C = \{\alpha_1, \dots, \alpha_7\}$, where $\alpha_1 = \langle b_1 \rangle, I_1^1 = O_1 = b_1, \alpha_2 = \langle b_2, b_3, b_4 \rangle, I_2^1 = b_2, O_2 = b_4, \alpha_3 = \langle b_5, b_6 \rangle, I_3^1 = b_5, O_3 = b_6, \alpha_4 = \langle b_7, b_8 \rangle, I_4^1 = b_7, O_4 = b_8, \alpha_5 = \langle b_9 \rangle, I_5^1 = O_5 = b_{10}, \alpha_6 = \langle b_{10}, b_{11} \rangle, I_6^1 = b_{10}, O_6 = b_{11}, \alpha_7 = \langle b_{12}, b_{13} \rangle, I_7^1 = b_{12}, O_7 = b_{13}$. An analysis of this set gives that $G = 7, F_{max} = 3, O(\Gamma) = \{b_1, b_4, b_6, b_8, b_9, b_{11}, b_{13}\}$. In our example we have $R_1 = 3, R_2 = 2, M = 13, R = 4$.



Rys. 3. Początkowa sieć działań Γ_1
 Fig. 3. Initial flow-chart of algorithm Γ_1

Encoding of EOLCs

This step is executed in a trivial way. Let the EOLCs of CMCU $U_2(\Gamma_1)$ are encoded in the following manner: $K(\alpha_1)=000$, $K(\alpha_2)=001, \dots, K(\alpha_7)=110$.

Encoding of the components of EOLC

Let the first component of each EOLC $\alpha_g \in C$ has code with all zeros. The codes of the next components are determined according to (2). In case of $U_2(\Gamma_1)$ we can get the following codes: $K(b_1)=K(b_2)=K(b_3)=K(b_5)=K(b_9)=K(b_{10})=K(b_{12})=00$; $K(b_4)=K(b_6)=K(b_8)=K(b_{11})=K(b_{13})=01$; $K(b_7)=10$.

Addressing of EMI

Control memory of CMCU $U_2(\Gamma_1)$ includes $M_1=7$ expanded microinstructions, where $Y_1=\{y_1, y_2\}$, $Y_2=\{y_0, y_1, y_2\}$, $Y_3=\{y_0, y_2, y_3, y_4\}$, $Y_4=\{y_2, y_3\}$, $Y_5=\{y_1, y_4\}$, $Y_6=\{y_E, y_3, y_6\}$, $Y_7=\{y_2, y_3, y_4\}$, $Y_8=\{y_E, y_1, y_4\}$. Thus $R_3=3$, $Z=\{z_1, \dots, z_3\}$, condition (11) holds and application of proposed method has sense.

Let us encode the EMIs Y_m ($m=1, \dots, 8$) in the following manner: $K(Y_1)=000$, $K(Y_2)=001, \dots, K(Y_8)=110$.

Construction of the CM content

The CM of CMCU U_2 is represented by a table with the columns [3, 4]: $K(Y_m)$, Y_m , b_i . In case of CMCU $U_2(\Gamma_1)$ this table includes $M_1=8$ lines (Table 1).

Tab. 1. Zawartość pamięci
 Tab. 1. Content of control memory

$K(Y_m)$	Y_m	b_i	$K(Y_m)$	Y_m	b_i
000	y_1, y_2	b_1	100	y_1, y_4	b_6
001	y_0, y_1, y_2	b_2, b_7, b_{10}, b_{12}	101	y_E, y_3, y_6	b_9
010	y_0, y_2, y_3, y_4	b_3, b_5	110	y_2, y_3, y_4	b_{11}
011	y_2, y_5	b_4, b_8	111	y_E, y_1, y_4	b_{13}

Optimization of the size of CM is possible when different nodes $b_i \in B_1$ contain the same expanded microinstructions.

Encoding of the classes of pseudoequivalent EOLCs

In case of CMCU $U_2(\Gamma_1)$ we have $\Pi_C=\{B_1, B_2, B_3, B_4, B_5\}$, where $B_1=\{\alpha_1\}$, $B_2=\{\alpha_2, \alpha_3\}$, $B_3=\{\alpha_4\}$, $B_4=\{\alpha_6\}$, $B_5=\{\alpha_5, \alpha_7\}$. Thus, $R_0=3$, $V=\{v_1, v_2, v_3\}$. Let us encode the classes $B_i \in \Pi_C$ in the following manner: $K(B_1)=000, \dots, K(B_5)=100$.

Construction of the table of transitions of CMCU

This table is the base to form the functions (4). A part of the CMCU U_2 transitions table is show in the Table 2.

Tab. 2. Fragment tabeli przejść CMCU U_2
 Tab. 2. Fragment of the CMCU U_2 transitions table

B_i	$K(B_i)$	I_g^j	$A(I_g^j)$	X_h	Ψ_h	h
B_1	000	I_2^1	00100	x_1	D_3	1
		I_3^1	01000	$/x_1x_2$	D_2	2
		I_7^1	11000	$/x_1/x_2$	D_1D_2	3

Construction of the table of address transformer

This table is the base to form the functions (9) and it includes the columns: b_m , $A(b_m)$, Y_m , $K(Y_m)$, Z_m , m . The fragment of the table of address transformer of CMCU U_2 is shown in the Table 3.

Tab. 3. Fragment tabeli konwertera adresów
 Tab. 3. Fragment of the table of address transformer

b_m	$A(b_m)$	Y_m	$K(Y_m)$	Z_m	m
b_1	00000	Y_1	000	-	1
b_2	00100	Y_2	001	z_3	2
b_3	00101	Y_3	010	z_2	3
b_4	00110	Y_4	011	z_2z_3	4

Construction of the table of code transformer

This table is the base to form system (5) and it includes the columns: α_g , $K(\alpha_g)$, B_i , $K(B_i)$, V_g , g . Here V_g is a collection of variables $v_j \in V$, which are equal to 1 in the code $K(B_i)$, where $\alpha_g \in B_i$ and $B_i \in \Pi_C$.

Construction of the system of functions Ψ, Z, V

The functions Ψ are formed using the table of transitions of CMCU U_2 . The functions Z are formed using the table of address transformer. The functions V are formed using the table of code transformer. Let us point out that the most natural way to implement both CM and TC is application of DMBs.

Implementation of the circuit of CMCU

The methods of circuit implementation on the base of CPLD, FPGA, PROM are well-know [2, 6, 7, 9]. These methods are out of the scope of our article.

4. Conclusion

The method design of CMCU with EMI and EOLC permits to decrease the hardware amount (number of LUT-elements) in the addressing circuit CC due to usage of pseudoequivalent operational linear chains of initial FCA Γ . The application of address transformer AT permits to use the method of codes sharing under any characteristics of FCA, which determine the values of R, R_1, R_2, R_3 . We should point out that existence of AT causes increasing of the cycle time of a digital system with CMCU. Thus, this method can be applied only if resulting project meets required time characteristics. These outcomes are true for all proposed structures of compositional microprogram control units.

5. Literatura

- [1] Barkalov A.A: Principles of optimization of logical circuit of Moore finite-state machine. - Cybernetics and system analysis. 1998, №1.c.65 - 72.
- [2] Barkalov A. A.: Synthesis of Control Units on PLDs, DonNTU, Donetsk, 2002 (in Russian). 262 p.
- [3] Barkalov A. A, Palagin A. V.: Synthesis of Microprogram Control Units, IC NAC of Ukraine, Kiev, 1997 (in Russian). 135 p.
- [4] Grushnitsky R., Mursaev A., Ugrjumov E.: Development of systems on chips with programmable logic. - SPb: BHV - Petersburg, 2002 (in Russian) 626 p.
- [5] H. Iwai., Future CMOS Scaling. - Proceeding of the 11th International Conference "Mixed Design of Integrated Circuits and System.", Szczecin, Poland, 2004.
- [6] Luba T.: Synteza układów logicznych. - Warszawa: WSIZ, 2001. 238 s.
- [7] Salcic Z.: VHDL and FPLDs in Digital Systems Design, prototyping and Customization. - Kluwer Academic Publishers, 1998. 412 pp.
- [8] Sasao T.: Switching Theory for Logic Synthesis. - Kluwer Academic Publishers, 1999. 316 pp.
- [9] Solovjev V.: Design of digital systems using the programmable logic integrate circuits. - Moscow: Hot line - Telecom, 2001 (in Russian). 636 p.