DESIGNING CPLD-BASED CONTROL UNITS FOR EMC

Titarenko L.A., Barkalov A.A.

Institute of Metrology, Electronics and Computer Science, University of Zielona Góra, Zielona Góra, Poland

E-mail: {l.titarenko, a.barkalo}@imei.uz.zgora.pl

Abstract

The method of hardware reduction is proposed which is oriented on compositional microprogram control units and CPLDs. The method is based on using multiple sources of addresses. This allows minimizing the number of PAL macrocells in the resulting circuit of control unit. The method targets control blocks for electromagnetic compatibility of radiotechnical devices

1. Introduction

One of very important problems connected with VLSI-based design is a providing the electromagnetic compatibility for different radiotechnical devices. It is very important, for example, for real-time embedded systems [1]. Special hardware blocks can help in solving this problem. These blocks can be represented by a model of compositional microprogram control unit (CMCU) [2]. Very often, these blocks are implemented using complex programmable logic devices (CPLDs) [3, 4]. CPLDs are based on macrocells having a wide fan-in. This specific can be used for optimizing the CMCU circuit [2].

This article proposes an approach for optimizing CMCU circuits. The method is based on a wide fan-in of PAL macrocells [3,4]. This approach aims at CMCU with address transformer [2].

2. Main idea of proposed method

CMCU includes a control memory (CM). In the case of CPLD, the CM is implemented using internal programmable read-only memory (PROM) blocks of a chip. The number of cell outputs (t) can be taken from the set {1, 2,4,8, 16} [3, 4]. We propose to address the components of OLC $\alpha_g \in C_1$ in such a manner that maximal possible amount of classes $B_i \in \Pi_C$ was represented by a single generalized interval of R-dimensional Boolean space.

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

$$\Pi_B = \emptyset \tag{1}$$

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

$$R_2 = \lceil \log_2(I_B + 1) \rceil \tag{2}$$

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

$$R_0 = \left\lceil \frac{N+2}{t} \right\rceil \quad (3)$$

memory blocks with enough amount of cells (not less than M). In this case, R_3 outputs of PROM are free, where

$$R_3 = R_0 * t - N - 2 \quad (4)$$

If condition

$$R_3 \geq R_2, \quad (5)$$

takes place, then there is no need in the block of address transformer (BAT). Otherwise, it is necessary to represent Π_B as $\Pi_E \cup \Pi_D$, where $I_E = |\Pi_E|$, $I_D = |\Pi_D|$. In this case

$$I_E = 2^{R_3} - 1, \quad (6)$$

$$R_4 = \lceil \log_2 (I_D + 1) \rceil. \quad (7)$$

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 relation $B_i \in \Pi_D$. Thus, only the outputs of OLC $\alpha_g \in B_i$ should be transformed. Based on this analysis, we propose a CMCU U_1 (Fig. 1).

In CMCU U_1, 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 U_2, block BMA implements functions

$$\Phi = \Phi(T, Z, V, X), \quad (8)$$

and block BAT implements functions

$$Z = Z(T). \quad (9)$$

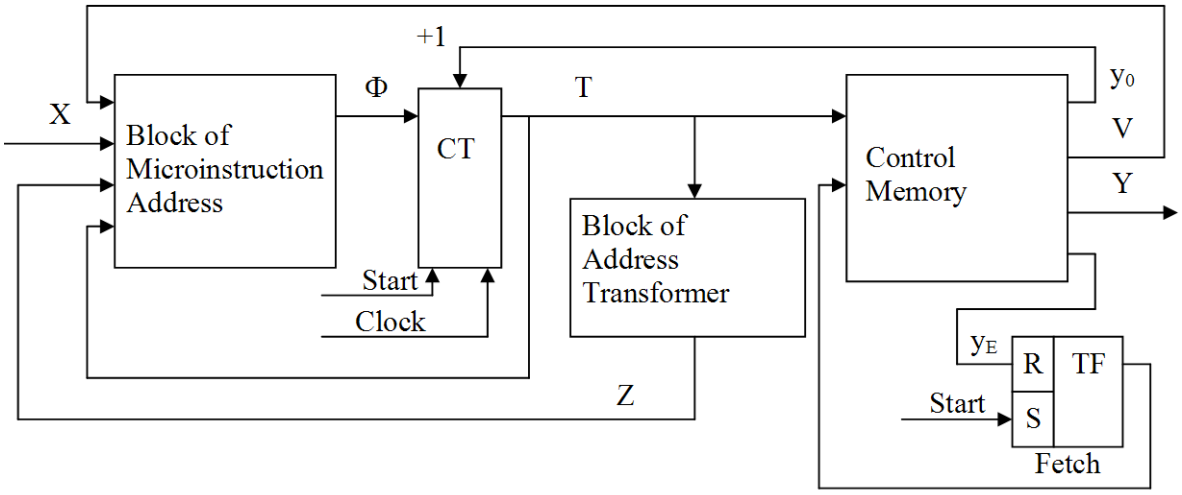

Fig. 1. Architecture of CMCU U_1

In this article the method of CMCU U_1 synthesis is proposed with the following steps: Construction of the sets C, C_1 and Π_C; Microinstruction addressing; Construction of the sets Π_A, Π_E and Π_D; Encoding of the classes $B_i \in \Pi_E \cup \Pi_D$; Construction of control memory content; Construction of transition table for CMCU; Construction of the table for address transformer; Synthesis of the logic circuit of CMCU.

3. Example of application of proposed method

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

To address microinstructions, we propose some modification of the approach [2]. It gives the addresses $A(b_1) = 00000,\ldots, A(b_{25}) = 110000$, $A(b_{26}) = 11100,\ldots, A(b_{28}) = 11110$, $A(b_{29}) = 11001,\ldots, A(b_{31}) = 11011$. It allows to get the following intervals for classes: 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.

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 Γ_1 and t=4 for PROM chips in use. In this case we can get R_3 = 1 and R_2 = 2. Thus, condition (7) is violated and block BAT should be used in CMCU $U_2(\Gamma_1)$. Let $\Pi_E = \{B_3\}$, then $\Pi_D = \{B_4\}$. Thus, the sets Π_A, Π_E and Π_D are constructed.

Obviously, there is $V = \{v_1\}, Z = \{z_1\}$, 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 control memory is constructed in a trivial way [2] and this step is here omitted. Let us point out, that the cells with addresses 10100 and 11000 include the variable v_1 = 1.

Let transitions for classes B_2, B_3, B_4 are described by the following system of generalized transition formulae [6]:

$$\begin{aligned} B_2 &\to x_3 b_9 \vee \overline{x_3} b_{26}; \\ B_3 &\to x_1 b_{18} \vee \overline{x_1} x_2 b_{20} \vee \overline{x_1}\,\overline{x_2} b_{26}; \\ B_4 &\to x_5 b_{27} \vee \overline{x_5} b_5. \end{aligned} \qquad (10)$$

The system (10) determines the fragment of transition table with 7 lines (Table 1).

Table 1.

Fragment of transition table for CMCU

B_i	$K_A(B_i)$	$K_E(B_i)$	$K_D(B_i)$	b_q	$A(b_q)$	X_h	Φ_h	h
	$T_1T_2T_3T_4T_5$	V_1	Z_1					
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 of this table and system (10) is a transparent one. Let us point out that the case $v_1 = z_1 = 0$ 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. This table is a base for construction of the corresponding system of Boolean functions. For example, the following parts of SOP can be obtained:

$$\begin{aligned} D_1 &= \overline{T_1}\,\overline{T_2}T_3\overline{v_1}\,\overline{z_1}\,\overline{x_3} \vee v_1\overline{z_1} \vee \overline{v_1}z_1x_5; \\ D_2 &= \overline{T_1}\,\overline{T_2}T_3\overline{v_1}\,\overline{z_1} \vee v_1\overline{z_1}\,\overline{x_1}\,\overline{x_2} \vee \overline{v_1}z_1x_5. \end{aligned}$$

The table of BAT is constructed for the classes $B_i \in \Pi_D$. It can be executed in a trivial way [2]. This table is the base to construct the system (14). In our case this system represented as the following one:

$$z_1 = T_1\overline{T_2}T_3\overline{T_4}\,\overline{T_5} \vee T_1T_2\overline{T_3}\,\overline{T_4}\,\overline{T_5}.$$

Synthesis of logic circuit of CMCU is reduced to implementation of systems (13)-(14) using PAL macrocells and control memory using PROM chips. These problems are well-known [2,6] and they are not discussed in our article.

4. Conclusion

The proposed method targets on decrease in hardware amount (the number of macrocells) in the circuit of 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 does 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 block BAT can be eliminated if discussed in the article conditions take places.

Our experiments show that the number of macrocells in block BAT is decreased up to 60-70% in comparison with known methods of CMCU design. The total decrease in hardware amount is up to 10% in comparison with CMCU with a base structure [2].

We think that the proposed method can be used when CMCU circuit is implemented using FPGA chips. In this case, our approach can reduce the number of look-up table elements in FPGA-based circuits interpreting the linear algorithms. These approaches can be used in the design of various hardware blocks providing electromagnetic compatibility in telecommunications.

References

1 A.Barkalov, L.Titarenko, M.Mazurkiewicz. Foundations of embedded systems. – Berlin: Springer, 2019.

2. A.Barkalov, L.Titarenko. Logic synthesis for compositional microprogram control units. - Berlin: Springer, 2008.

3. http://www.altera.com/products/devices/common/dev-family_overview.html.

4. Xilinx CPLDs http://www.xilinx.com/products/silicon_solutions/cplds/index.htm.

5. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с.

6. Baranov S. Logic Synthesis for Control Automata. – Kluwer Academic Publishers, 1994. – 312 pp.