МАШИНОСТРОЕНИЕ И ТЕХНОСФЕРА XXI ВЕКА

Сборник трудов XV международной научно-технической конференции

Том 4

2008 ДОНЕЦК

Министерство образования и науки Украины
Донецкая областная и городская администрации
Международный союз машиностроителей
Фонд поддержки прогрессивных реформ
Национальная металлургическая академия Украины (НИИСТ)
Донецкий и Севастопольский национальные технические университеты
Брянский государственный технический университет
Московский государственный университет инженерной экологии
Таганрогский технологический институт Южного федерального университета
Азербайджанский, Жешувский, Остравский, Силезский, Ясский
технические университеты, Политехника Любельская
Технический университет Молдовы, Политехника Ченстохова
Магдебургский, Портсмутский, Тульский университеты
Бухарестская военно-техническая академия
Институт международного сотрудничества, Российско-Украинский университет
Институт механики и сейсмологической стабильности АН РУ
Севастопольский центр профессионально-технического образования
Донецкий институт холодильной техники
Ассоциация металловедов и термистов Украины
Научно-технический союз машиностроения Болгарии
Научный центр проблем механики машин НАН Беларуси
Издательство «Машиностроение», ОАО НИИ «Изотерм», ОАО «ДЗГА»
АО «НОРД», ЗАО «НКМЗ», ЧП «Технополис», Снежнянский машзавод

МАШИНОСТРОЕНИЕ И ТЕХНОСФЕРА XXI ВЕКА

Сборник трудов

XV

МЕЖДУНАРОДНОЙ НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ

Том 4

15 - 20 сентября 2008 г. в городе Севастополе

Донецк-2008

ББК К5я54
УДК 621.01(06)

Машиностроение и техносфера XXI века // Сборник трудов XV международной научно-технической конференции в г. Севастополе 15-20 сентября 2008 г. В 4-х томах. – Донецк: ДонНТУ, 2008. Т. 4. – 338 с.

ISBN 966-7907-23-6

В сборник включены материалы XV международной научно-технической конференции «Машиностроение и техносфера XXI века», отражающие научные и практические результаты в области обработки изделий прогрессивными методами, создания нетрадиционных технологий и оборудования. Представлены современные достижения и перспективные направления развития технологических систем, металлорежущего инструмента и оснастки. Освещены современные проблемы материаловедения в машиностроении. Рассмотрены вопросы механизации и автоматизации производственных процессов, управления качеством и диагностики технических систем. Приведены сведения об особенностях моделирования, экономических проблемах производства, вопросах инженерного образования и других актуальных проблемах техносферы.

Предназначен для научно-технических работников, ИТР и специалистов в области машиностроения и техносферы.

Издается при содействии Международного союза машиностроителей

Адрес международного организационного комитета:

Украина, 83000, г. Донецк, ул. Артема, 58, ДонНТУ, кафедра ТМ.
Тел.: +38 (062) 305-01-04, факс: +38 (062) 305-01-04
E-mail: tm@mech.dgtu.donetsk.ua
http://www.dgtu.donetsk.ua

ISBN 966-7907-23-6

SYNTHESIS OF CONTROL UNIT WITH CODE SHARING AND MODIFIED LINEAR CHAINS

Barkalov A.A., Kovalyov S.A., Bieganowski J., Miroshkin A.N.
(DonNTU, Donetsk, Ukraine; University of Gelena Gora, Poland)

The method of design for compositional microprogram control units with code sharing is proposed. The method is oriented on reduction in the number of PAL macrocells in the combinational part of control unit. The method is based on modification of operational linear chains by some additional control microinstructions, which contain codes of the classes of pseudoequivalent chains. Proposed method is illustrated by an example.

1. Introduction

One of the most important blocks of any digital system is its control unit [1] responsible for interplaying of all other system blocks. If an interpreted control algorithm is a linear one, it can be interpreted using the model of compositional microprogram control unit (CMCU) [2]. Recently, the complex programmable logic devices (CPLD) [3, 4] are widely used for implementation of logic circuits [5, 6]. One of the important tasks connected with control unit design is minimization of hardware amount. In case of CPLD, this task can be solved due to decrease of the number of Programmable Array Logic (PAL) macrocells. To solve this problem, the number of terms in sum-of-products (SOP) should be diminished for address functions of CMCU [5, 6]. In this article one of the ways for this problem solution is proposed. The method targets on CMCU with code sharing [2].

2. Peculiarities of CMCU with code sharing

Let a control algorithm to be interpreted be represented by a graph-scheme of algorithm (GSA) Γ [7].Let this GSA be characterized by the set of vertices $B=\{b_0,b_E\}\cup E_1\cup E_2$ and the set of arcs E, where b_0 is an initial vertex, b_E is a final vertex, E_1 is a set of operator vertices, and E_2 is a set of conditional vertices. Each operator vertex $b_q\in E_1$ contains a collection of microoperations $Y(b_q)\subseteq Y$, where $Y=\{y_1,\dots,y_N\}$ is a set of data-path microoperations. Each conditional vertex $b_q\in E_2$ contains some element $x_l\in X$, where $X=\{x_1,\dots,x_L\}$ is a set of logical conditions (input signals). A GSA Γ is named a linear GSA [2] if the number of its operator vertices exceeds 75% of the total their number in GSA.

Let the set $C=\{\alpha_1,\dots,\alpha_G\}$ be constructed for GSA Γ, where $\alpha_g\in C$ is an operational linear chain (OLC) [2]. Any component b_{g_i} of OLC $\alpha_g\in C$ belongs to the set E_1 $(i=1,\dots,F_g)$. Each pair of adjacent components b_{g_i}, $b_{g_{i+1}}$ corresponds to the arc $<b_{gi},b_{gi+1}>\in E$, where $i=1,\dots,F_g-1$, $g=1,\dots,G$. Each OLC $\alpha_g\in C$ has only one output O_g and the arbitrary number of inputs. Formal definitions of OLC, its input and output can be found in [2]. Each vertex $b_q\in E_1$ corresponds to microinstruction MI_q kept in the cell of control memory (CM) with address A_q. It is enough

$$R=\lceil\log_2 M\rceil \qquad (1)$$

bits for microinstruction addressing, where $M=|E_1|$. Let each OLC $\alpha_g\in C$ include F_g components and $Q=\max(F_1,\dots,F_G)$. Let each OLC $\alpha_g\in C$ be encoded by binary code $K(\alpha_g)$ having

$$R_1=\lceil\log_2 G\rceil \qquad (2)$$

bits and variables $\tau_r \in \tau$ be used for such encoding, where $|\tau| = R_1$. Let each component $b_q \in B_1$ be encoded by binary code $K(b_q)$ having

$$R_2 = \lceil \log_2 Q \rceil \quad (3)$$

bits and variables $T_r \in T$ be used for this encoding, where $|T| = R_2$. Encoding of components is executed in such a manner that condition

$$K(b_{gi+1}) = K(b_{gi}) + 1 \quad (4)$$

takes place for each OLC $\alpha_g \in C$ $(i = 1, ..., F_g - 1)$. If condition

$$R_1 + R_2 = R \quad (5)$$

takes place, then the model of CMCU with code sharing U_1 can be used for interpretation of GSA Γ (Fig. 1).

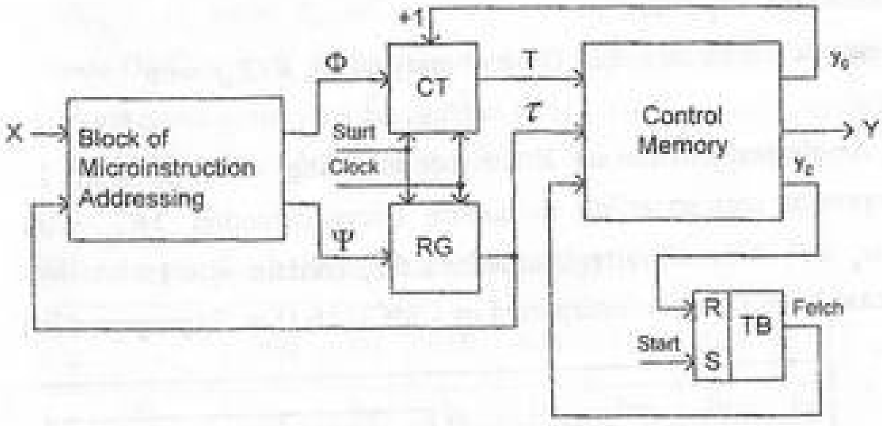

Fig. 1. Structure diagram of CMCU U_1

In CMCU U_1, a block of microinstruction addressing (BMA) implements the system of input memory functions for counter and register RG:

$$\Phi = \Phi(\tau, X),$$
$$\Psi = \Psi(\tau, X). \quad (6)$$

Let us point out that in the case of CMCU U_1 an address of microinstruction is represented as the following one:

$$A(b_q) = K(\alpha_g) * K(b_q), \quad (7)$$

where b_q is a component of OLC $\alpha_g \in C$ and "*" is a sign of concatenation. The CMCU U_1 operates in the following order.

If Start=1, then an initial address (all zeros) is loaded into RG and CT. In the same time a flip-flop TF is set up which causes Fetch=1, then microinstructions can be read out of control memory. Each cell of CM keeps microoperations $y_n \in Y$ and special variables y_0 and y_E. If $y_0 = 1$, then a current content of CT is incremented, otherwise both CT and RG are loaded from BMA. The first case corresponds to transition from any OLC component except of its output. The second case corresponds to transition from OLC output. If $y_E = 1$, then flip-flop TF is reset, signal Fetch=0 and operation of CMCU is terminated. It corresponds to transition from the vertex $b_q \in E_1$, where $< b_q, b_E > \in E$. Pulse Clock is used for timing of CMCU.

Let us point out that OLC $\alpha_i, \alpha_j \in C$ are pseudoequivalent OLC [2] if their outputs are connected with input of the same vertex of GSA Γ. The hardware amount in logic circuit of BMA can be decreased due to introduction of a special block for transforming the OLC codes into the codes of the classes of pseudoequivalent OLC (POLC) named as a code transformer (TC) [2]. But TC consumes some resources of the chip in use.

In this article we propose to use free cells of CM for such transformation. In results in decrease of hardware amount in both blocks BMA and TC without increase of the number of PROM chips in CM.

3. Main idea of proposed approach

Let $C_1 \subset C$ be a set of OLC such that their outputs are not connected with the vertex b_E. Let us find the partition $\Pi_C = \{B_1, ..., B_I\}$ of the set C_1 by the classes of POLC. Let condition

$$2^{K_1} > F_g \tag{8}$$

take place for each OLC $\alpha_g \in C_1$.

Let us encode the classes $B_i \in \Pi_C$ by binary codes $K(B_i)$ with

$$R_3 = \lceil \log_2 I \rceil \tag{9}$$

bits and let us use elements of the set Z for such encoding, where $|Z| = R_3$. Let us insert an additional component corresponding to control microinstruction MC_g with $y_0 = 0$ and $K(B_i)$, where $\alpha_g \in B_i$. Now all microinstructions MI_q contain $y_0 = 1$.

In this case GSA Γ can be interpreted by CMCU U_2 (Fig. 2) proposed in this article.

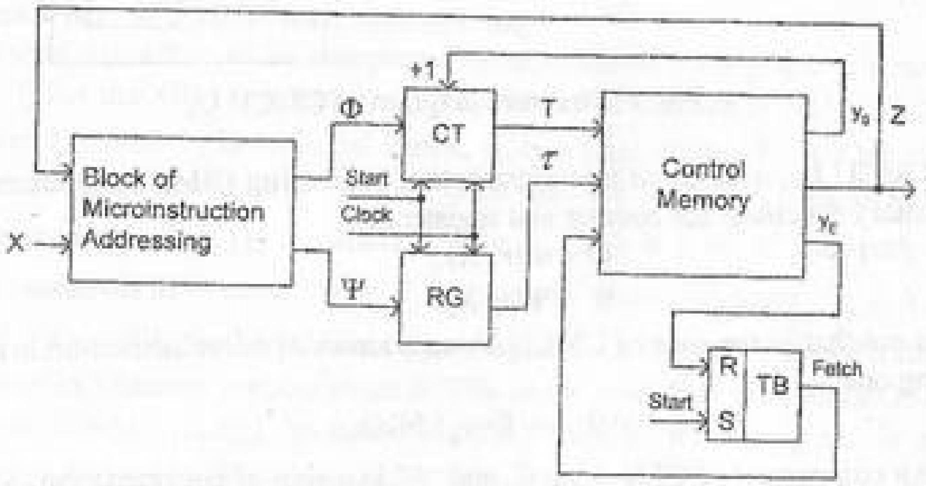

Fig. 2. Structural diagram of CMCU U_2.

In CMCU U_2, the block BMA implements functions

$$\Phi = \Phi(Z, X), \tag{10}$$

$$\Psi = \Psi(Z, X), \tag{11}$$

all other components of U_2 have the same meaning as their counterparts in U_1.

Functions (10)–(11) are generated if contents of RG and CT represent an address of control microinstruction. In this case a data-path of controlled system is in the idle state. It can be achieved, for example, if a data-path synchronization is stirred by variable y_0.

In this article we propose the following method of CMCU U_2 design:

1. Construction of the sets C, C_1, Π_C for GSA Γ.
2. Including of additional components into OLC $\alpha_g \in C_1$.
3. Encoding of OLC, their components and classes.
4. Construction of control memory content.

5. Construction of transition table of CMCU.
6. Implementation of CMCU logic circuit.

4. Example of proposed method application

Let GSA Γ_1 be characterized by sets $C=\{\alpha_1,...,\alpha_7\}$, $C_1=\{\alpha_1,...,\alpha_6\}$, $\Pi_C=\{B_1,B_2,B_3\}$, where $B_1=\{\alpha_1\}$, $B_2=\{\alpha_2,\alpha_3\}$, $B_3=\{\alpha_4,\alpha_5,\alpha_6\}$, $\alpha_1=<b_1,b_2,b_3>$, $\alpha_2=<b_4,b_5>$, $\alpha_3=<b_6,b_7,b_8>$, $\alpha_4=<b_9,b_{10},b_{11}>$, $\alpha_5=<b_{12},b_{13},b_{14}>$, $\alpha_6=<b_{15},b_{16}>$, $\alpha_7=<b_{17},b_{18},b_{19},b_{20}>$. It means that G=7, R_1=3, $\tau=\{\tau_1,\tau_2,\tau_3\}$, Q=4, R_2=2, $T=\{T_1,T_2\}$, M=20, R=5 and conditions (5) and (8) take places. Therefore, application of proposed method has sense. Let us point out that R_3=2, $Z=\{z_1,z_2\}$.

After introducing of additional components into OLC $\alpha_g \in C_1$ we have: $\alpha_1=<b_1,b_2,b_3,MC_1>$, $\alpha_2=<b_4,b_5,MC_2>$, $\alpha_3=<b_6,b_7,b_8,MC_3>$, $\alpha_4=<b_9,b_{10},b_{11},MC_4>$, $\alpha_5=<b_{12},b_{13},b_{14},MC_5>$, $\alpha_6=<b_{15},b_{16},MC_6>$.

Let us encode OLC $\alpha_g \in C$ in arbitrary manner, namely: K(α_1) = 000, ..., K(α_7) = 110. Let code 00 is assigned to the first component of any OLC $\alpha_g \in C_1$, code 01 to the second, code 10 to the third, and code 11 to the fourth. Now microinstruction addresses are shown in Table 1.

Table 1. Microinstruction addresses for CMCU $U_2(\Gamma_1)$

T_1T_2 \ $\tau_1\tau_2\tau_3$	000	001	010	011	100	101	110
00	b_1	b_4	b_6	b_9	b_{12}	b_{15}	b_{17}
01	b_2	b_5	b_7	b_{10}	b_{13}	b_{16}	b_{18}
10	b_3	MC_2	b_8	b_{11}	b_{14}	MC_6	b_{19}
11	MC_1	*	MC_3	MC_4	MC_5	*	b_{20}

The symbol $U_i(\Gamma_j)$ stands for the case when GSA Γ_j is interpreted by CMCU U_i. We can derive from Table 1, for example, A(b_1)=00000, A(b_8)=01010, A(MC_2)=00110.

Let us encode classes $B_i \in \Pi_C$ by the following codes: K(B_1)=00, K(B_2)=01, K(B_3)=10. Let microoperations $y_n \in Y$ are distributed among the operator vertices in the following manner:
$Y(b_1)=Y(b_5)=\{y_1,y_2\}$, $Y(b_2)=Y(b_6)=Y(b_{14})=\{y_3\}$, $Y(b_3)=Y(b_7)=Y(b_{12})=\{y_1,y_4\}$, $Y(b_4)=Y(b_9)=Y(b_{13})=\{y_2,y_3\}$, $Y(b_8)=Y(b_{15})=Y(b_{17})=\{y_5\}$, $Y(b_{10})=Y(b_{16})=Y(b_{19})=\{y_4\}$, $Y(b_{11})=Y(b_{18})=\{y_1,y_5\}$, $Y(b_{20})=\{y_6\}$.

In this case the control memory content for CMCU $U_2(\Gamma_1)$ is shown in Table 2.

Table 2. Content of control memory for CMCU $U_2(\Gamma_1)$

T_1T_2 \ $\tau_1\tau_2\tau_3$	000	001	010	011	100	101	110
00	$y_0 y_1 y_2$	$y_0 y_2 y_3$	$y_0 y_3$	$y_0 y_2 y_3$	$y_0 y_1 y_4$	$y_0 y_5$	$y_0 y_5$
01	$y_0 y_3$	$y_0 y_1 y_2$	$y_0 y_1 y_4$	$y_0 y_4$	$y_0 y_2 y_3$	$y_0 y_4$	$y_0 y_1 y_5$
10	$y_0 y_1 y_4$	z_2	$y_0 y_5$	$y_0 y_1 y_5$	$y_0 y_3$	z_1	$y_0 y_4$
11	–	*	z_2	z_1	z_1	*	$y_E y_6$

Let us point out that transition from Table 2 to control memory implementation is a straight – forward one.

Let transition from OLC outputs of GSA Γ_1 are represented by the following system of generalized transition formulae [2]:

$$\begin{aligned} &B_1 \rightarrow x_1 b_4 \vee \bar{x}_1 x_2 b_6 \vee \bar{x}_1 \bar{x}_2 b_{11}; \\ &B_2 \rightarrow x_2 x_3 b_9 \vee x_2 \bar{x}_3 b_{13} \vee \bar{x}_2 x_4 b_{12} \vee \bar{x}_2 \bar{x}_4 b_{15}; \\ &B_3 \rightarrow x_4 b_{17} \vee \bar{x}_4 x_5 b_{11} \vee \bar{x}_4 \bar{x}_5 b_{20}. \end{aligned} \quad (12)$$

Such a system is the base to construct the transition table of CMCU U_2 with the following columns: $B_i, K(B_i), b_q, A(b_q), X_h, \Psi_h, \Phi_h, h$. Here X_h is a conjunction of some elements $x_l \in X$, determining the transition from $B_i \in \Pi_C$ into microinstruction MI_q; Ψ_h is a set of input memory functions to form the code $K(\alpha_g)$ into RG, where $\alpha_g \in B_i$; Φ_h is a set of input memory functions to form the code $K(b_q)$ into CT; h is the number of transition, where $h = 1, ..., H_2(\Gamma_j)$. The subscript 2 underlines that we deal with CMCU U_2. The number $H_2(\Gamma_j)$ is equal to the number of terms in the system of the type (12). In our case, $H_2(\Gamma_1) = 10$ and some part of the table is shown in Table 3.

Table 3. Fragment of transition table for CMCU $U_2(\Gamma_1)$.

B_i	$K(B_i)$	b_q	$A(b_q)$	X_h	Ψ_h	Φ_h	h
B_3	10	b_{17}	11000	x_4	$D_1 D_2$	-	1
		b_{11}	01110	$\bar{x}_4 x_5$	$D_2 D_3$	D_4	2
		b_{20}	11011	$\bar{x}_4 \bar{x}_5$	$D_1 D_2$	$D_4 D_5$	3

The connections between Table 3 and system (12) as well as with Table 1 are obvious. After minimization we can get the parts of system (10) and (11) from Table 3:

$$\begin{aligned} &D_1 = z_1 \bar{z}_2 x_4 \vee z_1 \bar{z}_2 \bar{x}_4 \bar{x}_5; \\ &D_2 = z_1 \bar{z}_2; \qquad D_3 = z_1 \bar{z}_2 \bar{x}_4 x_5; \\ &D_4 = z_1 \bar{z}_2 \bar{x}_4; \qquad D_5 = z_1 \bar{z}_2 \bar{x}_4 \bar{x}_5. \end{aligned} \quad (13)$$

Implementation of logic circuit of CMCU $U_2(\Gamma_j)$ is reduced to implementation of systems (10) – (11) using PAL macrocells and implementation of control memory using PROM chips. In our case Table 2 is used to implement the control memory.

Let us point out that for GSA Γ_1 we have $H_1(\Gamma_1)=20$, and $\eta = H_1(\Gamma_1)/H_2(\Gamma_1) = 2$. As some experiments show [2], the number of terms in systems (10) – (11) is η times less than its number for system (6). Obviously, the number of PAL macrocells in logic circuit of block BMA can be found if we know the number of terms per cell. But the ratio of the numbers of macrocells is approximately equal to η [8].

5. Conclusions

The proposed method of modification of OLC targets on decrease in hardware amount (the number of PAL macrocells) in the block of microinstructions addressing of CMCU with code sharing. This optimization does not increase the number of PROM chips used for implementation of the control memory of CMCU.

The method is based on encoding of the classes of pseudoequivalent OLC permitting decrease of the transition table lines in comparison with equivalent CMCU U_1 without modification of OLC.

The drawback of this method is increase in the number of cycles needed for execution of control algorithm in comparison with CMCU U_1. But decrease in the macrocell number can lead to decrease in the number of layers in combinational part of CMCU. It results in decrease of the cycle time. Thus, the final conclusion about algorithm execution time should be made after implementation of logic circuits for $U_1(\Gamma_j)$ as well as for $U_2(\Gamma_j)$. Our experiments show that the number of macrocells is decreased up to 30% and the number of layers is decreased up to 3. Of course, application of proposed method is possible only for interpretation of linear GSA when condition (8) takes place.

The next steps in this research are development of CAD tools for CMCU design and exploration of possibility for given method application in case of FPGA [9].

References: 1. De Micheli G. Synthesis and Optimization of Digital Circuits. – NY: McGraw-Hill, 1994. – 636 pp. 2. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах: Донецк ДНТУ, 2002 – 262 с. 3. MAX II CPLD Family: The Lowest-Power, Lowest-Cost CPLDs: http://www.altera.com/products/devices/cpld/ max2/mx2-index.jsp. 4. CoolRunner-II 1.8 Volt CPLD: http://www.xilinx.com/products/silicon_solutions/cplds/coolrunner_series/coolrunner_ii_cpld s/overview.htm. 5. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем с использованием микросхем программируемой логики. – СПб: БХВ. – Петербург, 2002. – 608 с. 6. Соловьев В.В. Проектирование цифровых схем на основе программируемых логических интегральных схем. – М.: Горячая линия-ТЕЛЕКОМ, 2001. – 636 с. 7. Baranov S. Logic Synthesis for Control Automata – Boston: Kluwer Academic Publishers, 1994 – 312 pp. 8. Баркалов А.А, Бабаков Р.М, Ахмад Бадер. Исследование аппаратурных характеристик автомата Мили с кодированием фрагмента микроопераций по VHDL- моделям // Исскуственный интеллект. – 2007, №1, - С. 117-122. 9. Maxfield C. The Design Warriors Guide for FPGA – Amsterdam : Elseteir, 2004. –541 pp.

OPTIMIZATION OF CONTROL UNIT BASED ON PECULIARITIES OF CPLD

Barkalov A.A., Zelenyova I.Y., Kovalyov S.A., Lavrik A.S.
(University of Zielona Gora, DonNTU, Zielona Gora, Donetsk, Poland, Ukraine)

The method of hardware reduction is proposed which is oriented on compositional microprogram control units and CPLD chips. The method is based on a wide fan-in of PAL macrocells allowing using more than one source of microinstruction address. Such approach permits to minimize the number of PAL macrocells used for transformation of microinstruction address. Conditions for this method application and example of its application are given.

1. Introduction

A control unit is one of the very important parts of any digital system [1]. 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) [2]. The programmable logic devices with programmable array logic (PAL) macrocells are widely used for implementation of logic cir-