UDC 004.3

A.A. Barkalov[1], I.J. Zelenyova[2], Hathot Biayrek[2], A.S. Lavrik
[1]University of Zielona Gora, Poland
A.Barkalov@iie.uz.zgora.p1
[2]Donetsk National Technical University, Donetsk, Ukraine
allah_karem90@yahoo.com

Implementation of logic circuit of compositional microprogramming control unit with elementary operational linearchains on custom-made matrices

The modified method of compositional microprogramming control unit realization on custom matrix is proposed. Analysis of proposed method and results of modeling for flow-charts of control algorithm are given.

***Key words**: custom matrix, compositional microprogram control unit, elementarization, operational linear chain.*

Introduction

The properties of the interpreted control algorithm have great influence on the hardware amount of corresponding control units. One of such properties is the existence of operational linear chains (OLC) corresponding to the graph-scheme of algorithm (GSA) paths, which include operator vertices only [1]. There are many of the approaches for linear graph-scheme of algorithm interpretation and one of them is the use of compositional microprogram control units (CMCU), which can be viewed as a composition of the finite state machine and microprogram control unit [2]. In this case control unit can be implemented as composition of automaton with “hardwired” logic, realizing addressing of transitions between OLCS, and automaton with “programmable” logic. The given article explains the method of CMCU synthesis with optimization of CMCU with elementary OLC on custom-made matrix.

The purpose of the research is realization of logic circuit of control unit on custom-made VLSI as interpretation of linear control algorithm.

The task of the research is development of method to optimization for CMCU with microinstructions elementary OLC that allows reducing the area of custom-made matrix for its logic circuit. The control algorithm is represented in the form of GSA.

Basic definitions and general provisions

If all OLC $\alpha_g \in$ C have only one input , then an address of OLC input will be equal to “0”.So, we don’t need to write code into counter .Operational linear chain with one input is called elementary OLC. Transformation of microinstruction addresses is oriented at the reduction of hardware amount in CMCU logic circuit. In this method we will construct the set of elementary OLC and encoding the elementary OLCS and the components, later we will use the transformation of microinstruction addresses of operational linear chains outputs into codes of the classes of pseudoequivalent OLC.

Let the set of inputs be $B_1 = \{b_0,..,b_{M-1}\}$, and $A(b_q)$ be a binary address of microinstructions corresponding $b_q \in B_1$ includes $R_{B_1} = \lceil \log_2 M \rceil$, $M = |B_1|$, where M is the total number of microinstructions kept in control memory, and R_{B_1} lets for their addressing. And set of conditional vertices B_2. Vertices $b_q \in B_1$ contains microoperations $Y(b_q) \subseteq Y$, where $Y = \{y_1,....,y_N\}$ is the set of microoperations. In the vertices $b_q \in B_2$ elements from the set of logic conditions $X = \{x_1,....,x_L\}$ are contained. Vertices GSA are connected by arches $\langle b_t, b_q \rangle$, forming the classes set of pseudoequivalent for elementary OLC $E = \{E_1,...,E_d\}$. Any OLC α_g can has more than one input, let’s designate j-й input OLC α_g like I_g, every OLC has only one output O, entering in the set $O = \{O_1,...,O_G\}$ of OLC outputs.

$C = \{\alpha_1,..,\alpha_G\}$ is a set of operator vertices of initial graph-scheme of algorithm GSA with minimal possible G with out elementary, $C_E = \{\alpha_1,..,\alpha_{G_E}\}$ is a set of elementary OLC of GSA, including G_E elements.

Each OLC $\alpha_g \in C$ corresponds to a binary code $K(\alpha_g)$ with the following number of bits:

$$R_\alpha = \lceil \log_2 G \rceil, \qquad (1)$$

where $G = |C|$.

Let L_g be the number of components in OLC $\alpha_g \in C$. We find the following relation:

$$L_{\max} = \max(L_1, ..., L_G). \qquad (2)$$

Let us encode each EOLC $\alpha_g \in C_E$ by a binary code $K(\alpha_g)$ with

$$R^e_\alpha = |\log_2 G_E| \qquad (3)$$

bits and use variables $\tau_r \in \tau$ for this encoding, where $|\tau| = R^e_\alpha$. Let components $b_q \in D^g$ of OLC $\alpha_g \in C$ be encoded by binary codes $K(b_q)$ with the following number of bits:

$$R^e_g = \lceil \log_2 L\max \rceil. \qquad (4)$$

Let some variables $T_r \in T$, where $|T| = R^e_g$, be used for this encoding. Let us encode the components in such a way that any pair of adjacent components $b_t, b_q \in D^g$ of OLC $\alpha_g \in C$, where $\langle b_t, b_q \rangle \in E$, satisfies the condition:

$$K(b_q) = K(b_t) + 1. \qquad (5)$$

In this case an address $A(b_q)$ of microinstruction, corresponding to vertex $b_q \in D^g$ *(g=1,….., G_E)*, can be represented by a concatenation of the OLC codes $\alpha_g \in C$ and by its component $K(b_q)$:

$$A(b_q) = K(\alpha_g) * K(b_q), \qquad (6)$$

Where * is a sign concatenation. CMCU implemented on FC with elementary OLCs or with EOLC, we denote by symbol U_E in (Fig. 1.), where CMCU is built using principle of codes division with optimal coding of EOLC, in fig.1. structural diagram of *CMCU* U_E is shown and represents basic of the CMCU with EOC.

On a Start signal in RG and CT zero codes are written; on signal y_0 CT:=CT+1, addressing the next component of OLC; if y_0 = 0 CT:=0 ,which corresponds to input of the next OLC.

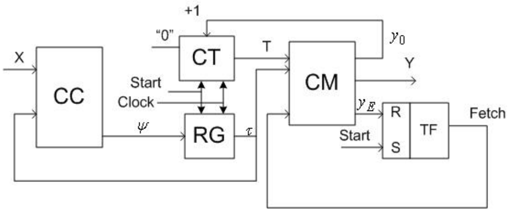


Figure 1. Structure of CMCU with elementary EOLC.

In the *CMCU* U_E, combinational circuit CC implements the following input memory functions for counter CT

$$\phi = \phi(\tau, X) \qquad (7)$$

and input memory functions for register RG

$$\psi = \psi(\tau, X). \qquad (8)$$

Control memory CM of CMCU U_E implements the system of output functions

$$Y = Y(\tau, Y) \qquad (9)$$

$$y_0 = y_0(\tau, T); \qquad (10)$$

$$y_E = y_E(\tau, T).$$

Method of synthesis of CMCU U_E includes the following stages:

1) Addition of y_0, y_E to transformation of initial GSA.

2) Construction of the set elementary OLC C_E For GSA U_E.

3) Encoding of elementary OLC $\alpha_g \in C_E$ and the components for elementary OLC $\alpha_g \in C_E$.

4) Construction of the on Classes of pseudoequivalent for elementary OLC $E = \{E_1, E_2, E_3, E_4\}$.

5) Optimal encoding using karnaugh map apply on Classes of pseudoequivalent for elementary OLC $E = \{E_1, E_2, E_3, E_4\}$.

6) Construction of the control memory content.

7) Construction of transition table U_E.

8) Synthesis of logical circuit of CMCU U_E.

Matrix realization of CMCU with elementary operational linear chains on custom-made matrix

For implementation of the CMCU scheme on custom-made matrices we proposed to represent the CC circuit in the form of conjunctive matrix M_1 and disjunctive matrix M_2, CM is presented as matrix M_3

and disjunctive matrix M_4 while the converter device is introduced as M_5 matrix. (Fig. 2):

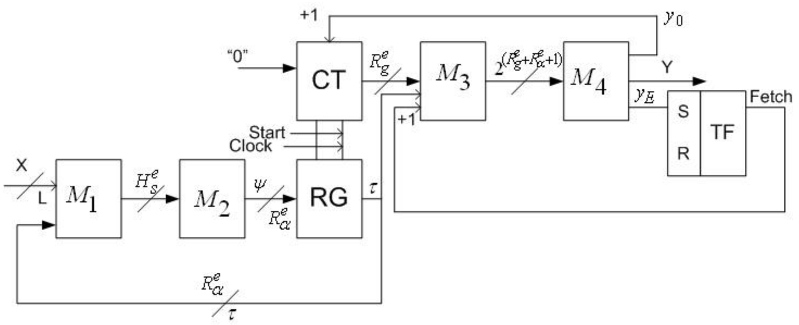


Figure 2 —Matrix realization of CMCU U_E

.

From Fig. 2 we can calculate S_{CMCU}^e , where S_{CMCU}^e is the area of CMCU U_E :

$$S_{CMCU}^e = S_{M_1} + S_{M_2} + S_{M_3} + S_{M_4} \qquad (11)$$

$$S_{M_1} = 2*(L + R_\alpha^e)*H_S^e \qquad (12)$$

$$S_{M_2} = H_S^e * R_\alpha^e \qquad (13)$$

$$S_{M_3} = 2*(R_\alpha^e + R_g^e)*2^{(R_g^e + R_\alpha^e)} \qquad (14)$$

$$S_{M_4} = 2^{(R_g^e + R_\alpha^e)}*(N+2) \qquad (15)$$

Example of application of CMCU with elementary operational linear chains on custom-made matrix

Let us take the example for explaining the idea of our method by applying the stages of synthesis of CMCU U_E (Fig 3). In the first stage we should add $y_0 : CT = CT + 1$, y_E : End of reading ROM, and determine the set of microoperations $Y = \{y_1,....,y8\}$, this is meaning the number of microoperations N=8. And the set of logic conditions $X = \{x_1, x_2\}$, L=2, where L is the number of logic conditions. And the set of microinstructions $B_1 = \{b_0,..,b_{17}\}$, M=18, this is leading $R_{B_1} = \lceil \log_2 18 \rceil = 5$. We will notice the number of OLC with out elementary G=4, $R_{\alpha=2}$ but the number of OLC with elementary G_E =5, $R_\alpha^e = 3$, $\tau_r = \{\tau_1, \tau_2, \tau_3\}$. $L_{\max} = 4$, $R_g^e = \lceil \log_2 4 \rceil$, $R_g^e = 2$, $T_r = \{T_1, T_2\}$. When R_B, R_α^e, R_g^e will be known, we should give attention to R_B must be equal to $R_\alpha^e + R_g^e$.

On second stage we will construct the set of elementary OLC C_E for GSA U_E, $C_E = \{\alpha_1^e, \alpha_2^e, \alpha_3^e, \alpha_4^e, \alpha_5^e\}$.

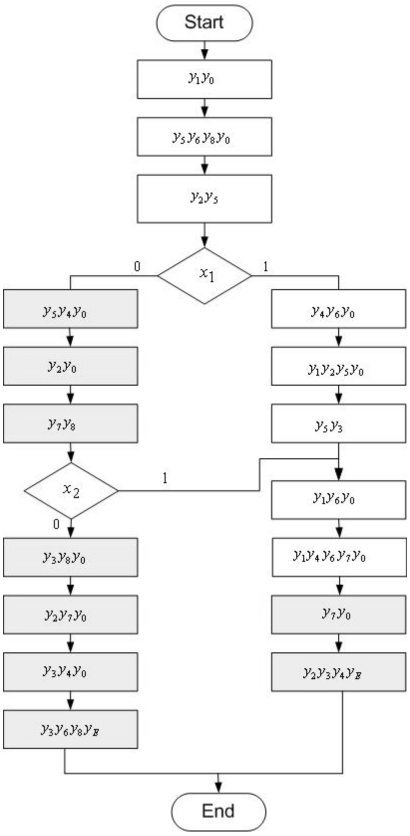


Figure 3 – Initial GSA for CMCU U_E

IOn third stage we should encode the elementary of OLC $\alpha_g \in C_E$ and the components for elementary OLC $\alpha_g \in C_E$ and it is show in in table 1.

Table 1. Microinstruction addresses for CMCU U_E

$\tau_1\tau_2\tau_3$ \ T_1T_2	α_1^e 000	α_2^e **001**	α_3^e **010**	α_4^e **011**	α_5^e **100**	α_6^e **101**	α_7^e **110**	α_8^e 111
00	b_1	b_4	b_7	b_{10}	b_{14}	*	*	*
01	b_2	b_5	b_8	b_{11}	b_{15}	*	*	*
10	b_3	b_6	b_9	b_{12}	b_{16}	*	*	*
11	*	*	*	b_{13}	b_{17}	*	*	*

On fourth stage we will Construct the Classes of pseudoequivalent for elementary OLC: $E=\{E_1,E_2,E_3,E_4\}$, $E_1=\{\alpha_1\}$, $E_1=\{\alpha_2\}$, $E_3=\{\alpha_4\}$, $E_1=\{\alpha_3,\alpha_5\}$.

And on fifth stage we will apply the optimal encoding using the karnough map on Classes of pseudoequivalent elementary OLC $E=\{E_1,E_2,E_3,E_4\}$ (Table 2).

Table 2.optimal encoding for Classes of pseudoequivalent.

$\tau_2\tau_3$ \ τ_1	00	01	11	10
0	α_1^e	α_2^e	α_3^e	α_5^e
1	*	α_4^e	*	*

$K(E_1)=\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3$, $K(E_2)=\bar{\tau}_1\bar{\tau}_2\tau_3$, $K(E_3)=\tau_1\bar{\tau}_2\tau_3$, $K(E_4)=*\tau_2*$.

On sixth stage we should build the control memory content (Table 3.)

With the table we can build system function, for example:

$$D_1 = F_3 \vee F_5 = \bar{\tau}_1\bar{\tau}_2\tau_3 x_2 \vee \tau_1\bar{\tau}_2\tau_3 . \qquad (16)$$

In seventh stage we should build transition table U_E (Table 4.)

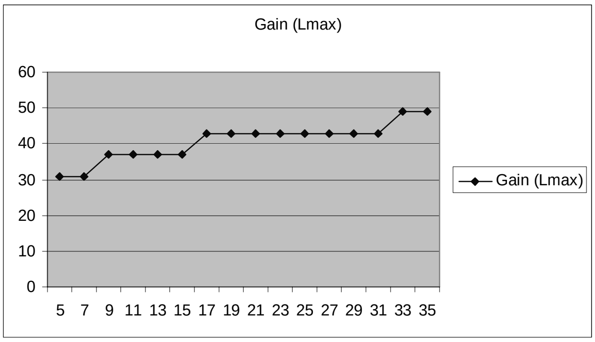
Table 3. Control memory content CMCU U_E.

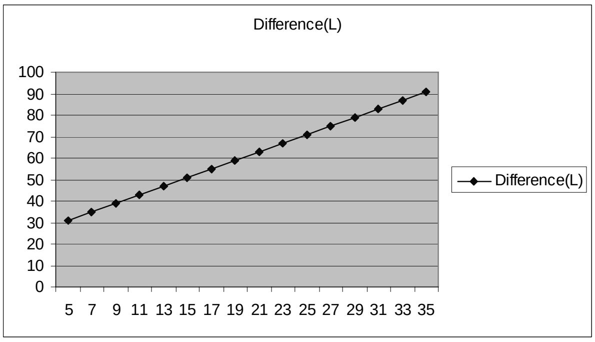
Address	Content										B_I	Π_C	I_g^i	O_g
$\tau_1\tau_2\tau_3T_1T_2$	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_E	B_I	C_E	I_g^i	O_g
00000	1	1	0	0	0	0	0	0	0	0	b_1	α_1^e	I_1	-
00001	1	0	0	0	0	1	1	0	1	0	b_2	α_1^e	-	-
00010	0	0	1	0	0	1	0	0	0	0	b_3	α_1^e	-	O_1
00011	0	0	0	0	0	0	0	0	0	0	-	-	-	-
00100	1	0	0	0	1	1	0	0	0	0	b_4	$\alpha_2{}^e$	I_2	-
00101	1	0	1	0	0	0	0	0	0	0	b_5	$\alpha_2{}^e$	-	-
00110	0	0	0	0	0	0	0	1	1	0	b_6	$\alpha_2{}^e$	-	O_2
00111	0	0	0	0	0	0	0	0	0	0	-	-	-	-
01000	1	0	0	1	0	0	0	0	1	0	b_{10}	$\alpha_4{}^e$	I_3	-
01001	1	0	1	0	0	0	0	1	0	0	b_{11}	$\alpha_4{}^e$	-	-
01010	1	0	0	1	1	0	0	0	0	0	b_{12}	$\alpha_4{}^e$	-	-
01011	0	0	0	1	0	0	1	0	1	1	b_{13}	$\alpha_4{}^e$	-	O_3
01100	1	0	0	0	1	0	1	0	0	0	b_7	$\alpha_3{}^e$	I_4	-
01101	1	1	1	0	0	1	0	0	0	0	b_8	$\alpha_3{}^e$	-	-
01110	0	0	1	1	0	0	0	0	0	0	b_9	$\alpha_3{}^e$	-	O_4
01111	0	0	0	0	0	0	0	0	0	0	-	-	-	-
10000	1	1	0	0	0	0	1	0	0	0	b_{14}	$\alpha_5{}^e$	I_5	-
10001	1	1	0	0	1	0	1	1	0	0	b_{15}	$\alpha_5{}^e$	-	-
10010	1	0	0	0	0	0	0	1	0	0	b_{16}	$\alpha_5{}^e$	-	-
10011	0	0	1	1	1	0	0	0	0	1	b_{17}	$\alpha_5{}^e$	-	O_5

Table 4. Transitions for CMCU U_E with optimal encoding.

E_m	$K(E_m)$	I_g	b_s	$K(b_s)$		X_h	ϕ_h	H_s^e
				$\tau_1\tau_2\tau_3$	T_1T_2			
E_1	$\bar{\tau}_1\bar{\tau}_2\bar{\tau}_3$	I_4	b_7	011	00	X_1	D_2D_3	1
		I_2	b_4	001	00	$\bar{X}_1$	D_3	2
E_2	$\bar{\tau}_1\bar{\tau}_2\tau_3$	I_5	b_{14}	100	00	X_2	D_1	3
		I_3	b_{10}	010	00	$\bar{X}_2$	D_2	4
E_3	$\tau_1\bar{\tau}_2\tau_3$	I_5	b_{14}	100	00	1	D_1	5

Results of the modeling





Conclusion

Our suggested method aims to markdown in hardware amount, used area of the custom-made matrices. The method is based on optimal encoding of the classes of pseudoequivalent OLC allows to reduce of the transition table lines in comparison with CMCU with base structure. Our method will be effective for any GSA if it has optimal encoding in of pseudoequivalent for elementary OLC, where number of α_g^e should be $2^{n-1} < G_E << 2^n$, we should notice the initial GSA has $R_B = R_g^e + R_\alpha^e$, $R_\alpha^e = R_a$, and if it is possible to get on minimal amount of OLC code.

References

1. Barkalov A.A., Titarenko L.A. Synthesis of Operational and Control Automata. – Donetsk: Unitech, 2005. – 256 pp.
2. Barkalov A.A., Titarenko L.A. Logic Synthesis for Compositional Microprogram Control Units Berlin Springer,2008. – 272 pp.
3. Smith M. Application – Specific Integrated Circuits. – Boston: Addison Wesley, 1997. – 836 pp.
4. Baranov S. Logic Synthesis For Control Automata. – Boston: Kluwer Academic Publisher, 1954. – 312 pp.

Надійшла до редакції 30.01.2011