Alexander BARKALOV, Larysa TITARENKO, Olena HEBDA
UNIVERSITY OF ZIELONA GÓRA,
Licealna str. 9, 65-417 Zielona Gora

Synthesis of Moore finite state machine with transformation of extended state codes

Prof. dr hab. inż. Alexander BARKALOV

Prof. Alexander A. Barkalov worked in Donetsk National Technical University (DNTU) from 1976 till 1996 as a tutor . He cooperated actively with Kiev Institute of Cybernetics (IC) named after Victor Glushkov. He got his degree of doctor of technical sciences (Informatics) in 1995 from IC. From 1996 till 2003 he worked as a professor of DNTU. From 2003 he has been working as a professor on Department of Electrotechnics, Informatics and Telecommunications of University of Zielona Gora.

e-mail: A.Barkalov@iie.uz.zgora.pl

Dr hab. inż. Larysa TITARENKO

Prof. Larysa Titarenko has got her degree of doctor of technical sciences (Telecommunications) in 2005 from Kharkov National University of Radioelectronics (KNURE). Till September, 2003 she worked as a professor of KNURE. From 2005 she has been working as a professor on Department of Electrotechnics, Informatics and Telecommunications of University of Zielona Gora.

e-mail: L.Titarenko@iie.uz.zgora.pl

MA Olena Hebda

MA Olena Hebda studied at National aerospace university 'Kharkiv Aviation Institute' from 2003 till 2009 and has received higher education on speciality "Biotechnical and Medical Apparatuses and Systems" and qualification of the research engineer (electronics, telecommunications). In 2009 she has started PhD study on Department of Electro-technics, Informatics and Telecommunications of University of Zielona Gora.

e-mail: O.Shapoval@weit.uz.zgora.pl

Abstract

The method for reduction of the number of programmable array logic macrocells in a microprogrammed Moore finite state machine circuit is proposed. It is based on representation of the state code as a concatenation of a code for the class of pseudoequivalent states and a code of states inside this class. Such an approach allows eliminating the dependence between states and microoperations. The special code converter is used for formation of microoperations. As a result, both circuits for generation of input memory functions and microoperations are optimized. An example of the proposed method application is given.

Keywords: Moore FSM, CPLD, pseudoequivalent states, classes, PAL macrocells, hardware reduction.

Synteza skończonego automatu stanu typu Moore'a z transformacją rozszerzonej przestrzeni kodowej

Streszczenie

Model skończonego automatu stanu typu Moore'a jest często stosowany w jednostkach sterujących [1]. Postęp w technologii półprzewodnikowej powoduje pojawienie się coraz bardziej złożonych układów cyfrowych, takich jak złożone programowalne układy cyfrowe, gdzie funkcje logiczne są implementowane przy użyciu programowalnych bloków logicznych (ang. Programmable Array Logic, PAL). Jedną z osobliwości PAL jest ograniczona ilość termów [6, 7]. Dla tego jest potrzebna minimalizacja realizowanych funkcji. Układ automatu Moore'a składa się z bloku funkcji wzbudzenia pamięci (BFWP) i bloku mikrooperacji (BMO) (rys. 1), które są implementowane przy użyciu makrokomórek PAL. Znane metody syntezy automatu Moore'a mogą optymalizować tylko jeden z bloków. W artykule proponowana jest metoda zorientowana na redukcję ilości makrokomórek PAL potrzebnych do implementacji skończonego automatu stanu typu Moore'a. Ta metoda bazuje na przedstawieniu kodu stanu jako konkatenacji kodu klasy stanów pseudo-równoważnych i kodu stanów wewnątrz tej klasy. Takie podejście pozwala usunąć zależność między stanami oraz mikrooperacjami. Dla formowania mikrooperacji został użyty specjalny przetwornik kodów (rys. 2). Zaproponowane podejście pozwala zoptymalizować blok wejściowych funkcji pamięci i blok mikrooperacji . Artykuł przedstawia także przykład zastosowania proponowanej metody.

Słowa kluczowe: automat typu Moore'a, CPLD, stany pseudorównoważne, makrokomórka PAL.

1. Introduction

The Moore finite state machine (FSM) model [1] is often used during design of digital control systems [2, 3]. The development of microelectronics has led to appearance of different programmable logic devices [4], one of which is CPLD (complex programmable logic devices) [4, 5]. The basis of CPLD is a macrocell PAL (programmable array logic). The cells are connected by the programmable array of interconnections.

One of PAL singularities is a limited number of terms that necessitates minimization of implemented functions [6, 7]. The Moore FSM logic circuit consists of block of input memory functions and a block of microoperations that are implemented on PAL. The known methods of synthesis of Moore FSM allow optimizing only one of FSM blocks. In this work an optimization method is proposed. It is based on introduction of an additional code converter (CC) and nonstandard representation of the Moore FSM state code.

2. The general aspects and the basic idea of a proposed method

Let Moore FSM be represented by the structure table (ST) with columns [1]: a_m, $K(a_m)$, a_s, $K(a_s)$, X_h, Φ_h, h. Here a_m is an initial state of FSM; $K(a_m)$ is a code of state $\mathrm{a_m} \in \mathrm{A}$ of capacity $R = \lceil \log_2 M \rceil$, to code the states the internal variables from the set $\mathrm{T} = \{\mathrm{T_1},..,\mathrm{T_R}\}$ are used; a_s, $K(a_s)$ are a state of transition and its code respectively; X_h is an input, which determines the transition $\langle a_m, a_s \rangle$, and is equal to conjunction of some elements (or their complements) of a set of logic conditions $\mathrm{X} = \{\mathrm{x_1},..,\mathrm{x_L}\}$; Φ_h is a set of input memory functions for flip-flops of FSM memory, which are equal to 1 for memory switching from $K(a_m)$ to $K(a_s)$, $\Phi_\mathrm{h} \subseteq \Phi = \{\varphi_1,..,\varphi_\mathrm{R}\}$; $h = 1, \ldots, H$ is the number of transitions. In the column a_m a set of microoperations Y_q is written. It is generated in the state $\mathrm{a_m} \in \mathrm{A}$, where $Y_q \subseteq Y = \{y_1,..,\mathrm{y_N}\}$, $q = 1,...,Q$. This table is a basis for forming the system of functions

$$\Phi = \Phi(T, X), \tag{1}$$

$$Y = Y(T), \tag{2}$$

which determines an FSM logic circuit. Systems (1)-(2) describe the model of Moore FSM U_1, shown in Fig.1. In this model a block of input memory functions (BIM) realizes the system (1), a register RG keeps states' codes, block of microoperations (BMO) realizes the system (2).

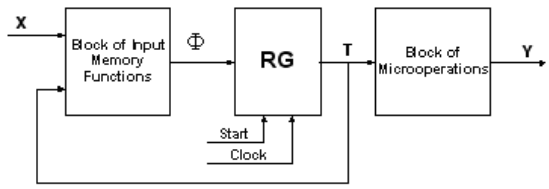

Fig. 1. Structural diagram of Moore FSM U_1

Rys. 1. Struktura diagramu automatu Moore'a U_1

In the Moore FSM the number of transitions H exceeds considerably this number of the equivalent Mealy FSM H_0 [2]. However, the appropriate encoding of states allows reducing this parameter, which leads to minimizing the number of PAL macrocells in BIM [8]. Various methods of states encoding for reduction macrocells PAL in BMO are used [3, 8]. But an effective method for optimizing both part of logic circuit FSM does not exist.

In this paper a method for reduction of hardware amount in both parts of the Moore finite state machine logic circuit is proposed.

One of Moore FSM features is the existence of pseudoequivalent states [2], which are the states with the same transitions by the effect of the same inputs. Such states correspond to the control algorithm operator vertices [1], outputs of which are connected with an input of the same vertex.

Let $\Pi_A = \{B_1,..,B_I\}$ be a partition of a set A on the classes of pseudoequivalent states. Let us code classes $B_i \in \Pi_A$ by binary codes $K(B_i)$ having R_1 bits, where

$$R_1 = \lceil \log_2 I \rceil . \tag{3}$$

Let use variables $\tau_r \in \tau$ for encoding classes of pseudoequivalent states, where $|\tau| = R_1$.

Let classes $B_i \in \Pi_A$ have M_i states and let

$$M_0 = \max(M_1,\ldots,M_I) . \tag{4}$$

Let us code state $a_m \in B_i$ by binary codes $C(a_m)$ having R_2 bits, where

$$R_2 = \lceil \log_2 M_0 \rceil , \tag{5}$$

and let us use variables $T_r \in T$ for it, where $|T| = R_2$.

In this case state $a_m \in A$ is unambiguously determined by the following expanded code

$$K(a_m) = K(B_i) * C(a_m) , \tag{6}$$

where * is a concatenation sign, $m = \overline{1,M}$, $i = \overline{1,I}$.

Let initial GSA Γ include Q different collections of microoperations (CMO) $Y_q \subseteq Y$. Let us code set Y_q with binary code $K(Y_q)$ having R_3 bits, where

$$R_3 = \lceil \log_2 Q \rceil . \tag{7}$$

Let us use variables $z_r \in Z$ for encoding CMO, where $|Z| = R_3$. In this case model of Moore FSM U_2 for interpretation of GSA Γ is offered (Fig.2).

In FSM U_2 the block BIM realizes functions

$$\Phi = \Phi(\tau, X) \tag{8}$$

that include $R_1 + R_2$ elements. The block CC forms variables $z_r \in Z$ and the functions

$$Z = Z(\tau, T) \tag{9}$$

are generated. The block BMO generates microoperations $y_n \in Y$ as functions

$$Y = Y(Z) . \tag{10}$$

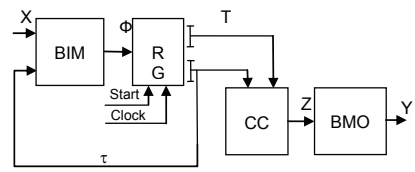

Fig. 2. Structural diagram of Moore FSM U_2

Rys. 2. Struktura diagramu automatu Moore'a U_2

The offered approach has the following advantages:

1. The number of terms in system (8) is always equal to H_0 for any coding of classes $B_i \in \Pi_A$.
2. The encoding of sets $Y_q \subseteq Y$ can be done so that to minimize numbers of PAL macrocells in BMO.

The evident disadvantages of FSM U_2 are increase in the number of functions Φ if the following condition

$$R_1 + R_2 > R \tag{11}$$

is fulfilled and existence of the block CC. It should be noted that encoding of the classes $B_i \in \Pi_A$ and states $a_m \in B_i$ can be done so that to minimize the numbers of PAL macrocells in the block CC. If the block BMO is implemented on customized matrices [9] or memory macrocells [3] and the following condition

$$R_3 < R \tag{12}$$

Is fulfilled, then the reduction in the hardware amount occurs.

3. Synthesis method of Moore FSM U_2

In this work a method of Moore FSM U_2 synthesis using a GSA Γ is proposed. The method includes the next stages:

1. Marking of the GSA Γ and creation of the set A.
2. Partition of the set A on classes of pseudoequivalent states.
3. Construction of microoperation collections $Y_q \subseteq Y$.
4. Optimal coding of microoperation collections $Y_q \subseteq Y$.
5. Coding of the classes $B_i \in \Pi_A$ and states $a_m \in B_i$.
6. Construction of the transformed structure table of Moore FSM U_2 .
7. Construction of the functions (8) – (9).
8. The synthesis of FSM logic circuit with given logic elements.

Let us discuss an example for a Moore FSM U_2 (Γ_1) synthesis, where a GSA Γ_1 is shown in Fig.3. The symbol $U_i(\Gamma_j)$ stands for interpretation of a GSA Γ_j with an FSM model U_i. As follows from Fig.3, the set A includes $M=10$ elements distributed among $I=4$ classes $B_1=\{a_1\}$, $B_2=\{a_2,a_3,a_4\}$, $B_3=\{a_5,...,a_8\}$, $B_4=\{a_9,a_{10}\}$. So $R=4$, $R_1=2$.

There are $Q=7$ different collections of microoperations in the operator vertices of the GSA Γ_1, namely $Y_1=Y(a_1)=\varnothing$, $Y_2=\{y_1,y_2,y_7\}$, $Y_3=\{y_3,y_5,y_6\}$, $Y_4=\{y_1,y_3,y_7\}$, $Y_5=\{y_1,y_2,y_6,y_8\}$, $Y_6=\{y_2,y_4,y_5\}$, $Y_7=\{y_2,y_4,y_6\}$. It is enough $R_3=3$ variables for coding these collections, it means that $Z=\{z_1,z_2,z_3\}$. Note that condition (12) holds.

Let us call the "optimal" coding such a coding of microoperation collections that function (10) has minimal possible number of terms. One of the variants of the optimal coding is presented in Fig. 4.

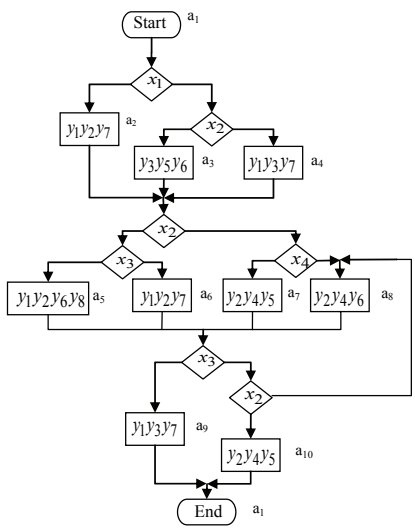

Fig. 3. Initial GSA Γ_1
Rys. 3. Sieć działań Γ_1

z_1 \ z_2z_3	00	01	11	10
0	Y_1	Y_3	Y_4	*
1	Y_7	Y_6	Y_2	Y_5

Fig. 4. Optimal codes of microoperation collections
Rys. 4. Kodowanie optymalne zbioru mikrooperacji

Subject to these codes we can get that

$$y_1=Y_2\vee Y_4\vee Y_5=z_2;\ y_2=Y_2\vee Y_6\vee Y_5\vee Y_7=z_1;$$
$$y_3=Y_3\vee Y_4=\bar{z}_1z_3;\ y_4=Y_6\vee Y_7=z_1\bar{z}_2;$$
$$y_5=Y_3\vee Y_6=\bar{z}_2z_3;\ y_6=Y_3\vee Y_5\vee Y_7=\bar{z}_1\bar{z}_2z_3\vee z_1\bar{z}_3;\quad (13)$$
$$y_7=Y_2\vee Y_4=z_2z_3;\ z_8=Y_5=z_2\bar{z}_3.$$

Classes $B_i\in\Pi_A$ and states $a_m\in B_i$ should be encoded so that to reduce number of terms in functions (9). It leads to optimization of the number of PAL macrocells in the block CC. Let us construct the table that shows connection between the codes of microoperation collections and the codes of states of Moore FSM $U_2(\Gamma_1)$ (Table 1).

Tab. 1. The connection between the codes Y_q and the states a_m
Tab. 1. Związek między kodami zbiorów mikrooperacji Y_q i stanami a_m

$K(Y_q)$	001	011	100	101	110	111
a_m	a_3	a_4, a_9	a_8, a_{10}	a_7	a_5	a_2, a_6
Y_q	Y_3	Y_4	Y_7	Y_6	Y_5	Y_2

In each column of Table 1 the first position of code $K(Y_q)$ corresponds to z_1, the second corresponds to z_2 and the third corresponds to z_3. It allows constructing the next function system

$$z_1=a_2\vee a_5\vee a_6\vee a_7\vee a_8\vee a_{10};$$
$$z_2=a_2\vee a_4\vee a_6\vee a_9;\quad (14)$$
$$z_3=a_2\vee a_3\vee a_4\vee a_6\vee a_7\vee a_9.$$

The states $a_m\in A$ should be encoded so that the code $K(B_i)$ is the same for all $a_m\in B_i$.

Optimization of the system (14) is a typical task of symbolic encoding. Methods for solving the similar tasks are described in [1]. One of the variants of optimal encoding the classes $B_i\in\Pi_A$ and the states $a_m\in B_i$ is shown in Fig. 5.

$\tau_1\tau_2$ \ T_1T_2	00	01	11	10
00	a_1	*	*	*
01	*	a_2	a_4	a_3
11	a_5	a_6	a_7	a_8
10	*	*	a_9	a_{10}

Fig. 5. Encoding the classes and states of Moore FSM $U_2(\Gamma_1)$
Rys. 5. Kodowanie klasów i stanów automatu Moore'a $U_2(\Gamma_1)$

The following system of equations can be obtained using the system (14) and the codes of Fig. 5

$$z_1=\tau_2\overline{T_1}\vee\tau_1\tau_2\vee\tau_1T_1\overline{T_2};$$
$$z_2=\overline{T_1}T_2\vee\overline{\tau}_2T_2;\quad (15)$$
$$z_3=T_2\vee\overline{\tau}_1\tau_2.$$

If PALs with $q=3$ terms are used for realization of the block CC, then each function $z_r\in Z$ is implemented in one PAL macrocell.

The system of generalized formulae of transitions is needed for construction of the transformed structure table [3]. In the case of $U_2(\Gamma_1)$ this system is

$$B_1\to x_1a_2\vee\bar{x}_1x_2a_3\vee\bar{x}_1\bar{x}_2a_4;$$
$$B_2\to x_2x_3a_5\vee x_2\bar{x}_3a_6\vee\bar{x}_2x_4a_7\vee\bar{x}_2\bar{x}_4a_8;\quad (16)$$
$$B_3\to x_3a_9\vee x_2\bar{x}_3a_{10}\vee\bar{x}_2\bar{x}_3a_8;$$
$$B_4\to a_1.$$

This system leads to the transformed table of transitions for FSM $U_2(\Gamma_1)$ (Table 2).

The system (8) can be obtained from Table 2.Then it should be minimized for realization on PAL macrocells. Let F_h be a term that corresponds to the line h of the transformed table of transitions. For example, $F_1=\bar{\tau}_1\bar{\tau}_2x_1$, $F_{10}=\tau_1\tau_2\bar{x}_3\bar{x}_2$. The function $D_1=F_4\vee\ldots\vee F_{10}$ can be found from Table 2. After

minimization the final function $D_1 = \tau_2$ is received. Similarly, it is possible to receive remaining functions of the system (8)

The last stage of synthesis is connected with implementation of circuits of the Moore FSM blocks on the obtained systems of functions. This stage is thoroughly discussed in the literature [6, 7]. There are the industrial packets [4, 5] oriented on solving the similar tasks. This stage is beyond the scope of this paper.

It should be noted that the transformed table of transitions for FSM $U_1(\Gamma_1)$ has $H = 23$ lines and each function $D_r \in D$ will have till 23 terms. Application of the proposed method reduces parameter H in 2,3 times and the number of functions $D_r \in \varphi$ will be the same for $U_1(\Gamma_1)$ and $U_2(\Gamma_2)$ because the condition (11) does not take place.

Tab. 2. Transformed table of transitions for FSM $U_2(\Gamma_1)$

Tab. 2. Tabela przejść automatu Moore'a $U_2(\Gamma_1)$

B_i	$K(B_i)$	a_s	$K(a_s)$	X_h	Φ_h	h
B_1	00	a_2	0101	x_1	D_2D_4	1
		a_3	0110	$\bar{x}_1x_2$	D_2D_3	2
		a_4	0111	$\bar{x}_1\bar{x}_2$	$D_2D_3D_4$	3
B_2	01	a_5	1100	x_2x_3	D_1D_2	4
		a_6	1101	$x_2\bar{x}_3$	$D_1D_2D_4$	5
		a_7	1111	$\bar{x}_2x_4$	$D_1D_2D_3D_4$	6
		a_8	1110	$\bar{x}_2\bar{x}_4$	$D_1D_2D_3$	7
B_3	11	a_9	1011	x_3	$D_1D_3D_4$	8
		a_{10}	1010	$\bar{x}_3x_2$	D_1D_3	9
		a_8	1110	$\bar{x}_3\bar{x}_2$	$D_1D_2D_3$	10

4. Conclusion

The proposed method of transformation of extended state codes is oriented on decrease in the hardware amount under implementation of the Moore FSM logic circuit. This approach guarantees the decrease in the number of PAL macrocells in the blocks of input memory function and microoperations. However, it is connected with implementation of additional block CC, which results in consumption of some resources of a CPLD crystal and adding delay in the Moore FSM logic circuit. These disadvantages can be reduced due to the optimal encoding of extended state codes.

Investigations of the proposed method effectiveness on standard examples [10] show that the proposed method allows chip space decrease to 37% (on average) for 98,2% of examples. Thus performance of FSM U_2 practically coincides with this parameter in FSN U_1 .It is because the number of levels in the circuit of the Moore FSM U_2 blocks is decreased.

The further direction of our research is application of the proposed method to cases of FPGA and customized matrices.

5. References

[1] DeMicheli G.: Synthesis and Optimization of Digital Circuits. NJ: McGraw-Hill, 1994. 636 pp.

[2] Baranov S.: Logic and System Design of Digital Systems. Tallinn: TUT Press, 2008. 267 pp.

[3] Barkalov A., Titarenko L.: Logic Synthesis for FSM-Based Control Units. Springer. Lectures Notes in Electrical Engineering. №53. Berlin: Verlag Heidelberg, 2009. 233 pp.

[4] Altera Corporation webpage. http//www.altera.com

[5] Xilins Corporation webpage. http//www.xilinx.com

[6] Соловьев В.В.: Проектирование цифровых систем на основе программируемых логических интегральных схем. – 2-е изд., М.: Горячая линия – Телеком, 2007. 636 с. ил.

[7] Kania D.: Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL. – Gliwice: Politechnika Śląska, 2004. 212 str.

[8] Barkalov A., Titarenko L., Chmielewski S.: Hardware reduction for Moor FSM implemented with CPLD. KEiT PAN, 2009. – 212 pp.

[9] Barkalov A., Titarenko L., Hebda O.: Matrix implementation of Moore FSM with expansion of coding space. Pomiary Automatyka Kontrola.vol.56, № 7, 694 – 696 pp, 2010.

[10] Yang S.: Logic Synthesis and Optimization Benchmarks user guide. Technical report №1991 IWLS-UG Saeyang. – Microelectronics center of North Carolina.

[11] Maxfield C.: The Design Warrior's Guide to FPGAs. Amsterdam: Elseveir, 2004. 541 pp.

[12] Smith M.: Application Specific Integrated Circuits. Boston: Addison-Wesley, 1997

otrzymano / received: 13.03.2011
przyjęto do druku / accepted: 04.05.2011 *artykuł recenzowany*

INFORMACJE

Informacja redakcji dotycząca artykułów współautorskich

W miesięczniku PAK od numeru 06/2010 w nagłówkach artykułów współautorskich wskazywany jest autor korespondujący (Corresponding Author), tj. ten z którym redakcja prowadzi wszelkie uzgodnienia na etapie przygotowania artykułu do publikacji. Jego nazwisko jest wyróżnione drukiem pogrubionym. Takie oznaczenie nie odnosi się do faktycznego udziału współautora w opracowaniu artykułu. Ponadto w nagłówku artykułu podawane są adresy korespondencyjne wszystkich współautorów.

Wprowadzona procedura wynika z międzynarodowych standardów wydawniczych.

Redakcja