Alexander BARKALOV, Larysa TITARENKO, Sławomir CHMIELEWSKI
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT INFORMATYKI I ELEKTRONIKI

DECREASE OF HARDWARE AMOUNT IN LOGIC CIRCUIT OF MOORE FSM

prof. dr hab. inż. Alexander BARKALOV, prof. UZ

Prof. Alexander A. Barkalov w latach 1976-1996 był pracownikiem dydaktycznym w Instytucie Informatyki Narodowej Politechniki Donieckiej. Współpracował aktywno 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

mgr inż. Sławomir CHMIELEWSKI

Mgr inż. Sławomir Chmielewski absolwent Uniwersytetu Zielonogórskiego. Ukończył studia informatyczne o specjalizacji inżynieria komputerowa. Obecnie student drugiego roku na studiach doktoranckich.

e-mail: S.Chmielewski@weit.uz.zgora.pl

Abstract

Method of decrease in the number of PAL macrocells in logic circuit of Moore FSM is proposed. The method is based on simultaneous application of refined state assignment and transformation of the codes of pseudoequivalent states into codes of their classes. The proposed approach permits to decrease the hardware amount without decrease of digital system performance. The results of experiments are shown.

Keywords: Moore finite-state-machine, PAL macrocells, CPLD, flow-chart of algorithm

Słowa kluczowe: automat Moore'a, PAL makro-komórka, CPLD, schemat blokowy algorytmu

Streszczenie

W pracy przedstawiona została metoda zmniejszania ilości makro-komórek w układach typu PAL przy pomocy automatów Moore'a FSM. Metoda ta bazuje na wykorzystaniu wyznaczonych stanów i przekształceniu kodu klasy pseudo-równoważnych stanów w odpowiedni kod danej klasy. Zaproponowane podejście pozwala zmniejszyć ilość wymaganego zużycia sprzętowego bez zmniejszenia wydajności systemów cyfrowych. Podany również jest przykład aplikacji zaproponowanego rozwiązania.

1. Introduction

Control unit of any digital system can be implemented of Moore finite-state-machine (FSM) [1]. Rapid evolution in semiconductor technology has resulted in appearance of sophisticated VLSI such as complex programmable logic devices (CPLD) where logic functions are implemented using programmable array logic (PAL) macrocells [2,3,4]. One of the issues of the day in case of FSM implementation using CPLD is decrease in the number of PAL macrocells in logic circuit of control unit [5]. A proper state assignment [6] can be used to solve this problem. Let us point out that such characteristics of FSM as cost/area, power consumption and maximum frequency depend heavily on this step outcome. Because of their importance, state assignment methods are continually developed. There are effective methods based on symbolic minimization [7,8,9], genetic algorithms [10,11] and other heuristics [12,13]. In this article we propose some method of decrease in the number of PAL macrocells where state assignment is oriented on optimization of microoperation block of FSM.

dr hab. inż. Larysa Titarenko, prof. UZ

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 2007 pracuje jako profesor na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego.

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

Block of input memory functions is optimized due to transformation of the codes of pseudoequivalent states [12] into codes of their classes and use of wide fan-in of PAL macrocells [14,15].

2. Background of Moore FSM

Let Moore FSM is represented by structure table [1] with the columns: a_m is a current state, $a_m \in A$, where $A = \{a_1,...,a_M\}$ is a set of internal states; $K(a_m)$ is a code of state a_m having $R = \lceil log_2 M \rceil$ bits; a_s is a state of transition; $K(a_s)$ is a code of state $a_s \in A$; X_h is a conjunction of some elements of the set of logical conditions X (or their complements) determining the transition $\langle a_m, a_s \rangle$ where $X = \{x_1,...,x_L\}$, Φ_h is a collection of input memory functions from set $\Phi = \{D_1,...,D_R\}$ which are equal to *1* to switch the automaton memory from $K(a_m)$ into $K(a_s)$; $h = 1$, …, H is number of line. The column a_m contains collection of microoperations $Y(a_m) \subseteq Y$, which are generated in the state $a_m \in A$, where $Y = \{y_1,...,y_N\}$. Such table determines Moore FSM U_1 shown in Fig. 1.

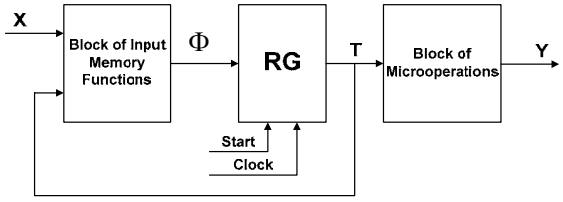

Fig. 1. Structural diagram of Moore FSM U_1
Rys. 1. Struktura diagramu Moore'a FSM U_1

In case of U_1, block of input memory functions (BIMF) generates functions

$$\Phi = \Phi(T, X) \qquad (1)$$

and block of microoperations (BMO) generates functions

$$Y = Y(T). \qquad (2)$$

Here $T = \{T_1,...,T_R\}$ is a set of state variables used to encode the states $a_m \in A$. Pulse Start is used to load the code of initial state $a_1 \in A$ into register RG. Pulse Clock causes change of RG content.

The hardware amount in logic circuit of FSM U_1 can be decreased using one of the following state assignment approaches

[2,12]. In the case of optimal state assignment [2] classes of pseudoequivalent states are represented by generalized intervals of R-dimensional Boolean space. It decreases the number of terms in system (1) up to the number of transitions of equivalent Mealy FSM. Let us remind that states a_s, $a_s \in A$ are pseudoequivalent states if identical inputs result in identical next states for both a_m and a_s. In the case of refined state assignment [12] each microoperation $y_n \in Y$ is represented by generalized interval of R-dimensional Boolean space. It decreases the number of terms in system (2) up to N. In both cases, such well-known method as ESPRESSO [6] can be used for state assignment. Obviously, it is impossible to apply both methods simultaneously. It means that hardware amount can by decreased either for BIMF, or for BMO. In our article we propose some method allowing hardware decrease for both combinational blocks of Moore FSM U_1.

3. Main idea of proposed method

Let $\Pi_A = \{B_1,...,B_I\}$ be a partition of the set A by the classes of pseudoequivalent states. Let $\Pi_A = \Pi_B \cup \Pi_C$, where $B_i \in \Pi_B$ if $|B_i| \geq 2$ and otherwise $B_i \in \Pi_C$. Let us encode classes $B_i \in \Pi_B$ by binary codes $K(B_i)$ using

$$R_1 = \lceil log_2 I_B \rceil \qquad (3)$$

variables $\tau_r \in \tau$, where $I_B = |\Pi_B|$. Due to wide fan-in of PAL macrocells [3,4], it is possible to use more than one source of state code for block of input memory functions [14,15]. It makes possible Moore FSM U_2 shown in Fig. 2

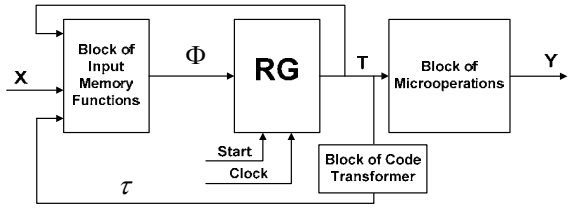

Fig. 2. Structural diagram of Moore FSM U_2
Rys. 2. Struktura diagramu Moore'a FSM U_2

In FSM U_2, BIMF implements functions

$$\Phi = \Phi(T, \tau, X), \qquad (4)$$

block of code transformer (BCT) transforms codes of pseudoequivalent states $a_m \in B_i$ into codes of the classes $K(B_i)$. To do it, BCT generates functions

$$\tau = \tau(T). \qquad (5)$$

The refined state assignment allows decrease of the numbers of PAL macrocells in both BMO and BCT.

Let control algorithm to be interpreted is represented by graph-scheme of algorithm [1]. In this case proposed design method includes the following steps:

1. Construction of Moore FSM transition table (without codes of states).
2. Construction of partitions Π_A, Π_B and Π_C.
3. Encoding of the classes $B_i \in \Pi_B$.
4. Refined state assignment minimizing the total number of terms in systems (2) and (5).
5. Construction of transformed structure table where current states $a_m \in B_i$ are replaced by corresponding classes $B_i \in \Pi_A$ and codes $K(a_m)$ are replaced by corresponding codes $K(B_i)$.
6. Construction of table for block BCT and system (5).
7. Construction of systems (2) and (4).
8. Implementation of Moore FSM U_2 logic circuit using systems (2), (4) and (5) and some CPLD chips.

4. Example of proposed method application

Let some Moore FSM S_1 is represented by its structure table, from which the following information can be obtained: $M = 13$, $N = 8$, $\Pi_A = \{B_1,...,B_7\}$, where $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3\}$, $B_3 = \{a_4\}$, $B_4 = \{a_5, a_6, a_7\}$, $B_5 = \{a_8, a_9\}$, $B_6 = \{a_{10}\}$ and $B_7 = \{a_{11}, a_{12}, a_{13}\}$. Let system (2) be represented as the following one:

$$Y_1 = A_4 \vee A_8 \vee A_9 \vee A_{10};\ Y_2 = A_5 \vee A_{11};$$

$$Y_3 = A_2 \vee A_3 \vee A_{12};\ Y_4 = A_6 \vee A_7 \vee A_8;$$

$$Y_5 = A_3 \vee A_7 \vee A_8 \vee A_{11};\ Y_6 = A_3 \vee A_7;$$

$$Y_7 = A_4 \vee A_6 \vee A_{10};\ Y_8 = A_3 \vee A_7 \vee A_{12}. \qquad (6)$$

In this system variables A_m correspond to conjunctions of state variables $T_r \in T$ determined by state codes $K(a_m)$ $(m = 1,...,M)$. Let transitions for classes $B_2, B_3 \in \Pi_A$ are the following ones:

$$B_2 \to x_3 a_4 \vee \bar{x}_3 x_4 a_7 \vee \bar{x}_3 \bar{x}_4 a_6;.$$

$$B_3 \to a_9. \qquad (7)$$

In discussed case $R = 4$, $T = \{T_1,...,T_4\}$, $\Phi = \{D_1,...,D_4\}$. Using the algorithm ESPRESSO [6] the following refined state assignment can be obtained in case of Moore FSM S_1 (Fig. 3).

T_1T_2 \ T_3T_4	00	01	11	10
00	a_1	a_2	*	a_4
01	a_5	*	a_6	a_{10}
11	a_{11}	a_3	a_7	a_8
10	a_{13}	a_{12}	*	a_9

Fig. 3. Refined state assignment for Moore FSM S_1
Rys. 3. Stany automatu Moore'a FSM S_1

Using these codes and insignificant state assignments marked by *, we can transform the system (6) into the following system:

$$Y_1 = T_3\bar{T}_4;\ Y_2 = T_2\bar{T}_3\bar{T}_4;\ Y_3 = \bar{T}_3T_4;$$

$$Y_4 = T_3T_4 \vee T_1T_2T_3;\ Y_5 = T_1T_2;$$

$$Y_6 = T_1T_2T_4;\ Y_7 = \bar{T}_1T_3;\ Y_8 = T_1T_4. \qquad (8)$$

In practice, each PAL macrocell has not less than 8 inputs [3,4]. It means that each function from (8) can be implemented using only one macrocell. Obviously, there is no solution with less hardware amount.

An analysis of Fig. 3 shows that the codes $K(a_8)$ and $K(a_9)$ belong to single generalized interval of Karnaugh map. It means that class $B_5 \in \Pi_B$, because states $a_8, a_9 \in B_5$. Thus, there are the partitions $\Pi_B = \{B_1, B_3, B_5, B_6\}$, $\Pi_C = \{B_2, B_4, B_7\}$, it means that $I_C = 3$, $R_1 = 2$ and $\tau = \{\tau_1, \tau_2\}$.

The classes $B_i \in \Pi_C$ can be encoded using algorithm ESPRESSO [6], but in our simple case they can be encoded in the

following way: $K(B_4)=01$, $K(B_2)=10$, $K(B_3)=11$. Let us point out that code 00 is used as a pointer that $B_i \notin \Pi_C$.

Transformed structure table of Moore FSM includes the following columns: B_i, $K(B_i)$, a_S, $K(a_S)$, X_h, Φ_h, h. In our case the following part of this table (Table 1) correspond to system (7).

Tab. 1. Fragment of transformed structure table for S_1
Tab. 1. Fragment tabeli z przekształceniem dla S_1

B_i	$K(B_i)$	$K(a_m)$	$K(a_S)$	X_h	Φ_h	h
B_2	10****	-	0010	x_3	D_3	1
			1111	$/x_3x_4$	$D_1D_2D_3D_4$	2
			0111	$/x_3/x_4$	$D_2D_3D_4$	3
B_3	00001*	0001	1010	1	D_1D_3	4

In this table the first R_1 bits in column $K(B_i)$ correspond to variables $\tau_r \in \tau$ and the last R bits to $T_r \in T$. If $\tau_1 = \tau_2 = 0$, then register RG is a source of the codes $K(B_i)$. Otherwise, these codes are taken from block BCT. From this table we can get system (4). For example, Table 1 gives the following equation: $D_1 = \tau_1\overline{\tau}_2\overline{x}_3x_4 \vee \overline{\tau}_1\overline{\tau}_2\overline{T}_1\overline{T}_2T_3$.

Table of BCT includes the columns a_m, $K(a_m)$, B_i, $K(B_i)$, τ_m, m. Here the column τ_m includes variables $\tau_r \in \tau$, which are equal to *1* for code $K(B_i)$ from m-th line of the table. In our case table of BCT includes $I_M = 7$ lines (Table 2).

Tab. 2. Table of block BCT for Moore FSM S_1
Tab. 2. Fragment tabeli BCT Moore'a FSM S_1

a_m	$K(a_m)$	B_i	$K(B_i)$	τ_m	m
a_2	0001	B_2	10	τ_1	1
a_3	1101				2
a_5	0100	B_4	01	τ_2	3
a_6	0111				4
a_7	1111				5
a_{11}	1100	B_7	11	$\tau_1\tau_2$	6
a_{12}	1001				7

This table is used for construction of system (5). For example the following equation can be derived from Table 2: $\tau_1 = \overline{T}_3T_4 \vee T_1T_2\overline{T}_3$.

Implementation of logic circuit for Moore FSM U_2 is reduced to implementation of systems (2), (4) and (5) using some standard package [3,4].

5. Conclusion

The proposed method is based on use of transformation of state codes into codes of the classes of pseudoequivalent states of Moore FSM. It is oriented toward decrease in the number of terms in each from the input memory functions. In the same time it allows application of refined state assignment leading to hardware amount decrease for blocks of microoperations and code transformation. Our experiments show that proposed method always produces logic circuits with fewer amounts of PAL macrocells than any known methods for Moore FSM design.

Let us point out, that decrease in hardware amount is very often connected with decrease in combinational circuit levels. It results in decrease of FSM cycle time and, therefore, increase of its performance. Because code transformation is executed in the same time when system data-path executes some operation, it does not lead to slowing down of digital system operation.

Our future research is connected with exploration of possibility for the proposed method application when a control unit is implemented using technology of FPGA. In this case we are going to use some standard benchmarks from [16].

5. References

[1] S. Baranov, "Logic Synthesis for Control Automata," Boston: Kluwer, 1994.
[2] A. Barkalov and M. Węgrzyn, "Design of Control Units with Programmable Logic," Zielona Gora: University of Zielona Gora Press, 2006.
[3] http://www.altera.com.
[4] http://www.xilinx.com.
[5] D. Kania, "Logic Synthesis Oriented on Programmable Logic Devices of the PAL type," Gliwice: Silesian Technical University (in Polish), 2004.
[6] G. De Micheli, "Synthesis and Optimization of Digital Circuits," New York: McGraw Hill, 1994.
[7] S. Devadas, H.-K. Ma, R. Newton, A. Sangiovanni-Vincentelli, "State Assignment of Finite State Machines Targeting Multilevel Logic Implementations," *IEEE Transactions on Computer-Aided Design*, 1988, pp.1290-1300.
[8] T. Kam, T. Villa, R. Brayton, A. Sangiovanni-Vincentelli, "Synthesis of Finite State Machines: Functional Optimization," Kluwer Academic Publishers, Boston/London/Dordrecht, 1998.
[9] T. Villa, T. Kam, R. Brayton, A. Sangiovanni-Vincentelli, "Synthesis of Finite State Machines: Logic Optimization," Kluwer Academic Publishers, Boston/London/Dordrecht, 1998.
[10] S. Chattopadhyay, "Area Conscious State Assignment with Flip-Flop and Output Polarity Selection for Finite State Machine Synthesis," A Genetic Algorithm Approach, The Computer Journal, vol. 48, No. 4, 2005, pp. 443-450.
[11] Y. Xia and A. Almaini, "Genetic algorithm based state assignment for power and area optimization," *IEEP. – Comput. Dig. T.*, 149, 2002, pp. 128-133.
[12] A. Barkalov, "Principles of Optimization of logical circuit of Moore FSM. Cybernetics and system analysis," No.1, 1998, pp. 65-72 (in Russian).
[13] V. Solovjev, "Design of Digital System Using the Programmable Logic Integrated Circuits," Moscow: Hotline – Telecom (in Russian), 2001.
[14] A. Barkalov, L. Titarenko, S. Chmielewski, "Optimization of Moore FSM on CPLD," *Computer – Aided design of Discrete Devices,* Proceedings of the Sixth International conference, Minsk, vol. 2, pp. 39-45, November 2007.
[15] A. Barkalov, L. Titarenko, S. Chmielewski, "Optimization of Moore FSM on System-on-Chip," *IEEE East-West Design & Test Symposium,* Yerevan, Armenia, Kharkov, 2007, pp. 105-109.
[16] S. Yang, "Logic Synthesis and Optimization Benchmarks User Guide," Microelectronics Center of North Carolina, Research Triangle Park, North Carolina, 1991.

Title: Zmniejszenie zasobów sprzętowych w układach cyfrowych przy użyciu automatów Moore'a FSM

Artykuł recenzowany