Alexander BARKALOV, Larysa TITARENKO, Olena HEBDA
UNIWERSYTET ZIELONOGÓRSKI,
ul. Licealna 9, 65-417 Zielona Góra

Optimization of Moore finite-state-machine matrix circuit

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 area of matrix implementation of the Moore finite state machine (FSM) circuit is proposed. The method is based on optimal state coding and decomposition of a matrix in two sub-matrices. Thus, classes of the pseudoequivalent states are used. Such approach allows reducing number of lines of the Moore FSM transition table to that of the equivalent Mealy FSM. As a result, the area of the matrices forming the excitation function of a states memory register is optimized. An example of the proposed method application is given.

Keywords: Moore FSM, graph-scheme of algorithm, pseudoequivalent states, customized matrices, logic circuit.

Optymalizacja macierzowego układu skończonego automatu stanu typu Moore'a

Streszczenie

Model skończonego automatu stanu typu Moore’a jest często stosowany w jednostkach sterujących [1]. Postęp technologii półprzewodnikowej pozwala na tworzenie coraz bardziej złożonych układów cyfrowych. W przypadku produkcji masowej szeroko stosowane są układy ASIC (ang. Application-Specified Integrated Circuits). W układach ASIC automaty skończone są projektowane przy użyciu struktur macierzowych (rys. 1). Jednym z głównych problemów syntezy automatów skończonych ze strukturami macierzowymi jest zmniejszenie powierzchni układu scalonego zajmowanej przez układ logiczny automatu Moore’a. W artykule proponowana jest metoda, która jest ukierunkowana na redukcję zasobów sprzętowych potrzebnych do implementacji skończonego automatu stanu typu Moore’a implementowanego w układach o strukturze macierzowej. Ta metoda jest oparta na optymalnym kodowaniu stanów i rozbijaniu macierzy termów na dwie podmacierze (rys. 2). Takie podejście pozwala zmniejszyć liczbę linii w tabeli przejść automatu Moore’a do liczby linii równoważnej automatowi z wyjściami typu Mealy’ego (tab. 2). Artykuł przedstawia także przykład zastosowania proponowanej metody.

Słowa kluczowe: automat typu Moore'a, sieć działań, stany pseudorównoważne, układ logiczny.

1. Introduction

The model of Moore finite state machine (FSM) [1] is often used during the digital control systems realization [3, 7]. The development of microelectronics has led to appearance of different programmable logic devices [3] which are used for implementing FSM logic circuit. But in the case of mass production of microelectronics products, there are widely use so called customized circuits called ASIC (Application-Specified Integrated Circuits) [4]. In the case of ASIC, the logic circuits of FSM are designed, as a rule, using so called matrix structures. In these customized matrices, the principle of distributed logic is used [6].

One of the important problems of FSM synthesis with matrix structures is the decrease in chip space occupied by FSM logic circuit. In this work the optimization method is proposed. It is based on using two sources of state codes.

2. The general aspects and basic idea of the 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 $a_m \in A$ having the capacity $R = \lceil \log_2 M \rceil$ (to code the states the internal variables from the set $T = \{T_1,..,T_R\}$ 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 it is equal to conjunction of some elements (or their complements) of a set of logic conditions $X = \{x_1,..,x_L\}$; Φ_h is a set of input memory functions for flip-flops of FSM memory, which are equal to 1 to change the content of the memory from $K(a_m)$ to $K(a_s)$, $\Phi_h \subseteq \Phi = \{\varphi_1,..,\varphi_R\}$; $h = 1, \ \ldots, \ H$ is a number of transition. In the column a_m a set of microoperations Y_q is written, these microoperations are generated in the state $a_m \in A$ ($Y_q \subseteq Y = \{y_1,..,y_N\}$, $q = 1,...,Q$). This table is a basis to form the following systems of functions

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

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

These systems specify an FSM logic circuit. In the case of matrix implementation, systems (1)-(2) determine the model of Moore FSM U_1, shown in Fig. 1.

In the FSM U_1, a conjunctive matrix M_1 implements the system of terms $F = \{F_1,...,F_H\}$. Each term F_h corresponds to one line of ST. F_h is determined as

$$F_h = A_m X_h \quad (h = \overline{1, H_1}). \tag{3}$$

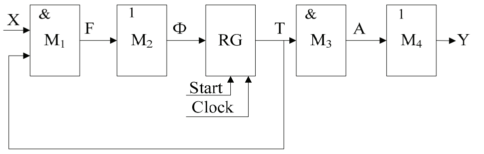


Fig. 1. Matrix circuit of Moore FSM U_1
Rys. 1. Macierzowy układ automatu Moore'a U_1

In the equation (3) variable $A_m = \overset{R}{\underset{r=1}{\wedge}} T_r^{l_{mr}}$ corresponds to conjunction of internal variables $T_r \in T$ that is defined by code $K(a_m)$ of initial state for line h of ST. $l_{mr} \in \{0,1\}$ is a value of the r th position of code $K(a_m)$, $T_r^0 = \overline{T}_r$, $T_r^1 = T_r$ ($r = \overline{1,R}$).

A disjunctive matrix M_2 implements system (1); a conjunctive matrix M_3 implements terms A_m ($m = 1,\ldots,M$), corresponding to FSM states; a disjunctive matrix M_4 implements system (2). The register RG keeps state codes; it is controlled by signals Start (clearing) and Clock (changing content depending on functions Φ). The matrices M_1 and M_2 determine the block of input memory functions (BIM), whereas the matrices M_3 and M_4 determine the block of microoperations (BMO).

Complexity of each matrix is defined by the area $S(M_i)$ of a crystal demanded for its implementation ($i = \overline{1,4}$). In theoretical papers this area is defined in arbitrary units [1, 6]. Next estimations can be obtained for FSM U_1:

$$\begin{aligned} &S(M_1) = 2(L+R)H_1; S(M_2) = H_1 R; \\ &S(M_3) = 2R \cdot M; S(M_4) = M \cdot N. \end{aligned} \tag{4}$$

The area $S(U_1)$ that is occupied by a logic circuit of FSM U_1 is defined as a sum of the areas (4).

The evident disadvantage consists in the considerable excess of parameter H_0 defined by a number of ST lines of the equivalent Mealy FSM. In addition the number of states of Moore FSM can exceed appreciably number of states of the equivalent Mealy FSM. Thus, it is frequent in practice that next conditions are taken place:

$$H_0 < H_1; R_0 = \lceil \log_2 M_0 \rceil < R. \tag{5}$$

One of Moore FSM features is 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 [3], outputs of which are connected with an input of the same vertex. The known methods of optimization of the matrix circuit U_1 are based on existence of pseudoequivalent states. There are two main optimization methods.

Method of optimal state encoding. The state $\mathrm{a_m} \in \mathrm{A}$ is encoded so that each class of pseudoequivalent states $B_i \in \Pi_A$ is represented by a single generalized interval of R-dimensional Boolean space. It leads to Moore FSM U_2. The structures of both U_1 and U_2 are the same. But in U_2 M_1 realizes only $H_0 = H_2 < H_1$ terms. However, such coding that leads to $H_2 > H_0$ is not always possible [3].

Method of transformation of the states codes. The classes $B_i \in \Pi_A$ are encoded by the binary codes $K(B_i)$ with $R_1 = \lceil \log_2 I \rceil$ bits. Let us point out that $I = M_0$, where M_0 is the number of the states of the equivalent Mealy FSM. The variables $\tau_r \in \tau$ are used for such encoding, where $|\tau| = R_1$. Next, a system of the function is formed

$$\tau = \tau(I). \tag{6}$$

The system (6) specifies the law of transforming the codes $K(a_m)$ into the codes $K(B_i)$. ST is transformed in such way that the states $a_m \in B_i$ are replaced with the classes $B_i \in \Pi_A$ in the column of the initial state. Such approach allows decreasing ST to H_0 and the number of variable feedback is reduced to $R_1 < R$.

This approach leads to a Moore FSM U_3, with a code transformer (CT). The number of transitions of the Moore FSM U_3 is equal to H_0. The drawback of U_3 is the existence of a block of the code transformer that consumes additional resources of embedded memory blocks (in comparison with U_1).

Let us encode states $\mathrm{a_m} \in \mathrm{A}$ so that the maximum possible number of classes $B_i \in \Pi_A$ is represented by single interval R-dimensional Boolean space. Let us include such classes in set Π_{RG} and remaining classes in set Π_{CT}, where $|\Pi_{CT}| = I_1$. Let us encode classes $B_i \in \Pi_{CT}$ by binary codes $K(B_i)$ having $R_2 = \lceil \log_2(I_1 + 1) \rceil$ bits. This approach leads to a Moore FSM U_4 (Fig. 2).

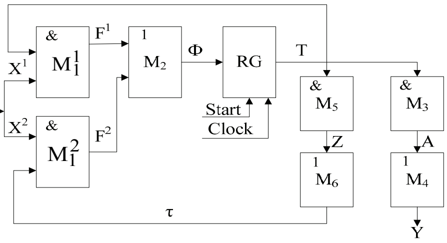


Fig. 2. Matrix circuit of Moore FSM U_4
Rys. 2. Macierzowy układ automatu Moore'a U_4

In U_4 the matrix M_1^1 implements the system of terms (3) that depends on variables $T_r \in T$. Register RG is a source of the codes of classes $B_i \in \Pi_{RG}$. The matrix M_1^2 realizes the system of terms $F_h = (\overset{R_1}{\underset{r=1}{\wedge}} \tau^{l_{ir}}) \cdot X_h$ for which CT is a source of the codes. The matrix M_5 realizes a system of terms Z and matrix M_6 realizes function (6). The set of terms is divided into classes F^1 and F^2. Terms of class F^1 are determined as

$$F_h = (\overset{R}{\underset{r=1}{\wedge}} T_r^{l_{ir}}) \cdot X_h, \tag{7}$$

where $l_{ir} \in \{0,1,*\}$ is a value of the r th position of code $K(B_i)$, $T_r^0 = \overline{T}_r$, $T_r^1 = T_r$, $T_r^* = 1$ ($r = \overline{1,R}$, $h = \overline{1,|F^1|}$).

Such an approach allows decreasing the total number of terms in the systems F^1 and F^2 till H_0. Thus, the total area of matrices M_1^1, M_1^2 and M_2 will be less than that of matrices M_1 and M_2 in U_1. As ST implements only the partial transformation of the state codes (only for $a_m \in B_i$, where $B_i \in \Pi_{CT}$), it occupies the smaller space than in U_3.

Let us estimate the areas of the matrixes of U_4 (except M_3 and M_4):

$$\begin{aligned} &S(M_1^1) = 2(L_1 + R) \cdot H_0^1; \quad S(M_1^2) = 2(L_2 + R_1) \cdot H_0^2; \\ &S(M_2) = H_0 \cdot R; \quad S(M_5) = 2R \cdot H_z; \quad S(M_6) = H_z \cdot R_2. \end{aligned} \tag{8}$$

In the system (8) $H_0^1 = \left|F^1\right|$, $H_0^2 = \left|F^2\right|$, $L_1 = \left|X^1\right|$, $L_2 = \left|X^2\right|$, $H_Z = \left|Z\right|$, $H_0^1 + H_0^2 = H_0$, $X^1 \cup X^2 = X$.

3. Synthesis method of Moore FSM U_4

In this paper there is proposed the following design method for Moore FSM U_4:

1. Marking of the GSA Γ and creation of the set A.
2. Partition of the set A on classes of pseudoequivalent states Π_A.
3. Optimal encoding of the states and forming the sets Π_{RG}, Π_{CT}.
4. Coding of the classes $B_i \in \Pi_{CT}$.
5. Construction of CT for classes $B_i \in \Pi_{RG}$ and $B_i \in \Pi_{CT}$.
6. Construction of code transformation table.
7. Formation of systems of functions setting matrices of U_4.
8. Implementation of FSM matrix circuit on customized matrices.

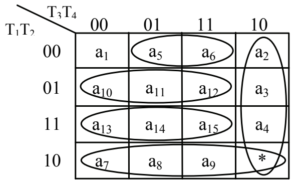
Let us for some control algorithm Γ_1 assume $A = \{a_1, \ldots, a_{15}\}$, $X = \{x_1, \ldots, x_8\}$, $Y = \{y_1, \ldots, y_{12}\}$ that is $M = 15$, $L = 8$ and $N = 12$, $R = 4$ and $T = \{T_1, \ldots, T_4\}$. It can be found from GSA Γ_1, that $H_1 = 53$ and $H_0 = 21$. From the system of functions (4) the following areas of GSA Γ_1 can be found: $S(U_1) = 1904$; $S(M_1) = 2(8+4)\cdot 53 = 1272$; $S(M_2) = 53 \cdot 4 = 212$; $S(M_3) = 2 \cdot 8 \cdot 15 = 240$; $S(M_4) = 15 \cdot 12 = 180$. Thus, the matrices M_1 and M_2 occupy 67 % of the total area of a crystal.

There is the partition $\Pi_A = \{B_1, \ldots, B_6\}$, where $B_1 = \{a_1\}$, $B_2 = \{a_2, a_3, a_4\}$, $B_3 = \{a_5, a_6\}$, $B_4 = \{a_7, a_8, a_9\}$, $B_5 = \{a_{10}, a_{11}, a_{12}\}$, $B_6 = \{a_{13}, a_{14}, a_{15}\}$. Let us encode the states $a_m \in A$ in the optimal way (Fig. 3). The following codes can be found from the Karnaugh map (Fig. 3): $K(B_1) = 000$, $K(B_2) = {**}10$, $K(B_3) = 10{**}$, $K(B_4) = 00{*}1$. Class B_5 and class B_6 require two intervals of 4-dimensional Boolean space. It can be found that $H_2 = 28$ and $S(M_1) = 2(8+4)\cdot 28 = 672$, $S(M_2) = 28 \cdot 4 = 112$, $S(U_2) = 1204$. Now the matrices M_1 and M_2 occupy 56 % of the total area of a crystal.

So $\Pi_{RG} = \{B_1, \ldots, B_4\}$ and $\Pi_{CT} = \{B_5, B_6\}$, $I_1 = 2$ and $R_2 = 2$, $\tau = \{\tau_1, \tau_2\}$. Let code 00 be used for identification of sign $B_i \notin \Pi_{CT}$ and let $K(B_5) = {*}1$, $K(B_6) = 1{*}$.

The system of generalized formulae of transitions is needed for construction of the transformed structure table. In the case of Γ_1 for classes $B_i \in \Pi_{RG}$ this system is

$$\begin{aligned}
&B_1 \to x_1 a_2 \vee \bar{x}_1 x_2 a_3 \vee \bar{x}_1 \bar{x}_2 a_4; B_2 \to x_3 a_5 \vee \bar{x}_3 x_4 a_6 \vee \bar{x}_3 \bar{x}_4 a_4;\\
&B_3 \to x_4 x_5 a_7 \vee x_4 \bar{x}_5 a_8 \vee \bar{x}_4 x_6 a_9 \vee \bar{x}_4 \bar{x}_6 a_{10};\\
&B_4 \to x_1 x_3 a_{11} \vee x_1 \bar{x}_3 a_7 \vee \bar{x}_1 x_4 a_{12} \vee \bar{x}_1 \bar{x}_4 a_9.
\end{aligned} \tag{9}$$



T_1T_2 \ T_3T_4	00	01	11	10
00	a_1	a_5	a_6	a_2
01	a_{10}	a_{11}	a_{12}	a_3
11	a_{13}	a_{14}	a_{15}	a_4
10	a_7	a_8	a_9	*

Fig. 3. The state codes for FSM U_4
Rys. 3. Kody stanów automatu U_4

In the case of Γ_1 for classes $B_i \in \Pi_{CT}$ this system is

$$\begin{aligned}
&B_5 \to x_5 x_6 a_{13} \vee x_5 \bar{x}_6 a_{14} \vee \bar{x}_5 x_7 a_{15} \vee \bar{x}_5 \bar{x}_7 a_{10};\\
&B_4 \to x_3 x_8 x_6 a_{13} \vee x_3 x_8 \bar{x}_6 a_{14} \vee x_3 \bar{x}_8 a_1 \vee \bar{x}_3 a_{10}.
\end{aligned} \tag{10}$$

Thus, $H_0^1 = 14$ and $H_0^2 = 8$, $S(M_2) = (14+8)\cdot 4 = 88$. As $X^1 = \{x_1, \ldots, x_6\}$ and $X^2 = \{x_3, x_5, x_6, x_7, x_8\}$, $L_1 = 6$ and $L_2 = 5$. Thus, $S(M_1^1) = 2\cdot(6+4)\cdot 14 = 280$, $S(M_1^2) = 2\cdot(5+2)\cdot 8 = 112$, whereas $S(M_3) + S(M_4) = 240 + 180 = 420$ (as in FSM U_1).

In this case, the system (6) has $H_z = 4$ terms. Thus, there can be found areas $S(M_5) = 2\cdot 4\cdot 4 = 32$ and $S(M_6) = 4 \cdot 2 = 8$. For implementation of matrices M_3 and M_4, it is necessary to generate system (2). For example, $y_1 = A_2 \vee A_4 \vee A_5 \vee A_9$. Implementation of the obtained systems of equations is a trivial task and, as such, is not considered in this paper.

The analysis shows that the proposed method allows finding the most efficient decision. Let us note that this conclusion remains valid for 83 % FSMs presented in [4] as standard examples for comparison of various synthesis methods. The models U_2 and U_3 were the most efficient for 11% and 6% examples from [4], respectively.

4. Conclusions

The offered method has two advantages. Firstly, the decrease in the number of terms in the system of Moore FSM input memory functions to that of the equivalent Mealy FSM is guaranteed. Secondly, using two sources of state codes allows reducing complexity of the state code transformer circuit (in comparison with the circuit with one source). In addition, decomposition of the transitions table in two sub-tables allows decreasing complexity of the circuit of forming terms of the excitation function system. It is connected with reduction in the number of the logical conditions defining terms for each sub-table in comparison with the known methods of implementation. The method is ineffective only at approaching the result of optimum state coding to two extreme points.

In the first case all classes are represented as a single generalized interval. Thus, the least area of implementation is ensured using the model U_2. In the second case all classes of pseudoequivalent states are liable to transformation. Meanwhile the model U_3 is more efficient. Though it has been found that models U_2 and U_3 are better than model U_4 only for 17% of test examples from [4].

The further direction of our research is connected with application of the proposed method for the cases when FSM logic circuits are implemented using standard elements such as CPLD and FPGA.

5. References

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

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

[3] Nababi Z.: Embedded Core Design with FPGA. NY: McGraw-Hill, 2008.

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

[5] Yang S.: Logic Synthesis and Optimization Benchmarks user guide. Technical report. Microelectronics center of North Carolina, 1991.

[6] Barkalov A.A., Titarenko L.A.: Sintez mikroprogramnych awtomatow na zakaznych i programirujemych SBIS. Donieck, UNITECH, 2009, 336 s.

[7] Soloviev W.W.: Projektirowanie cifrowych sistem na osnowie programirujemych logiczeskich integralnych schem. Wyd. 2, M – Goraczaja linia – Telekom, 2007, 636 s.

otrzymano / received: 13.05.2011
przyjęto do druku / accepted: 04.07.2011 *artykuł recenzowany*