1

UDC 004.3

SYNTHESIS OF FSM WITH TRANSFORMATION OF CODES OF OBJECTS

Barkalov A.A.
Institute of Cybernetics of NAN of Ukraine
Barkalovaa@gmail.com

The general method of transformation of codes of states into the codes of microoperations' sets and vice versa is proposed. The structures of circuits for Mealy and Moore FSMs are proposed and method of their design is discussed. The further hardware reduction is connected with increasing of the circuit's levels. The possible solutions are given.

1 Introduction

A control unit (CU) of any digital system can be represented either as Mealy or Moore finite state machine (FSM). The minimization of hardware in the circuits of control units (CU) implemented on the Programmable Logic Devices (PLD) can be worked out thanks to the increasing of the number of levels in the circuit of FSM [1]. In this case optimization of the circuit that implements the microoperations of the digital system is connected with formation of some additional variables [2, 3]. The method that permits to decrease the amount of additional variables and, therefore, an amount of PLDs in the circuit of FSM is proposed in this article.

2 The main definitions and an idea of the proposed method

Let a control unit be represented as a finite state machine S with set of internal states $A = \{a_1,\ldots,a_M\}$, set of logic conditions $X = \{x_1,\ldots,x_L\}$, set of microoperations $Y = \{y_1,\ldots,y_N\}$ and terms $F = (F_1,\ldots,F_M\}$ and let each term corresponds to one line of the direct structural table (DST) [1]. Let DST has Z different sets of microoperations $Y_Z \subseteq Y$ and let states $a_m \in A$ are encoded using special set of internal variables $T_r \in T = \{T_1,\ldots,T_R\}$, where R =]log$_2$M[. Let each set Y_z corresponds to the binary code K(Y_z) with Q =]log$_2$Z[bits and let these encoding variables form set $V = (v_1,\ldots,v_Q\}$. The states $a_m \in A$ correspond to the binary codes K(a_m) with R bits. Let's name state $a_m \in A$ and set $Y_Z \subseteq Y$ as the objects of FSM S. The main idea of the proposed method is following.

One of the objects (state or set of microoperations) is a function on the terms of DST and second object is the function of the first one and - may be- some additional elements. Such approach is based on the insertion of special code transformer (CT) in the structure of FSM. If code transformer CT implements the matching A $\rightarrow$ Y, then we'll name such FSM as PC$_A$Y – FSM. If code transformer CT implements the matching Y $\rightarrow$ A, then we'll name such FSM as PC$_Y$Y – FSM. The figures 1 and 2 show the structures of PC$_A$Y- FSM and PC$_Y$Y – FSM that are the Moore FSM.. Here W is the set of variables that are necessary for one-to-one identification of the objects [4].

Because of the independence of the output functions of Moore FSM from the logic conditions, the subcircuit Y implementing the system of output functions Y can be connected with the outputs of register RG whether directly (PC$_Y$Y – FSM) or through the circuit CT (PC$_A$Y – FSM). Here subcircuit P forms the excitation functions $\Phi = \Phi(T,E)$ and special functions $\Psi = \Psi(T,X)$ that are needed to form in the RG whether the functions T(PC$_A$Y – FSM) or the functions V and W (PC$_Y$Y – FSM). PC$_A$Y – FSM does not form functions Ψ because of relation

$$M \geq Z. \qquad (1)$$

Condition (1) is true only for particular case of Mealy FSM. It is the reason to form the functions W in Mealy PC$_A$Y- FSM (Figure 3) and PC$_Y$Y – FSM (Figure 4).

If codes of the objects are form by subcircuit P such objects are named as primary objects. If codes of the objects are the functions from other objects they are named as the secondary objects.

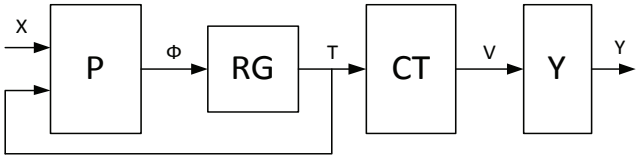

Figure 1. The structural circuit of Moore PC_AY- FSM

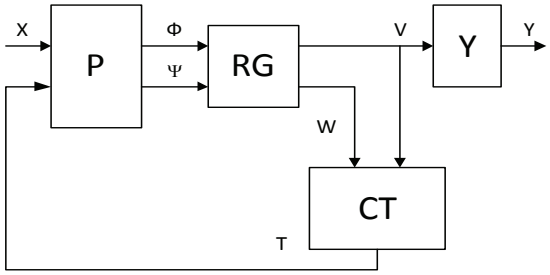

Figure 2. The structural circuit of Moore PC_YY-FSM

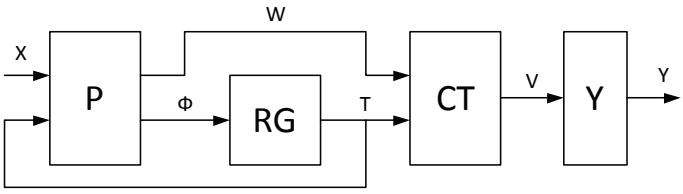

Figure 3. The structural circuit of Mealy PC_AY-FSM

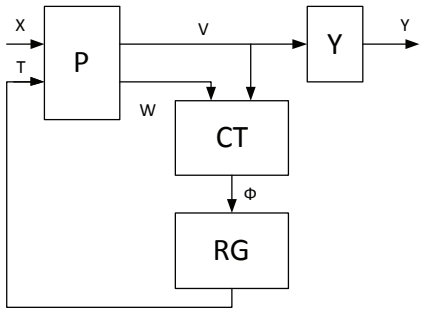

Figure 4. The structural circuit of Mealy PC_YY-FSM

3 Common method of design of FSM with transformation of objects

The method of design for any automaton includes the same steps:

1. Formation of the direct structural table of FSM. In this case an initial information about a control algorithm is represented as flow-chart marked by the states of the corresponding FSM.

2. Determination of the set of variables *I* to the one-by-one identification of the secondary objects by primary ones [4]. The amount of the variables in the set *I* should be minimal one to decrease the total amount of the required outputs of the subcircuit P.

3. Encoding of the sets of microoperations $Y_Z \subseteq Y$ using the elements of the set *V*. Formation of table of microoperations. This step is executed for further design of the transformed DST. The amount of variables in the set *V* should be minimal to minimize the amount of outputs in the subcircuit P.

4. Encoding of the variables $I_k \in I = \{I_1,...,I_K\}$ using the elements of the set $W = \{w_1,\ldots,w_P\}$, where $P =]\log_2 K[$. Such approach permits to minimize the total amount of outputs in the subcircuit P and the amount of the inputs in the code transformer.

5. One-to-one identification of the secondary objects by the codes of prime objects and codes $K(I_K)$ of variables $I_k \in I$.

6. Formation of the transformed DST by excluding of the column of secondary objects and insertion of the column of variables for identification.

7. Formation of the system of functions implemented by subcircuit P using the transformed DSP.

8. Formation of the table of code converter and the system of its functions.

9. Design of the logical circuit of FSM in the given base.

The examples of application of this procedure with details for design of particular structure of FSM can be found in literature [4, 5].

Let us discuss an example of Moore FSM design using Table 1. This table contains T=4 sets of microoperations: $Y_1=\varnothing$, $Y_2=\{y_1,y_2\}$, $Y_3=\{y_3\}$, $Y_4=\{y_3,y_4\}$, and Q=2 variables is sufficient to encode them so $Z=\{z_1, z_2\}$.Let $K(Y_1)=00$, $K(Y_2)=01$, …, $K(Y_4)=11$. Then tables of microoperations and code transformer are given in the tables 2 and 3 respectively.

Functional circuit of FSM is shown on the figure 5. The application of this method is profitable if

$$R > Q. \qquad (2)$$

Table 1. Initial DST of Moore FSM

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Φ_h	h
a_1 (–)	000	a_2	001	x_1	D_3	1
		a_3	010	$\overline{x_1}x_2$	D_2	2
		a_4	011	$\overline{x_1}\overline{x_2}$	$D_2 D_3$	3
a_2 (y_1,y_2)	001	a_5	100	x_3	D_1	4
		a_6	101	$\overline{x_3}$	$D_1 D_3$	5
a_3 (y_3)	010	a_5	100	x_3	D_1	6
		a_6	101	$\overline{x_3}$	$D_1 D_3$	7
a_4 (y_1,y_2)	011	a_5	100	x_3	D_1	8
		a_6	101	$\overline{x_3}$	$D_1 D_3$	9
a_5 (y_3,y_4)	100	a_7	110	x_4	$D_1 D_2$	10
		a_1	000	$\overline{x_4}$	-	11
a_6 (y_1,y_2)	101	a_7	110	x_4	$D_1 D_2$	12
		a_1	000	$\overline{x_4}$	-	13
a_7 (y_3,y_4)	110	a_5	100	x_3	D_1	14
		a_6	101	$\overline{x_3}$	$D_1 D_3$	15

Table 2. Table of microoperations of Moore FSM

Y_t	$K(Y_t)$	y_1	y_2	y_3	y_4	t
Y_1	00	0	0	0	0	1
Y_2	01	1	1	0	0	2
Y_3	10	0	0	1	0	3
Y_4	11	0	0	1	1	4

Table 3. Table of the code transformer of Moore FSM

a_m	$K(a_m)$	Y_t	$K(Y_t)$	Z_m	m
a_1	000	Y_1	00	-	1
a_2	001	Y_2	01	z_2	2
a_3	010	Y_3	10	z_1	3
a_4	011	Y_2	01	z_2	4
a_5	100	Y_4	11	$z_1 z_2$	5
a_6	101	Y_2	01	z_2	6
a_7	110	Y_4	11	$z_1 z_2$	7

4 Optimization of FSM with transformation of the codes of objects

The further hardware optimization is possible by increasing of the number of the circuit's levels and application of method of encoding of the compatible microoperations [6]. Additionally Moore FSM can be

optimized taking into account the sets of pseudoequivalent states [7].

The replacement of logic conditions [1] yields FSM with MP-structure, where set X is replaced by the set $P = \{p_1,\ldots,p_G\}$, $G << L$. The optimization of subcircuit M is possible whether thanks to precise encoding of the states (it is FSM with MC –structure) or thanks to transformation of the codes of the states in the codes of logic conditions (FSM with ML- structure) [7]. In this case an amount of variables in the set P can vary from 1 till G and it is represented by the index g in the type of FSM. Therefore the replacement of logic conditions yields FSM with M_1-, M_1C-, M_1L-,…, M_G-, M_GC-, M_GL-structures.

The encoding of the fields of compatible microoperations yields FSM with PD-structure [1], where the system of microoperations is implemented using decoders. There are J classes of compatible microoperations in the particular FSM. The microoperations of the each class are implemented on the separate decoder. The procedure of verticalization of the control algorithm [6] permits to vary the number of the classes from 1 to J. It yields FSM with PD_1 -,PD_2-,…, PD_J-structure.

The optimization of Moore FSM is possible using [7]:

1.Optimal encoding of pseudoequivalent states that yields the FSM with P_E-structure. In this case the codes of the states for one class of pseudoequivalent states should be included in one generalized interval of Boolean space. It is possible only in limited amount of cases. If such solution exist, it permits to minimize the hardware amount without delay in the time of the cycle of FSM.

2. Transformation of the codes of the states in the codes of the classes of pseudoequivalent states that yields the FSM with P_C-structure. In this case the circuit of FSM includes a special transformer and it permits to compress the length of the DST of Moore FSM up to the length of equivalent Mealy FSM. But in this case the additional hardware is needed to solve the problem of optimization. In this case the optimal solution always exists. This solution is not connected with decreasing of the performance of FSM.

3. Transformation of the initial algorithm by including in it some additional operational nodes that yields FSM with P_Γ-structure. In this case we should insert additional operational node in initial flow-chart. Such approach leads to the increasing of the amount of cycles that are need for execution of algorithm.

All possible structures of the FSM with transformation of the codes of the objects are represented in the Table 4.

Table 4. The structures of the logical circuits of the finite state machines with transformation of the codes of objects

A	B	C
M_1 M_1C M_1L . . . . M_G M_GC M_GL	Mealy FSM: PC_A PC_Y	Y D_1 . . . . D_Y
	Moore FSM: PC_A PC_Y P_EC_A P_EC_Y P_CC_A P_CC_Y $P_\Gamma C_A$ $P_\Gamma C_Y$	

The structures S_i yielding by this table corresponds to the words B*C (double-level structures) или A*B*C (triple-level structures). From table 1 we can form 2(J+1) structures of Mealy FSM of the type B*C, 8(J+1) structures of Moore FSM of the type B*C, 6G(J+1) structures of Mealy FSM of the type A*B*C and 24G*(J+1) structures of Moore FSM of the type A*B*C. Therefore each arbitrary control algorithm can be implemented using

$$n = 2(15G + 5J + 15GJ + 5) \quad (3)$$

structures of the logical circuit of FSM with transformation of the codes of the objects.

For the FSM of middle complexness G=J=6 [1], therefore expression (2) determines n=1330 different structures. The particular structure S_i is set whether by the formula B*C or by the formula A*B*C. For example, formula $S_1=M_2LP_EC_AD_3$ determines the Moore FSM with optimal encoding of the states, transformation of the codes of the states in the codes of the logic conditions, replacement of logic conditions by two additional variables, three classes of compatible microoperations and transformation of the codes of the states in the codes of the sets of microoperations.

As we can see there are many possible solutions for the same problem. It means that there is a problem to compare the different proposed solutions and well-known solutions. In other words we should find the effectiveness of any proposed method. One of the approaches to solution of this problem is to compare the results of different methods' application for some test examples. But such approach permits to get decision only for some points of the design space and does not show the picture in common. More fruitful approach proposed by the authors is based on the probabilistically representation of the characteristics of an algorithm of control. Second point here is to find not absolute but relative characteristics to compare the different methods of design. And last point here is to use customized VLSI instead of standard PLD. It is known that relative estimation of hardware amount for two different structures of FSM is approximately the same for implementation of its circuits using standard and customized PLDs.

5 Conclusions

The method of transformation of the codes of objects permits to decrease the hardware amount in the circuit used for formation of the excitations functions of FSM. An application of this method is reasonable if total cost of subcircuit P and code transformer is less then cost of subcircuit P in the FSM with PY-structure. This cost can be estimated weather as money or as amount of chips in the circuit.

Transformation of the codes of the states in the codes of the sets of microoperations leads to the increasing of the latency time of FSM, therefore this method can be applied only when the criteria of the effectiveness of the FSM is the minimal cost. The researchers conducted by the author shown that application of the proposed approach permits to decrease the hardware amount up to 17-22% to compare with FSM SM with PY-structure.

References

[1] Baranov S. Logic and System Design of Digital Systems. – Tallinn: TUT Press, 2008. – 328 pp.

[2] Barkalov A.A., Titarenko L.A. Logic synthesis for FSM-based control units. – Berlin: Springer, 2009. – 233 pp.

[3] Соловьёв В.В, Климович А. Логическое проектирование цифровых систем на основе программируемых логических интегральных схем. – М.: Горячая линия – Телеком, 2008 – 376 с.

[4] Баркалов А.А., Баркалов А.А. Оптимизация схем автоматов Мура с кодированием наборов микроопераций / В кн. "Автоматизация проектирования дискретных систем". Материалы 4-й Международной конференции (14-16.11.2001, Минск). Том 2. – Минск: ИТК НАН Беларуси, 2001.– с.14-19.

[5] Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления.– Киев: ИК НАН Украины, 1997 – 135 с.

[6] Баркалов А.А. Принцип оптимизации логической схемы микропрограммного автомата Мура // Кибернетика и системный анализ. – 1998, №1. – с.65-72.

[7] Barkalov A., Barkalov A. Design of Mealy finite-state-machines with transformation of object codes // International Journal of Applied Mathematics and Computer Science. – 2005, V.15, №1.– pp.151-158.