Розробка засобів обчислювальної техніки та дослідження комп'ютерних мереж

УДК 004.274

A.A. Barkalov[1], professor,
L.A. Tytarenko[1], professor,
I.J. Zeleneva[2], PhD,
Kamil Mielcarek, [1]
[1]University of Zielona Gora,Poland,
[2]Zaporozgye National Technikal University, Ukraine
e-mail: A.Barkalov@iie.uz.zgora.pl
Irina.zeleneva@gmail.com

Design FSM Mealy with transformation of object codes based on embedded memory blocks

A design method is proposed for EMB-based Mealy FSM. The method is based on transformation of collections of output functions into state variables. Only two EMBs are necessary in the best case to implement the FSM logic circuit. The conditions of applying proposed method are given. An example of design is shown.

Key words: Mealy finite-state machine, structure table, design, FPGA, embedded memory block.

Introduction

The control unit of any digital system can be implemented using the model of Mealy finite state machine [1, 2]. Today the field programmable gate arrays (FPGA) are widely used for implementing the logic circuits of FSMs [3, 4]. One of the main problems connected with implementing logic circuits of FSMs is the problem of hardware reduction [5]. The solution of this problem leads to decreasing of such parameters as the chip area occupied by an FSM circuit, its propagation time and the power consumption [3].

To implement the logic circuit of a Mealy FSM, it is enough to use three components of an FPGA chip, such a logic elements (LE), embedded memory blocks (EMB) and a matrix of programmable interconnections. A logic element includes a look-up table (LUT) element and a programmable flip-flop [6, 7]. The flip-flop of LE could be bypassed, so the output of a LUT can be either registered or combinational. A LUT can be viewed as a random-access memory (RAM) device having up to 6 address inputs [6, 7]. A single LUT can represent a truth table of an arbitrary Boolean function having up to 6 arguments. If the number of arguments exceeds the number of address inputs, then more than one LE is necessary to implement a corresponding logic circuit. It leads to increasing for the number of logic layers in the final circuits and to complication for interconnections. In turn, it results in the increasing for the propagation time and power consumption [8]. To improve these parameters, some parts of an FSM circuit should be implemented using EMBs [8-11].

In this article we propose a design method for the EMB-based Mealy FSMs. The method is based on the transformation of object codes [3, 9].

Background of Mealy FSMs and relative works

Let a control algorithm of a digital system be specified by a graph-scheme of algorithm (GSA) Γ [1]. A GSA contains the initial (Start) vertex, the final (End) vertex, operator and conditional vertices. An operator vertex contains a collection of output functions $Y_t \subseteq Y$, where $Y=\{y_1,...,y_N\}$ is a set of output functions. A conditional vertex include one element of the set of logical conditions $X=\{x_1,...,x_L\}$. The rules [1] are used for creating the set of internal states $A=\{a_1,...,a_M\}$. Let us encode states $a_m \in A$ by binary codes $K(a_m)$ having R bits, where

$$R=\lceil \log_2 M \rceil. \qquad (1)$$

The states are encoded using state variables $T_r \in T$, where $T=\{T_1,...,T_R\}$.

A structure table can be used for representing the behaviour of FSM [1]. It includes the columns a_m, $K(a_m)$, a_s, $K(a_s)$, X_h, Y_h, Φ_h, h. Here a_m is a current state; $K(a_m)$ is a code of state $a_m \in A$; a_s is a state of transition (next state); $K(a_s)$ is a code of state $a_s \in A$; X_h is an input signal determining the transition from a_m into a_s; Y_h is a collection of output functions equal to 1 for the transition $<a_m, a_s>$; Φ_h is a subset of the set of input memory functions $\Phi=\{D_1,...,D_R\}$ equal to 1 to transform the code $K(a_m)$ into the code $K(a_s)$; $h=\overline{1,H}$ is a number of transitions.

The structure table is a base for deriving the following systems of Boolean functions:

$$Y=Y(T,X), \qquad (2)$$

$$\Phi= \Phi(T,X). \qquad (3)$$

The EMBs of up-to-day FPGAs have the property of configurability allowing the changing of such parameters as the numbers of cells (V) and outputs (t_F) [6,7]. Here are the following typical configurations of modern EMBs: 16Kx1, 8Kx2, 4Kx4, 2Kx8, 1Kx16 and 512x32 (bits) [6,7]. It means that the EMBs are very flexible. They can be tuned to meet demands of a particular design. Let an EMB contains V cells and t_F outputs. Let V_0 be a number of cells if t_F=1. The number V can be found as

$$V=\lceil V_0/t_F \rceil. \qquad (4)$$

Let the following condition takes place:

$$2^{L+R}(R+N)\leq V_0. \qquad (5)$$

In this case, a Mealy FSM is implemented in a trivial way [10] using only a single EMB (Fig.1a). As a rule, such model is named P FSM [3]. Here the EMB implements systems (2)-(3).

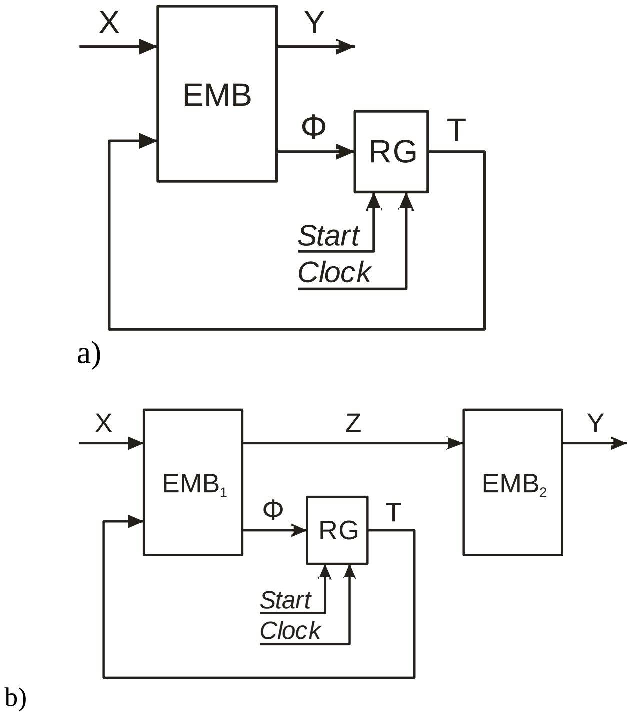

Fig.1. Structural diagram of P (a) and PY (b) Mealy FSM

All discussed FSMs use a register RG to keep the state codes. The RG includes R of D flip-flops [3, 4]. A pulse Start causes loading the zero code of the initial state $a_1 \in A$ into RG. A pulse Clock allows changing the content of RG.

The P FSM needs only a single EMB. It is the fastest solution. But it is possible only if the condition (5) takes place. If this condition is violated, then different methods of structural decomposition [3, 4] can be used for representing an FSM circuit. Let us discuss the method of encoding of collection of outputs functions.

Let T_0 different collections of output functions $Y_t \subseteq Y$ be written in operator vertices of GSA Г. Let us encode each collection Y_t by a binary code $C(Y_t)$ having R_Y bits:

$$R_Y=\lceil \log_2 T_0 \rceil. \qquad (6)$$)

Let us use variables from the set $Z=\{z_1,...,z_{RY}\}$ for encoding.

This approach leads to PY Mealy FSM (Fig.1b). Here the EMB1 implements the system (3) together with a system

$$Z=Z(T,X). \qquad (7)$$

The EMB2 implements the system of output functions represented as

$$Y=Y(Z). \qquad (8)$$

The PY FSM can be used if the following conditions take place:

$$2^{L+R}\cdot(R+R_Y)\leq V_0, \qquad (9)$$

$$2^{RY}\cdot N\leq V_0. \qquad (10)$$

Of course, this model can be used if the condition (5) is violated.

If the condition (5) is violated then different approaches based on the replacement of logical conditions can be used [13, 14]. We do not discuss these models in our article. We propose to use the transformation of codes $C(Y_t)$ into codes $K(a_m)$ based on results from [9].

Main idea of proposed method

In [9], states $a_m \in A$ and collections $Y_t \subseteq Y$ are named objects of Mealy FSMs. Let us represent states as some functions of collections $Y_t \subseteq Y$. To understand the principle, let us discuss the following example taken from [9].

Let a Mealy FSM S_1 be represented by the structure table (Table 1). The FSM has the following characteristics: $X=\{x_1,\ldots,x_4\}$, $Y=\{y_1,\ldots,y_7\}$, $A=\{a_1,\ldots,a_5\}$, M=5, L=4, N=7 and H=12.

Table 1 - Structure table of Mealy FSM S_1

a_m	$K(a_m)$	a_s	$K(a_s)$	X_h	Y_h	Φ_h	h
a_1	000	a_2	010	x_1	y_1y_2	D_2	1
		a_3	011	$/x_1$	y_3	D_2D_3	2
a_2	010	a_2	010	x_2	y_1y_2	D_2	3
		a_3	011	$/x_2x_3$	y_4	D_1D_3	4
		a_4	100	$/x_2/x_3$	y_1y_2	D_1	5
a_3	011	a_4	100	x_1	y_2y_5	D_1	6
		a_5	101	$/x_1$	y_6	D_1D_3	7
a_4	100	a_5	101	1	y_3y_7	D_1D_3	8
a_5	101	a_2	010	x_2x_3	y_1y_2	D_2	9
		a_3	011	x_2/x_3	y_3	D_2D_3	10
		a_5	101	$/x_2x_4$	y_3y_7	D_1D_3	11
		a_2	000	$/x_2/x_4$	-	-	12

Analysis of Table 1 gives us T_0=7 different collections of output functions: $Y_1=\varnothing$, $Y_2=\{y_1,y_2\}$, $Y_3=\{y_3\}$, $Y_4=\{y_4\}$, $Y_5=\{y_2,y_5\}$, $Y_6=\{y_6\}$, $Y_7=\{y_3,y_7\}$. Next, the collection Y_1 is generated only if the state a_1 is a state of transitions. But the collection Y_2 is generated during the transitions into states a_2 (lines 1, 3, 9) and a_4 (line 5).

It means that some identifier I_1 is necessary to determine an exact state of transition. If $I_1=0$, for example, the next state is equal a_2, otherwise ($I_1=1$) it is the state a_4.

Let it be enough K of identifiers to determine any state of transition. Let us construct the set of identifiers $I=\{I_1,\ldots,I_K\}$. Let’s encode an identifier $I_k\in I$ by a binary code $C(I_k)$ having R_I bits:

$$R_I=\lceil log_2 K \rceil. \quad (11)$$

Let us use the elements of the set $W=\{w_1,\ldots,w_{RI}\}$ for the encoding. In this case we propose the structural diagram of EMB-based $P_{YA}Y$ Mealy FSM (Fig.2).

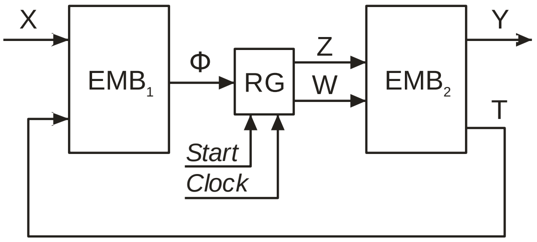

Fig.2. Structural diagram of EMB-based $P_{YA}Y$ Mealy FSM

In $P_{YA}Y$ FSM, the EMB1 generates the functions (3), the EMB2 the functions

$$Y=Y(Z,W), \quad (12)$$

$$T=T(Z,W). \quad (13)$$

This model can be used if the following conditions take places:

$$2^{L+R}\cdot(R_Y+R_I)\leq V_0, \quad (14)$$

$$2^{L+1+R}\cdot(R_Y+R_I)>V_0, \quad (15)$$

$$2^{RY+RI}\cdot(R+N)\leq V_0. \quad (16)$$

The conditions (14)-(15) show that the method can be used only if there is

$$R_Y+R_I=R. \quad (17)$$

If (17) is violated, then it is necessary more than a single EMB to implement the system (3). The condition (16) shows that systems (12)-(13) can be implemented using a single EMB.

In this article we propose a design method for $P_{YA}Y$ Mealy FSMs.

Proposed design method

The proposed design method includes the following steps:

1. Constructing the set of states A for a given GSA Γ.
2. State assignment.
3. Constructing the structure table of Mealy FSM.
4. Constructing the set of collections of output functions.
5. Encoding of collections $Y_t\subseteq Y$.
6. Constructing the set of identifiers I.
7. Encoding of identifiers $I_K\in I$.
8. Creating the table of EMB1.
9. Creating the table of EMB2.
10. Implementing FSM logic circuit using EMBs of a particular FPGS chip.

Let us discuss the application of the proposed method in the case of Mealy FSM S_1. Let us point out that the steps 1-4 are already executed.

In the common case, the step 1 is executed using the rules from [1]. The state a_1 is used to mark the output of the initial vertex, as well as the input of the final vertex. An unique state a_m is used to mark the input of any vertex following an operator vertex.

There are a lot of state assignment algorithms [3,5] targeting the optimization of FSM logic circuits. But there is no influence of this step’s outcome on the hardware amount of $P_{YA}Y$ FSM. In any case, its circuit includes two EMBs. The structure table is constructed using the rules from [1].

As it is found before, there is $T_0=7$. Therefore, it is necessary $R_Y=3$ variables for encoding of collections $Y_t\subseteq Y$. It follows from [5]. So, there is the set $Z=\{z_1,z_2,z_3\}$. The codes $C(Y_t)$ do not affect the number of EMBs using for generating the systems (12)-(13). Because of it, let us encode the collections in the trivial way: C(Y1)=000, $C(Y_2)=001,\ldots$, $C(Y_7)=110$.

Let $A(Y_t)$ be a set of states such that the collection $Y_t\subseteq Y$ is generated under the transition into the state $a_m\in A(Y_t)$. Obviously, there is the relation $A(Y_t)\subseteq A$. It is enough to use $m_t=|A(Y_t)|$ identifiers to distinct any element of $A(Y_t)$. Let us find the value of K, where

$$K=max(m_1,\ldots,m_{T0}). \quad (18)$$

The value from (18) is used in the expression (11).

In the discussed case, the following sets $A(Y_t)$ are found: $A(Y_1)=\{a_1\}$, $A(Y_2)=\{a_2,a_4\}$, $A(Y_3)=\{a_3\}$, $A(Y_4)=\{a_4\}$, $A(Y_5)=\{a_4\}$, $A(Y_6)=\{a_5\}$ and $A(Y_7)=\{a_5\}$. Therefore, there is K=2 (because of $|A(Y_2)|=2$). It gives the value of $R_I=1$ and the set $W=\{w_1\}$.There is the set $I=\{I_1,I_2\}$. Let us use the following codes of identifiers: $C(I_1)=0$ and $C(I_2)=1$.

Let us form a table reflecting the correspondence between states, collections and identifiers for the FSM S_1 (Table 2).

Table 2 - Correspondence between states, collections and identifiers for $P_{YA}Y$ Mealy FSM S_1

h	1	2	3	4	5	6
$\mathbf{a_s}$	a_2	a_3	a_2	a_3	a_4	a_4
$\mathbf{Y_t}$	Y_2	Y_3	Y_2	Y_4	Y_2	Y_5
$\mathbf{I_k}$	I_1	*	I_1	*	I_2	*
h	7	8	9	10	11	12
$\mathbf{a_s}$	a_5	a_5	a_2	a_3	a_5	a_1
$\mathbf{Y_t}$	Y_6	Y_7	Y_2	Y_3	Y_7	Y_1
$\mathbf{I_k}$	*	*	I_1	*	*	*

Table 2 is constructed on the base of Table 1. It contains numbers of rows (the first line), states of transition (second line), collections of output functions (the third line) and identifiers (the fourth line). If a set $A(Y_t)$ contains only a single state, that the corresponding cell of Table 2 includes the sign of asterisk. It corresponds to “don’t care” value of an identifier.

Using Table 2, the transformed ST can be constructed for $P_{YA}Y$ Mealy FSM. In the discussed case it is Table 3. The table includes the following columns: a_m, $K(a_m)$, X_h, Z_h (variables $z_r \in Z$ which are equal to 1 in the code $C(Y_t)$ for the column h of the table of correspondence), W_h (variables $w_r \in W$ which are equal to 1 in the code $C(I_k)$ for the column h of the table of correspondence), h.

Table 3 - Transformed ST of $P_{YA}Y$ Mealy FSM S_1

a_m	$K(a_m)$	X_h	Z_h	W_h	h
a_1	000	x_1	z_3	*	1
		$/x_1$	z_2	*	2
a_2	010	x_2	z_3	*	3
		$/x_2x_3$	z_2z_3	*	4
		$/x_2/x_3$	z_3	w_1	5
a_3	011	x_1	z_1	*	6
		$/x_1$	z_1z_3	*	7
a_4	100	1	z_1z_2	*	8
a_5	101	x_2x_3	z_3	-	9
		x_2/x_3	z_2	*	10
		$/x_2x_4$	z_1z_2	*	11
		$/x_2/x_4$	-	*	12

In the discussed case there is $R_Y=3$ and $R_I=1$. So, the set Φ should include 4 elements: $\Phi=\{D_1,\ldots D_4\}$. Let the first flip-flop of RG correspond to z_1, the second to z_2, the third to z_3 and the fourth to w_1. If EMBs are synchronized in a particular FPGA chip, that the register should be eliminated from the structural diagram of $P_{YA}Y$ Mealy FSM. We do not discuss this case in our article. Let us denote the corresponding model as $P_{YA}Y_R$ Mealy FSM (Fig.3).

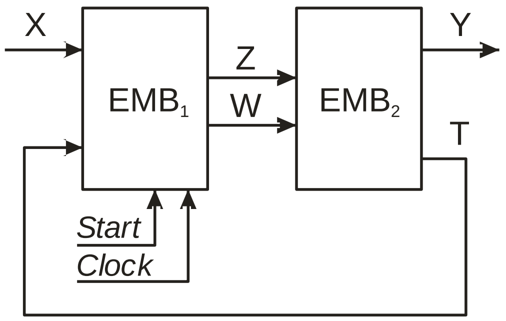

Fig.3. Structural diagram of $P_{YA}Y_R$ Mealy FSM

The table of EMB1 includes the columns $K(a_m)$, $K(X_h)$, Φ_h, h_1, h where $h_1=\overline{1,2^{R+L}}$. The columns $K(a_m)$ and $K(X_h)$ form an address of a cell, whereas the column Φ_h determines its content. Transitions from each state $a_m \in A$ are represented by 2^L rows of the table. For example, the transitions from state a_2 are shown in Table 4.

Table 4 - Part of table of EMB1 for $P_{YA}Y$ Mealy FSM S_1

$K(a_m)$			$K(X_h)$				Φ_h				h_1	h
T_1	T_2	T_3	x_1	x_2	x_3	x_4	D_1	D_2	D_3	D_4		
0	1	0	0	0	0	0	0	0	1	1	33	5
0	1	0	0	0	0	1	0	0	1	1	34	5
0	1	0	0	0	1	0	0	1	1	0	35	4
0	1	0	0	0	1	1	0	1	1	0	36	4
0	1	0	0	1	0	0	0	0	1	0	37	3
0	1	0	0	1	0	1	0	0	1	0	38	3
0	1	0	0	1	1	0	0	0	1	0	39	3
0	1	0	0	1	1	1	0	0	1	0	40	3
0	1	0	1	0	0	0	0	0	1	1	41	5
0	1	0	1	0	0	1	0	0	1	1	42	5
0	1	0	1	0	1	0	0	1	1	0	43	4
0	1	0	1	0	1	1	0	1	1	0	44	4
0	1	0	1	1	0	0	0	0	1	0	45	3
0	1	0	1	1	0	1	0	0	1	0	46	3
0	1	0	1	1	1	0	0	0	1	0	47	3
0	1	0	1	1	1	1	0	0	1	0	48	3

In the discussed case L=4, so it is necessary 16 rows to represent transitions from each state of FSM S_1. So, Table 4 starts from $h_1=33$. There are numbers of lines of Table 3 into the column h. The column $K(X_h)$ is a vector of logical conditions $<x_1,...,x_L>$. Let us explain the filling of Table 4.

As follows from a row h=3, there is $X_3=X_2$. So, there are eight different vectors <*,1,*,*> corresponding to row 3 in Table 4. They are the rows 37-40 and 45-48. All these rows contain $D_3=1$ corresponding to $z_3=1$ (row 3 of Table 3). All other rows are filled in the same manner.

The table of EMB2 is constructed on the base of the initial ST, table of correspondence and codes of collections and identifiers. It has $H=2^{RY+RI}$ rows. The tables includes the columns $C(Y_t)$, $C(I_k)$, Y_h, $K(a_m)$, h. In the discussed case, it has H=16 rows (Table 5).

For example, the address 0010 corresponds to the collection $Y_2=\{y_1,y_2\}$ and to the identifier I_1. From Table 2, we can find that $<Y_2,I_1>$ corresponds to the state a_2. So the cell with address 0010 contains 1’s for the outputs y_1,y_2 and T_2.Using the same approach, it can be found the content for all cells of EMB2.

The last step of proposed method is connected with using VHDL models of FSMs and standard CAD tools [4]. We do not discuss this step in our article.

Analysis of proposed method

At the start of analysis of the discussed example let us to repeat the characteristics of the Mealy FSM S_1: L=4, N=7, R=3, $R_Y=3$ and $R_I=1$. Let’s use the FPGA chip including EMBs with the following configurations: 512x1, 256x2, 128x4, 64x8 and 32x16.

Table 5 - Table of EMB2 for $P_{YA}Y$ Mealy FSM S1

C(Y$_t$)			**C(I$_k$)**	**Y$_h$**							**K(a$_m$)**			**h**
z$_1$	**z$_2$**	**z$_3$**	**w$_1$**	**y$_1$**	**y$_2$**	**y$_3$**	**y$_4$**	**y$_5$**	**y$_6$**	**y$_7$**	**T$_1$**	**T$_2$**	**T$_3$**	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	2
0	0	1	0	1	1	0	0	0	0	0	0	1	0	3
0	0	1	1	1	1	0	0	0	0	0	1	0	0	4
0	1	0	0	0	0	1	0	0	0	0	0	1	1	5
0	1	0	1	0	0	1	0	0	0	0	0	1	1	6
0	1	1	0	0	0	0	1	0	0	0	0	1	1	7
0	1	1	1	0	0	0	1	0	0	0	0	1	1	8
1	0	0	0	0	1	0	0	1	0	0	1	0	0	9
1	0	0	1	0	1	0	0	1	0	0	1	0	0	10
1	0	1	0	0	0	0	0	0	1	0	1	0	1	11
1	0	1	1	0	0	0	0	0	1	0	1	0	1	12
1	1	0	0	0	0	1	0	0	0	1	1	0	1	13
1	1	0	1	0	0	1	0	0	0	1	1	0	1	14
1	1	1	0	0	0	0	0	0	0	0	0	0	0	15
1	1	1	1	0	0	0	0	0	0	0	0	0	0	16

There is R+L=7, so it is necessary 128 cells to keep all rows of ST for S_1. Each cell should have t(P)=10 outputs. So, the condition (5) is violated. To implement the logic circuit of P FSM S_1, it is necessary h(P)=3 EMBs having the configuration 128x4. The number n(P) is obtained using the following formula:

$$n(P)=\lceil t(P)/t_F \rceil. \qquad (19)$$

There is $t(PY)=R+R_Y=6$. It means that it is necessary to use the configuration 128x6 for implementing the EMB1 of PY Mealy FSM S_1. We can find the number of EMBs for EMB1 using the following formula:

$$h(PY)=\lceil t(PY)/t_F \rceil. \qquad (20)$$

There is n(PY)=2 in the discussed case.

There is N=7, so we can choose the configuration 32x16 (or 64x8) to implement the EMB2. In both cases we will set n(Y)=1

$$n(Y)=\lceil N/t_F \rceil. \qquad (21)$$

So, it is necessary 3 EMBs for implementing the logic circuit of PY FSM S_1.

It is necessary a single EMB 128x4 to implement the block EMB1 of the $P_{YA}Y$ Mealy FSM S_1. It follows from the equalities R+L=7 and $R_Y+R_I=4$. There is R+N=10, so the EMB 32x16 can be used to implement the block EMB2 of the $P_{YA}Y$ FSM S_1.

So, the proposed method allows decreasing for the number of EMBs used for implementing the logic circuit of S_1. It requires 50% less of EMBs. Moreover, the number of interconnections is decreased, too. It means the decreasing for the propagation time and consumed power. Of course, it is true only for the FSM S_1 and EMBs used for implementing its circuit.

We investigate the benchmarks of the library [11, 12] to find out when the proposed method can be applied.

Conclusion

The proposed method is based on transformation of collections of output functions into the state variables of Mealy FSM. In some cases, it allows reducing the number of EMBs in the circuit of FSM in comparison with known methods (P FSM and PY FSM). The level of reduction depends on characteristics of both FSM and EMB.

If condition (14) is violated, then more than one EMB is necessary to implement the circuit of EMB1. To diminish this number, they can use the method of replacement of logical conditions [13, 14].

References

1. S. Baranov. Logic and System Design of Digital Systems. - Tallinn: TUT Press, 2008. - 276 pp.
2. R. Czerwiński, D. Kania. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. - Berlin: Springer, 2013. - 172 pp.
3. A. Barkalov, L. Titarenko. Logic Synthesis for FSM-based Control Units. - Berlin: Springer, 2009. - 234 pp.
4. S. Sklarov, I. Sklarova, A. Barkalov, L. Titarenko. Synthesis and Optimization of FPGA-based Systems. - Berlin: Springer, 2014. - 432 pp.
5. G. De Micheli. Synthesis and Optimization of Digital Circuits. - New York: McGraw-Hills, 1994. – 327 pp.
6. www.xilinx.com
7. www.altera.com

8. J. Cong, K. Yan. Synthesis for FPGAs with embedded memory blocks. - In: Proceedings of the 2000 ACM - SIGDA 8th International Symposium on FPGAs. - 2000, pp. 75-82.
9. A. Barkalov, A. Barkalov, Jr. Design of Mealy finite-state machines with the transformation of object codes // Int. Journal of Applied Mathematics and Computer Science. - 2005, V.15, № 1. - pp. 151-158
10. I. Garcia-Vargas, R. Senhadji-Navarro, A. Civit-Balcells, P. Guerra-Gutierrezz. ROM-based finite state machine implementation in low cost FPGAs. - In: IEEE International Symposium of Industrial electronics. - Vigo 2007. - pp. 2342-2347.
11. Yang S. Logic synthesis and optimization bench marks user guide. Technical report. – Microelectronic Center of North Carolina, 1991. – 44pp.
12. Баркалов А.А, Зеленева И.Я., Мирошкин А.Н., Товстоног А.А. Автоматизация процесса проектирования цифровых устройств управления
Известия ТТИ ЮФУ-ДонНТУ. Материалы Четырнадцатого Международного научно-практического семинара «Практика и перспективы развития партнерства в сфере высшей школы» В 3х кн. – Таганрог. Изд-во ТТИ ЮФУ. Кн. 2. 2013,– С. 10-12.
13. Баркалов А.А., Зеленёва И.Я., Мирошкин А.Н. Замена логических условий в композиционном микропрограммном устройстве управления с разделением кодов. \\ Материалы Десятого Международного научно-практического семинара «Практика и перспективы развития партнерства в сфере высшей школы». В 2-х кн. – ДонНТУ. Кн.1. 2009.- С. 172-180.
14. Barkalov A.A., Tytarenko L.A., Efimenko K.N., Zeleneva I.J. Optimization of the Addressing Scheme in Compositional Microprogram Control Unit with Code Sharing//Наукові праці ДонНТУ. Серія «Інформатика, кібернетика і обчислювальна техніка» (ІКОТ) №1(17). – Донецьк: ДонНТУ, 2013. – С. 5-10.

Надійшла до редакції 10.03.2015

О.О. БАРКАЛОВ[1], Л.О. ТИТАРЕНКО[1], І.Я. ЗЕЛЕНЬОВА[2], К. МЕЛЧАРЕК[1]
[1]Університет Зеленогурський (Польща)
[2]ДВНЗ «Запорізький національний технічний університет» (Україна)

СИНТЕЗ FSM МІЛІ ІЗ ПЕРЕТВОРЕННЯМ КОДІВ ОБ’ЄКТІВ НА БАЗІ ВБУДОВАНИХ БЛОКІВ ПАМ’ЯТІ

В роботі запропоновано метод синтезу кінцевого автомату Мілі, орієнтований на використання вбудованих блоків пам’яті мікросхем FPGA. Метод заснований на перетворенні наборів вихідних функцій у змінні станів. Метод забезпечує в кращому випадку використання лише двох блоків вбудованої пам’яті для реалізації логічної схеми автомата. Розглянуті умови застосування даного методу та наведено приклад синтезу.

Ключові слова: кінцевий автомат Мілі, структурна таблиця, синтез, FPGA, вбудований блок пам’яті.

А.А. БАРКАЛОВ[1], Л.А. ТИТАРЕНКО[1], И.Я. ЗЕЛЕНЕВА[2], К. МЕЛЧАРЕК[1]
[1]Университет Зеленогурский (Польша)
[2]ДВНЗ «Запорожский национальный технический университет» (Украина)

СИНТЕЗ FSM МИЛИ С ПРЕОБРАЗОВАНИЕМ КОДОВ ОБЪЕКТОВ НА БАЗЕ ВСТРОЕННЫХ БЛОКОВ ПАМЯТИ

В работе предложен метод синтеза конечного автомата Мили, ориентированный на использование встроенных блоков памяти микросхем FPGA. Метод основан на преобразовании наборов выходных функций в переменные состояний. Метод обеспечивает в лучшем случае использование только двух блоков встроенной памяти для реализации логической схемы автомата. Рассмотрены условия применения данного метода и приведен пример синтеза.

Ключевые слова: конечный автомат Мили, структурная таблица, синтез, FPGA, встроенный блок памяти.