Adam KLIMOWICZ
BIALYSTOK UNIVERSITY OF TECHNOLOGY
45 A Wiejska St., 15-351 Bialystok, Poland

Speed Targeted Minimization of Finite State Machines for CPLDs

Abstract

A method of the minimization finite state machines (FSM) is proposed. In this method, such optimization criterion as the critical delay path is taken into account already at the stage of minimizing internal states. The method is based on sequential merging of two internal states including the optimization criteria. The critical path is estimated for CPLD devices. In addition, the proposed method allows one to minimize the number of transitions and input variables of the FSM. Experimental results shows, that the maximum clock frequency of minimized FSMs is higher by 17% comparing to initial FSM.

Keywords: finite state machines, minimization, logic synthesis, performance, speed.

1. Introduction

The high-speed performance of the electronic system directly depends on performance of the main control unit (system clock frequency) and also on the control units of the separate parts of the system. Generally the control unit represents the sequential circuit which mathematical model is the finite state machine (FSM). The most perspective approach to increase the performance of the electronic equipment is an application of the synthesis methods allowing to build FSMs and control units with the maximum high-speed performance. The methods of a logic synthesis can be used for any technological basis, can be applied together with system methods, and do not depend on supply voltage.

The problem of the simultaneous minimization of area and signal delay on the critical path is considered in [1-4]. In [1], a structural model of the FSM called MAR model is proposed, which consists of an FSM and a combinational circuit with flip-flops in the feedback loops. In [2], codes with two unities (two-hot) and three unities (three-hot) are used. In [3], the minimization of power consumption and delay is considered for asynchronous FSMs. The concept of a low power semi-synchronous FSM operating on a high frequency is proposed that can be implemented and tested as an ordinary synchronous FSM. In [4], a two-level structural model is proposed to minimize the power consumption, area and delay. The first level of this model consists of sequential units, while the second level consists of combinational units of limited size.

The paper [5] proposes a timing optimization technique for a complex finite state machine that consists of not only random logic but also data operators. The proposed technique, based on the concept of catalyst, adds a functionally redundant block – which includes a piece of combinational logic and several other registers - to the circuits under consideration so that the timing critical paths are divided into stages.

The paper [6] proposes to use the evolutionary methodology to yield optimal evolvable hardware that implements the state machine control component. The evolved hardware requires a minimal hardware area and introduces a minimal propagation delay of the machine output signals. The paper [7] is concerned with the problem of state assignment and logic optimization of high speed finite state machines. The method is designed for PAL-based CPLD (*Complex Programmable Logic Devices*) implementation.

The analysis of available studies showed that there are only small number of works in which the number of internal states and the speed of FSM are simultaneously minimized. In this paper, we propose an heuristic state minimization method based on an operation of two states merging. In this method, such optimization criteria as the speed (critical delay path) and possibility of merging other states are taken into account already at the stage of minimizing internal states. In addition to reduction of internal states this method minimizes the number of FSM transitions and FSM input variables. Presented method takes in consideration parameters of target CPLD device for estimation of FSM speed. Similar method for FPGA devices was presented in [8].

2. Preliminaries

Let us denote by L the number of FSM input variables of a set $X = \{x_1,\ldots,x_L\}$, by N the number of FSM output variables of a set $Y = \{y_1,\ldots,y_N\}$, by M the number of FSM internal states of a set $A = \{a_1,\ldots,a_M\}$, and by R the minimal number of bits required to encode internal states, where $R = \text{intlog2}M$.

A FSM behavior can be described by the transition list. The transition list is a table with four columns: a_m, a_s, $X(a_m, a_s)$, and $Y(a_m, a_s)$. Each row of the transition list corresponds to one FSM transition. The column a_m contains a present state, the column a_s contains a next state, the column $X(a_m, a_s)$ contains a transition condition (an input vector), and the column $Y(_{am}, a_s)$ contains an output vector. A FSM output vector is represented by ternary vector. For example, $Y(a_m, a_s)$="01-0", where 0 denotes zero value, 1 denotes unity value, and dash ("-") denotes a don't care value of the corresponding output variable.

The transition condition may be described in the column $X(a_m, a_s)$ in the form of conjunction of FSM input variables. The transition condition can also be represented by a ternary vector. Since the FSM behaviour is deterministic, all the transition conditions from every FSM state should be mutually orthogonal. Two transition conditions are orthogonal if they have different significant values (0 or 1) at least in one position.

Two FSM states a_i and a_j can be merged, i.e. replaced by one state a_{i_j}, if they are equivalent. Equivalency of two FSM states means that FSM behaviour does not change when these states are merged in one. FSM behaviour does not change after merging, if the transition conditions from the states a_i and a_j that lead to different states are orthogonal. If there are transitions from states a_i and a_j that lead to the same unique state, then the transition conditions for such transitions should be equal. Moreover, the output vectors that are generated at these transitions should be not orthogonal. Note also that in case of two FSM states merging wait states can be formed. The detailed conditions of the states merging procedure are precisely described in paper [9].

3. Minimization algorithm

The main strategy of the proposed method consists in finding the set G of all the pairs of FSM states satisfying the merging conditions. Then for each pair of states from set G the trial merging is performed. Finally a pair (a_i, a_j) is selected for merging in such a way that leaves the maximal possibilities for other pairs of FSM states merging. This process repeats as long as there exists a possibility of merging for at least one pair of FSM states. The method was described more precisely in paper [9].

In distinction from [13], in the present paper we chose for merging at each step the pair (a_i, a_j) that best satisfies the optimization criteria in terms of speed, and leaves the maximum possibilities for merging other pairs of states. This procedure is repeated while at least one pair of states can be merged.

Let (a_s, a_t) be a pair of states in G, where S_{st} is the estimate of speed (critical delay path), and M_{st} is the estimate of the possibility to merge other states. Then, with regard to the above considerations, the Fig. 1 presents the main FSM minimization algorithm.

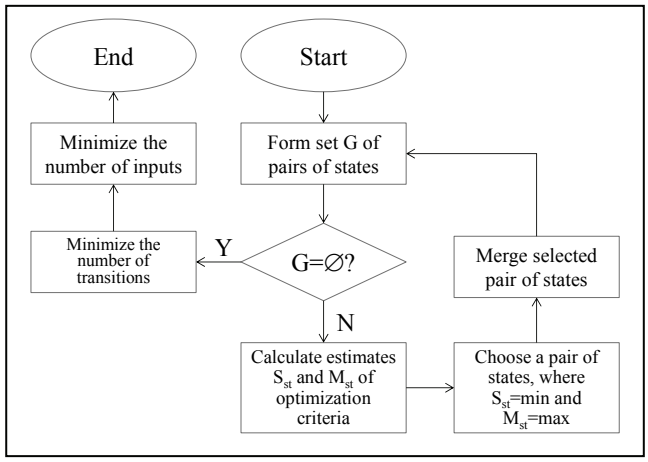

Fig. 1. Main FSM minimization algorithm

Merging of the states, minimization of the number of transitions and minimization of the number of input variables are performed as described in [9].

4. Optimization criteria

To estimate the optimization criteria, all pairs of states in G are considered one after another. For each pair of states (a_s, a_t) in G, a trial merging is performed. For the resultant FSM, its internal states are encoded using one of the available methods that will later be used in the synthesis of the FSM, and the system of Boolean functions corresponding to the combinational part of the FSM is built. Next, for the pair (a_s, a_t), the speed S_{st}, and the possibility of minimizing other states M_{st} are estimated. The optimization criteria for each pair of states (a_s, a_t) in G are estimated using the algorithm presented in Fig. 2.

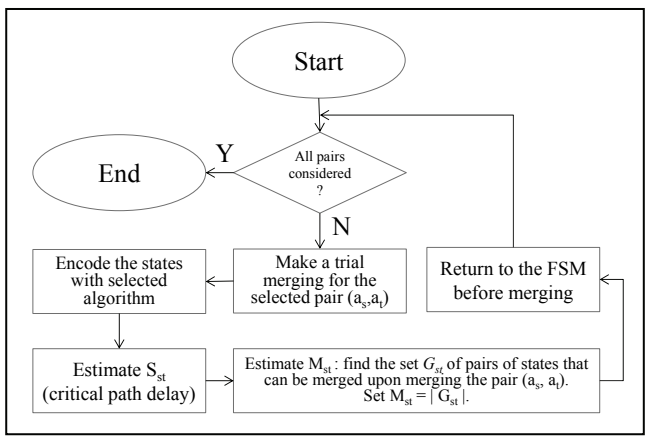

Fig. 2. Algorithm for estimation of optimization criteria

The speed of operation of an FSM is determined by the length of the critical path of its combinational part, which is equal to the number of CPLD macrocells. The CPLD architecture is a set of functional units of which each consists of two programmable arrays - AND and OR. The outputs of the array AND are connected to the inputs of the array OR; they are called terms. Typically, the number of inputs of functional blocks in CPLDs is sufficiently large and it usually exceeds the number of arguments of the functions implemented by the combinational part of the FSM. If the FSM is implemented on the basis of CPLD, there can be a large number of minterms in the DNF (*Disjunctive Normal Form*) of a function and a large number of arguments of a function. For that reason, when the FSM is implemented on the basis of CPLD, two critical paths are found, and the longest of them is chosen.

In the case of a large number of minterms in the DNF of a function, the linear decomposition [10] with respect to the minterms is used. The number of inputs of the OR gates is restricted by the parameter $q_{\max}$ - the maximum number of terms that can be connected to one CPLD macrocell (typically, $q_{\max}$ for different families of CPLDs is between 12 and 90). In the case of a large number of arguments, linear decomposition of the Boolean function with respect to the arguments is used, where the number n_{FB} of inputs of CPLD functional blocks is used as the restriction (for different families of CPLDs, n_{FB} is between 16 and 54).

Algorithm for the estimation of the FSM speed of operation

1. Find the maximum number $L_{\max}$ of arguments of the functions realized by the combinational part of the FSM. For each pair of states (a_s, a_t) in G, calculate the estimates S_{st}, and M_{st} of the optimization criteria

$$L_{\max} = \max_{w_i \in W} \left| L(w_i) \right|, \tag{1}$$

where $L(w_i)$ is a set of arguments of the function w_i. If the FSM is implemented on the basis of CPLD, then additionally the maximum number of minterms in the DNFs of the functions realized by the combinational part of the FSM is determined:

$$Q_{\max} = \max_{w_i \in W} \left| Q(w_i) \right|, \tag{2}$$

where $Q(w_i)$ is a set of minterms in the DNF of the function w_i.
2. Calculate the length of the critical path in the combinational part of the FSM. The lengths of two critical paths are found: one (S_L) depending on the maximum number of arguments:

$$S_L = 1 +] (L_{\max} - n_{FB})/(n_{FB} - 1) [, \tag{3}$$

where] x [is the minimum integer greater than or equal to x and the other (S_Q) depending on the maximum number of terms:

$$S_Q = 1 +] (Q_{\max} - q_{\max})/(q_{\max} - 1) [. \tag{4}$$

3. Set $S_{st} = \max(S_L, S_Q)$.

The estimate M_{st} is determined by the number of pairs of the FSM that can be merged after merging the pair (a_s, a_t). To provide the best possibilities for merging other states, M_{st} should be maximized. With regard to the above considerations, the algorithm for estimating the FSM speed of operation is as follows.

Algorithm for estimation of possibility of merging other states

1. Using the method described in [9], find the set G_{st} of pairs of states that can be merged upon merging the pair (a_s, a_t). For each pair of states (a_s, a_t) $\in G$, make a trial merging.
2. Set $M_{st} = | G_{st} |$.

5. Experimental results

To estimate the efficiency of the offered method we used MCNC FSM benchmarks [11]. Each of tested FSM benchmarks was encoded using binary encoding. The speed was calculated for the initial FSM, the full minimization program [9] and the method described in this paper.

The CPLD device model has following parameters, which can affect synthesis result: q_{max}=5 and n_{FB}=54. It corresponds to real CPLD devices, such as XC9500XL from Xilinx. The experiments were performed using Xilinx ISE 14.4 EDA tool and the target device was XC95288XL-6-TQ144 from the XC9500XL family. All the states were encoded using trivial binary method.

The experimental results are presented in Table 1, where M_0, C_0 and F_0 are, respectively, the number of internal states, the number of used macrocells and the maximum frequency (in MHz) of the initial FSM; M_1, C_1 and F_1 are, respectively, the same parameters after minimization using method described in [9] and M_2, C_2 and F_2 are, respectively the same parameters after minimization using proposed method (with taking in consideration speed of FSM).

Tab. 1. Experimental results for XC9500XL CPLD family

Name	M_0	C_0	F_0	M_1	C_1	F_1	M_2	C_2	F_2
Bbara	10	6	178.5	7	5	178.5	7	5	178.5
Bbsse	16	20	208.3	13	20	208.3	13	20	208.3
Beecount	7	7	178.5	5	7	208.3	5	7	208.3
Lion9	9	5	178.5	4	3	208.3	4	3	208.3
S27	6	4	208.3	5	4	178.5	5	4	178.5
sse	16	20	208.3	13	20	208.3	13	20	208.3
tbk	32	16	47.8	32	35	169.4	32	35	169.4
tma	20	26	178.5	18	24	178.5	18	24	178.5
Train11	11	12	208.3	4	3	208.3	4	3	208.3

In Table 2 values M_0/M_2, C_0/C_2 and F_2/F_0 are ratios of the corresponding parameters and *Mean* is the geometric mean value.

Tab. 2. Relative ratios of considering parameters

Name	M_0/M_2	C_0/C_2	F_2/F_0
Bbara	1.43	1.20	1.00
Bbsse	1.23	1.00	1.00
Beecount	1.40	1.00	1.17
Lion9	2.25	1.67	1.17
S27	1.20	1.00	0.86
sse	1.23	1.00	1.00
tbk	1.00	0.46	3.54
tma	1.11	1.08	1.00
Train11	2.75	4.00	1.00
Mean	1.43	1.17	1.17

The analysis of Tables 1 and 2 show that application of the proposed method using allows to reduce the number of internal states of the initial FSM 1.43 times. It can be also noticed that the reduction of states in 3 cases leads to critical path reduction of FSMs. The average gain of maximum operating frequency the FSM makes 1.17 times, and on occasion (benchmark t_{bk}) 3.54 times. In 4 cases there was also reduction of number of used macrocells (in average – 1.17 times).

Comparing to method from paper [9], there was no difference in results in number of states, cost and speed, so in these cases minimal representation of FSM is also the fastest one.

6. Conclusions

In this paper the method for FSM minimization was presented. In this method, the critical delay path is taken into account already at the stage of the minimization of the number of internal states. The presented method allows to reduce the number of internal states of FSM and additionally it allows to increase a speed of FSMs in some cases. The proposed method also allows to reduce the number of FSM transitions and input variables.

The main conclusion from experiments is that using this method for CPLD devices is less efficient than for FPGA devices, where obtained speed of synthesized FSMs was 7% greater than results obtained from other known methods.

In the offered approach to FSM minimization only two states merging is considered. The presented method can be modified in the future to join a group of states containing more states. Proposed method is the part of work on the complex minimization method, where speed, power consumption and area parameters are taken in consideration. In future, this method will serve to minimize power, cost and increase speed for FSM realization on programmable logic devices of different types.

The research has been done in the frame work of the grant S/WI/3/2018 and financed from the funds for science by MNiSW.

7. References

[1] Rama Mohan C., Chakrabarti P.: A new approach to synthesis of PLA-based FSM's, in Proc. of the 7th Int. Conf. on VLSI Design, Calcutta, India, 1994 (IEEE Computer Society, 1994), pp. 373–378.

[2] Gupta B. N. V. M., Narayanan H., Desai M. P.: A state assignment scheme targeting performance and area, in Proc. of the Twelfth Int. Conf. on VLSI Design, Goa, India, 1999 (IEEE Computer Society, 1999), pp. 378–383.

[3] Lindholm C.: High frequency and low power semi-synchronous PFM state machine, in Proc. of the IEEE Int. Symposium on Digital Object Identifier, Rio de Janeiro, 2011 (IEEE Computer Society, 2011), pp. 1868–1871.

[4] Liu Z., Arslan T., Erdogan A. T.: An Embedded low power reconfigurable fabric for finite state machine operations, in Proc. of the Int. Symposium on Circuits and Systems (ISCAS), Island of Kos, Greece, 2006 (IEEE Computer Society, 2006), pp. 4374–4377.

[5] Huang S. Y.: On speeding up extended finite state machines using catalyst circuitry, in proc. of the Asia and South Pacific Design Automation Conf.(ASAP-DAC), Yokohama, Jan.-Feb., 2001, p. 583-588.

[6] Nedjah N., Mourelle L.: Evolutionary Synthesis of Synchronous Finite State Machines, in proc. of the International Conference on Computer Engineering and Systems, Cairo, Egypt, 5-7 Nov. 2006, pp.19-24.

[7] Czerwiński R., Kania D.: Synthesis method of high speed finite state machines. Bulletin of the Polish Academy of Sciences: Technical Sciences. Vol. 58, (4), pp. 635–644 (2010).

[8] Klimowicz A.: Minimization method of finite state machines for low power design, in Proc. of Euromicro Conference on Digital System Design (DSD), Funchal, Portugal, pp 259-262 (2015).

[9] Klimowicz A., Solov'ev V. V.: Minimization of incompletely specified Mealy finite-state machines by merging two internal states. J. Comput. Syst. Sci. Int. Vol. 52 (3), pp. 400-409 (2013).

[10] Zakrevskij A. D.: Logic Synthesis of Cascade Circuits (Nauka, Moscow, 1981) [in Russian].

[11] Yang S.: Logic synthesis and optimization benchmarks user guide. Version 3.0. Technical Report. North Carolina. Microelectronics Center of North Carolina (1991).

Received: 12.10.2017 Paper reviewed Accepted: 01.12.2017

Adam KLIMOWICZ, PhD

He graduated from the Institute of Computer Science at the Bialystok University of Technology. He defended his doctoral dissertation in 2007 at the Faculty of Computer Science at the Bialystok University of Technology, where he holds the position of assistant professor. His scientific interests include design of embedded systems and methods of synthesis of combinational and sequential circuits based on CPLD/FPGA devices and microcontrollers. Co-author of the ZUBR package for supporting the design process of digital systems.

e-mail: a.klimowicz@pb.edu.pl