УДК 681.326

Miroschnyk M. A.[1], Pakhomov Y. V.[2], Shkil A. S.[3], Kulak E. N.[3], Kucherenko D. Y.[3]

[1]Dr. Sc., Professor, Head of the Department of Information Technologies, Ukrainian State University of Railway Transport, Kharkiv, Ukraine
[2]Assistant of Department of Gas and Heating Systems Operation, Kharkiv National University of Urban Economy, Kharkiv, Ukraine
[3]Ph.D, Associate Professor of Design Automation Department, Kharkiv National University of Radioelectronics, Kharkiv, Ukraine

DESIGN AUTOMATION OF EASY-TESTED DIGITAL FINITE STATE MACHINES

Context. The relevance of the work is to provide minimal additional hardware costs during design automation of easy-tested digital devices, which are represented by models of control finite state machines on hardware description languages.

Objective. To develop procedures of models' constructing of easy-tested control finite state machines on hardware description languages and estimate hardware costs for different methods of hardware redundancy introduction to HDL-models of finite state machines.

Method. The introduction to HDL-models of control finite state machines, which are presented in the form of the FSM template, hardware redundancy (additional fragments of the HDL-code), providing the forcing setting of finite state machine into an arbitrary state without the use of synchronizing sequences. For implementation of this approach, the method of FSM's state table extending is applied, which ensures the mode of bypassing of all nodes of FSM' state diagram in the diagnostic mode.

Results. Simulation of extended VHDL-models of the control FSM using Active-HDL confirmed the operability of this approach. Synthesis of these models using CAD XILINX ISE confirmed the receipt of testable structures and showed the minimum hardware costs for the method associated with the extension of the state table, in comparison with the organization of the shift register in the Scan Path mode.

Conclusions. The task of computer-aided design of testable control finite state machine on the basis of application of FSM' setting methods into given state is solved in the work. The optimal way of the setting organization into an arbitrary state of the control FSM is to expand the state table, which improves the controllability of FSM' states and leads to the structure' transformation of their HDL-models into easy-tested ones.

The scientific novelty of the work is the transformation of control FSM' models on hardware description languages, which is realized by introduction of the additional symbol to the state table, providing the settings of the FSM into an arbitrary state without the use of synchronizing sequences.

The practical significance of obtained results is to confirm the optimality, in terms of additional hardware costs, of the setting method of the control FSM into an arbitrary state by introducing the additional symbol into the state table.

Keywords: control finite state machine, state table, shift register, scanned path, hardware description language, CAD, Active-HDL, XILINX ISE.

TERMINOLOGY

A – state alphabet of FSM;
CLK – synchronization signal;
FSM – finite state machine;
HDL – hardware description language;
MUX – multiplexor;
TDI – test data input;
VHDL – very high speed integrated circuits HDL;
X – input alphabet of FSM;
Y – output alphabet of FSM;
$\delta(a_i, x_k) = a_j$ – transition function of FSM;
$\lambda(a_i, x_k) = y_\alpha$ – output function of FSM;
DE – diagnostic experiment;
CP – combinational part;
PLD – programmable logic device;
CAD – computer-aided design system;
ST – state table;
DD – digital device.

INTRODUCTION

Computer-aided design of digital devices (DD) based on specification which is represented in the form of hardware description languages (HDL). Structural methods of diagnosis, which are oriented to the detection of constant faults, are not effective. At the same time take into consideration that modern DD have a big size which makes the application of functional methods practically impossible.

Specialized data processing and control DD as a rule described by finite state machine (FSM). The forms of FSM representation are state table (ST) and state diagram (SD).

Due to the complexity of the diagnostic experiments (DE) under FSM, various methods of testability assurance were proposed. Namely, modification of FSM models providing for the introduction of hardware redundancy, both at structural level (additional inputs and outputs to ensure the simplicity of DE) and at functional level (additions and changes in the functional description of the FSM, namely state diagram).

Thanks to this approach testable FSM becomes easy-tested FSM. For easy-tested FSM, tasks of test diagnosis are simply solved within the limits of the established design costs. By reducing the cost of one or more main factors, determining the complexity of the test diagnosis, you can make easy-tested FSM.

1 PROBLEM STATEMENT

At a high-level design of digital control devices based on finite state machines, the form of specification' representation is the state table or the state diagram.

The form of FSM description using VHDL language is the FSM template, i.e. the code where the transitions function and outputs function are collected into separate processes, and the assignment of the new state is carried out in a special process associated with synchronization.

During design of testable control FSM, the hardware redundancy, which provides easy-testability, is advisable

© Miroschnyk M. A., Pakhomov Y. V., Shkil A. S., Kulak E. N., Kucherenko D. Y., 2018
DOI 10.15588/1607-3274-2018-2-13

to introduce at the initial stage of design (HDL coding). An easy-tested FSM is a finite state machine for which it is possible to construct a diagnostic experiment of minimum length by ensuring that the machine will go to any state for a minimum number of cycles.

Thus, a functional model of an abstract FSM in the form of a ST or a SD is given and on its basis a VHDL model is constructed in the form of an FSM template. It is necessary to consider different ways of introducing hardware redundancy into the VHDL model to provide easy- testability, and to choose the optimal method in terms of additional hardware costs. Hardware redundancy in VHDL-models is provided by adding to the HDL-code additional conditional operators that guarantee the construction of a scanned path in the memory part of the FSM, which is confirmed by the results of automated synthesis. In terms of additional hardware costs the optimal method is that, which provides minimal additional hardware costs by Quine' assessment of the circuit gate equivalence, which is synthesized into FPGA chip in automatic mode by the CAD tools.

2 LITERATURE REVIEW

Testability increasing methods by introducing hardware redundancy into the circuit implementation are sufficiently developed and widely used in the design. In the classical work about the development and classification of structural methods of testable design [1], the concepts of "controllability" and "observability" were introduced as the basis for structural testability evaluation. The principles of shift registers organization in the memory part of DD and the construction of a scanned path based on them are described. In addition, modifications of known structural methods for constructing tests using scanning methods are proposed. Practical recommendations on the use of structural methods to increase the testability of DD and the construction of diagnostic algorithms based on them are outlined in [2, 3].

In the classical article [4], methods of built-in self-test (BIST) are considered as the basis for ensuring the testability of designed devices. The application of BIST technology in designing of digital systems on programmable logic devices (PLD) is shown. In [5], ideas for testable design based on Boundary Scan technology using the JTAG port were further developed. Standards of boundary scanning IEEE 1149.x, IEEE 1500 and IEEE 1532, as well as the standard of testable design IEEE 1687 are considered. Technologies of testable design and testing of digital-analog circuits are proposed in [6]. Methods of functional decomposition of circuits that provide automated construction of testable digital-analog structures are considered.

Functional methods of the testable design of DD are considered in detail in [7]. For the digital automata presented in the form of ST, concepts of the diagnosed and definitely diagnosed classes of FSM are introduced and methods for bringing state tables of FSMs to the specified classes are proposed. Procedures for carrying out of DE with FSM using setting, synchronizing, diagnostic and transfer sequences are considered. In addition, the idea of improving the testability of the FSM is justified by introducing hardware redundancy by expanding the input alphabet, the output alphabet and the states alphabet. The problems of DE organization by bypassing all nodes and arcs of the state diagram are considered in [8]. Estimates of the length and completeness of DE with different variants of the diagram bypassing are given. The procedures for increasing the FSM' testability by equivalent transformations and expanding the input alphabet are proposed.

In [9], structural models of FSM are considered that allow us to use the values of output variables as codes of internal states. This approach was used during the synthesis of Mealy FSM on PLD, which allowed to reduce the cost of implementation by 1.5-2 times. Also, the proposed method of state coding allows to reduce the power consumption of the circuit implementations of FSMs [10].

In [11], the methods of testable design of DD based on the PLD and the organization of DE in the networks of cellular automata are described. The problems of automating the DE realization under HDL-models of FSM in the form of a FSM template are considered in [12]. The procedures for the DE organization in the verification system of CAD for PLD are considered and procedures for localization of design errors in HDL models are proposed.

Thus, the actual task is the development of procedures for design automation of the of easily-tested FSM, which are represented in hardware description languages, the comparison additional hardware costs for structural and functional methods of increasing testability based on comparison of synthesis results of testable HDL-models using CAD system.

3 MATERIALS AND METHODS

A finite state machine as an abstract model of DD with memory is determined by the six parameters $W = < X, A, Y, \delta, \lambda >$, where $X = \{x_1, x_2, ..., x_m\}$ – input alphabet; $A = \{a_1, a_2, ..., a_n\}$ – set of FSM' states; $Y = \{y_1, y_2, ..., y_r\}$ – output alphabet; $\delta(a_i, x_k) = a_j$ – the transition function, which is defined as $\lambda(z_i, x_k) = y_\alpha$ – the output function of the FSM.

The circuit is testable if the following parameters:

– generating procedures of test sets;

– estimating of their effectiveness;

– implementation of the test diagnostic [1] subject to the compliance financial costs, time costs in specified limits.

DE under FSM is a process of setting the input sequences, observing the corresponding output sequences, and making conclusions which is based on these observations. Experiments are classified by 3 groups: the identification of the FSM' states, the identification of the FSM' input sequence and FSM identification with n states which is different from all other FSM with the same number of states. The diagnostic experiment, during which all nodes of the diagram are bypassed, is used to detect design errors in models of control FSMs, which are represented by the state table or by the state diagram.

Thus, the easy-tested FSM is FSM which can be set to any state less than for n cycles, where n is the number of FSM states without using synchronizing sequences [13].

Let's consider conditions which are necessary to meet to create easy-tested FSMs based on FSM template:

– opportunity to switch FSM to the testing mode and vice versa at any cycle of the machine operation;

– setting FSM to any state in (*n*–1) clock cycles in the testing mode, where n is the number of FSM' states; Hamiltonian cycle can be organized for any of states;

– opportunity to build the easy-tested FSM in an automated mode by CAD tools.

The base of structural testable design is the method of scan path which allows to solve the problem of testability by reducing the complexity of the circuit structure.

The idea of scan path method is following:

1) first of all, we have to check memory elements;

2) next step is to check the combinational part (CP) for the possibility of setting internal variables to any state (regardless of their previous state).

A simple technique for creating the scan path is placing a multiplexer 2-to-1 just ahead of each flip-flop. The multiplexer control line, connected to scan mode selection signal. Scanning mode gives the opportunity to set memory units to any state by putting a sequence of signals to the input of the scanned data.

Let's consider the VHDL-model of Mealy FSM, which is represented by the state table (fig. 1a) and the state diagram (fig. 1b).

In the future (unless otherwise stated) we will consider FSMs whose controlled by generator of synchronization pulses (CLK), i.e. FSM has not only inputs ($x_1, \dots, x_n$), but also CLK input, which is used to set synchronization sequence. Therefore, the input signal on the transition (a_m, a_s) corresponding to the path $a_m X(a_m, a_s) – Y(a_m,a_s) a_s$ will be not $X(a_m,a_s)$, but the conjunction $CLK \cdot X(a_m, a_s)$. Thus, for $X(a_m, a_s) = 1$ (unconditional transition), the FSM operates only from the synchronization pulse. Usually CLK signal is not added to arcs of state diagram.

Let's consider a structural method to organize the scanning path in HD- models of finite state machines by manually setting the shift register directly in the model.

The difference between this model and FSM template is codes of FSM' states have to be set by force.

The fragment of the VHDL-model of Mealy FSM in the form of FSM template [12] with a shift register in the memory part is shown in Fig. 2. The signal A = 1 set the standard mode of FSM operation, A = 0 – the shift mode in the register from the external input TDI (test data input) in the direction to the higher discharges. FSM states in this model are coded forcibly according tp the number of state (during states' internal signals declaration) using the minimum number of flip-flops (3 in this case) to store the code of the state.

```
architecture FSM_MX of FSM_Mealy_MX is
signal  State, NextState: STD_LOGIC_vector (2
downto 0);
signal a1: STD_LOGIC_vector (2 downto 0):=″001";
signal a2: STD_LOGIC_vector (2 downto 0):=″010";
signal a3: STD_LOGIC_vector (2 downto 0):=″011";
signal a4: STD_LOGIC_vector (2 downto 0):=″100";
signal a5: STD_LOGIC_vector (2 downto 0):=″101";
begin
Sreg0_CurrentState: process (Clk, Reset)
begin
     if Reset=′1' then State <= a1;
     elsif Clk′event and Clk = ‘1′ then
          if A=′1' then State <= NextState;
     else State <= State(1 downto 0) & TDI;
          end if;
     end if;
end process;
```

Figure 2 – The fragment of the VHD-model of Mealy FSM with shift register (FSM_Mealy_MX)

The time diagram of Mealy FSM with the shift register in the memory part and the external input TDI (FSM_Mealy_MX) is shown in Fig. 3.

Let's analyze the result of simulation:

– signal *State* presented in decimal form;

– from 0 ns till 500 ns, the FSM operates in the normal mode (A=1);

– from 500 ns till 2000 ns, the FSM operates in the shift mode (A=0).

After synthesis of Mealy FSM with shift register and input TDI (FSM_MX) Xilinx ISE will create a circuit, the fragment of which is shown in Fig. 4.

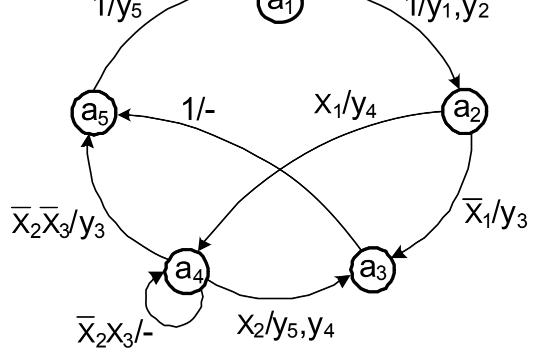
The Boolean equation for trigger D2 is:

$$D_2 = \{[(x_3 \vee \overline{x_3})\, y_3\, \overline{x_2} \vee (x_1\, y_1 \vee y_2)]\, A \vee Q_2\, \overline{A}\}\, \overline{\mathrm{Re}\,set}\,.$$

Actually we got multiplexor relatively to the control signal A, which fully corresponds to the circuit structure with the path scanning.

a	1	x_1	$\overline{x_1}$	x_2	$\overline{x_2}x_3$	$\overline{x_2}\,\overline{x_3}$
a_1	a_2/ y_1,y_2					
a_2		a_3/y_3	a_4/y_4			
a_3	a_5/–					
a_4				a_3/ y_4,y_5	a_4/–	a_5/ y_3
a_5	a_1/ y_5					

a



b

Figure 1 – The state table and the state diagram of Mealy FSM

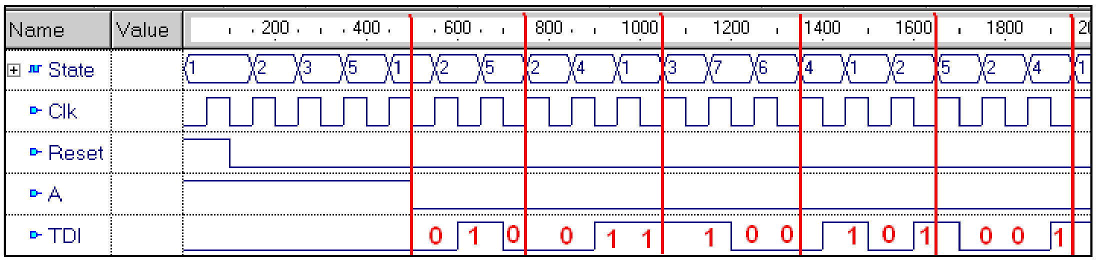


Figure 3 – Timing diagram of Mealy FSM from 500 ns till 2000 ns with shift register (FSM_Mealy_MX)

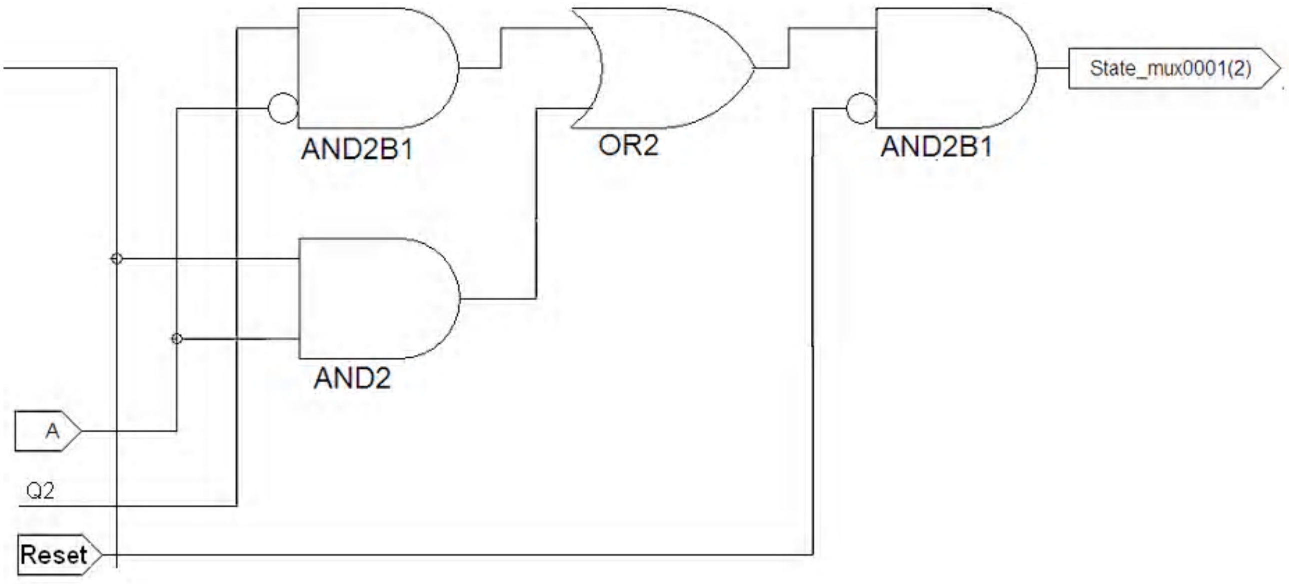


Figure 4 – The circuit' fragment of Mealy FSM with the shift register (FSM_Mealy_MX)

Rotate shift register is used to implement Hamiltonian cycle of diagram's bypassing. Operation modes of Mealy FSM with the shift register are following: signal A = 1 corresponds to the mode of the "normal" operation of the FSM (parallel load), A = 0 – the mode of rotate shift to the left. The FSM encoding style is "one-hot" code (unitary code): a1 (00001), a2 (00010), a3 (00100), a4 (01000), a5 (10000). Thus, for each state there is a trigger. Such model allows transferring into new state during each new clock cycle, and moreover, to bypass all states during the number of cycles equal to the number of states in a shift mode.

```
if A='1'   then State <= NextState;
else   State <= State(3 downto 0) & State(4);
end if;
```

The proposed methodology for converting FSM into easy-tested by introducing shift registers solves assigned problem, but also has disadvantages.

1. TDI input is used for organizing the BoundaryScan structure, but in the standard operating mode of the FSM, the control of this input is not provided.
2. During FSM scanning an open shift register creates difficulties for organizing a Hamiltonian cycle.
3. Using arbitrary encoding style for FSM's states, a shift register reduces the speed in scanning mode.
4. Although this approach effectively solves the problem of the Hamiltonian cycle construction of diagram's bypassing, but due to unitary state coding leads to significant hardware costs and performance decreasing.

Thus, the proposed methodology does not solve all assigned tasks. The solution lies in the field of functional transformation of state table by expanding the input alphabet to ensure testability.

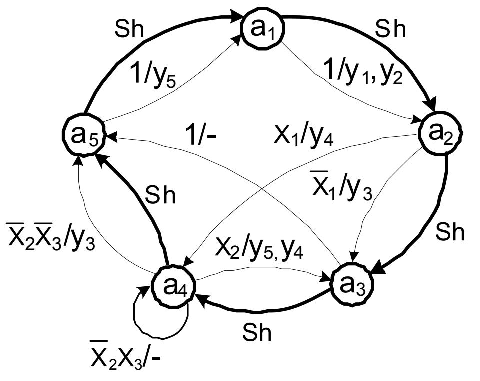
During a functional approach to designing (analyzing) of digital devices which specified by the finite state machine $W = < X, A, Y, \delta, \lambda >$, increasing the testability of an FSM is possible only by expanding the input alphabet X, the alphabet of states A, or the output alphabet Y [7]. On the one hand, this leads to the introduction of hardware redundancy into the circuit of DD, and on the other hand it ensures the preservation of the algorithm for the operation of the FSM, which is usually specified by the state table or state diagram.

In [14] authors of this paper proposed and theoretically justified the introduction of the additional column (symbol) in the state table of FSM by providing for this symbol the transition function of the state diagram of the shift register, which allows you to set the FSM to any given state.

Let's extend the ST of Mealy FSM (Fig. 1) by adding the column Sh (Shift). If Sh=1, the FSM operates in the setting mode in any given state, and if Sh=0, the FSM realizes the given algorithm. The expanded ST and the modified SD are shown in fig. 5.

a	1	x_1	$\bar{x}_1$	x_2	$\bar{x}_2 x_3$	$\bar{x}_2 \bar{x}_3$	Sh
a_1	a_2/ y_1,y_2						a_2/–
a_2		a_3/ y_3	a_4/ y_4				a_3/–
a_3	a_5/–						a_4/–
a_4				a_3/ y_4,y_5	a_4/–	a_5/y_3	a_5/–
a_5	a_1/ y_5						a_1/–

a



b

Figure 5 – Extended ST and modified SD of Mealy FSM with Sh signal

4 EXPIREMENTS

To test the effectiveness of the proposed method of increasing the testability of the FSM by expanding the input alphabet and introducing an additional column into the state diagram, let’s consider the modification of the VHDL-model shown in fig. 5.

The fragment of VHDL architecture of the easy-tested Mealy FSM with the additional signal Sh is shown in Fig. 6. The main advantage of this model is the possibility of transition from the settings mode to the operation mode of the FSM at any moment of time.

```
– Formation of combinational part of FSM – the
description of states’ transitions
Sreg0_NextState: process (State, Sh, x1, x2,
x3)
begin
y1 <= ‘0’; y2 <= ‘0’; y3 <= ‘0’; y4 <= ‘0’; y5
<= ‘0’;
  case State is
when a1 => if Sh = ‘1’ then NextState <= a2;
else NextState <= a2; y1 <= ‘1’; y2 <= ‘1’;
end if;
when a2 => if Sh = ‘1’ then NextState <= a3;
elsif x1=’1' then NextState <= a4; y4<=’1';
else NextState <= a3; y3 <= ‘1’;
end if;
when a3 => if Sh = ‘1’ then NextState <= a4;
else NextState <= a5;
end if;
when a4 => if Sh = ‘1’ then NextState <= a5;
elsif x2=’1' then NextState<=a3; y4<=’1'; y5<=
‘1’;
elsif x3=’1' then NextState <= a4;
else NextState <= a5; y3 <= ‘1’;
end if;
when a5 => if Sh = ‘1’ then NextState <= a1;
else  NextState <= a1; y5 <= ‘1’;
end if;
when others => NextState <= a1;
end case;
end process;
```

Figure 6 – The architecture of the VHDL-model of the easy-tested Mealy FSM (FSM_Mealy_Sh)

The simulation result of the VHDL model of the easy-tested FSM (FSM_Mealy_Sh), which provides the bypass of the state diagram on the Hamiltonian cycle, is shown in Fig. 7. If Sh = 0 (from 0 ns till 400 ns), the FSM operates in the standard mode, if Sh = 1 (from 400 ns till 950 ns) – bypassing of all states cyclically (Hamilton cycle).

Let’s note one feature of this waveform is the absence of output signals in the bypassing mode of transitions of state diagram (Hamiltonian cycle). This is due to the fact that the output signals y_i of Mealy FSM are associated with the arcs of the state diagram, and the introduction of additional arcs Sh doesn’t provide the presence of any output signal.

Of course, when constructing of the VHDL model of the easy-tested FSM, each arc Sh could be matched with an additional output signal, but this would lead to an expansion of the output alphabet and to additional hardware costs, which does not correspond to the formulation of the problem.

The result of the VHDL model synthesis with the signal Sh (FSM_Mealy_Sh) is shown in Fig. 8. Matching signals in the CAD XILINX ISE: Sh – In0, x1 – In1, x2 – In2, x3 – In3, Reset – Rst, Clk – Clk, Q2 “ FSM_FFd2 (internal signal from the output of flip-flop D2), Q3 “ FSM_FFd3 (internal signal from the output of flip-flop D3), D2 “ FSM_FFd2:In (internal signal to the input of flip-flop D2).

The complete equation of the excitation function for discharge D2 will be:

$$D_2 = Q_2 Q_3 x_3 \overline{x_2}\,\overline{Sh} \vee \overline{Q_2} Q_3 x_1 \overline{Sh} \vee Q_2 Q_3 x_2 \overline{Sh} \vee$$
$$\vee \overline{Q_2} Q_3 \overline{x_1}\,\overline{Sh} \vee (Q_2 \overline{Q_3} \vee \overline{Q_2} Q_3) Sh \,.$$

From this expression it is simple to do the conclusion that with respect to the signal Sh, the multiplexor was received. This is proves the synthesis of easy-tested circuit with has possibility of shift register implementation in the memory part of the FSM.

5 RESULTS

Let’s consider the different variants of the circuit synthesis of the easy-tested FSM in terms of additional hardware costs, which can be estimated by Quine’s criterion. Quine costs are defined as the total number of inputs of all gates in the circuit, since the number of inputs of the gate is proportional to the number of transistors in it.

Analyses of results are shown in Table 1. As can be seen from the table, in terms of hardware costs the optimal

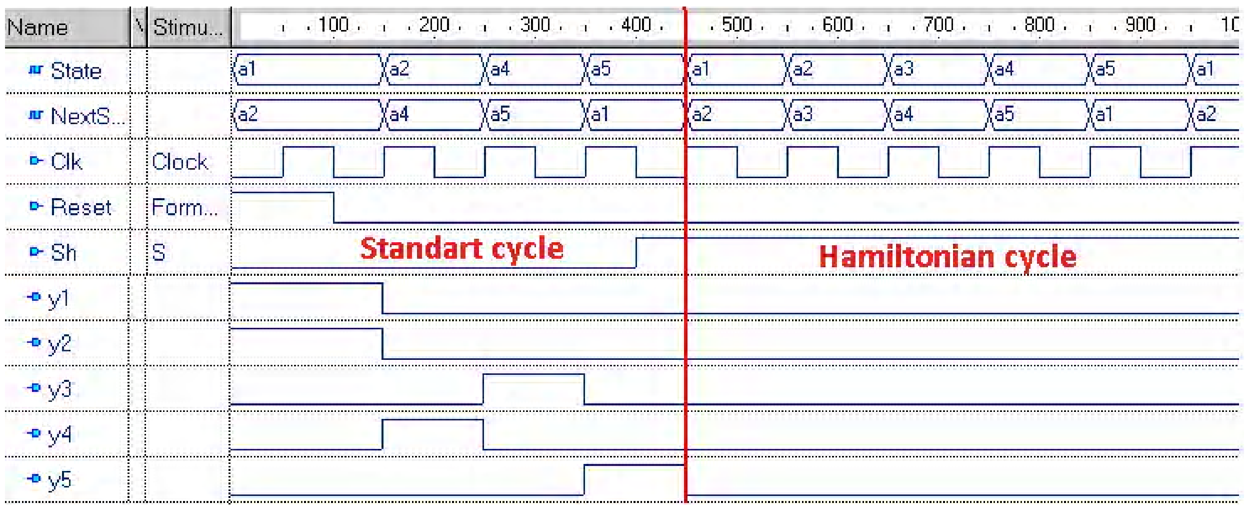


Figure 7 – The time diagram of Mealy FSM with the signal Sh (FSM_Mealy_Sh)

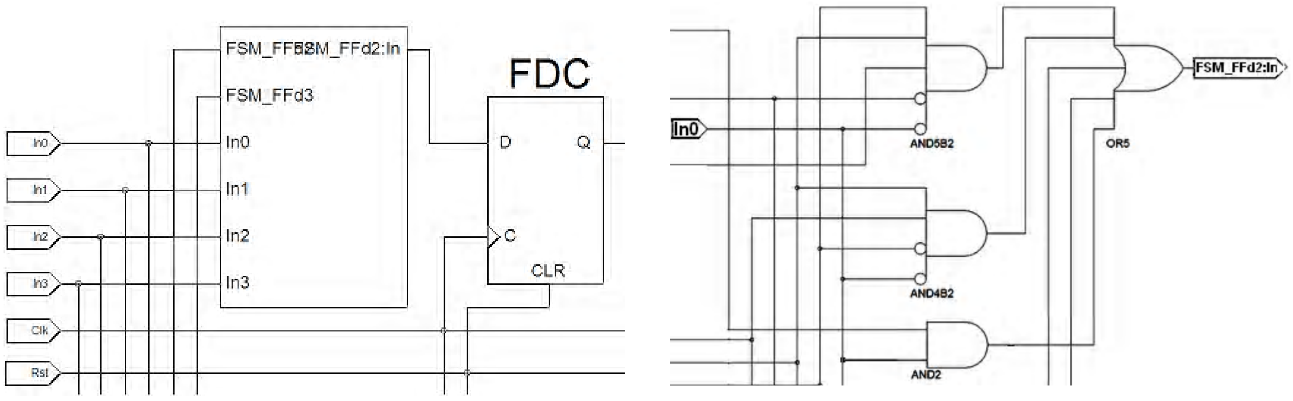


Figure 8 – The fragment of the synthesized circuit of Mealy FSM with the signal Sh (FSM_Mealy_Sh)

Table 1 – Hardware costs for circuits of easy-tested Mealy FSM

№	Type of FSM	Project name in Active-HDL	Number of flip-flops	Total costs by Quine
1	Mealy, standard	FSM_Mealy	3	110
2	Mealy with shift register	FSM_Mealy_MX	3	137
3	Mealy with shift reg., one-hot encoding style	FSM_MX_unit	5	193
4	Mealy with additional column in state table	FSM_Mealy_Sh	3	130

is the introduction of an additional column in the ST of the FSM, which increases the controllability of the FSM' states. Hardware costs in this case increase by 20–25%, depending on the type of machine.

6 DISCUSSION

This work is the first in a series of works related to the development of functional methods of automated testable design of the DD, which is represented in the form of FSM using hardware description languages. In existing CAD systems based on hardware description languages, such as Active-HDL or Riviera from Aldec Inc., there are software modules with a visual interface for automated generation of HDL models of finite state machines based on the SD (State Diagram Editor) and the conduction of diagnostic experiment on bypassing of nodes and arcs of the SD (ASFTEST) [12]. It would be advisable to supplement these modules with procedures for constructing easy-tested HDL-models of FSM.

Thus, the prospect of these researches consists in the development of a software module with a visual interface for entering the state diagram of the FSM, which will implement the proposed ideas related to the expansion of the input alphabet of the ST. This module can be integrated into simulation system of HDL-models Active-HDL and Riviera from Aldec Inc. If we talk about further theoretical researches, they can be associated with increasing testability by expanding the output alphabet and the states alphabet in models of FSM, and also by developing methods of automating testable encoding of FSM's states. A separate area of research can be the development of designing methods for easy-tested FSM, which are optimal from the point of view of time parameters of conducting DE.

CONCLUSIONS

The introduction of hardware redundancy underlies the method of automated design of the easy-tested FSM, which is the subject of this article. The form of FSM model in VHDL is FSM template. Additional fragments of the VHDL code ensure the forced setting of the FSM into an arbitrary state without synchronizing sequences. The shift register in the memory part of the FSM was considered for path scanning. Diagnosis mode involves the mode of bypassing of all nodes of the FSM' state diagram, which is provided by the proposed method state table expansion.

Due to the application of this method, the controllability of the FSM states, and consequently the testability, is increased. Active-HDL confirms the operability of this approach by simulation of the extended VHDL-models of the FSM.

CAD XILINX ISE confirmed the receipt of testable structures and showed the minimum hardware costs for the method associated with the extension of the state table, in comparison with the organization of the shift register in the Scan Path mode.

The scientific novelty of the work consists of the further development of the testability increasing method of FSM, by expanding the input alphabet, which gave the possibility to automatize of the design process of easy-tested FSM using HDL.

The practical significance of the obtained results is to develop the procedures for adding redundancy and expanding the input alphabet of HDL-models by adding additional conditional operators in the HDL-code, which is conduct to an arbitrary state.

Developed procedures can be applied during the development of an additional program module for CAD, which will automatically generate the HDL code of the easy-tested FSM.

ACKNOWLEDGMENTS

The work was carried out within the framework of the state budget research topic of the Kharkov National University of Radioelectronics "Cyber-physical system Smart cloud management of transport" (state registration number 0115U-000712 from 04.03.2015, order of the Ministry of Education of Ukraine № 105 from 09.02.15) on the basis of the Design automation department.

REFERENCES

1. Беннеттс Р. Дж. Проектирование тестопригодных логических схем : пер. с англ. / Р. Дж. Беннеттс. – М. : Радио и связь, 1990. – 176 с.
2. Пархоменко П. П. Основы технической диагностики (Оптимизация алгоритмов диагностирования, аппаратурные средства) / П. П. Пархоменко, Е. С. Согомонян ; под ред. П. П. Пархоменко. – М. : Энергия, 1981. – 320 с.
3. Горяшко А. П. Проектирование легко тестируемых дискретных устройств: идеи, методы, реализация / А. П. Горяшко // Автоматика и телемеханика. – 1984. – № 7. – С. 5–35.
4. Stroud C. E. A designer's guide to built-in self-test / Charles E. Stroud. – Kluwer Academic Publishing, 2002. – 319 p.
5. Городецкий А. Введение в технологии JTAG и DFT. Тестирование в технологиях граничного сканирования и тестопригодное проектирование / Ами Городецкий. – Palmarium Academic Publishing, Germany, 2012. – 308 с.
6. Mosin S. Methodology to Design-For-Test Automation for Mixed Signal Integrated Circuit / S. Mosin // Proceedings of the International Symposium. EWDTS'2013, September 27–30, Rostov on Don, Russia, 2013. – P. 178–183.
7. Тоценко В. Г. Алгоритмы технического диагностирования цифровых устройств / В. Г. Тоценко. – М. : Радио и связь, 1985. – 240 с.
8. Богомолов А. М. Контроль и преобразования дискретных автоматов / А. М. Богомолов, И. С. Грунский, Д. В. Сперанский. – К. : Наукова думка, 1975. – 175 с.
9. Solov'ev V. V. Minimization of mealy finite-state machines by using the values of the output variables for state assignment / V. V. Solov'ev // Journal of Computer and Systems Sciences International. – January 2017.– Volume 56, Issue 1. – P. 96–104.
10. Solov'ev V. V. Minimization of Power Consumption of Finite State Machines by Splitting Their Internal States / V. V. Solov'ev, T. N. Grzes // Journal of Computer and Systems Sciences International.– 2015. – Vol. 54, No. 3. – P. 367–374.
11. Мирошник М. А. Проектирование диагностической инфраструктуры вычислительных систем и устройств на ПЛИС: монография / М. А. Мирошник. – Х. : ХУПС, 2012. – 188 с.
12. Шкиль А.С. Автоматизация поиска ошибок проектирования в HDL-моделях конечных автоматов / А. С. Шкиль, Г. П. Фастовец, А. С. Серокурова // АСУ и приборы автоматики. – 2014. – Вып. 168. – С. 43–52.
13. Miroshnyk M. A. Design automation of testable finite state machines / [M. A. Miroshnyk, Д. Е. Кучеренко, Ю. В. Пахомов и др.] // 15th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS-2017). Харьковский национальный университет радиоэлектроники. – 2017. – P. 203–208
14. Бережная М. А. Синхронизирующие последовательности в конечных детерминированных автоматах / М. А. Бережная // Вестник НТУ «ХПИ». – 2008. – № 57. – С. 7–15.
15. Методы синтеза легкотестируемых цифровых автоматов / [М. А. Мирошник (Бережная), Ю. В. Пахомов А. С. Гребенюк, И. В. Филиппенко] // Інформаційно-керуючі системи на залізничному транспорті. – 2016. – № 5. – С. 28–39.

Article was submitted 22.09.2017.
After revision 06.11.2017.

Мірошник М. А.[1], Пахомов Ю. В.[2], Шкіль О. С.[3], Кулак Е. М.[3], Кучеренко Д. Ю.[3]

[1]Д-р техн. наук, професор, зав. кафедрою інформаційних технологій Українського державного університету залізничного транспорту, Харків, Україна

[2]Асистент кафедри експлуатації газових і теплових систем Харківського національного університету міського господарства, Харків, Україна

[3]Канд. техн. наук, доцент кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна

АВТОМАТИЗАЦІЯ ПРОЕКТУВАННЯ ЛЕГКОТЕСТОВАНИХ ЦИФРОВИХ АВТОМАТІВ

Актуальність роботи полягає в забезпеченні мінімальних додаткових апаратурних витрат при автоматизованому проектуванні легкотестованих цифрових пристроїв, представлених моделями кінцевих автоматів, що управляють, на мовах опису апаратури.

Мета роботи. Розробити процедури побудови моделей легкотестованих автоматів, що управляють, на мовах опису апаратури і оцінити апаратурні витрати для різних способів надання апаратурної надмірності в HDL-моделі автоматів.

Метод. Внесення в HDL-моделі кінцевих автоматів, що управляють, представлених у формі автоматного шаблону, апаратурної надмірності (додаткових фрагментів HDL-коду), які забезпечують примусову установку автомата в довільний стан без використання синхронізуючих послідовностей. Для реалізації цього підходу застосований метод розширення таблиці переходів-виходів автомата, який забезпечує режим обходу усіх вершин графа переходів автомата (станів) в режимі діагностування.

Результати. Моделювання розширених HDL-моделей автомата, що управляє, засобами Active-HDL підтвердило працездатність цього підходу. Синтез цих моделей інструментальними засобами автоматизованого проектування XILINX ISE підтвердив отримання тестопридатних структур і показав мінімальні апаратурні витрати для методу, пов'язаного з розширенням таблиці переходів-виходів, в порівнянні з організацією зсувного регістра в режимі Scan Path.

Висновки. У роботі вирішено завдання автоматизованого проектування тестопридатних автоматів, що управляють, на основі застосування методів установки автоматів в заданий стан. Оптимальним з точки зору апаратурних витрат способом організації установки в довільний стан автоматів, що управляють, є розширення таблиці переходів-виходів, яке підвищує керованість станів автомата і призводить до перетворення структури їх HDL-моделей в легкотестовані.

Наукова новизна роботи полягає у подальшому розвитку методів підвищення тестопридатності кінцевих автоматів за рахунок розширення вхідного алфавіту в HDL-моделях у формі автоматного шаблону, що дало можливість автоматизувати процес проектування легкотестованих автоматів з використанням мов опису апаратури.

Практична цінність отриманих результатів полягає у розробці процедур внесення надлишковості і розширення вхідного алфавіту в HDL-моделях кінцевих автоматів у формі автоматного шаблону шляхом внесення додаткових умовних операторів у HDL-код, які забезпечують встановлення автомату у довільний стан. Розроблені процедури можуть бути застосовані при розробці додаткового програмного модуля САПР цифрових пристроїв, який буде в автоматизованому режимі формувати HDL-код легкотестованого кінцевого автомату.

Ключові слова: автомат, що управляє, таблиця переходів-виходів, зсувний регістр, сканований шлях, мова опису апаратури, САПР, Active-HDL, XILINX ISE.

Мирошник М. А.[1], Пахомов Ю. В.[2], Шкиль А. С.[3], Кулак Э. Н.[3], Кучеренко Д. Е.[3]

[1]Д-р техн. наук, профессор, зав. кафедрой информационных технологий Украинского государственного университета железнодорожного транспорта, Харьков, Украина

[2]Ассистент кафедры эксплуатации газовых и тепловых систем Харьковского национального университета городского хозяйства, Харьков, Украина

[3]Канд. техн. наук, доцент кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники, Харьков, Украина

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ЛЕГКОТЕСТИРУЕМЫХ ЦИФРОВЫХ АВТОМАТОВ

Актуальность работы состоит в обеспечении минимальных дополнительных аппаратурных затрат при автоматизированном проектировании легкотестируемых цифровых устройств, представленных моделями конечных управляющих автоматов на языках описания аппаратуры.

Цель работы. Разработать процедуры построения моделей легкотестируемых управляющих автоматов на языках описания аппаратуры и оценить аппаратурные затраты для разных способов введения аппаратурной избыточности в HDL-модели автоматов.

Метод. Внесение в HDL-модели конечных управляющих автоматов, представленных в форме автоматного шаблона, аппаратурной избыточности (дополнительных фрагментов HDL-кода), обеспечивающих принудительную установку автомата в произвольное состояние без использования синхронизирующих последовательностей. Для реализации данного подхода применен метод расширения таблицы переходов-выходов автомата, который обеспечивает режим обхода всех вершин графа переходов автомата (состояний) в режиме диагностирования.

Результаты. Моделирование расширенных HDL-моделей управляющего автомата средствами Active-HDL подтвердило работоспособность данного подхода. Синтез данных моделей инструментальными средствами автоматизированного проектирования XILINX ISE подтвердил получение тестопригодных структур и показал минимальные аппаратурные затраты для метода, связанного с расширением таблицы переходов-выходов, по сравнению с организацией сдвигового регистра в режиме Scan Path.

Выводы. В работе решена задача автоматизированного проектирования тестопригодных управляющих автоматов на основе применения методов установки автоматов в заданное состояние. Оптимальным с точки зрения аппаратурных затрат способом организации установки в произвольное состояние управляющих автоматов является расширение таблицы переходов-выходов, которое повышает управляемость состояний автомата и приводит к преобразованию структуры их HDL-моделей в легкотестируемые.

Научная новизна работы состоит в дальнейшем развитии метода повышения тестопригодности конечных автоматов за счет расширения входного алфавита в HDL-моделях в форме автоматного шаблона, что дало возможность автоматизировать процесс проектирования легкотестируемых автоматов с использованием языков описания аппаратуры.

Практическая ценность полученных результатов заключается в разработке процедур внесения избыточности и расширения входного алфавита в HDL-моделях конечных автоматов в форме автоматного шаблона путем внесения дополнительных условных операторов в HDL-код, которые обеспечивают установку автомата в произвольное состояние. Разработанные процедуры могут быть применены при разработке дополнительного программного модуля САПР цифровых устройств, который будет в автоматизированном режиме формировать HDL-код легкотестируемого конечного автомату.

Ключевые слова: управляющий автомат, таблица переходов-выходов, сдвиговый регистр, сканируемый путь, язык описания аппаратуры, САПР, Active-HDL, XILINX ISE.

REFERENCES

1. Bennetts R. G. Proektirovanie testoprigodnyh logicheskih shem: per. s angl. Moscow, Radio i svjaz’, 1990, 176 p.
2. Parhomenko P. P., Sogomonjan E. S.; pod red. Parhomenko P. P. Osnovy tehnicheskoj diagnostiki (Optimizacija algoritmov diagnostirovanija, apparaturnye sredstva). Moscow, Jenergija, 1981, 320 p.
3. Gorjashko A. P. Proektirovanie legko testiruemyh diskretnyh ustrojstv: idei, metody, realizacija, *Avtomatika i telemehanika*, 1984, No. 7, pp. 5–35.
4. Stroud C. E. A designer’s guide to built-in self-test. Kluwer Academic Publishing, 2002, 319 p.
5. Gorodetsky A. Introduction to JTAG and DFT technology. Testing in edge scanning technologies and testable design. Palmarium Academic Publishing, Germany, 2012, 308 p.
6. Mosin S. Methodology to Design-For-Test Automation for Mixed Signal Integrated Circuit, *Proceedings of the International Symposium. EWDTS’2013, September 27–30, Rostov on Don.* Russia, 2013, pp.178–183.
7. Tocenko V. G. Algoritmy tehnicheskogo diagnostirovanija cifrovyh ustrojstv. Moscow, Radio i svjaz’, 1985, 240 p.
8. Bogomolov A. M., Grunskij I. S., Speranskij D. V. Kontrol’ i preobrazovanija diskretnyh avtomatov. Kiev, Naukova dumka, 1975, 175 p.
9. Solov’ev V. V. Minimization of mealy finite-state machines by using the values of the output variables for state assignment, *Journal of Computer and Systems Sciences International*, 2017, January, Volume 56, Issue 1, pp. 96–104.
10. Solov’ev V. V., Grzes T. N. Minimization of Power Consumption of Finite State Machines by Splitting Their Internal States, *Journal of Computer and Systems Sciences International*, 2015, Vol. 54, No. 3, pp. 367–374.
11. Miroshnik M. A. Proektirovanie diagnosticheskoj infrastruktury vychislitel’nyh sistem i ustrojstv na PLIS: monografija. Xar’kov, HUPS, 2012, 188 p.
12. Shkil’ A. S., Fastovec G. P., Serokurova A. S. Avtomatizacija poiska oshibok proektirovanija v HDL-modeljah konechnyh avtomatov, *ASU i pribory avtomatiki*, 2014, Vol. 168, pp. 43–52.
13. Miroshnyk M. A., Kucherenko D. E., Paxomov Yu. V., German E’. E., Shkil’ A. S., Kulak E’. N. Design automation of testable finite state machines, *15th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS-2017)*. Xar’kovskij nacional’nyj universitet radioe’lektroniki, 2017, pp. 203–208.
14. Berezhnaja M. A. Sinhronizirujushhie posledovatel’nosti v konechnyh determinirovannyh avtomatah, *Vestnik NTU “HPI”,* 2008, No. 57, pp. 7–15.
15. Miroshnik (Berezhnaja) M. A., Pahomov Ju. V., Grebenjuk A. S., Filippenko I. V. Metody sinteza legkotestiruemyh cifrovyh avtomatov, *Informacijno-kerujuchi sistemi na zaliznichnomu transporti*, 2016, No. 5, pp. 28–39.