

Synchronizing Sequences For Verification Of Finite State Machines

Maryna Miroshnyk
dept. Specialized Computer Systems
Ukrainian State University of Railway
Transport
Kharkiv, Ukraine
marinagmiro@gmail.com
0000-0002-2231-2529

Olexander Shkil
dept. Computer Engineering Design
Kharkiv National University of Radio
Electronics
Kharkiv, Ukraine
oleksandr.shkil@nure.ua
0000-0003-1071-3445

Elvira Kulak
dept. Computer Engineering Design
Kharkiv National University of Radio
Electronics
Kharkiv, Ukraine
elvira.kulak@nure.ua
0000-0002-8441-5187

Dariia Rakhlis
dept. Computer Engineering Design
Kharkiv National University of Radio
Electronics
Kharkiv, Ukraine
dariia.rakhlis@nure.ua
0000-0002-6652-1840

Yurii Pahomov
dept. Computer Engineering Design
Kharkiv National University of Radio
Electronics
Kharkiv, Ukraine
abc050073@gmail.com
0000-0002-3584-2107

Anatolii Miroshnyk
dept. Automation and Control in
Technical Systems of the National
Technical University "Kharkov
Polytechnic Institute"
Kharkiv, Ukraine
anatoliimiro@gmail.com

Abstract—The method of detection and localization of design errors in HDL-models of finite state machines with arbitrary output functions was proposed. The diagnostic experiment is carried out bypassing all arcs of the Mealy machine, starting from the initial vertex, including for machines of the "non-exclusive" class. To ensure the return of the machine with a possible design error in the initial state, it is suggested to use synchronizing sequences. Diagnostic experiments were performed in the Active-HDL design environment.

Keywords— verification, state machine, state diagram, diagnostic experiment, HDL, tests, waveform.

I. INTRODUCTION

The most complex and costly stage (70% of the total time) in the current designing cycle of digital devices (DD) is functional verification, i.e. in the process of detection, localization and elimination of errors in the system model. One of the most common methods in this original description of the specialized DD of data processing, and control is the finite state machine (FSM) [1]. The form of its representation is the transition-output table (state table), and the graph of the state machine transitions (state diagram, SD), constructed on its basis. The main form of projects description of DD in the systems of computer-aided design is the Hardware Description Language (HDL), so the verification object is the model of the DD using hardware description language, i.e. HDL-code. Based on this, the actual task is the development of verification procedures (search and localization) of design errors in HDL-models of FSMs for tabular or graph forms of specification [2].

One of the ways to describe the models of DD in the form of finite state machines in the VHDL language is the FSM template, i.e. a way to describe models of the control FSM, the specification of which is specified by the state table or by the state diagram. This is a special structure of the VHDL-model, in which the functions of transitions and outputs are isolated into separate processes (process), and the assignment of the new state is carried out in a special process associated with synchronization [3].

The selection of fragments of HDL-code, describes the behavior of FSM, with the style of "FSM template", allows

you to identify design errors such as "incorrect transition in the state diagram". For their detection and localization, it's possible to conduct the standard diagnostic experiment (DE) on FSMs (bypassing all arcs for the Mealy FSM, bypassing all nodes for the Moore FSM). The class of FSM under consideration and methods for describing, it must satisfy the conditions described in [4]. Especially it should be noted that in order to carry out the DE, the faultless and all faulty FSMs must be an "exceptional" class, i.e. have different output functions for each state or transition [5]. For a "non-exclusive" class of FSM, a single traversal of all arcs in the state diagram does not guarantee the detection of all design errors [6].

For HDL-models of finite state machines, the model of design error of several types is introduced, such as: "operator replacement" (logical or arithmetic), which corresponds to an error in the transition or output function, and "operand replacement" (in the assignment or conditional statement) that corresponds to an error in assigning of a new state or an output signal [7].

II. PREPARATION OF THE DIAGNOSTIC EXPERIMENT

Based on the state diagram bypass, a test is constructed and based on the results of its passage and comparison with the specification (output function), a vector of experimental tests (VEC) is constructed. Then, by analyzing the route table of graph bypassing (test matrix) and by analyzing the state diagram, an erroneous arc (arc) is/are found. In this case, the error in the HDL-code is most likely inside the fragment of the FSM template code, associated with the node (state), from which the erroneous arc originates. As an elementary test P_i during carrying out of the DE, the realization of a certain route of the graph bypass is used. A decision tree (the tree of bypassing of the state diagram) is constructed for the implementation of strategy of bypassing all arcs in the state diagram. The result of carrying out of the DE on the bypassing of the state diagram is the vector of experimental tests $V = (v_1, v_2, \dots, v_m)$, where m – the number of terminal nodes of the decision tree. Wherein, $v_i = 0$, if the output

function of the FSM coincides with the specification (the test passed) and $v_j = 1$ if it does not match (the test failed). The localization of a single error in the route of state diagram bypassing is on the left side of the formula (1), and the multiple – on the right, where M_j – the j th row of the route matrix of the state diagram bypassing and the VEC.

$$D = \cap_{v_j=1} M_j - \cup_{v_j=0} M_j \quad (1)$$

$$D = \cup_{v_j=1} M_j - \cup_{v_j=0} M_j$$

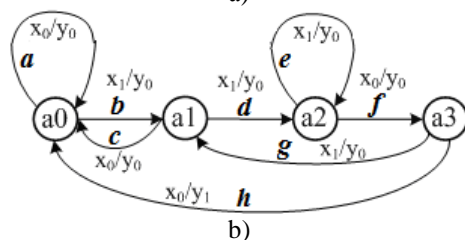
To construct the test, a strategy of bypassing all arcs of the state diagram is implemented, starting from the initial node, subject to the admissibility of more than one arc $a_i \Rightarrow a_j$ (mixed strategy). This approach involves a so-called “indestructible” experiment in which, in the end of each test the FSM automatically or forcefully returns back to its initial state. In this case, all single faults of these transitions are tested, as well as the correctness of the excitation functions of the FSM providing these transitions. The DE over the HDL-model of the finite state machine consists of applying input effects to it in accordance with the chosen bypassing strategy of the state diagram of the FSM, obtaining output reactions in the form of output signals or the states’ list of the FSM on the waveform or on the list of state diagram bypassing in the file, and comparison of the obtained reactions with a specification by visual or program way. Based on this analysis of the comparison results, it’s concluded that the HDL-model corresponds to the specification. DE is conducted in the Active-HDL, FSM models are presented in VHDL [4].

Quite often, in the specification of the finite state machine, only output values for certain input effects are specified (state table or output function), and the state diagram will be constructed during the subsequent design stages. In this case, the faultless or faulty FSMs can go beyond the limits of an exclusive class of FSMs, and a test built for a reference state diagram may not identify certain design errors. This problem is the subject of this article and will be discussed in detail below.

For example of conducting DE to search for design errors, the VHDL model of the Mealy FSM, which performs recognition of the set “1100” in the input bit sequence, is used. In Fig. 1 the state table (a) and the state diagram (b) of the given Mealy machine are represented.

a	x0	x1
a0	a0, y0	a1, y0
a1	a0, y0	a2, y0
a2	a3, y0	a2, y0
a3	a0, y1	a1, y0

a)



b)

Figure 1. State table (a) and state diagram (b) of Mealy FSM

Note that in the VHDL model, the input signal x0 in the code corresponds to data=0, and the signal x1 – data=1, and, respectively, output signal y0 in the code corresponds to y0=0, and y1 in the code – y=1. This is due to the features of the graphic editor State Diagram in Active-HDL.

To describe the state diagram, it is suggested to use a modified adjacency matrix (fig. 2), where for each node of the diagram (rows of the matrix) in the cells of the matrix, the names of the arcs connecting the given node to the successors (columns) are indicated. In this case, more than one arc can exist in one cell (multigraph) [2].

	a0	a1	a2	a3
a0	a	b		
a1	c		d	
a2			e	f
a3	h	g		

Figure 2. Adjacency matrix for state diagram on fig. 1

To implement the strategy of bypassing all arcs of the diagram (in the “manual” test building mode), a binary decision tree (the graph bypass tree) is constructed using the following algorithm.

1. The construction of the diagram bypassing tree begins with the initial node a_0 . The first line of the adjacency matrix is analyzed and a list of successor nodes is compiled.

2. The first node is selected from the list of successors and included in the tree. The corresponding cell in the matrix (arc) is marked.

3. For the line corresponding to this number, a list of unmarked successors is compiled.

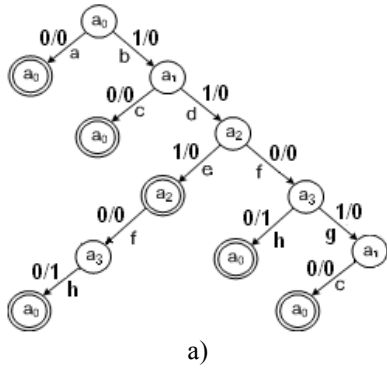
4. The construction of the route in the diagram ends with a terminal (finite) node. The node in the tree becomes terminal if:

- the incoming arc is a loop (an element of the main diagonal of the adjacency matrix);
- the next successor node is the initial node a_0 (the cycle of the FSM operation is completed);
- all cells in the next line (during performing step 3) will be marked.

5. Steps 2, 3 and 4 are repeated until all arcs (cells of the adjacency matrix) are labeled.

6. For terminal nodes with loops, return routes to the initial node are being completed.

The decision tree for bypassing of state diagram arcs (a) and bypass routes (b) for the reference model (without design errors) is shown in Fig. 3. It should be noted that the number of bypass routes of state diagram is equal to the number of terminal nodes (nodes that do not have successors) of the decision tree. In Fig. 3 (a) terminal nodes are marked by a double circle. Taking into account input effects and output values, the strategy for bypassing all arcs of diagram can be represented in the form of test matrix, which is shown in Fig. 3 (c). In the last column of this matrix, the reference values of the outputs function of finite state machine are indicated. Analyzing the test matrix, it should be noted that the test P_2 supplements P_4 to return the machine to its initial state, but P_3 completely includes P_5 . This allows constructing a matrix of combined tests (fig. 4).



a)

1	a0-a-a0
2	a0-b-a1-c a0
3	a0-b-a1-d-a2-e-a2-f-a3-h-a0
4	a0-b-a1-d-a2-f-a3-g-a1-c-a0
5	a0-b-a1-d-a2-f-a3-h-a0

b)

M	data	a	b	c	d	e	f	g	h	y
P_1	00	1								0
P_2	10		1	1						0
P_3	11100		1		1	1	1		1	1
P_4	11010		1	1	1		1	1		0
P_5	1100		1		1		1		1	1

c)

Figure 3. The strategy of bypassing all arcs of state diagram: a) – the decision tree; b) – bypass routes of the diagram; c) – the matrix of elementary tests

B	data	a	b	c	d	e	f	g	h	y
P_1	00	1								0
P_2	10		1	1						0
$P_4 + P_5$	11010		1	1	1		1	1		0
$P_3 + P_5$	11100		1		1	1	1		1	1

Figure 4. Matrix of combined tests for the reference diagram

In this case, the VEC will be $V = (P_1, P_2, P_4 + P_5, P_3 + P_5)$. Simulation results of the test matrix for the reference model are shown in Fig. 5.

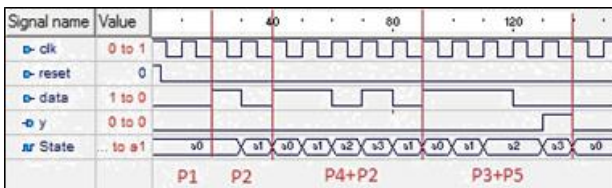


Figure 5. Waveform of the reference HDL-model

III. DIAGNOSTIC EXPERIMENT OVER THE HDL MODEL OF THE FSM

We will perform several DEs in search of design errors in VHDL-models of FSM.

Consider an erroneous VHDL-model with a design error, such as “erroneous new state assignment” (code snippet in fig. 6).

Simulation results of the VHDL model with the assignment error are shown in Fig. 7. The result of the fourth test does not coincide with the reference, because y can be

equal to 1 only when the sequence “1100” is set on input.

```

when a2 =>
  if data='0' then nextstate<=a3; y<='0';
  else
    nextstate<= a3; y<='0';
    -- (instead of nextstate <= a2)
  end if;

```

Figure 6. Fragment of the VHDL-model of the FSM with an erroneous assignment statement

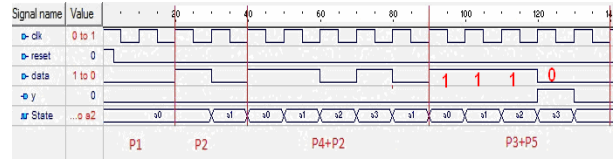


Figure 7. Waveform for a new state assignment error

In Fig. 7 we see that $y = 1$ when the input sequence “1110” is set, i.e. the VEC will be $V = (0, 0, 0, 1)$. The location of the error in the bypass route of state diagram is given by the formula 1. $D = \{b, d, e, f, h\} - \{a\} \cup \{b, c\} \cup \{b, c, d, f, g\} = \{b, d, e, f, h\} - \{a, b, c, d, f, g\} = \{e, h\}$ and the erroneous arcs e and h in the state diagram are defined.

To find the location of the design error, you need to return to the VHDL code of the model and perform a visual inspection of the code area to calculate the error arc. Based on this results obtained by comparing the reference waveform, and the waveform obtained after simulation of the code with an error, an erroneous operator is found (`else nextstate <= a3;`).

Let’s consider the erroneous VHDL model with a design error such as “error in the conditional if statement” (code snippet in fig. 8). Simulation results of this erroneous model are shown in fig. 9.

```

when a2 =>
  if data='1' then nextstate<=a3;y<='0';
  -- (instead of data='0')
  else
    nextstate<=a2;y<='0';
  end if;

```

Figure 8. Fragment of the VHDL model with an erroneous statement if

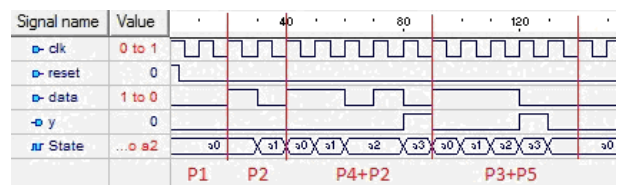


Figure 9. Waveform for the error in a conditional statement if data='1' instead of data='0'

Based on the waveform analysis, the VEC will be $V = (0, 0, 1, 1)$. The location of the error in the bypass route of the state diagram is given by the formula 1: $D = \{b, c, d, e, f\} \cap \{b, d, e, h\} - \{a\} \cup \{b, c\} = \{b, d, e\} - \{a, b, c\} = \{d, e\}$, i.e. erroneous arcs d and e in the state diagram are defined.

Let’s consider an erroneous VHDL-model with a design error such as “error in the assignment of the output signal” (code snippet in fig. 10).

```

when a3 =>
  if data='0' then nextstate<= a0; y<='1';
  else
    nextstate<= a1; y<='1';
  -- (instead of y<='0')
  end if;

```

Figure 10. Fragment of the VHDL model of the FSM with an error in the assignment of the output signal

Simulation results of the VHDL model with an error in the assignment of the output signal are shown in fig. 11.

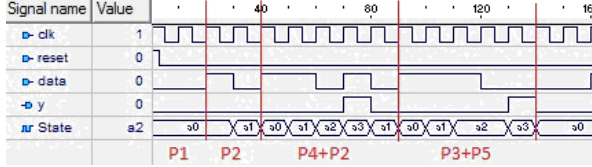


Figure 11. Waveform for an error in the assignment of the output signal $y=1$ instead of $y=0$

Based on the waveform analysis, the VEC will be $V = (0, 0, 1, 0)$. The location of the error in the bypass route of the state diagram is given by the formula 1: $D = \{b, c, d, f, g\} - \{a\} \cup \{b, c\} \cup \{b, d, e, f, h\} = \{b, c, d, f, g\} - \{a, b, c, d, e, f, h\} = \{g\}$, i.e. the erroneous arc g in the state diagram are defined.

Let's consider one more erroneous VHDL-model with a design error such as "new state assignment error" (code snippet in fig. 12).

```

when a1 =>
  if data='1' then nextstate<= a2; v<='0';
  else
    nextstate<= a3; v<='0';
  -- (instead of nextstate <= a0)
  end if;

```

Figure 12. Fragment of the VHDL model of the FSM with an erroneous assignment statement

Simulation results of the VHDL model with a "new state assignment error" are shown in fig. 13.

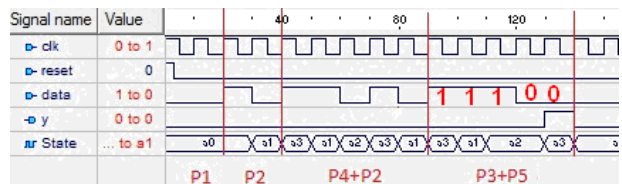


Figure 13. Waveform for assignment error a_3 instead of a_0

The result of the output function (y) coincided with the reference, which implies no error in the VHDL model. But if you carefully analyze the waveform, you can see that after the third test ($P_4 + P_2$), the FSM does not return to the initial state a_0 , but the output function gives the correct result. So this result is due to the fact that this FSM does not belong to the so-called "exclusive" class of FSM, which have different output functions for each state (each arc for the Mealy FSM). In this FSM, for all arcs except the arc h , the output $y = 0$ (fig. 1).

Considered design error ($nextstate<=a_3$ instead of $nextstate<=a_0$) generates a new arc c from the state a_1 , but the output function is the same for them.

This example confirms that the construction of the test based on a simple strategy, "single bypassing of all arcs of the diagram" on the reference state diagram does not guarantee finding all design errors in the HDL-model for a non-exclusive class of FSM.

IV. USING SYNCHRONIZING SEQUENCES IN DIAGNOSTIC EXPERIMENTS

As mentioned above, the main task in carrying out an "indestructible" DE with the possibility of returning the FSM to the initial state, regardless of the result of the next test. For these purposes it is expedient to use synchronizing sequences (SS) of FSMs [4]. The input sequence X_s of the FSM, which sets it to a certain final state, regardless of the state of the output and the initial state, is called SS. If an FSM is given by state table, then it follows from the definition that an FSM has a SS if and only if there exists an input sequence X_s such that $\delta(a_i, X_s) = a_0, \forall a_i \in A, a_0 \in A$. The set of transitions $\delta(a_i, X_s) = a_0, \forall a_i \in A$ of an FSM determines the mapping of the set of its states A into a certain state a_0 when the input sequence X_s is set to the FSM, i.e. $a_i \xrightarrow{X_s} a_0$.

A SS for a given FSM can be found from a synchronizing tree, which is a successor tree constructed according to certain rules [4]. In Fig. 14 the construction of the SS for the considered Mealy FSM is shown. It should be noted that there is no SS for the state a_3 .

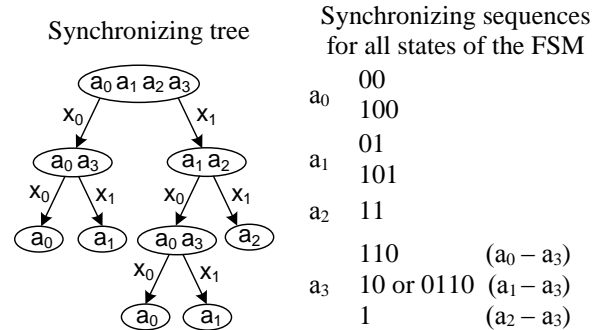


Figure 14. Synchronizing sequences for Mealy FSM

Thus, if we are speaking about the forced return back to the state a_0 , then synchronizing sequences $\{00, 100\}$ must be at the end of each input word of elementary test. Using the construction principle of combined tests [5], the reference tests matrix (fig. 5) takes the form shown in fig. 15, where for tests P_2 and $(P_4 + P_2)$ fragments of "supplements" of synchronizing sequences are highlighted.

B	data	a	b	c	d	e	f	g	h	y
P_1	00	1								0
P_2	100	1	1	1						0
$P_4 + P_2$	110100	1	1	1	1		1	1		0
$P_3 + P_5$	11100		1		1	1	1	1	1	1

Figure 15. Matrix of reference combined tests taking with addition of synchronization sequences

Simulation results of the VHDL model with the error of new state assignment error, using the SS to return to the state a_0 , are shown in fig. 16.

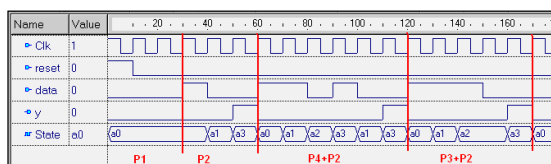


Figure 16. Waveform for assignment error a3 instead of a0 with SS

Analysis of this waveform allows you to make two conclusions. First, after completing each test, even for the erroneous VHDL model, the FSM returns to its initial state, i.e. an “indestructible” DE is performed. Secondly, for tests P_2 and $(P_4 + P_2)$ with SS, the output value y does not coincide with the reference, i.e. the VEC will be $V = (0, 1, 1, 0)$.

The location of the error in the bypass route of the state diagram is given by the formula 1: $D = \{a, b, c\} \cap \{b, c, d, f, g\} - \{a\} \cup \{b, d, e, f, h\} = \{b, c\} - \{a, b, d, e, f, h\} = \{c\}$, i.e. the erroneous arc c in the state diagram is found. Considering the VHDL-code corresponding to arcs coming from a_1 , we find the erroneous assignment statement `nextstate<=a3`.

ACKNOWLEDGMENT

The diagnostic experiments carried out under the VHDL models of finite state machines, which is represented in the form of an FSM template, showed that a single bypass of all arcs of the state diagram of the Mealy machine with a return to the initial state, allows to detect and localize design errors in HDL-code such as “operator replacement” or “operand replacement”.

For an exclusive class of FSM (each arc has a unique output function), a single bypass can detect single and multiple design errors that change the functions of the transitions and outputs of the FSM. For a non-exclusive class of FSM for carrying out an “indestructible experiment” (with organization of return to the initial node a_0), it is suggested to use synchronizing sequences that transfer an FSM from any state to a given state.

The proposed approach made it possible to detect and localize design errors for HDL-models of finite state machines with an arbitrary output function.

REFERENCES

- [1] Barcalov A. Logic Synthesis for FSM-Based Control Units /Alexander Barkalov, Larysa Titarenko. – Berlin: Springer Science & Business Media,. 2009. – 233 p.
- [2] Shkil A. Search for design errors in HDL-models of digital state machines /S. Almadhoun, Y. Syrevitch, A. Shkil // Bulletin of the Kherson STU . – 2013. – №2 (46). – P. 377-383.
- [3] Gill A. Introduction to the theory of finite state machine.– M.: Science, 1966. – 272 p.
- [4] Berejnaia M. Synchronizing sequences in finite determinate machines / M. Berejnaia // Bulletin NTU “KPP”. – 2008. – № 57. – P. 7-15.
- [5] Shkil A. Searching methods for design errors in language models of control FSM / A. Shkil, E. Kulak, A. Grebenyuk (Serokurova) // Collection of articles of the scientific-information center “Znanie” based on the materials of the XIXth International conference: “The development of science in the XXI century”, part 2, Kharkov, November 2016. – P. 57-65. .
- [6] Shkil A. Design of Logical Control Units Based on Finite State Machines' Patterns/ M Miroshnyk; S. Poroshyn; A. Shkil; E. Kulak; I. Filippenko; D. Kucherenko; Y. Pakhomov; J. Salfetnikova; M. Goga//Proceedings of the 2018 IEEE East-West Design & Test Symposium.– 6 p. [Електронний ресурс] / IEEE Xplore Digital Library. – Режим доступу: www / URL: <https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?filter=issueId%20EQ%20%228524135%22&refinements=Author:Alexander%20Shkil&pageNumber=1&resultAction=REFINE>.
- [7] Shkil A. /Design automation of testable finite state machines. / Miroshnik M.A., Shkil A.S., Kulak E.N., Filippenko I.V., Kucherenko D.Y., E.E. German // Proceedings of IEEE East-West Design & Test Symposium (EWDTS), 2017. – P. 203-208. .