

Testability Increasing Method by Introducing Hardware Redundancy in the Easy-tested Finite State Machines

Marina Miroshnyk

*Department of Specialized Computer Systems
Ukrainian State University of Railway Transport,
Ukraine, Kharkov,
marinagmiro@gmail.com*

Olga Zaichenko

*Department of Design and Operation of Electronic Devices
Kharkiv National University of Radio Electronics,
Ukraine, Kharkov,
olha.zaichenko@nure.ua*

Pavlo Galkin

*Department of Design and Operation of Electronic Devices,
Kharkiv National University of Radio Electronics,
Ukraine, Kharkov,
pavlo.halkin@nure.ua*

Roman Tsekhmistro

*Dep. of Media Engineering and Information Radioelectronic Systems,
Kharkiv National University of Radio Electronics,
Ukraine, Kharkov,
tsekhmistorroman@gmail.com*

Abstract—Testability increasing methods by introducing hardware redundancy into the circuit implementation are sufficiently developed and widely used in the design. Since the construction of the testing sequence is based on the use of automaton diagrams, it eliminates the need to analyze the circuit implementation of the remote control when building a diagnostic experiment. This approach allows us to extend the class of detectable faults, which in structural-analytical test generation methods is limited, as a rule, to a multitude of single constant faults. The use of automaton models in the construction of tests allows to detect any malfunction that changes the automaton diagram of a serviceable remote control and does not increase the number of states of remote control memory elements. There was described finite state machine using hardware description language. The method of computer-aided design of the easytested control FSM by introducing the hardware redundancy is presented in the paper. The FSM model is represented in VHDL in the form of the FSM template. The solution way is to add additional fragments of the VHDL code, which ensure the forced setting of the FSM into an arbitrary state without the use of synchronizing sequences. The use of the shift register in the memory part of the control FSM for organizing the path scanning was considered. The method of FSM state table expansion, which ensures the mode of bypassing all nodes of the FSM' state diagram in the diagnostic mode was proposed.

Keywords—easy tested finite state machine, Hamiltonian cycle, distinguishing sequence, homing sequence, shift register.

I. INTRODUCTION

During design of a digital devices such model of a digital device with memory as finite state machine (FSM) is widespread [1-3]. Finite state machine, like other digital devices models are presented in languages VHDL, Verilog, intended for designing digital circuits on modern basis of field-programmable gate array (FPGA). According to VHDL descriptions of finite state machine is synthesized synchronous logical circuits in a given basis of logical elements. Today, the process of synthesis is automated [4-5] and the most important problem when creating projects and

system-on-a-chip (SoC) is the problem of verifying the original submitted to the VHDL. The actual task is the further development of procedure of computer-aided design of easy tested finite state machines, which are presented using hardware description language, and their verification.

II. TESTABILITY OF FINITE STATE MACHINE

Testability increasing methods by introducing hardware redundancy into the circuit implementation are sufficiently developed and widely used in the design. In technical diagnostics, the direction associated with the use of the classical theory of experiments with finite state machines for constructing testing sequences is known [6].

An experiment with finite state machines refers to the process of applying input sequences, observing the corresponding output sequences, and deriving conclusions based on these observations. A characteristic feature of the theory of experiments with finite state machines is the use of the functional approach and the transition-output tables of the automaton model of remote control to build a complete testing sequence.

The fundamental work, in the theory of experiments with finite state machine, is the work [7]. In this paper, it is proposed to use a number of characteristic sequences that allow identification of the transition table-output of an finite state machine. The procedure proposed in [7] gives the best results for a class of minimal strongly connected finite state machine having distinguishing. In the same paper, the class of installation synchronization and characteristic sequences is defined, which allows one to construct verifying experiments for finite state machine that do not have distinguishing sequences. In the future, this direction was developed in the work [8].

The main advantage of this direction is the simplicity of constructing a testing sequence for a given finite state machine and a wide class of detectable faults, which is limited only by faults that increase the number of states of the machine. The complexity of constructing a testing experiment with an finite state machine is determined by the properties of its diagram. To easily tested finite state machine [6-10], we will assign finite state machine, for which the tasks of test diagnostics are solved as simply as possible within the established costs. Before discussing the design of checking experiments, it is appropriate to recall several definition relating to experiments performed on sequential circuits or finite state machines. A synchronizing sequence for a given sequential circuit is an input sequence whose application is guaranteed to leave the circuit in a certain final state, regardless of the particular initial state of the circuit. Some circuits have synchronizing sequences, others do not. A homing sequence for a given sequential circuit is an input sequence whose application makes it possible to determine the final state of the circuit by observing the output sequence that the circuit produces. Both synchronizing and homing sequences have to do with the determination of the final state of a circuit. A distinguishing sequence is an input sequence whose application makes it possible to determine the initial state of the circuit by observing the output sequence that the circuit produces [7]. The opinion of experts in the field of the theory of experiments with finite state machine is that the class of finite state machine having distinguishing, synchronizing and homing sequences of minimal length is a class of easily tested finite state machine. This class of finite state machine is a model of testable discrete devices with memory elements. Therefore, the analysis of the properties of finite state machine models is directly related to the analysis of the conditions for constructing testable devices, that is, with the development of methods for testable design. Currently, testability is one of the main criteria by which the design level is assessed, and is informally defined as follows: financial costs, time and values of indicators characterizing the object's ability to detect faults, search for a fault site and implement test diagnostics.

III. FUNCTIONAL APPROACH TO SYNTHESIS OF DIAGNOSTIC EXPERIMENTS WITH FINITE STATE MACHINES

An experiment with an finite state machines is the process of applying the input sequences, observing the corresponding output sequences and deriving conclusions based on these observations. Depending on the purpose of the experiment, experiments on identification of the states of the finite state machines, identification of the input sequence of the finite state machines, and identification of the finite state machines with the states differing from all other finite state machines with the same number of states differ.

In the development of approaches, principles, methods and used mathematical models in the technical diagnostics of complex discrete or digital objects, several directions were formed. The first direction includes structural-analytical test generation methods based on knowledge of the circuit implementation and explicit models of logical faults.

The second direction, based on the functional approach and a wider class of implicit models of detectable faults, involves the use of automatic models and the construction of

a test sequence that allows to check the conformity of the diagram or transition-output table of the tested device to the transition- output table of the healthy device.

In the development of the second direction of test diagnostics, the methods of the classical theory of experiments with finite state machines are widely used. A characteristic feature of this theory is the use of the functional approach and the transition-output tables of the automaton model to build a complete testing sequence. Since the construction of the testing sequence is based on the use of diagrams, it eliminates the need to analyze the circuit implementation when building a diagnostic experiment. This approach allows to extend the class of detectable faults, which in structural- analytical test generation methods is limited, as a rule, to a multitude of single constant faults. The use of finite state models in the construction of verification tests allows detecting any malfunction that changes the diagram of a healthy device and does not increase the number of states of the elements of the memory [10-11].

IV. GRAPH MODEL OF EASY TESTED FINITE STATE MACHINES

The methods of graph theory was used during development graphic models of easy tested finite state machine, methods of the theory of digital machines was used for the organization of diagnostic experiments with automaton models of functional modules.

The shift register (SR), as a standard, homogeneous and easy-to-test structure, is widely used in discrete devices of various uses. The structural methods of increasing the reliability of the test, the method of the path scanning, as well as some methods of converting automated device models into easily tested ones, ultimately, are reduced to providing in the diagnostic mode a controlled reconfiguration of memory elements in one shift register, the serial input and output of which are the external poles of the device. In addition, the device implemented on shift registers is easy to test, since shift register has a distinguishing minimum-length sequence. In this connection, the study of the diagnostic properties of sequences generated by shift register is an important task directly related to solving the problem of synthesis of easily tested devices and systems. automaton models for certain rules of construction trees and truncating its finite vertices [10].

V. REALIZATION OF EASY TESTED FINITE STATE MACHINE

Computer-aided design of digital devices (DD) based on specification which is represented in the form of hardware description language (HDL). Specialized data processing and control DD as a rule described by finite state machines (FSM). The form of finite state machine representation are state table (ST) and state diagram (SD). The FSM template, i.e. a way to describe of the 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 HDL-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 [9-11]. One of ways to describe DD models in the form of FSM using VHDL language is the FSM template, i.e. a way to describe of the 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 HDL-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 (Fig.1).

```

architecture FSM_MX of FSM_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) & TD1;
        end if;
    end if;
end process;
end architecture;

```

Fig. 1. The architecture fragment of the VHDL model of Moore FSM with shift register.

Let's extend the ST of Mealy finite state machine (Fig. 2) by adding shift register Sh. If Sh=1, the finite state machine operates in the setting mode in any given state, and if Sh=0, the finite state machine realizes the given algorithm.

Simulation of extended VHDL-models of the control finite state machine using Active-HDL confirmed the operability of this approach. Synthesis of these models using CAD XILINX ISE confirmed the receipt of testable structures.

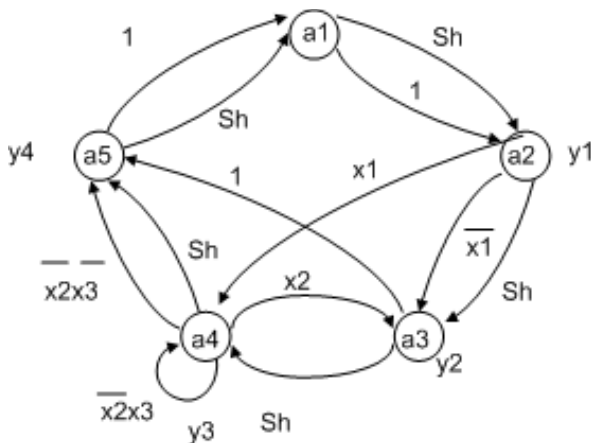


Fig. 2. Graph model of Moore FSM with shift register [10].

VI. CONCLUSION

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 finite state machine.

REFERENCES

- [1] P. Bibilo and V. Romanov, "Construction of compact tests for functional verification of VHDL descriptions of finite automata", *Control systems and machines*, №1, 2017, pp.35-45.
- [2] V. Semenets, G. Krivulya, A. Gorbatyuk, and A. Shilverst, *VHDL for designing computer devices*. Kharkov: Publishing House of Kharkov Technical University of Radio electronics, 2001, p.164.
- [3] P. Bibilo *Synthesis of logic circuits using the VHDL language*. M: SOLON-P, 2009, p.384.
- [4] A. Dolmatov and A. Petrov, *XILINX XST synthesis technology. Laboratory work 3*. Ekaterinburg: Publishing House of Ural State Technical University, 2005, p.23.
- [5] ISE. In-Depth Tutorial, UG695 (v14.1). 2012. Xilinx, p.150.
- [6] P. Parkhomenko and E. Soghomonyan, *Basics of technical diagnostics*. M: Energy, 1981, p.320.
- [7] F. Hennine, "Fault detecting experiments for sequential circuits," 5th Annual Symposium on Switching Circuit Theory and Logical Design, 1964, pp.95-100. doi: 10.1109/SWCT.1964.8.
- [8] V. Totsenko, *Algorithms for technical diagnostics of discrete devices*. M: Radio and communication, 1983, p.240.
- [9] M. Miroshnyk, T. Korytchinko, O. Demihev, V. Krylova, D. Karaman and I. Filippenko, "Practical methods for de Bruijn sequences generation using non-linear feedback shift registers," 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Slavske, 2018, pp. 1157-1161. doi: 10.1109/TCSET.2018.8336400.
- [10] M. Miroshnyk, Y. Pakhomov, E. German, A. Shkil, E. Kulak and D. Kucherenko, "Design automation of testable finite state machines," 2017 15th IEEE East-West Design and Test Symposium (EWDTS), Novi Sad, 2017, pp.1-6. doi: 10.1109/EWDTS.2017.8110034.
- [11] M. Miroshnyk, S. Poroshin, A. Shkil, E. Kulak, I. Filippenko and D. Kucherenko, "Design logical control units with finite state machine patterns," 2018 16th IEEE East-West Design and Test Symposium (EWDTS), Kazan, 2018, pp.203-208. doi: 10.1109/EWDTS.2018.8524869.
- [12] O. Drozd, M. Kuznietsov, O. Martynyuk and M. Drozd, "A method of the hidden faults elimination in FPGA projects for the critical applications," 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2018, Kiev, pp. 218-221. doi:10.1109/DESSERT.2018.8409131.
- [13] G. Kuchuk, V. Kharchenko, A. Kovalenko and E. Ruchkov, "Approaches to selection of combinatorial algorithm for optimization in network traffic control of safety-critical systems," 2016 IEEE East-West Design and Test Symposium (EWDTS), 2016, Yerevan, pp. 1-6. doi: 10.1109/EWDTS.2016.7807655.
- [14] I. Furman, M. Malinovsky, S. Bovchaliuk, A. Allashev, "Retrospective analysis of the development of a parallel-action PLC architecture," *Bulletin of Kharkiv National Technical University for Petro Vasylenka*, 2016, №176, pp.35-38.