

Метод синтеза тестовых программ для аналого-цифровых интегральных схем с применением сети автоматов

С. Г. Мосин

Казанский федеральный университет, smosin@ieee.org

Аннотация — Предложен метод синтеза тестовых программ в виде сети автоматов, обеспечивающих выполнение тестов с различной разрешающей способностью при проведении тестирования аналого-цифровых интегральных схем (АЦИС) на автоматическом тестовом оборудовании. Определены две базовые модели описания процесса иерархического тестирования АЦИС в виде сети автоматов. Представлено описание предложенного метода в виде набора базовых операций.

Ключевые слова — автоматизация проектирования, тестопригодное проектирование, тестовая программа, аналого-цифровая интегральная схема, сеть автоматов.

I. ВВЕДЕНИЕ

Тестирование, обеспечивающее отбраковку негодных изделий и гарантирующее качество партии произведенных микросхем, – одна из сложных и дорогостоящих операций в ходе производства АЦИС. Тестирование играет важную роль на ранних стадиях производства – до массового выпуска – при выявлении места и причины нарушения работы АЦИС. В зависимости от цели тестирования применяют контролирующие или диагностические тесты. С ростом функциональной и структурной сложности современных микросхем наблюдается переход к концепции тестопригодного проектирования (*DFT – Design-For-Testability*), когда разработку схемы и способов ее тестирования осуществляют одновременно, начиная с ранних стадий процесса проектирования. Следует отметить, что для цифровых подсхем разработаны и активно используются эффективные способы тестирования [1-3]. В свою очередь, организация тестирования аналоговых подсхем представляет существенную сложность в силу отсутствия универсальных методов и необходимости разработки или адаптации тестовых мероприятий для различных классов аналоговых схем или даже конкретного устройства. Отсюда следует, что тестирование АЦИС, основанное на совместной проверке цифровой и аналоговой подсхем, – сложная задача, которая требует согласования между собой методов тестирования, используемых для обеих подсхем [4-8]. В работе предложен метод формирования тестовых программ в виде сети автоматов, обеспечивающих выполнение тестов с различной разрешающей способностью при проведении тестирования АЦИС на автоматическом тестовом оборудовании, который соответствует методологии

тестопригодного проектирования аналого-цифровых интегральных схем.

II. МЕТОД СИНТЕЗА ТЕСТОВОЙ ПРОГРАММЫ В ВИДЕ СЕТИ АВТОМАТОВ

Привлечение подхода тестопригодного проектирования позволяет осуществить выбор наиболее подходящих методов тестирования для каждой из подсхем с учетом минимизации временных и стоимостных затрат на его проведение, обеспечивая требуемое покрытие неисправностей [9]. При построении и использовании диагностических тестов, как правило, применяют иерархическое тестирование, обеспечивающее поиск неисправностей на уровне отдельных компонентов, функциональных блоков (ФБ), подсхем или устройства в целом. Иерархическое тестирование АЦИС направлено на проверку работоспособности аналоговой подсхемы и ее ФБ, цифровой подсхемы и ее ФБ, схем преобразователей АЦП и ЦАП, а также всего устройства в целом. Элементы схемы, участвующие в иерархическом тестировании, образуют множество E . Процесс тестирования АЦИС сводится к подаче тестовых сигналов на вход проверяемого элемента и измерению выходных реакций, на основании которых принимается решение о корректности функционирования схемы, т.е. носит событийный характер [10]. Для описания и синхронизации процессов реализации тестирования АЦИС предложено использовать алгебраическую теорию автоматов, где для описания процессов тестирования отдельных элементов используется конечный автомат следующего вида:

$$S = (A, Z, W, \delta, \lambda, a_0),$$

в котором $A = \{a_0, \dots, a_m, \dots, a_M\}$ – множество состояний, определяющих стадии процесса тестирования; $Z = \{z_1, \dots, z_f, \dots, z_F\}$ – множество входных сигналов управления тестированием и осведомления; $W = \{w_1, \dots, w_g, \dots, w_G\}$ – множество выходных сигналов, управляющих подсистемами автоматического тестового оборудования (АТО); $\delta: A \times Z \rightarrow A$ – функция переходов, реализующая отображение $D_\delta \subseteq A \times Z$ в A ; $\lambda: A \times Z \rightarrow W$ – функция выходов, реализующая отображение $D_\lambda \subseteq A \times Z$ в W ; a_0 – начальное состояние автомата.

Процесс совместного тестирования элементов АЦИС предложено представить сетью автоматов

$$Net = (Z, \{S_i\}, W, \{f_i\}, \{\psi_i\}, g),$$

где $\{S_i = (A_i, Z_i, W_i, \delta_i, \lambda_i, a_{i0})\}$, $1 \leq i \leq n$ – множество автоматов; $\{f_i\}$ – множество функций соединения автоматов; $\{\psi_i\}$ – множество входных функций; g – выходная функция сети.

Применительно к иерархическому тестированию АЦИС можно выделить две основные модели описания процесса тестирования [11]:

1) сеть автоматов, описывающая множество независимых параллельных процессов, обеспечивающих тестирование отдельных компонентов или ФБ аналоговой и цифровой подсхем, а также преобразователей (рис. 1, а).

2) Сеть автоматов, описывающая множество параллельных процессов, обеспечивающих совместное, согласованное во времени тестирование ФБ аналоговой и цифровой подсхемы. Согласование во времени реализуется введением в сеть синхронизирующих состояний (рис. 1, б).

Частый случай Модели 2 – комплексное тестирование всей АЦИС с подачей тестовых сигналов на первичные входы аналоговой и цифровой подсхем и измерением откликов относительно первичных выходов.

Предлагаемый метод формирования тестовых программ в виде сети конечных автоматов с учетом представленных моделей включает следующий набор опе-

раций (рис. 2):

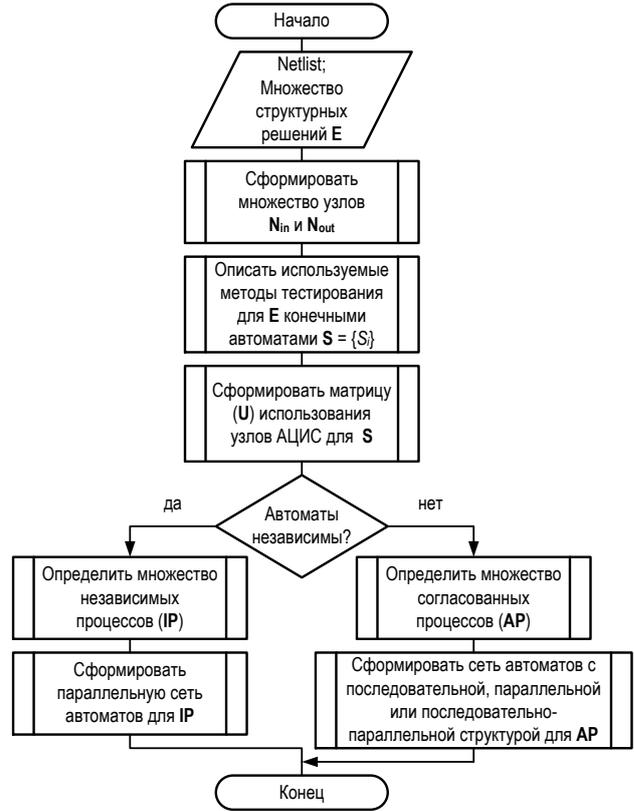


Рис. 2. Блок-схема формирования сети автоматов для тестовой программы

1) сформировать множество входных и выходных узлов АЦИС (N), используемых при тестировании, с разделением по функциональному признаку на информационные (подают тестовые данные) и управляющие (подают сигналы управления процессом тестирования)

$$N = N_{in} \wedge N_{out},$$

где $N_{in} = N_{in}^{data} \wedge N_{in}^{ctrl}$ – множество первичных информационных и управляющих входных узлов, $N_{out} = N_{out}^{data} \wedge N_{out}^{ctrl}$ – множество первичных информационных и управляющих выходных узлов.

2) Методы тестирования для всех элементов АЦИС из E описать конечными автоматами $\{S_i\}$, учитывая особенности тестируемой схемы и используемого тестового оборудования.

3) Для множества построенных автоматов $S = \{S_i\}$ сформировать матрицу использования узлов АЦИС во время тестирования

$$U = (u_{ij} \in [0,1]), 1 < i \leq sn, 0 < j \leq k,$$

где sn – суммарное количество узлов, используемых S во время тестирования, k – количество временных точек в ходе тестирования (рис. 3).

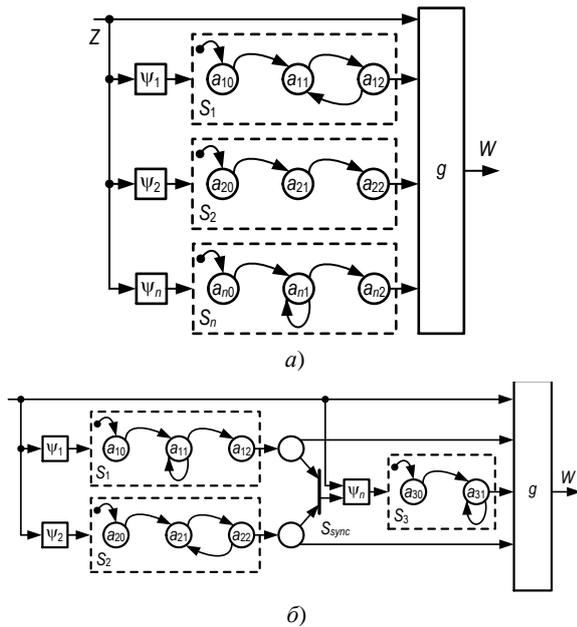


Рис. 1. Модель описания процесса тестирования в виде сети: независимых во времени автоматов (а), согласованных во времени автоматов (б)

Узел	Автомат	Время						
		t_0	t_1	t_2	t_3	t_4	...	t_k
n_1	S_1	1	1	1	0	0	...	0

	S_n	0	1	1	1	1	...	0
n_2	S_1	0	0	0	1	1	...	1
	S_2	0	1	1	0	0	...	0
n_i	S_1	0	0	1	1	0	...	0
	S_3	0	0	0	0	1	...	1

Рис. 3. Матрица использования узлов во время тестирования

4) Для Модели 1 определить множество независимых процессов, в которых в ходе тестирования отсутствует конфликт использования узлов

$$\mathbf{IP} = \{IP_i = \{S_p, S_q\} : \forall (p, q), p \neq q, N_{S_p} \cap N_{S_q} = \emptyset\},$$

$$N_{S_p} = \{n : z(n) \in Z \vee w(n) \in W, Z \Delta W \in S_p\}, \quad (1)$$

$$N_{S_q} = \{n : z(n) \in Z \vee w(n) \in W, Z \Delta W \in S_q\},$$

где N_{S_p} – множество тестовых узлов, используемых в конечном автомате S_p ; N_{S_q} – множество тестовых узлов, используемых в конечном автомате S_q .

5) Для множеств $IP_i \in \mathbf{IP}$ сформировать параллельную сеть автоматов Net , обеспечивающую одновременное тестирование отдельных компонентов или ФБ аналоговой и цифровой подсхем, а также преобразователей.

6) Выполнить декомпозицию матрицы \mathbf{U} в соответствии с множеством совместно тестируемых ФБ и подсхем. Для Модели 2 определить множество согласованных процессов, в которых в ходе тестирования отсутствует конфликт использования узлов

$$\mathbf{AP} = \{AP_i = \{S_p, S_q\} : \forall (p, q), p \neq q, \bar{n}_p \cap \bar{n}_q = \emptyset\},$$

$$\bar{n}_p = \{n_p(t) \in [0,1] : n_p \in N_{S_p}, 0 \leq t \leq t_k\}, \quad (2)$$

$$\bar{n}_q = \{n_q(t) \in [0,1] : n_q \in N_{S_q}, 0 \leq t \leq t_k\},$$

где \bar{n}_p – вектор использования узла n_p во время тестирования, \bar{n}_q – вектор использования узла n_q во время тестирования, t_k – конечное время тестирования.

7) Для множеств $AP_i \in \mathbf{AP}$ сформировать сеть автоматов Net , обеспечивающую совместное согласованное во времени тестирование ФБ аналоговой и цифровой подсхемы. Формируемая сеть может обладать последовательной, параллельной или последовательно-параллельной структурой и при необходимости включать синхронизирующие состояния, обеспечивающие согласование работы автоматов $S_j \in AP_i$ в ходе рабо-

ты в соответствии с выбранным методом тестирования.

Полученная в результате сеть автоматов обеспечивает детерминированный синтез сигналов управления процессом тестированием АЦИС с различной разрешающей способностью, а также сигналов активизации соответствующих тестовых сигналов.

Тестирование современных АЦИС выполняют на автоматическом тестовом оборудовании (АТО), процесс тестирования для которого задают с помощью тестовых программ. Такая программа определяет входные узлы микросхемы и номер тестового сигнала, формируемого внутренним генератором АТО, который подается на схему, выходные узлы, относительно которых будет происходить измерение выходных откликов, и их эталонные значения, а также длительность всех процессов. Таким образом, тестовые программы синхронизируют процессы подключения входных и выходных щупов АТО к контактным площадкам пластины или выводам интегральной схемы, подачи тестовых сигналов, измерения выходных откликов, сравнения их с эталонными значениями и принятия решения о корректности прохождения теста. Несмотря на существенные различия методов тестирования аналоговых и цифровых ИС, подготовка и исполнение тестовых программ для них носит схожий характер.

Трансляция полученной сети автоматов в команды и описания, соответствующие формату используемого тестового оборудования, лежит в основе написания тестовой программы для АТО. Эффективным инструментом подобного преобразования, которое может быть выполнено в автоматическом режиме, являются switch- или автоматное программирование.

Экспериментальное исследование предложенного метода проводилось на АЦИС, включающей аналоговый фильтр, цифровой компаратор и сканирующую цепь [11]. Комплексное тестирование устройства основано на иерархическом подходе, в ходе которого реализуются следующие процессы тестирования:

- 1) тестирование фильтра с использованием метода реконfigurирования в автогенератор [12];
- 2) функциональное тестирование цифрового компаратора;
- 3) тест сканирующей цепи битовой последовательностью «0011...»;
- 4) совместное тестирование функциональных блоков устройства.

Множество внешних узлов схемы, используемых для тестирования, включает $\mathbf{N} = \{Init, Mode, TDI, TDO, Test\}$, причем, $\mathbf{N}_{in}^{data} = \{TDI\}$, $\mathbf{N}_{in}^{ctrl} = \{Init, Mode, Test\}$, $\mathbf{N}_{out}^{data} = \{TDO\}$, где $Init$ – сигнал инициализации, $Mode$ – выбор режима работы сканирующей цепи, TDI и TDO – входные и выходные данные сканирующей цепи, $Test$ – сигнал выбора режима работы аналоговой подсхемы (нормальный/тестирование).

В иерархическом тестировании участвуют сканирующая цепь (e_1), цифровой компаратор (e_2) и аналоговый фильтр (e_3), которые образуют множество $\mathbf{E} = \{e_1, e_2, e_3\}$. Методы тестирования для каждого элемента из \mathbf{E} представим конечными автоматами ($\mathbf{S} = \{S_1, S_2, S_3\}$), которые формируют необходимые значения для управляющих сигналов из \mathbf{N}_{in}^{ctrl}

В соответствии с (1) $\mathbf{IP} = \{\{S_1\}, \{S_2\}, \{S_3\}\}$ – множество независимых процессов, в которых в ходе тестирования отсутствует конфликт использования узлов. Сеть автоматов для каждого множества IP_i будет представлена собственно конечным автоматом S_i , поскольку мощность полученных множеств IP_i равна единице. Иерархическое тестирование схемы в соответствии с моделью 1 возможно только в последовательном режиме для каждого элемента из \mathbf{E} .

В соответствии с (2) сформировано множество согласованных процессов, в которых в ходе тестирования отсутствует конфликт использования узлов: $\mathbf{AP} = \{\{S_1, S_3\}, \{S_2, S_3\}\}$. Сеть автоматов для каждого множества AP_i представлена параллельной структурой конечных автоматов $S_k \in AP_i$, которая обеспечивает одновременное тестирование элементов $e_k \in \mathbf{E}$, не требующее синхронизации процессов.

Для решения задачи внутрисхемной оценки попадания частоты самовозбуждения фильтра в допустимый диапазон значений, реализуемой в режиме тестирования, можно задействовать цифровой компаратор и сканирующую цепь. Таким образом, все элементы из множества \mathbf{E} будут совместно участвовать в процессе тестирования. $\mathbf{AP} = \{\{S_2, S_3\}, \{S_1\}, \{S_2\}, \{S_1\}\}$ – множество согласованных процессов без конфликта использования узлов. При формировании сети автоматов для \mathbf{AP} конечные автоматы каждого AP_i образуют параллельные структуры, которые соединяют последовательно. При этом выходы параллельных структур должны быть синхронизированы. В рассматриваемом случае выход параллельно исполняемых автоматов S_2 и S_3 поступает на вход дополнительного синхронизирующего состояния, которое обеспечивает ожидание завершения обоих процессов перед переходом в следующий последовательный процесс S_1 .

Описание и моделирование конечных автоматов и сети автоматов \mathbf{IP} и \mathbf{AP} выполнено в среде имитационного моделирования *Simulink* пакета математических расчетов *MATLAB*. Детерминированность и устойчивость формирования тестовых сигналов для схемы были подтверждены полученными результатами анализа сформированных моделей, описывающих процесс иерархического тестирования аналого-цифровой схемы.

III. ЗАКЛЮЧЕНИЕ

Тестовые программы, формируемые с привлечением аппарата сетей конечных автоматов в соответствии с предложенным методом, демонстрируют адекватность процессу иерархического тестирования АЦИС. Предложенный метод обеспечивает возможность автоматизации синтеза тестовых программ для проведения тестирования аналого-цифровых интегральных схем с различной разрешающей способностью, позволяя согласовать между собой используемые методы тестирования аналоговой и цифровой подсхем в рамках маршрута тестопригодного проектирования. Применение представленных моделей описания процесса иерархического тестирования АЦИС и метода формирования тестовых программ расширяет возможности автоматизации тестопригодного проектирования аналого-цифровых интегральных схем, обеспечивая снижение временных и стоимостных затрат на этапе тестирования.

ЛИТЕРАТУРА

- [1] Гришкин М., Лопаткин Г. С., Михайлов А. Н., Овсяников Д. А. Интерфейсный метод построения моделей входных воздействий для тестирования электронных цифровых модулей // Вопросы радиоэлектроники, 2013, Т. 1, № 1, с. 80-89.
- [2] Матросова А. Ю., Митрофанов Е. В. Синтез легко тестируемых последовательных схем // Вестник томского государственного университета. Управление, вычислительная техника и информатика. 2013, № 2 (23), С. 140-147
- [3] Матюшин Д. В., Курганский С. И. Моделирование системы конфигурирования и тестирования программируемой логической интегральной схемы // Системы управления и информационные технологии, 2011, Т. 46, № 4.1, С. 151-154.
- [4] Austin T. Creating a Mixed-Signal Simulation Capability for Concurrent IC Design and Test Program Development // Proc. International Test Conference, 1993, pp. 125 – 132.
- [5] Diamant P. E., Harrold S. J. Automatic Generation of Mixed-Signal Test Programs from Simulation Data // IEE Colloquium on Testing Mixed Signal Circuits, 1992, pp. 6/1 - 6/4.
- [6] Huang Wei-Hsing, Wey Chin-Long. ATPRG: an Automatic Test Program Generator Using HDL-A for Fault Diagnosis of Analog/Mixed-Signal Integrated Circuits // IEEE Transactions on Instrumentation and Measurement, Vol. 47, No. 2, 1998, pp. 426 – 431.
- [7] Keady A., Lyden C., Ryan J., Claffey S., Murphy N., Long B. Mixed-Signal Automatic Test Program Generation // Proc. of European Test Conference, 1993, pp. 528 – 529.
- [8] Richardson A., Lechner A., Olbrich T. Design for Testability Strategies for Mixed Signal & Analogue Designs-from Layout to System // Proc. IEEE Int. Conference on Electronics, Circuits and Systems, Vol. 2, 1998, pp. 425 - 432.
- [9] Mosin S. G. Design-for-testability automation of mixed-signal integrated circuits // Proc. IEEE 26th International SOC Conference (SOCC 2013), 2013, pp. 244–249.
- [10] Мосин С. Г. Структурные решения тестопригодного проектирования заказных интегральных схем // Информационные технологии. - 2008. - № 11. - С. 2–10.
- [11] Мосин С. Г. Метод формирования тестовых программ в виде сети конечных автоматов для проведения

Generating test programs for mixed-signal integrated circuits using automata network

S.G. Mosin

Kazan Federal University, smosin@ieee.org

Keywords — design automation, design-for-testability, test program, mixed-signal integrated circuit, automata network.

ABSTRACT

Testing plays important role at early stages of implementation before mass production at detection the place and reason of violation in a Circuit-under-Test (CUT) operation. Checking and diagnostic tests are used in dependence of the testing purpose. The active application of the design-for-testability concept (DFT) is observed nowadays with rising functional and structural complexity of the state-of-the-art IC, where circuit development is realized simultaneously with the corresponding tests at early stages of design process.

The effective test methods for digital IC have been developed and are actively used now. In turn, the testing of analog circuits is associated with the sufficient complexity due to absence of the universal methods and necessity to develop and adapt test techniques for different classes of analog circuits or even for individual devices. Hence, testing of mixed-signal integrated circuits (MSIC) based on the joint test of digital and analog subcircuits is complex task requiring the matching of test methods used for both subcircuits.

Efficient generation of test programs ensuring comprehensive testing and diagnosis of MSIC, effective utilization of automated test equipment, reduction of time and money costs on test preparation and execution, etc. is one of the challenges in the area of mixed-signal testing.

The method of test program generation in the form of automata network providing test execution with various resolving abilities for MSIC using the automated test equipment is proposed.

Hierarchical testing ensuring fault detection at the level of individual components, functional blocks (FB), subcircuits or of the whole device is typically used at preparation and application the diagnostic tests.

Hierarchical testing of MSIC is aimed at test of analog subcircuits and its FB, digital subcircuits and its FB, converter circuits ADC and DAC, as well as the entire device. The circuit elements involved in the hierarchical testing

form a set E. The testing process of MSIC is reduced to application of the test signal to the input of the checked element and measuring the output responses, based on which decision on the correctness of the circuit operation is made. So, the testing of analog and digital subcircuits can be considered as event-driven process. Algebraic automata theory is proposed to describe and synchronize the processes of MSIC test implementation. Finite-state machine is used to describe the processes of testing individual elements.

Two basic models for description of the test process of MSIC with respect to hierarchical testing are proposed.

1) Automata network describing a set of independent parallel processes which provide testing individual elements or FB of analog and digital subcircuits as well as converters.

2) Automata network describing a set of parallel processes ensuring joint and matched in time testing of analog and digital functional blocks. Matching in time is realized by including additional synchronization conditions into the network. The comprehensive testing of MSIC with application of test signals to the primary inputs of analog and digital subcircuits and responses acquisition at the primary outputs is a particular case of model 2.

The proposed method of test program generation in the form of automata network with a view of models considered above is described as the set of operations:

1. Prepare a set of input and output nodes of MSIC used at testing.
2. Describe the test methods for all elements of MSIC from the set E as finite state machines taking into account the features of circuit under test and the used test equipment.
3. Generate the matrix of MSIC nodes utilization during the testing for the set of constructed automata.
4. For model 1 define the set of independent processes in which there are no conflicts of node utilization during the testing.
5. Generate parallel automata network Net for the sets of independent processes ensuring the simultaneous testing

of individual elements or FB of analog and digital subcircuits as well as converters.

6. For model 2 define the set of matched processes in which there are no conflicts of node utilization during the testing.

7. Generate automata network Net for the sets of matched processes ensuring joint and matched in time testing of functional blocks of analog and digital subcircuits. The generated network may possess a serial, parallel or serial-parallel structure and, as required, includes synchronization conditions providing the matching of automata operation in accordance with selected test methods.

The automata network obtained as the result ensures deterministic synthesis of control signals for MSIC testing process with different resolving abilities, as well as activation signals for the used test signals.

Translation of obtained automata network into instructions and scripts relevant to the formats of the used test equipment underlies the generating test program for ATE. Switch- or automata programming is the effective tool for such translation which can be done in automated mode.

An application of the proposed method for mixed-signal circuit including analog filter, digital comparator and scan-chain is considered. The CUT comprehensive testing is based on the hierarchical approach which comprises the following test processes:

- 1) Analog filter testing using *OBIST* method;
- 2) Functional testing of the digital comparator;
- 3) The scan-chain testing by the bit sequence "0011...";
- 4) Joint testing of the CUT functional blocks.

Description and simulation of the generated finite-state machines and the automata network have been done in the *Simulink* toolbox of *MATLAB* software. The analysis results for the models describing the process of MSIC hierarchical testing have confirmed the determinacy and stability of test signal generation.

Experimental results have confirmed the efficiency of the proposed method for generating test programs for hierarchical testing of MSIC on automated test equipment. The proposed models for description of the test processes and the method of test program generation correspond to the methodology of design-for-testability automation of MSIC

and increase the efficiency of DFT concept application for mixed-signal integrated circuits design.

REFERENCES

- [1] Grishkin M., Lopatkin G. S., Mikhailov A. N., Ovsyannikov D. A. Interface-based method of generation the input stimuli for testing electronic digital modules – *Voprosy radioelektroniki*, 2013, no. 1, pp. 80-89 (in Russian).
- [2] Matrosova A. Yu., Mitrofanov E. V. Delay testable sequential circuit design – *Tomsk State University Journal of Control and Computer Science*, 2013, No. 2 (23), C. 140-147 (in Russian).
- [3] Matyushin D.V., Kurganskii S.I. Simulation of the configuring and testing system of the field programmable gate array - *Control systems and information technologies*, 2011, vol. 46, No. 4.1, pp. 151-154 (in Russian).
- [4] Austin T. Creating a Mixed-Signal Simulation Capability for Concurrent IC Design and Test Program Development // *Proc. International Test Conference*, 1993, pp. 125 – 132.
- [5] Diamant P. E., Harrold S. J. Automatic Generation of Mixed-Signal Test Programs from Simulation Data // *IEE Colloquium on Testing Mixed Signal Circuits*, 1992, pp. 6/1 - 6/4.
- [6] Huang Wei-Hsing, Wey Chin-Long. ATPRG: an Automatic Test Program Generator Using HDL-A for Fault Diagnosis of Analog/Mixed-Signal Integrated Circuits - *IEEE Transactions on Instrumentation and Measurement*, Vol. 47, No. 2, 1998, pp. 426 – 431.
- [7] Keady A., Lyden C., Ryan J., Claffey S., Murphy N., Long B. Mixed-Signal Automatic Test Program Generation - *Proc. of European Test Conference*, 1993, pp. 528 – 529.
- [8] Richardson A., Lechner A., Olbrich T. Design for Testability Strategies for Mixed Signal & Analogue Designs-from Layout to System - *Proc. IEEE Int. Conference on Electronics, Circuits and Systems*, Vol. 2, 1998, pp. 425 - 432.
- [9] Mosin S.G. Design-for-testability automation of mixed-signal integrated circuits - *Proc. IEEE 26th International SOC Conference (SOCC 2013)*, 2013, pp. 244–249.
- [10] Mosin S.G. Structural Solutions on Design-for-Testability of the Application Specific Integrated Circuits - *Information technologies*, 2008, No. 11, pp. 2–10 (in Russian).
- [11] Mosin S. G. The technique of test program generation in the form of automata network for testing the mixed-signal integrated circuits - *Dynamics of Complex Systems – XXI century*, 2015, vol. 9, No. 3, pp. 29-35 (in Russian).
- [12] Mosin S.G. A Built-in Self-Test Circuitry Based on Reconfiguration for Analog and Mixed-Signal IC - *Information Technology and Control*, vol. 40, no. 3, 2011, pp. 260 – 264.