KHARKOV NATIONAL UNIVERSITY OF RADIOELECTRONICS

Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2011)

Technically Co-Sponsored by

Sevastopol, Ukraine, September 9 – 12, 2011

IEEE EAST-WEST DESIGN AND TEST SYMPOSIUM 2011
ORGANISING COMMITTEE

General Chairs
V. Hahanov – Ukraine
Y. Zorian – USA

General Vice-Chairs
R. Ubar – Estonia
E. Pashkov – Ukraine
P. Prinetto – Italy

Program Chairs
S. Shoukourian – Armenia
D. Speranskiy – Russia

Program Vice-Chairs
M. Renovell – France
Z. Navabi – Iran

Steering Committee
M. Bondarenko – Ukraine
V. Hahanov – Ukraine
R. Ubar – Estonia
Y. Zorian – USA

Publicity Chairs
G. Markosyan – Armenia
S. Mosin – Russia

Public Relation Chair
V. Djigan – Russia

Program Committee
E. J. Aas – Norway
J. Abraham – USA
M. Adamski – Poland
A . Barkalov – Poland
R. Bazylevych – Ukraine
A. Chaterjee – USA
V. Djigan – Russia
A. Drozd – Ukraine
E. Evdokimov – Ukraine
E. Gramatova – Slovakia
A. Ivanov – Canada
M. Karavay – Russia
V. Kharchenko – Ukraine
K. Kuchukjan – Armenia
W. Kuzmicz – Poland
A. Matrosova – Russia
V. Melikyan – Armenia
L. Miklea – Romania
O. Novak – Czech Republic
Z. Peng – Sweden
A. Petrenko – Ukraine
D. Puzankov – Russia
J. Raik – Estonia
A. Romankevich – Ukraine
A. Ryjov – Russia
R. Seinauskas – Lithuania
S. Sharshunov – Russia
A. Singh – USA
J. Skobtsov – Ukraine
V. Tverdokhlebov –- Russia
V. Vardanian – Armenia
V. Yarmolik – Byelorussia

Organizing Committee
Yu. Gimpilevich – Ukraine
S. Chumachenko – Ukraine
E. Litvinova – Ukraine

EWDTS 2011 CONTACT INFORMATION

Prof. Vladimir Hahanov
Design Automation Department
Kharkov National University of Radio Electronics,
14 Lenin ave,
Kharkov, 61166, Ukraine.

Tel.: +380 (57)-702-13-26
E-mail: hahanov@kture.kharkov.ua
Web: www.ewdtest.com/conf/

9th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS 2011)

Sevastopol, Ukraine, September 9-12, 2011

The main target of the IEEE East-West Design & Test Symposium (EWDTS) is to exchange experiences in the field of design, design automation and test of electronic circuits and systems, between the technologists and scientists from Eastern and Western Europe, as well as North America and other parts of the world. The symposium aims at attracting attendees especially from the Newly Independent States (NIS) and countries around the Black Sea and Central Asia.

We cordially invite you to participate and submit your contribution(s) to EWDTS'11 which covers (but is not limited to) the following topics:

- Analog, Mixed-Signal and RF Test
- Analysis and Optimization
- ATPG and High-Level TPG
- Built-In Self Test
- Debug and Diagnosis
- Defect/Fault Tolerance and Reliability
- Design for Testability
- Design Verification and Validation
- EDA Tools for Design and Test
- Embedded Software Performance
- Failure Analysis, Defect and Fault
- FPGA Test
- HDL in test and test languages
- High-level Synthesis
- High-Performance Networks and Systems on a Chip
- Low-power Design
- Memory and Processor Test
- Modeling & Fault Simulation
- Network-on-Chip Design & Test
- Modeling and Synthesis of Embedded Systems
- Object-Oriented System Specification and Design
- On-Line Test
- Power Issues in Testing
- Real Time Embedded Systems
- Reliability of Digital Systems
- Scan-Based Techniques
- Self-Repair and Reconfigurable Architectures
- Signal and Information Processing in Radio and Communication Engineering
- System Level Modeling, Simulation & Test Generation
- Using UML for Embedded System Specification

CAD Session:

- CAD and EDA Tools, Methods and Algorithms
- Design and Process Engineering
- Logic, Schematic and System Synthesis
- Place and Route
- Thermal, Timing and Electrostatic Analysis of SoCs and Systems on Board
- Wireless Systems Synthesis
- Digital Satellite Television

The EWDTS'2011 will take place in Sevastopol, Ukraine. Sevastopol is a port city, located on the Black Sea coast of the Crimea peninsula. The city, formerly the home of the Soviet Black Sea Fleet, is now home to a Ukrainian naval base and facilities leased by the Russian Navy and used as the headquarters of both the Ukrainian Naval Forces and Russia's Black Sea Fleet.

The symposium is organized by Kharkov National University of Radio Electronics in cooperation with Sevastopol National Technical University and Tallinn University of Technology. It is technically co-sponsored by the IEEE Computer Society Test Technology Technical Council (TTTC) and financially supported by Virage Logic, Synopsys, Aldec, Kaspersky Lab, DataArt Lab, Tallinn Technical University, Cadence.

CONTENTS

Diagnosis Infrastructure of Software-Hardware Systems

Tiecoura Yves (Cote d'Ivoire)[1], Vladimir Hahanov[1], Omar Alnahhal (Livan)[1], M. Maksimov[1], Dmitry Shcherbin[1], Dmitry Yudin[1]
[1] *Computer Engineering Faculty, Kharkov National University of Radioelectronics, Kharkov, Ukraine, hahanov@kture.kharkov.ua*

Abstract

This article describes an infrastructure and technologies for diagnosis. A transactional graph model and method for diagnosis of digital system-on-chip are developed. They are focused to considerable decrease the time of fault detection and memory for storage of diagnosis matrix by means of forming ternary relations in the form of test, monitor, and functional component. The following problems are solved: creation of digital system model in the form of transaction graph and multitree of fault detection tables, as well as ternary matrices for activating functional components in tests, relative to the selected set of monitors; development of a method for analyzing the activation matrix to detect the faults with given depth and synthesizing logic functions for subsequent embedded hardware fault diagnosing.

1. TAB-model for diagnosing faulty components of SoC

Motivation is determined by the following: 1) the creation of simple and applicable models, methods and engines for diagnosis of multilayer software and hardware systems; 2) market appeal of matrix or table method for fault detection of SoC components (hardware and software) as the most effective one, which is focused on parallel processing and makes it possible to considerably reduce the time of diagnosis in the event of faults or non-functional mode.

Aim of this article is creation of model and method for considerable decrease the time of testing and memory for storage of diagnosis matrix by means of forming ternary relations (test – monitor – functional component) in a single table TAB: Tests – Assertions – Blocks. The problems are: 1) development of digital system model in the form of transaction graph, as well as multi-level model and engine for diagnosing software and hardware modules, based on the use of activation matrix of functional components on the tests relative to the selected monitor set [1-6]; 2) development of a method for analyzing activation matrix to detect faults with a given depth [4-7]; 3) Synthesis of logic functions for embedded fault diagnosis [8-11].

Model for testing a digital system is represented by the following transformation of the initial diagnosis equation, defined by xor-relation of the parameters <test – functionality – faulty blocks>:

$$T \oplus F \oplus B = 0 \to B = T \oplus F \to B = \{T \times A\} \oplus F \to$$
$$\to B = \{T \times A\} \oplus \{F \times m\},$$

which is transformed in ternary matrix relation of the components:

$$M = \{\{T \times A\} \times \{B\}\} \leftarrow M_{ij} = (T \times A)_i \oplus B_j.$$

Here, the coordinate of matrix (table) is equal to 1, if the pair test–monitor $(T \times A)_i$ checks (activates) faults of the functional block $B_j \in B$.

A model of digital system is presented in the transaction graph:

$$G =< B, A >, B = \{B_1, B_2, ..., B_i, ..., B_n\},$$
$$A = \{A_1, A_2, ..., A_j, ..., A_m\},$$

where sets of arcs – functional blocks and nodes – monitors for observation of the digital system variables are defined.

For diagnosing a set of test segments $T = \{T_1, T_2, ..., T_r, ..., T_k\}$ is overlapped on the graph model, which activates the transaction paths in the graph. In general, the testing model is represented by the Cartesian product $M =< B \times A \times T >$ that has the dimension $Q = n \times m \times k$. To reduce the amount of diagnostic information it is offered to assign a monitor to each test, which answers for visualization of an activation way for functional blocks, which makes it possible to decrease the dimension of model (matrix) to $Q = n \times k$ and retain all features of the triad relationship $M =< B \times A \times T >$. For the pair «test – monitor» not only one-to-one correspondence is possible $< T_i \to A_j >$, but functional $< \{T_i, T_r\} \to A_j >$ and

injective ones $< T_i \to \{A_j, A_s\} >$. Such variety of correspondences makes it possible to duplicate one test segment for different monitors, as well as assign several tests to the same monitor. At that the matrix cell $M_{ij} = \{0,1\}$ always preserves its dimension, equal to 1 bit.

The analytical generalized model for matrix diagnosing by using monitor engine, focused to achieving a given depth of fault detection, is presented in the following form:

$$M = f(G, L, T, B, A, t),$$
$$B = \{B_1, B_2, ..., B_i, ..., B_m\};$$
$$L = \{L_1, L_2, ..., L_i, ..., L_n\};$$
$$A(t) = \{A_1, A_2, ..., A_i, ..., A_k\};$$
$$A \subseteq L;\ G = L \times B;\ k \le n;$$
$$T = \{T_1, T_2, ..., T_i, ..., T_p\}.$$

Here B_i is a group of code statements, assigned to the node L_i (variable, register, counter, memory), determining its state; G is functionality, presented by the transaction graph $G = (L, A) \times B$ in the form of the Cartesian product of node and arc sets; A is a set of monitors, as a subset of transaction graph nodes $A \subseteq L$. The method for detecting faults of the functional blocks (FB) uses pre-built activation table (matrix) ATFB $M = [M_{ij}]$, where row is the relation between the test segment and a subset of activated blocks

$$T_i \to A_j \approx (M_{i1}, M_{i2}, ..., M_{ij}, ..., M_{in}), M_{ij} = \{0,1\},$$

observed on the monitor A_j. Column of the table describes the relation between the functional block, test segments and monitors $M_j = B_j \approx f(T, A)$. In the monitor engine the simulated time can be introduced, which complicates the activation matrix, indicating real or simulated cycle, on which monitoring of the node or functional block states on the test segment $A_j = f(T_i, B_j, t_j)$ is performed.

For fault diagnosis at the modeling stage the generalized response (column vector) $m = \{m_1, m_2, ..., m_i, ..., m_p\}$ of monitor engine A on the test segments T is determined, by forming $m_i = f(T_i, A_i)$. Searching faulty functional blocks is based on the definition of xor-operation between the assertion state vector and columns of the functional failures table $m \oplus (M_1 \vee M_2 \vee ... \vee M_j \vee ... \vee M_n)$.

The choice of solution is realized by using a method for xor-analyzing columns, to choose a set of vectors B_j with minimum number of unit coordinates:

$$B = \min_{j=1,n} [B_j = \sum_{i=1}^{p} (B_{ij} \overset{p}{\underset{i=1}{\oplus}} m_i)],$$

forming the functional blocks with faults, verified on the test segments. In addition to the model for matrix diagnosing is necessary to describe the following important features of the matrix:

$$1)\ M_i = (T_i - A_j);$$
$$2)\ \overset{m}{\underset{i=1}{\vee}} M_{ij} \to \forall M_j = 1;$$
$$3)\ M_{ij} \overset{n}{\underset{j=1}{\oplus}} M_{rj} \ne M_{ij};$$
$$4)\ M_{ij} \overset{k}{\underset{i=1}{\oplus}} M_{ir} \ne M_{ij};$$
$$5)\ \log_2 n \le k \leftrightarrow \log_2 |B| \le |T|$$
$$6)\ B_j = f(T, A) \to B \oplus T \oplus A = 0.$$

The features means: 1) Each row of the matrix is a match or subset of the Cartesian product (test – monitor). 2) Disjunction of all rows of the matrix gives a vector equal to one over all the coordinates. 3) All rows are distinct, which eliminates the test redundancy. 4) All columns of the matrix are distinct, which exclude the existence of equivalent faults. 5) The number of matrix rows must be greater than the binary logarithm of the number of columns that determines the potential diagnosability of all blocks. 6) Diagnosis function for block depends on the complete test and monitors, which must be minimized without reduction the diagnosability.

2. Design diagnosability

As for the quality of the model for diagnosing functional failures, it shows the efficiency of the use of pair (test, assertions) for a given diagnosis depth. Evaluation of the model quality is functionally dependent on the length of the test $|T|$, a number of assertions $|A|$, and a number of detected blocks with functional failures N_d on the total number of software blocks N:

$$Q = E \times D = \frac{]\log_2 N[}{|T| \times |A|} \times \frac{N_d}{N}.$$

The diagnosis efficiency is the ratio of the minimum number of bits needed for identification (recognition) of all the blocks to the real number of code bits, presented by the product of test length by number of assertions in each of them. If the first fraction of estimate is equal to 1 and all the blocks with functional failures

are detected ($N_d = N$), it means a test and assertions are optimal that gives value of 1 for quality criterion of diagnosis model.

Evaluation of the structure quality for the design code is interesting from the perspective of the diagnosability of software blocks. The purpose of the analysis is determining the quantitative assessment of the graph structure and a node for placement of assertion monitors, which make possible to obtain maximum diagnosis depth of functional failures of the software components. It is important not controllability and observability as in testability, but the distinguishability of the software components with functional failures, in the limit it is zero blocks with the equivalent (indistinguishable) failures. Such an assessment may be useful to compare the graphs implemented the same functionality. It is necessary to evaluate the graph structure from the position of potential detection depth of software functional failures. One possible option is diagnosability of ABC-graph as a function depending on adjacent arcs of each node (the number N_n), one of which is incoming, other one is outgoing. These arcs form paths without reconvergent fan-outs and branching (N is total number of arcs in the graph):

$$D = \frac{N - N_n}{N}.$$

Each node, joining two arcs entering in the number N_n, is called transit one. The estimation N_n is the number of indistinguishable functional failures of the software components. Potential locations of monitors for distinguishing functional failures are transit nodes. Given the above estimation of the diagnosability D the diagnosis model quality for software takes the form:

$$Q = E \times D = \frac{]\log_2 N[}{|T| \times |A|} \times \frac{N - N_n}{N}.$$

Rules for synthesis of diagnosable software: 1) Test (testbench) must create minimum number of one-dimensional activation paths, covered all the nodes and arcs of ABC-graph. 2) The base number of monitor-assertions equals the number of end nodes of the graph with no outgoing arcs. 3) An additional monitor can be placed in each node, which has one incoming and one outgoing arc. 4) Parallel independent code blocks have n monitors and a single test or one integrated monitor and n tests. 5) Serially connected blocks have one activation test for serial path and n-1 monitor or n tests and n monitors. 6) The graph nodes, which have different numbers of input and output arcs, create the conditions for the diagnosability of current section by one-dimensional activation tests without having to install additional monitors. 7) The set of test segments (testbench) has to be 100% functional coverage, given by the nodes of ABC-graph. 8) Diagnozability function is directly proportional to the test length, the number of assertions and inversely proportional to the binary logarithm of the number of software blocks:

$$D = \frac{N - N_n}{N} = f(T, A, N) = \frac{|T| \times |A|}{]\log_2 N[}.$$

Diagnosability as a function depending on the graph structure (for software), test and assertion monitors can always be reduced to unit value. For this purpose there are two alternative ways. The first one is increase of test segments, activating new paths for distinguishing equivalent faults without increasing assertions, if the software graph structure allows the potential links. The second is placement of additional assertion monitors in transit nodes of the graph. A third hybrid variant is possible, based on the joint application of two above ways. The relation of three components (the number of software blocks, the power of assertion engine and the test length) forms the set of optimal solutions

$$D = 1 \to \frac{|T| \times |A|}{]\log_2 N[} = 1 \to]\log_2 N[= |T| \times |A|,$$

when quality of the diagnosis and diagnosability model is equal to 1. It can be useful for choosing an quasioptimal variant of alternative way for providing the full distinguishability of software functional failures on a pair $|T| \times |A|$.

3. Multilevel method (engine) for diagnosis of digital system

Process model or method for searching faults by diagnosis multitree is reduced to creation of the engine (Fig. 1) for traversal of tree branch on the depth, specified by the user:

$$B_j^{rs} \oplus A^{rs} = \begin{cases} 0 \to \{B_j^{r+1,s}, R\}; \\ 1 \to \{B_{j+1,}^{rs}, T\}. \end{cases}$$

Here vector xor-operation is executed between the columns of the matrix and the output response vector A^{rs}, which is determined by the functionality response taken from the monitors (assertions or bits of boundary scan register) under all test patterns. If at least one coordinate of vector xor-sum is equal to zero $B_j^{rs} \oplus A^{rs} = 0$ then one of the following action is per-

formed: the transition to the activation matrix of lower level $B_j^{r+1,s}$ or repair of the functional block B_j^{rs}. At that analysis is carried out, what is the most important: 1) time – then repair of faulty block is performed; 2) money – then a transition down is carried out to specify the fault location, because replacement of smaller block substantially decreases the repair cost. If at least one coordinate of the resulting xor-sum vector is equal to one $B_j^{rs} \oplus A^{rs} = 1$, then transition to the next matrix column is performed. When all coordinates of the assertion monitor vector are equal to zero $A^{rs} = 0$, fault-free state of a device is fixed. If all vector sums are not equal to zero $B_j^{rs} \oplus A^{rs} = 1$, it means a test, generated for check the given functionality, has to be corrected.

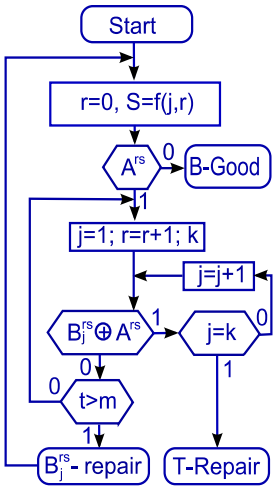

Fig. 1. Engine for traversal of diagnosis multitree

Thus, the graph shown in the Fig. 1, allows realizing efficient infrastructure IP for complex technical systems. The advantages of the engine, which is invariant to the hierarchy levels, are the simplicity of preparation and presentation of diagnostic information in the form of minimizing activation table of functional blocks on the test patterns.

4. Verification of models and method for diagnosis

To illustrate the performance of the proposed model and method the functionalities of three modules of the digital filter of Daubechies [11] are considered below. The first component is component Row_buffer; its transaction graph, based on RTL-model, shown in Fig. 2. Nodes are presented by the states of variables and monitors, which are responsible for node incoming transactions or arcs, corresponding to the functional blocks.

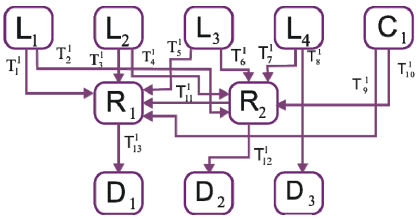

Fig. 2. Component Row_buffer of transaction graph

An activation table for functional blocks is generated by using graph, obtained in the simulation. Table rows are activation paths for blocks to the given monitor-node. A table is a coverage all columns or functional blocks by rows of paths. In this case it should not have at least two identical columns. The difference of table is creation of the pair <test – observed node>, making it possible to considerably reduce the dimension of the table with 100% detection of all faulty blocks. The main feature of the proposed model is the ability to describe the following relations by using the table: distinct tests – one node, one test – distinct nodes:

A_{ij}	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}
$t_1 \to D_3$	.	.	.	.	.	.	.	1	.	.	.	.	.
$t_2 \to D_1$	1	.	.	.	.	.	.	.	.	.	.	.	1
$t_3 \to D_1$	.	.	1	.	.	.	.	.	.	.	.	.	1
$t_4 \to D_1$	.	.	.	.	1	.	.	.	.	.	.	.	1
$t_5 \to D_1$	.	.	.	.	.	.	.	.	.	.	1	.	1
$t_6 \to D_1$	.	.	.	.	.	.	.	.	1	.	.	.	1
$t_7 \to D_2$	.	1	.	.	.	.	.	.	.	.	.	1	.
$t_8 \to D_2$	.	.	.	1	.	.	.	.	.	.	.	1	.
$t_9 \to D_2$	.	.	.	.	.	1	.	.	.	.	.	1	.
$t_{10} \to D_2$	.	.	.	.	.	.	1	.	.	.	.	1	.
$t_{11} \to D_2$	.	.	.	.	.	.	.	.	.	1	.	1	.

The use of the activation matrix of functional blocks (transaction graph) and xor-method for detecting faults allows synthesizing logic functions for the combination circuit, which determines number of functional block with semantic errors in process of simulation:

$$D_3 = T_8^1;$$

$$D_1 = T_{13}^1 T_1^1 \vee T_{13}^1 T_3^1 \vee T_{13}^1 T_5^1 \vee T_{13}^1 T_{11}^1 \vee T_{13}^1 T_9^1;$$

$$D_2 = T_{12}^1 T_2^1 \vee T_{12}^1 T_4^1 \vee T_{12}^1 T_6^1 \vee T_{12}^1 T_7^1 \vee T_{12}^1 T_{10}^1.$$

This feature is possible due to the lack of equivalent faults or identical columns in the activation

matrix. Therefore, fixing the actual state of monitors at the nodes D_1, D_2, D_3 on 11 test patterns makes it possible to unambiguously identify an incorrect functional module by performing xor-operation between the assertion vector and the columns of activation matrix. Zero value of all the coordinates for the result of xor-operations determines the number of the column corresponding to a faulty module. The implementation of the model and method in a logical function allows identifying the faulty block before the completion of the diagnostic experiment, if it is possible. This means significant savings of diagnosis time for certain types of faults. For instance, the test-monitor $t_1 \to D_3$ allows identifying a fault of the block B_8 at the first test.

Second test case for the practical use of the activation model and xor-method for searching faults is presented below. Synthesis of the diagnosis matrix for discrete cosine transform module from Xilinx library in the form of functional coverage is shown in Listing 1.

Listing 1. Part of functional coverage

```
c0: coverpoint xin
{
bins minus_big={[128:235]};
bins minus_sm={[236:255]};
bins plus_big={[21:127]};
bins plus_sm={[1:20]};
bins zero={0};
}
c1: coverpoint dct_2d
{
bins minus_big={[128:235]};
bins minus_sm={[236:255]};
bins plus_big={[21:127]};
bins plus_sm={[1:20]};
bins zero={0};
bins zero2=(0=>0);
}
endgroup
```

For all 12 modules the transaction graphs, activation tables, and logic functions are developed for testing and fault detection in the discrete cosine transform. Graph with the activation matrix and logic function (Fig. 3) are presented below.

This graph is associated with the following diagnosis matrix:

A_{ij}	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}
$P_1 \to F_7$	1	.	1	.	1	.	1	.	.	.	.	.	.	.
$P_2 \to F_8$	.	1	.	1	1	.	.	1	.	.	.	.	.	.
$P_3 \to F_9$	1	.	1	.	.	1	.	.	.	.	1	.	.	.
$P_4 \to F_{10}$	.	1	.	1	.	1	.	.	.	.	.	1	.	.
$P_5 \to F_{12}$	1	.	1	.	1	.	.	.	1	.	.	.	1	.
$P_6 \to F_{13}$	.	1	.	1	.	1	.	.	.	1	.	.	.	1
$P_1 \to F_2$	1	.	.	.	.	.	.	.	.	.	.	.	.	.
$P_2 \to F_3$	.	1	.	.	.	.	.	.	.	.	.	.	.	.

The system of diagnosis functions is presented below:

$$F_7 = T_1^1 T_3^1 T_5^1 T_7^1;\ F_8 = T_2^1 T_4^1 T_5^1 T_8^1;$$

$$F_9 = T_{11}^1 T_6^1 T_1^1 T_3^1;$$

$$F_{10} = T_4^1 T_5^1 T_6^1 T_{12}^1;\ F_{12} = T_1^1 T_3^1 T_5^1 T_9^1 T_{13}^1;$$

$$F_{13} = T_2^1 T_4^1 T_6^1 T_{10}^1 T_{14}^1;$$

$$F_2 = T_1^1;\ F_3 = T_2^1;$$

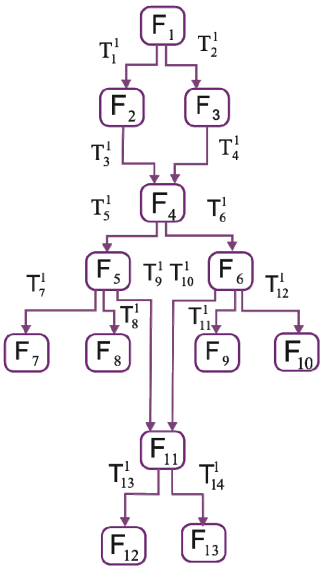

Fig. 3. Transaction graph of main-RTL module

Fragment of monitor engine is presented by Listing 2.

Listing 2. Code fragment of monitor engine

```
sequence first( reg[7:0] a, reg[7:0]b);
 reg[7:0] d;
 (!RST,d=a)
##7 (b==d);
 endsequence
property f(a,b);
 @(posedge CLK)
 // disable iff(RST||$isunknown(a)) first(a,b);
!RST |=> first(a,b);
endproperty
odin:assert property (f(xin,xa7_in))
 // $display("Very good");
else $error("The end, xin =%b,xa7_in=%b", $past(xin,
7),xa7_in);
```

Testing of discrete cosine transformation in the environment Riviera, Aldec detects incorrectness in seven rows of HDL-models:

```
//add_sub1a <= xa7_reg + xa0_reg;//
```

Subsequent correcting code allowed obtaining the following code (Listing 3).

Listing 3. Corrected code fragment

```
   add_sub1a <= ({xa7_reg[8],xa7_reg} +
{xa0_reg[8],xa0_reg});
   add_sub2a <= ({xa6_reg[8],xa6_reg}
+{xa1_reg[8],xa1_reg});
   add_sub3a <= ({xa5_reg[8],xa5_reg}
+{xa2_reg[8],xa2_reg});
   add_sub4a <= ({xa4_reg[8],xa4_reg} +
{xa3_reg[8],xa3_reg});
   end
   else if (toggleA == 1'b0)
   begin
   add_sub1a <= ({xa7_reg[8],xa7_reg} -
{xa0_reg[8],xa0_reg});
   add_sub2a <= ({xa6_reg[8],xa6_reg} -
{xa1_reg[8],xa1_reg});
   add_sub3a <= ({xa5_reg[8],xa5_reg} -
{xa2_reg[8],xa2_reg});
   add_sub4a <= ({xa4_reg[8],xa4_reg} -
{xa3_reg[8],xa3_reg});
```

5. Conclusion

1. Infrastructure and technology for cyberspace analysis are presented. Proposed transactional graph model and method for diagnosis of digital systems-on-chips are focused to considerable reducing the time of fault detection and memory for storing the diagnosis matrix by forming ternary relations in the form of test, monitor, and functional component.

2. An improved process model for definition of functional failures in software or hardware is proposed. It is characterized by using the xor-operation, which makes it possible to improve the diagnosis performance for single and multiple faults (functional failures) on the basis of parallel analysis of the fault table, boundary scan standard IEEE 1500, and vector operations and, or, xor.

3. A model for diagnosing the functionality of digital system-on-chip in the form of multitree and method for tree traversal, implemented in the engine for detecting faults with given depth, are developed. They considerably increase the performance of software and hardware Infrastructure IP.

4. Test verification of proposed diagnosis method is performed by three real case studies, presented by SoC functionalities of a cosine transform filter, which showed the consistency of the results in order to minimize the time of fault detection and memory for storing diagnostic information, as well as increase the diagnosis depth for digital products.

6. References

[1] Technical diagnosis basics / Editor P.P. Parchomenko.- M.: Energy.– 1976.– 460 p.

[2] Parchomenko P.P., Sogomonyan E.S. Technical diagnosis basics (Optimization of diagnosis algorithms, hardware tools) / Editor P.P. Parchomenko.– M.: Energy.– 1981.– 320 p.

[3] Infrastructure for brain-like computing / M.F. Bondaryenko, O.A. Guz, V.I. Hahanov, Yu.P. Shabanov-Kushnaryenko.– Kharkov: Novoye Slovo.– 2010.– 160 p.

[4] Design and Verification of digital systems on chips / V.I. Hahanov, I.V. Hahanova, E.I. Litvinova, O.A. Guz.– Kharkov: Novoye Slovo. – 2010. – 528 p.

[5] Semenets V.V., Hahanova I.V., Hahanov V.I. Design of digital systems by using VHDL language.– Kharkov: KHNURE.– 2003.– 492 p.

[6] Hahanov V.I., Hahanova I.V. VHDL+Verilog = synthesis for minutes. – Kharkov: KHNURE.– 2006.– 264 p.

[7] IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture IEEE Std 1149.7-2009. – 985 p.

[8] Da Silva F., McLaurin T., Waayers T. The Core Test Wrapper Handbook. Rationale and Application of IEEE Std. 1500™. –Springer.– 2006.– XXIX.– 276 p.

[9] Marinissen E.J., Yervant Zorian. Guest Editors' Introduction: The Status of IEEE Std 1500.–IEEE Design & Test of Computers.– 2009.– No26(1).– P.6-7.

[10] Benso A., Di Carlo S., Prinetto P., Zorian Y. IEEE Standard 1500 Compliance Verification for Embedded Cores // IEEE Trans. VLSI Syst.– 2008.– No 16(4).– P. 397-407.

[11] Hahanov V.I., Litvinova E.I., Chumachenko S.V., Guz O.A. Logic associative computer // Electronic simulation.– No 1.– 2011.– P.73-83.

Camera-ready was prepared in Kharkov National University of Radio Electronics
Lenin Ave, 14, KNURE, Kharkov, 61166, Ukraine

Approved for publication: 26.08.2011. Format 60×841/8.
Relative printer's sheets: 42. Circulation: 150 copies.
Published by SPD FL Stepanov V.V.
Ukraine, 61168, Kharkov, Ak. Pavlova st., 311

Матеріали симпозіуму «Схід-Захід Проектування та Діагностування – 2011»
Макет підготовлено у Харківському національному університеті радіоелектроніки
Редактори: Володимир Хаханов, Світлана Чумаченко, Євгенія Литвинова
Пр. Леніна, 14, ХНУРЕ, Харків, 61166, Україна

Підписано до публікації: 26.08.2011. Формат 60×84$^{1}/_{8}$.
Умов. друк. Арк. 42. Тираж: 150 прим.
Видано: СПД ФЛ Степанов В.В.
Вул. Ак. Павлова, 311, Харків, 61168, Україна