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

A Diagnostic Model for Detecting Functional Violation in HDL-Code of System-on-Chip

Ngene Christopher Umerah, Vladimir Hahanov
Computer Engineering Faculty, Kharkov National University of Radioelectronics, Kharkov, Ukraine, umerahlove@yahoo.co.uk

Abstract

The design of System-on-Chip (SoC) is becoming more difficult by the day with the increase in complexity of consumer requirements and time-to-market pressures. The use of HDLs in the design of digital system has become more ubiquitous and challenging as ever if timely delivery of product with increased yield is to be achieved. A technological and process-efficient models and methods for diagnosis of functional violations in software and/ or hardware products are proposed. The assertion-based transaction graph used in this model can be transformed into a tabular data structure that focuses on parallel execution of logic operations when searching for defective components or blocks with functional violation in HDL models.

1. Introduction

The problem of synthesis or analysis of the components of any system can be formulated by the interaction of its model with input patterns and reactions in a space. This is similar to determining the symmetric difference of the three components – model, test patterns and reaction of the model when the test pattern is applied to it. The distance or relationship between two or more objects in a space can be determined by the well known Cartesian or polar coordinate systems. In case of Boolean variables the Hamming distance has been used to determine how close or far apart are two binary variables of any length. Using Hamming distance results in cardinality or a number; but the beta metric proposed in this paper gives a broader view of how two or more binary variables of any length relates to each other in a cyberspace. And Hamming distance is only a particular case of beta metric. In this paper we define cyberspace as a set of interacting information processes and phenomena which conforms to a predefined metric using computer systems and networks as a vehicle. This paper is organised as follows: In section 2 we discuss the beta metric used in defining the relationship between objects in a space. The analysis of interactive graph of components of technical diagnostics is presented in section 3 followed by a model to search for functional violation in HDL-code in section 4.

2. Beta Metric System

The interaction of a model of a system (F) and the test patterns (T) in cyberspace gives rise to a reaction (L). This relationship can be expressed as the symmetric difference between the three components which is analogous to XOR-operation on Boolean variables.

$$f(F,T,L)=\varnothing \rightarrow F\Delta T\Delta L=\varnothing \quad (1)$$

The components of the space can be represented by a k-dimensional (tuple) vector in a binary alphabet:

$$a=(a_1,a_2,...,a_j,...,a_k),\ a_j=\{0,1\}. \quad (2)$$

A zero- vector is a k-tuple with all coordinates equal to zero: $a_j=0,\ j=\overline{1,k}$.

The β-metric of binary cyberspace is uniquely determined by the equation, which forms a zero-vector for the XOR-sum of the Hamming distances between the non-zero and finite number of points (objects) in a cycle:

$$\beta=\bigoplus_{i=1}^{n} d_i=0. \quad (3)$$

The Hamming distance between two objects (vectors) *a* and *b* is defined as the derivative of the vector:

$$d_i=d(a,b)=a_j\bigoplus_{j=1}^{k} b_j \quad (4)$$

In other words: the metric β of a vector-logical space is a logical binary number equal to a zero vector of the XOR-sum of distances between a finite

number of points (vertices) of the graph, which forms a loop. The sum of n-dimensional binary vectors that define the coordinates of the cycle is equal to a zero vector. This definition of metric operates on the relationships, which helps to reduce the system of axioms from three to one and extends its effect on any designs of n-dimensional cyberspace. The classic reference metric for determining the interaction of one, two and three points in a vector logical space is a special case of β-metric when $i = 1,2,3$ correspondingly:

$$M=\begin{cases} d_1 = 0 \leftrightarrow a = b; \\ d_1 \oplus d_2 = 0 \leftrightarrow d(a,b) = d(b,a); \\ d_1 \oplus d_2 \oplus d_3 = 0 \leftrightarrow d(a,b) \oplus d(b,c) = d(a,c). \end{cases}$$

The metric β of the multi-valued cyberspace, where each coordinate of the vector (object) is defined in the alphabet, constituting the Boolean on the universe of primitives with a power of p:

$$a_j = \{\alpha_1, \alpha_2, ..., \alpha_r, ..., \alpha_m\}, m = 2^p,$$

is equal to the $\varnothing$-vector (for all coordinates), the symmetric difference of instances between a finite number of points that form a cycle:

$$\beta = \underset{i=1}{\overset{n}{\Delta}} d_i = \varnothing \qquad (5)$$

The equality to empty vector of symmetric difference of set-theoretic interactions (5) emphasizes the equivalence of the components (distances), which form the equation, where the only co-ordinate operation $d_{i,j} \Delta d_{i+1,j}$ used, for example, is a four-valued Cantor model , determined by the corresponding Δ - tables:

Table 1. Cantor model and Set theoretic operations

Δ	0	1	x	$\varnothing$
0	$\varnothing$	x	1	0
1	x	$\varnothing$	0	1
x	1	0	$\varnothing$	x
$\varnothing$	0	1	x	$\varnothing$

$\cap$	0	1	x	$\varnothing$
0	0	$\varnothing$	0	$\varnothing$
1	$\varnothing$	1	1	$\varnothing$
x	0	1	x	$\varnothing$
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

$\cup$	0	1	x	$\varnothing$
0	0	x	x	0
1	x	1	x	1
x	x	x	x	x
$\varnothing$	0	1	x	$\varnothing$

a	0	1	x	$\varnothing$
$\tilde{a}$	1	0	$\varnothing$	x

Shown here also is the truth table for the other basic set-theoretic operations, used in the text. The number of primitive characters that forms a closed loop in relation to set-theoretic coordinate alphabet operations can be increased. The power of the alphabet (Boolean) is given by the expression $m = 2^p$, where p – number of primitive characters. This metric is not only of theoretical interest, but has a practical focus on synthesis and classification problems of technical diagnostics by creating a model XOR-relations on the set of four main components.

3. Interactive graphs of components of technical diagnostics

Synthesis procedure for tests, fault simulation and defect detection can be reduced to XOR-relationships on a graph (Figure 1) for the complete interaction of the four vertices (the device functional model, physical device, test patterns and defects) $G = \{F, U, T, L\}$

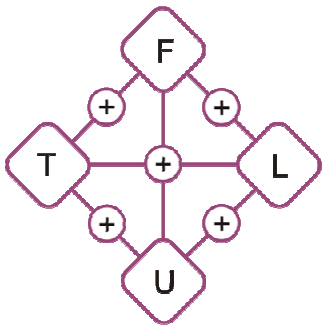

Figure 1. Interactive graph of four components

Such a graph gives rise to four basic triangles that form the 12 practically useful triad relationships, which formulates the problem of technical diagnostics:

Table 2. Relationship between models, test patterns, physical devices and defects

1	2	3	4
T⊕F⊕L= 0	T⊕L⊕U = 0	T⊕F⊕U = 0	F⊕L⊕U = 0
1) T = F⊕L 2) F = T⊕L 3) L = T⊕F	4) T = L⊕U 5) L = T⊕U 6) U = T⊕L	7) T = F⊕U 8) F = T⊕U 9) U = T⊕F	10) F = L⊕U 11) L = F⊕U 12) U = F⊕L

The introduction of vertex U in the graph between the components of technical diagnostics extends the functionality of the model giving rise to new properties of this system. Introduction of the new

vertex into the structure must have strong arguments in favour of its usefulness. The problems described by these formulae can be classified into four groups as follows.

Group 1 - theoretical experiments (on the functional model), without the physical device: 1) Synthesis of the test on the functional model for a specified list of faults 2) The functional model is based on a given test, and the list of faults. 3) Modelling of functional fault on a given test vectors.

Group 2 - the actual experiments (on the device), with no functional model: 4) Synthesis of the test by means of a physical emulation of defects in the device. 5) Definition of fault lists present on the device during diagnostic experiment. 6) Verification of tests and defects during experiment on a real device.

Group 3 - test experiments (verification), with no defects: 7) Test synthesis by means of comparing the results of simulation of the functional model and real device. 8) Synthesis of the functionality of a real device and a given test. 9) Verification test and model functionality with respect to the real device with existing faults.

Group 4 – carry out experiments on the device during its operation under the influence of operators: 10) Check for correct behaviour of a real device on the existing or specified defects. 11) Check the workability of the device in relation to the existing model during operation.

12) Verification of functionality and list of defects in relation to the behaviour of a real device. The most popular tasks of the above list are: 1, 3, 5, 8, 9.

All designs presented in these relationships, have a remarkable property of reversibility. A component that is calculated using the other two can be used as an argument to determine any of the two original ones. That is why we can speak of a transitive reversibility of each of the triad of relations on a complete graph, when any two components are always uniquely possible to restore or to determine the third. In this situation the format for the representation of each component must be identical in shape and dimension (vectors, matrices). Based on the proposed metrics and testing models a more detailed methods for diagnosing defects or functional violation are further discussed in the next section.

4. A Model to Search for functional violation in HDL-code

In order to discuss this search model we shall consider an equation describing how four components (functionality, test vectors, physical device and fault list) interact in a space.

$$f(F,T,L,U) = 0 \rightarrow F \oplus T \oplus L \oplus U = 0, \qquad (6)$$

Transforming expression (6) we have:

$$L = (T \oplus F) \oplus (T \oplus U) \qquad (7)$$

The Diagnosis of defects (functional violations) is reduced to comparing the model $(T \oplus F)$ and the physical experiment $(T \oplus U)$, which generates a list of functional violations L that is present in the object of diagnosis. The process model formula for finding the block F_i with functional violation is reduced to the choice of solutions through the identification of XOR-interaction between the three components:

$$L = F_i \leftarrow [(T \oplus F_i) \overset{p}{\underset{i=1}{\oplus}} (T \oplus U_i)] = 0. \qquad (8)$$

Analytical model for HDL-code verification with the use of temporal assertions (additional lines of observation) is aimed at achieving a given diagnostic resolution and is presented as follows:

$M = f(F, A, B, S, T, L)$, $F = (A * B) \times S$; $S = f(T, B)$;

$A = \{A_1, A_2, ..., A_i, ..., A_n\}$; $B = \{B_1, B_2, ..., B_i, ..., B_n\}$;

$S = \{S_1, S_2, ..., S_i, ..., S_m\}$; $S_i = \{S_{i1}, S_{i2}, ..., S_{ij}, ..., S_{ip}\}$;

$T = \{T_1, T_2, ..., T_i, ..., T_k\}$; $L = \{L_1, L_2, ..., L_i, ..., L_n\}$.

Here $F = (A * B) \times S$ – functionality provided by a graph (Figure 2) transaction program blocks (Code-Flow Transaction Graph - CFTG), where $S = \{S_1, S_2, ..., S_i, ..., S_m\}$ - the vertexes or the state of the software simulation test segments. Otherwise the graph can be identified as ABC-graph - Assertion Based Coverage Graph. Each state $S_i = \{S_{i1}, S_{i2}, ..., S_{ij}, ..., S_{ip}\}$ is determined by the values of essential variables of the project (Boolean, register variables, memory). Oriented arc of the graph shows a set of software units or modules:

$$B = (B_1, B_2, ..., B_i, ..., B_n), \ \bigcup_{i=1}^{n} B_i = B; \ \bigcap_{i=1}^{n} B_i = \varnothing$$

Each of them can be put in correspondence to assertion $A_i \in A = \{A_1, A_2, ..., A_i, ..., A_n\}$.

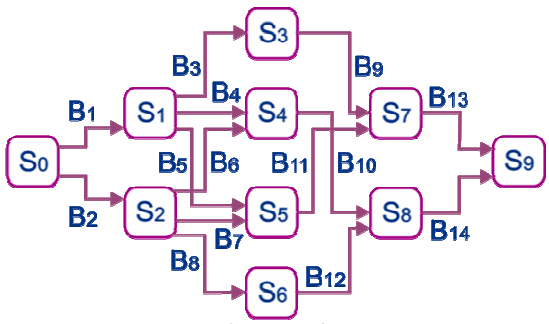

Figure 2. ABC graph for HDL-code

Each arc B_i - a group of operators of a code - forms the state of the vertex $S_i = f(T, B_i)$ depending on the test $T = \{T_1, T_2, ..., T_i, ..., T_k\}$. Each vertex can be put in correspondence with assertion monitor, which combines assertion [2] arc entering the vertex of the arc $A(S_i) = A_{i1} \vee A_{i2} \vee ... \vee A_{ij} \vee ... \vee A_{in}$. Vertex can have more than one incoming (outgoing) arc. A set of blocks with functional violations is represented by the lists $L = \{L_1, L_2, ..., L_i, ..., L_n\}$.

Table 3. Activation matrix

B_{ij}	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
T_1	1	.	1	.	.	.	.	.	1	.	.	.	1	.
T_2	1	.	.	1	.	.	.	.	.	1	.	.	.	1
T_3	1	.	.	.	1	.	.	.	.	.	1	.	1	.
T_4	.	1	.	.	.	1	.	.	.	1	.	.	.	1
T_5	.	1	.	.	.	.	1	.	.	.	1	.	1	.
T_6	.	1	.	.	.	.	.	1	.	.	.	1	.	1

The activation matrix (table 3) is constructed from the ABC-graph and it shows clearly the minimal set of test patterns for the graph (figure 2) and which block each of the patterns activates. We can use the activation matrix to determine which blocks with functional violations are distinguishable or falls into equivalent classes. The set of test patterns and assertions required to ensure that adequate diagnostic resolution should be greater than $\log_2 N$, where N is the number of blocks in the model. Diagnostic resolution can be increased by adding more assertions both at the appropriate edges and vertices.

The graph $F = (A * B) \times S$ has the following features that can be leveraged for the verification and diagnosis of both software and hardware as compared to those presented in [1, 4].

It has a minimum number of edges which equals the number linear blocks(modules) of a software product; 2) Each edge corresponds to assertion which answers for correct functioning of a module or a number of modules; 3) The vertex into which an edge enters represents a minimum set of only those variables which are modified by the program modules that enters into the vertex; 4) As long as the graph is a macrostructure of a software product, it can be simply constructed automatically; 5)The presence on the graph of a model time essentially reduces the size of the matrix meant for verification and diagnosis of functional violation; 6) The presence of the structural model of the software product makes it possible to use algorithmic methods of synthesis of tests of activation of all the vertexes and edges of the graph; 7) The program module is the argument and the vertex is the derivative of it. This kind of relationship defines a linear computational complexity for the synthesis of the graph for the purpose of verification of a software product.

5. Conclusion

The β-metric of binary cyberspace represented by a zero-sum cycle distances of binary codes creates a fundamental basis for all logical and associative problems of synthesis and analysis related to searching, recognition and decision making. The proposed generalized graph model for testing process is based on the XOR-interaction of four main components of technical diagnostics. The model allows the generation of analytical forms of technological processes and efficient structure synthesis and analysis of tests and diagnosis of functional violations in the software and / or hardware products. The HDL-model code provided in the form of ABC-graph displays not only the structure of the code, but also test segments of the functional coverage, which is formed by the program components included in the vertex under consideration during simulation.

The Transaction graph in conjunction with the activation matrix of a code allows you use the instrument of design for testability for assessing the quality of the product; evaluate the cost of creating tests, diagnosis and correction of functional violations; and to optimize the synthesis of test by solving the problem of covering a minimal set of activated paths of all arcs (vertices).

6. References

[1] Rafe V; Rafeh R; Miralvand M.R.Z, Verification and validation of activity diagrams using graph transformation. Computer technology and development, 2009. ICCTD'09 pages 201 – 205.

[2] H.D. Foster, A. C. Krolnik, D. J. Lacey: Assertion-based design, 2nd ed: Kluwer Academic Publishers, 2004.

[3] J. Bergeron, Writing Testbenches: Functional Verification of HDL Models, Second Edition, Kluwer Academic Publishers, 2003.

[4] Zhongjun Du, Zhengjun Dang, A new algorithm based graph-search for workflow verification. Information engineering and computer science (ICIECS), 2010 2nd International Conference, page(s) 1-3.

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