An Extended HDL SoC TAB-model for Diagnosability and Repair

Vladimir Hahanov, Ka Lok Man, Baghdadi Ammar Awni Abbas, Eugenia Litvinova, Svetlana Chumachenko, Eng Gee Lim, Mark Leach

***Abstract*— This article describes technology for diagnosis SoC HDL-models, based on transaction graph. Methods for diagnosis is focused on decreasing the time of fault detection and memory for storage of diagnosis matrix by means of forming ternary relations between test, monitor, and functional component. The following problems are solved: creation of digital system model in the form of transaction graph and multi-tree of fault detection tables, as well as ternary matrices for activating functional components of the selected set of monitors by using test patterns; development of a method for analyzing the activation matrix to detect the faulty blocks with given depth and synthesis logic functions for subsequent embedded hardware fault diagnosis.**

***Index Terms*— HDL SoC model; diagnosis; faulty blocks detection; transaction graph**

I. TAB-Model for Diagnosis Faulty SoC Components

The main objective is the realization of TAB-matrix model (Tests – Assertions – Blocks functional model) and diagnosis methods to reduce the time of testing and memory for storage by means of forming ternary relations (test – monitor – functional component) within one table.

The challenges involve:

1) Development of digital system HDL-model in the form of a transaction graph for diagnosing functional blocks by using assertion set [1-6,15];

2) Development method for analyzing TAB-matrix to detect minimal set of fault blocks [4-7,13];

3) Synthesis of logic functions for embedded fault diagnosis procedure [8-11,14].

The xor-relation between the parameters <test – functionality – faulty blocks B*> is a model for testing digital system HDL-code represented as follows:

$$\begin{aligned} &T \oplus B \oplus B^* = 0; \\ &B^* = T \oplus B = \{T \times A\} \oplus B, \end{aligned} \qquad (1)$$

which transforms the relationship of the components in the TAB-matrix:

$$M = \{\{T \times A\} \times \{B\}\}, M_{ij} = (T \times A)_i \oplus B_j. \qquad (2)$$

The coordinate of the matrix will be 1, if the pair test–monitor $(T \times A)_i$ detects or activates some faults of the functional block $B_j \in B$.

Verification by the use of temporal assertion is focused on the specified diagnosis depth and presented as follows:

$$\begin{aligned} &\Omega = f(G, A, B, S, T), \\ &G = (A^*B) \times S; S = f(T, B); \end{aligned} \qquad (3)$$

Here $G = (A^*B) \times S$ is functionality, represented by Code-Flow Transaction (CFT) Graph (Fig. 1); $S = \{S_1, S_2, ..., S_i, ..., S_m\}$ are software states or nodes when simulating test segments. Otherwise the graph can be considered as an ABC-graph – Assertion Based Coverage Graph. Each state $S_i = \{S_{i1}, S_{i2}, ..., S_{ij}, ..., S_{ip}\}$ is determined by the values of design essential variables (Boolean, register variables, memory). The oriented graph arcs are represented by a set of software blocks:

$$B = \{B_1, B_2, ..., B_i, ..., B_n\}; \bigcup_{i=1}^{n} B_i = B; B_i \bigcap_{i \neq j} B_j = \varnothing. \qquad (4)$$

The assertion $A_i \in A = \{A_1, A_2, ..., A_i, ..., A_n\}$ can be put in each block B_i – a sequence of code statements which determines the state of the graph node $S_i = f(T, B_i)$ depending on the test pattern $T = \{T_1, T_2, ..., T_i, ..., T_k\}$. The assertion monitor, uniting the assertions of incoming arcs $A(S_i) = A_{i1} \vee A_{i2} \vee ... \vee A_{ij} \vee ... \vee A_{iq}$ can be inserted on each node.

The ABC-graph model of HDL-code describes both the software structure, and also test segments of the functional coverage, generated using software blocks, incoming to the given node. In the aggregate, all nodes have to be full state coverage space of software variables, which determines the test quality, equal to 100%: $Q = \text{card} \bigcup_{i=1}^{m} S_i^r / \text{card} \bigcup_{i=1}^{m} S_i^p = 1$. Furthermore, the assertion set $< A, S >$ that exists in the graph, allows monitoring arcs (code-coverage) $B = (B_1, B_2, ..., B_i, ..., B_n)$ and nodes (functional coverage) $S = \{S_1, S_2, ..., S_i, ..., S_m\}$.

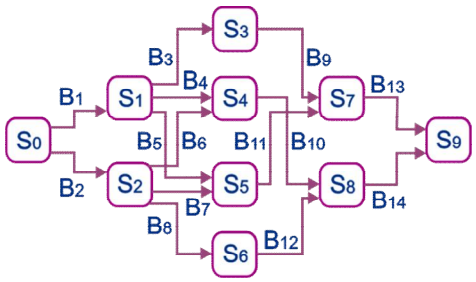
The ABC-graph makes possible the following: 1) to minimize the costs for generating tests, diagnosing and correcting the functional failures by using assertions; 2) to estimate the software quality via diagnosability design; 3) to

Manuscript received January 08, 2015; revised February 10, 2015.

V. Hahanov, E. Litvinova, S. Chumachenko are with the National University of Radioelectronics, Kharkov, Ukraine (e-mail: hahanov@kture.kharkov.ua)

K. L. Man, E. G. Lim and M. Leach are with Xi'an Jiaotong-Liverpool University, Suzhou, China (e-mail: {ka.man, enggee.lim , mark.leach}@xjtlu.edu.cn)

B. A. Awni Abbas is with Baghdad University, Iran



$$B = (B_1B_3B_9 \vee (B_2B_7 \vee B_1B_5)B_{11})B_{13} \vee$$
$$\vee ((B_1B_4 \vee B_2B_6)B_{10} \vee B_2B_8B_{12})B_{14} =$$
$$= B_1B_3B_9B_{13} \vee B_2B_7B_{11}B_{13} \vee B_1B_5B_{11}B_{13} \vee$$
$$\vee B_1B_4B_{10}B_{14} \vee B_2B_6B_{10}B_{14} \vee B_2B_8B_{12}B_{14}.$$

Fig. 1. Example of ABC-graph of HDL-code

optimize test synthesis via coverage all arcs and nodes by a minimum set of activated test paths.

Generally, the testing model is represented by the Cartesian product $M = T \times A \times B$ that accordingly has the dimension $Q = k \times h \times n$. To reduce the amount of diagnosis data, separate monitor or assertion point for visualization functional blocks activation is assigned to each test segment. It is possible to decrease the matrix dimension to $Q = n \times k$ and retain all features of the triad relationship $M = < T \times A \times B >$. Pair «test – monitor» are represented by three possible forms:

$$< T_i \to A_j >, < \{T_i, T_r\} \to A_j >, < \{T_i\} \to \{A_j, A_s\} >. \quad (5)$$

The method for diagnosis of functional block failure uses pre-built TAB-matrix (table) $M = [M_{ij}]$, where the row is the relation between the test segment and a subset of activated blocks observed by the monitor A_j.

$$T_i \to A_j \approx (M_{i1}, M_{i2}, ..., M_{ij}, ..., M_{in}), M_{ij} = \{0,1\} \quad (6)$$

Column of the table describes the relation between the functional blocks, detected on test segments, relatively monitors $M_j = B_j(T_j, A_j)$.

Detecting faulty functional blocks is based on xor-operation between the real assertion response vector and TAB-matrix columns

$$A^* \oplus [M_1(B_1) \vee M_2(B_2) \vee ... \vee M_j(B_j) \vee ... \vee M_n(B_n)] \quad (7)$$

The faulty block is defined by a vector B_j, which gives result with minimal number of 1-unit coordinates:

$$B = \min_{j=1,n} [B_j = \sum_{i=1}^{h} (B_{ij} \oplus A_i^*)]. \quad (8)$$

As an addition to the diagnosis model, necessary to describe the following important features of the TAB-matrix:

1) $M_i = (T_i \times A_j)$;

2) $\bigvee_{i=1}^{m} M_{ij} \to \bigvee_{j=1}^{n} M_j = 1$;

3) $M_{ij} \oplus_{j=1}^{n} M_{rj} \neq M_{ij}$;

4) $M_{ij} \oplus_{i=1}^{k} M_{ir} \neq M_{ij}$;

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 mean: 1) Every row of the matrix is a subset of the Cartesian product between test and monitor. 2) Disjunction of all matrix rows gives a vector equal to 1-unit over the all coordinates. 3) All matrix rows are unique, which eliminates the test redundancy. 4) All matrix columns are distinct, which excludes the existence of equivalent faulty blocks. 5) The number of matrix rows must be greater than the binary logarithm of the number of columns that determines the potential diagnosability of every block. 6) The diagnosis function of every block depends on the complete test and monitors, which must be minimized without diagnosability reduction.

In accordance with 6 test segments activated, the following graph nodes paths relatively assertion point is S9:

$$T = S_0S_1S_3S_7S_9 \vee S_0S_1S_4S_8S_9 \vee S_0S_1S_5S_7S_9 \vee$$
$$\vee S_0S_2S_4S_8S_9 \vee S_0S_2S_5S_7S_9 \vee S_0S_2S_6S_8S_9, \quad (9)$$

It will be easy using graph structure to define all functional blocks (oriented arcs) activated by test:

$$B = B_1B_3B_9B_{13} \vee B_2B_7B_{11}B_{13} \vee B_1B_5B_{11}B_{13} \vee$$
$$\vee B_1B_4B_{10}B_{14} \vee B_2B_6B_{10}B_{14} \vee B_2B_8B_{12}B_{14}. \quad (10)$$

The next step allows creating 6 rows of TAB-matrix $M_{ij}(G_1)$ in the form of relations between test segments and blocks activated respectively:

TABLE I
TAB-MATRIX

$M_{ij}(G_1)$	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}
$T_1 \to S_9$	1	.	1	.	.	.	.	.	1	.	.	.	1	.
$T_2 \to S_9$	1	.	.	1	.	.	.	.	.	1	.	.	.	1
$T_3 \to S_9$	1	.	.	.	1	.	.	.	.	.	1	.	1	.
$T_4 \to S_9$	.	1	.	.	.	1	.	.	.	1	.	.	.	1
$T_5 \to S_9$	.	1	.	.	.	.	1	.	.	.	1	.	1	.
$T_6 \to S_9$	.	1	.	.	.	.	.	1	.	.	.	1	.	1
$T_1 \to S_3$	1	.	1	.	.	.	.	.	.	.	.	.	.	.
$T_6 \to S_6$	.	1	.	.	.	.	.	1	.	.	.	.	.	.

The TAB-matrix of paths activation shows the existence of equivalent failure blocks 3 and 9, 8 and 12, on 6 test segments with one assertion point in the graph node 9. The columns 3 and 9, 8 and 12 are equivalent. To resolve indistinguishability of two pairs faulty blocks it is necessary to create two additional monitors in the nodes S3 and S6 for test segments T1 and T6 respectively. As a result, three assertions in the nodes $A = (S_9, S_3, S_6)$ allow distinguishing all faulty blocks of software HDL-code. Thus, the graph

ISBN: 978-988-19253-9-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

enables not only to synthesize the optimal test, but also to determine the minimal number of assertion monitors in the nodes to detect faulty blocks with a given diagnosis depth.

II. Design for Diagnosability

Diagnosability is the relationship $D = N_d / N$ between the recognized faulty blocks amount N_d, (when there are not equivalent components, or the diagnosis depth is equal to 1), and the total number N of HDL-blocks.

For the expense E evaluation of the TAB-matrix model for detecting functional failures, it can use the pair test-assertions efficiency for a given diagnosis depth. Criterion E functionally depends on the relation between the ideal $]\log_2 N[\times N$ and real $|T|\times|A|\times N$ memory sizes or resources (where $|T|$ – the test length, $|A|$ – a number of assertions) for the corresponding TAB-matrices, which compose the relative expense reduced to 0-1 intervals:

$$E = \frac{]\log_2 N[\times N}{|T|\times|A|\times N} = \frac{]\log_2 N[}{|T|\times|A|}. \qquad (11)$$

The ABC-graph analysis of assertion monitor placement, makes it possible to obtain maximal diagnosis depth of fault blocks. Diagnosability of the ABC-graph is a function depending on the number N_n of transit not ended nodes. Where exist only two adjacent arcs, one of which is incoming, other one is outgoing. N is the total number of arcs in the graph:

$$D = \frac{N\text{-}N_n}{N} \qquad (12)$$

The estimation N_n is the number of unrecognizable or equivalent functional blocks. Potential installation of additional monitors for improving diagnosability of failure blocks is pure transit nodes composed N_n. The diagnosis quality criterion of the ABC-graph takes the form:

$$Q = E \times D = \frac{]\log_2 N[}{|T|\times|A|} \times \frac{N - N_n}{N} \qquad (13)$$

The last expression produces some practical rules for synthesis of diagnosable HDL-code: 1) Test must create a minimal number of single activation paths, and cover all the nodes and arcs in the ABC-graph. 2) Base number of monitors equals the end node number of the graph with no outgoing arcs. 3) Additional monitors can be placed on each non end node. 4) Parallel independent code blocks must have n monitors and a single concurrent test, or one integrated monitor and n serial 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 more than 1 number of input and output arcs, create good conditions for the diagnosability of the current section by single path activation tests without installation additional monitors. 7) The test pattern or testbench has to be 100% functional coverage for the nodes of the ABC-graph. 8) Diagnosis quality criterion as a function depending on the graph structure, test and assertion monitors can always be increased close to the 1-value. For this purpose there are two alternative ways. The first one is increasing test segments by activating new paths for recognition equivalent faulty blocks without increasing assertions, if the software graph structure allows the potential links. The second way is adding assertion monitors on transit nodes of the graph. A third so called hybrid variant is possible, based on the joint application of two above-mentioned ways.

III. Multilevel Diagnosis Method of Digital System

Multilevel model of the multi-tree B (Fig. 2.) is shown, where each node is represented by digital or computer system component, which has a three-dimensional activation TAB-matrix of functional unit subcomponents.

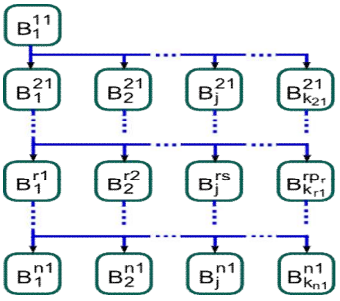


Fig. 2. Diagnosis multitree model

The outcoming from the node arcs are transitioning to a lower detailed level in diagnosing process, when replacing faulty block is too expensive:

$$B = [B_{ij}^{rs}],\ \mathrm{card}B = \sum_{r=1}^{n}\sum_{s=1}^{m_r}\sum_{i=1}^{p_{rs}}\sum_{j=1}^{k_{rs}} B_{ij}^{rs}, \qquad (14)$$

where n is a number of diagnosis multi-tree levels; m_r is a number of functional units or components at the level r; k_{rs} (p_{rs}) is a number of components (test length) in the table B^{rs}; $B_{ij}^{rs} = \{0,1\}$ is a component of an activation table, which is defined by 1-unit the detected faulty functionality under the test segment $T_{i\text{-}A_i}$ relatively to the observed monitor-assertion.

Method for faulty blocks diagnosis Hardware-Software HS-system, based on multi-tree model, allows creating the universal engine in form of algorithm (Fig. 3, block 6) for traversal of tree branches on the depth, specified a priory:

$$B_j^{rs} \oplus A^{rs} = \begin{cases} 0 \to \{B_j^{r+1,s}, R\}; \\ 1 \to \{B_{j+1,}^{rs}, T\}. \end{cases} \qquad (15)$$

Here $A_i^{rs} = m_i^{rs} \oplus g_i^{rs}$, $i = \overline{1, k_{rs}}$. If all coordinates of vector xor-sum $B_j^{rs} \oplus A^{rs} = 0$ then one of the following action is performed: the transition to the activation matrix of the lower level $B_j^{r+1,s}$ or repair of the functional block $B = B_j^{rs}$.

ISBN: 978-988-19253-9-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

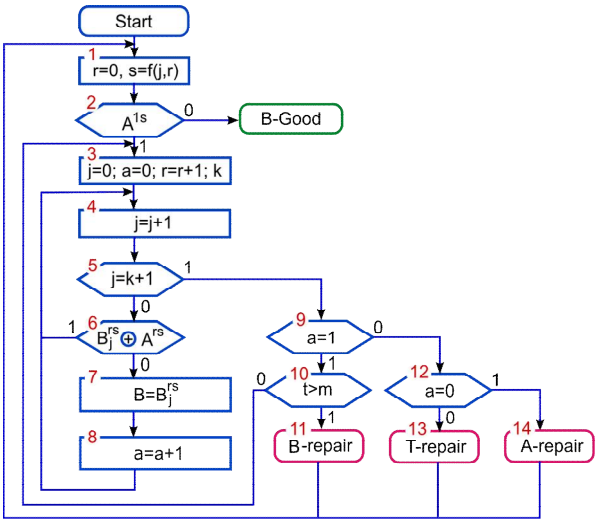


Fig. 3. Engine for traversal of diagnosis multitree

One of two analyses is executed, based on: 1) the time (t>m, block 10) – then repair of faulty block is performed; 2) the money (t<m) – then a transition down to specify a more exact fault location, because replacement of smaller block decreases the repair cost. If at least one coordinate of the resulting xor-sum vector $B_j^{rs} \oplus A^{rs} = 1$, then transition to the next matrix column is performed. When all coordinates of the assertion vector $A^{1s} = 0$, fault-free state of a HS-system is defined. So, the TAB-engine has four end-nodes, where one of them is B-good which indicates successful finishing of the testing. The other three means the intermediate results in the test process, which is necessary to take into account for the increasing a test quality and diagnosis depth by using extra assertions and/or additional test segments generation.

IV. Conclusion

Infrastructure and technology for digital systems analysis are presented. The proposed transactional graph model and method for diagnosis of digital systems-on-chips are focused to considerably reducing the time of faulty blocks detection and memory for storing the diagnosis compact matrix.

Finally, a new diagnosis quality criterion as a function depending on the graph structure, test, and assertion monitors is proposed. It allows making good choices in diagnosability by increasing test segments set for recognition equivalent faulty blocks or adding assertion monitors on transit nodes of the activation HDL-code graph.

References

[1] P.P. Parhomenko, “Technical diagnosis basics” Moscow: Energy, 1976.

[2] P.P.Parhomenko, and E.S. Sogomonyan, “Technical diagnosis basics (Optimization of diagnosis algorithms, hardware tools)”, Moscow: Energy, 1981.

[3] M.F. Bondarenko, O.A. Guz, V.I. Hahanov, and Yu.P. Shabanov-Kushnarenko, “Infrastructure for brain-like computing”, Kharkov: Novoye Slovo, 2010.

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

[5] V.V. Semenets, I.V. Hahanova, and V.I. Hahanov, “Design of digital systems by using VHDL language”, Kharkov: KHNURE, 2003.

[6] V.I. Hahanov, and I.V. Hahanova, “VHDL+Verilog = synthesis for minutes”, Kharkov: KHNURE, 2006.

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

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

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

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

[11] V.I. Hahanov, E.I. Litvinova, S.V. Chumachenko, and O.A. Guz, “Logic associative computer”, Electronic simulation, No 1, pp.73-83, 2011.

[12] Ngene Christopher Umerah, Hahanov V. A diagnostic model for detecting functional violation in HDL-code of SoC // Proc. of IEEE East-West Design and Test Symposium.– Sevastopol, Ukraine.– 19-20 September, pp. 299-302, 2011.

[13] Ubar R., Kostin S., Raik J. Block-Level Fault Model-Free Debug and Diagnosis in Digital Systems. DSD '09. 12th Euromicro Conference 2009, pp. 229 – 232.

[14] Benabboud Y., Bosio A., Girard P., Pravossoudovitch S., Virazel A., Bouzaida L., Izaute I. "A case study on logic diagnosis for System-on-Chip,"Quality of Electronic Design, 2009, pp.253,259.

[15] Datta K., Das P.P. Assertion based verification using HDVL. Proceedings 17th International Conference VLSI Design. 2004, pp. 319 – 325

ISBN: 978-988-19253-9-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)