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 of HDL-SoC models, based on transaction graph. Methods for diagnosis are 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 muliti-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. 

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 model for testing digital system HDL-code represented as follows:

\[ T \oplus B \oplus B^* = 0; \]
\[ B^* = T \oplus B = \{T \times A\} \oplus B, \]

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

\[ M = \{\{T \times A\} \times \{B\}\}, M_{ij} = (T \times A)\_i \oplus B\_j \] (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:

\[ \Omega = f(G, A, B, S, T), \quad G = (A \times B) \times S = f(T, B); \quad (3) \]

Here \(G = (A \times B) \times S\) is functionality, represented by Code-Flow Transaction (CFT) Graph (Fig. 1); \(S = \{S_1, S_2, \ldots, S_i, \ldots, S_m\}\) 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 \(S_i = \{S_{i1}, S_{i2}, \ldots, S_{ij}, \ldots, S_{ip}\}\) 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:

\[ B = \{B_{i1}, B_{i2}, \ldots, B_{in}\}; \quad \bigcup_{i=1}^{n} B_i = B; \quad B_i \cap B_j = \emptyset. \] (4)

The assertion \(A_1 \in A = \{A_1, A_2, \ldots, A_i, \ldots, A_n\}\) 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 \(T = \{T_1, T_2, \ldots, T_j, \ldots, T_k\}\). The assertion monitor, uniting the assertions of incoming arcs \(A(S_i) = A_{i1} \lor A_{i2} \lor \ldots \lor A_{ij} \lor \ldots \lor A_{i1}\) 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%: \(Q = card\bigcup_{i=1}^{m} S_{i1} / card\bigcup_{i=1}^{m} S_{ip} = 1\).

Furthermore, the assertion set \(<A, S>\) that exists in the graph, allows monitoring arcs (code-coverage) \(B = \{B_{i1}, B_{i2}, \ldots, B_{in}\}\) and nodes (functional coverage) \(S = \{S_{i1}, S_{i2}, \ldots, S_{in}\}\).

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
\[ B = (B_1 B_3 B_0 \lor (B_2 B_7 \lor B_5 B_3) B_1) B_{13} \lor \lor ((B_1 B_4 \lor B_2 B_6) B_{10} \lor B_2 B_8 B_{12}) B_{14} = B_3 B_5 B_9 B_{13} \lor B_2 B_7 B_{11} B_{13} \lor B_1 B_3 B_{11} B_{13} \lor \lor B_1 B_5 B_{10} B_{14} \lor B_2 B_6 B_{10} B_{14} \lor B_2 B_6 B_{12} B_{14}. \]

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 \( k \) functional blocks, detected on test segments, relatively separate monitor or assertion point for visualization.

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

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

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) \( \forall M_{ij} \rightarrow \forall M_j = 1 \);
3) \( M_{ij} \oplus M_{ir} \neq M_{ij} \);
4) \( M_{ij} \oplus M_{ir} \neq M_{ij} \);
5) \( \log_2 n \leq k \leftrightarrow \log_2 |B| \leq |T| \)
6) \( B_j = f(T, A) \rightarrow B \oplus T \oplus A = 0 \).

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 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:

\[ T = S_0 S_1 S_5 S_7 S_9 \lor S_0 S_1 S_4 S_8 S_9 \lor S_0 S_1 S_5 S_7 S_9 \lor S_0 S_1 S_6 S_9 S_9, \]

(9)

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

\[ B = B_1 B_3 B_5 B_{13} \lor B_2 B_7 B_{11} B_{13} \lor B_1 B_3 B_{11} B_{13} \lor B_2 B_6 B_{10} B_{14} \lor B_2 B_6 B_{10} B_{14} \lor B_2 B_6 B_{12} B_{14}. \]

(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:

<p>| TABLE I |</p>
<table>
<thead>
<tr>
<th>TAB-MATRIX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

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 = (S0, S3, S6) allow distinguishing all faulty blocks of software HDL-code. Thus, the graph...
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 \( D = \frac{N_d}{N} \) 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 \( \frac{|T| \times |A| \times N}{T \times A} \) 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:

\[
E = \frac{\log_2 N \times N}{|T| \times |A| \times N} \times \frac{N - N_n}{N} \quad (11)
\]

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:

\[
D = \frac{N - N_n}{N} \quad (12)
\]

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:

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

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

\[
B = [B^r_{ij}], \quad \text{card} B = \sum_{r=1}^{n} \sum_{s=1}^{m_r} \sum_{i=1}^{p_{rs}} \sum_{j=1}^{k_{rs}} B^r_{ij}, \quad (14)
\]

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 \( B^r_{ij} \); \( B^r_{ij} = \{0,1\} \) is a component of an activation table, which is defined by 1-unit the detected faulty functionality under the test segment \( T_{i-A_i} \) 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:

\[
B^r_j \oplus A^r_{i} = \begin{cases} 
0 \rightarrow [B^r_{j+1,s}, R]; \\
1 \rightarrow [B^r_{j+1,s}, T]. 
\end{cases} \quad (15)
\]

Here \( A^r_{i} = m^r_{i} \oplus b^r_{i}, i = 1, k_{rs} \). If all coordinates of vector xor-sum \( B^r_j \oplus A^r_{is} = 0 \) then one of the following action is performed: the transition to the activation matrix of the lower level \( B^r_{j+1,s} \) or repair of the functional block \( B = B^r_j \).
One of two analyses is executed, based on: 1) the time \((t>m, \text{block 10})\) – 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 \(B_j^{rs} \circ A^{rs} = 1\), then transition to the next matrix column is performed. When all coordinates of the assertion vector \(A^{ks} = 0\), 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


Fig. 3. Engine for traversal of diagnosis multtree