Comparing different approaches to the test of Reconfigurable Scan Networks

Comparing different approaches to the test of IEEE 1687 reconfigurable scan networks

R. Cantoro, A. Damljanovic, M. Sonza Reorda, G. Squillero
Politecnico di Torino, Torino, Italy

Abstract— **Due to the growing complexity of semiconductor devices integrating test and measurement instrumentation, IEEE 1687 standard has been introduced in order to enable flexible and effective access to embedded infrastructure through usage of reconfigurable scan network (RSN). Although the proper selection of valid network configurations can be used to ensure that any permanent fault is detected, the important and more challenging issue that has to be addressed is the duration of the test. Here we consider three different approaches used to generate the test sequence with minimum cost. They make use of a high-level fault-model, which allows independence on the detailed implementation. A comparison of the different approaches is given, based on a comprehensive set of experimental results.**

I. TEST PROCEDURE

The introduction of reconfigurable scan networks (RSN) [1] provided increased flexibility, but also made the generation of a test procedure more challenging. Apart from testing the ability of the flip-flops, which are forming Test Data Registers (TDRs), to correctly shift values, testing the reconfigurable elements themselves is also needed. Thus, it must be guaranteed that the network is properly reconfigured after applying valid configurations. In all of the proposed approaches, the same procedure is used to test RSN elements (i.e., TDRs, SIBs and ScanMuxes [2]) for permanent faults. The complete generated test procedure is organized as a set of *sessions*, composed out of alternating test and configuration vectors. Applying the test vector consists out of the following phases:

- shifting-in the first sequence consisting of 0s, while the length of the sequence is equal to the length of the longest path in the network;
- shifting-in the second sequence of alternating 0s and 1s (i.e. 0101…), while the length of the sequence is equal to the length of currently active path; as a sequence terminator two successive values (either 00 or 11) are added;
- exceptionally, the last test sequence is ended with shifting-out the values from the currently active path; the sequence has a length of the active path.

While shifting the second sequence, values at the output are being monitored and compared. While shifting the alternating sequence, one should expect all 0s appearing at the output, while after shifting the termination symbol, 1 and 0 should be observed. To change the configuration of the network, values required for the desired configuration are shifted-in, for the number of cycles corresponding to the length of the currently active path. Finally, applying the configuration vector demands an update operation. The duration of the test depends on the duration of each session. Furthermore, the duration of a session is determined by the length of TDRs included in the path, as well as by the previously imposed configuration. Although RSN testing and design validation [3] have certain similarities, in case of testing, stimuli duration is crucial.

II. FAULT MODEL

A high-level fault model can be used to represent possible hardware defects. The configuration of the network can differ from the expected one due to a fault affecting SIB and ScanMux modules. Consequently, a fault affecting them will most likely lead to an active path with a different length than in the fault-free RSN. This property is used to perform the test and distinguish a good circuit from a faulty one. Permanent faults affecting the SIB modules are labelled as *stuck-at asserted* and *stuck-at de-asserted.* Additionally, the effect of a fault affecting ScanMux element is such that one of the input segments is always selected, regardless of the values in the selection cells. Faults that affect scan bits of the selection cells and the update logic of the reconfigurable modules are detected by implication by testing the described high-level faults. Moreover, proposed test methodology is not being used to examine internal instruments connected to the TDRs. In addition, it is presumed that the TAP controller is fault-free.

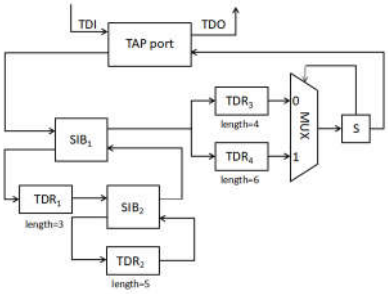

Figure 1 Example of IEEE 1687 RSN

Table 1 Set of configurations for the network in Fig.1

Conf.	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
SIB_1	D		D		A		A	
SIB_2	D	A	D	A	D		A	
Scan Mux	0		1		0	1	0	1
Length	6		8		10	12	15	17

Considering the example of a reconfigurable scan network shown in Fig.1, with all its valid configurations in Table 1,

stuck-at assert fault affecting SIB_2 module is manifested by always selecting the path containing TDR_2, regardless the configuration we try to apply. In order to detect the fault, particular module has to be excited. If SIB_1 is de-asserted (e.g. network is in the configuration C_2), no change in length can be detected, by applying proposed test procedure. Thus, fault is currently transparent. Moving the network to the configuration where SIB_1 is asserted and SIB_2 de-asserted (e.g., configuration C_5), we detect the fault by observing the mismatch between the length of the faulty path (17) and the expected length (12).

III. Proposed Approaches

Three approaches have been proposed.

A. Depth-first approach

This is a sub-optimal approach [4]. The main idea behind this approach is trying to generate the sequences in such a way that each TDR is accessed at least once; each SIB is at least once asserted and once de-asserted; for each ScanMux all possible configurations are taken into account. It is a heuristic method based on the direct graph network representation, which is built from three types of vertices (TDR, SIB and ScanMux). The optimized sequence is in this case obtained by using the depth-first approach for visiting the vertices of the graph. If it exists, any adjacent unvisited vertex is visited. On the other hand, if it does not, the algorithm continues from the last available unvisited adjacent vertex.

B. Evolutionary metaheuristic approach

The second approach is referred to as the evolutionary one. It is based on using and evolutionary engine [5] to identify the list of configurations required to achieve full coverage with minimum cost additional functions have also been implemented, such as producing the list of faults covered by a given set of configurations and returning the array of intermediate configurations able to move the network from one configuration to another one. The evolutionary engine generates the population of individuals, representing sequences of valid configurations. The fitness values provided by the evaluation engine are used to evaluate the individuals (in terms of test-cost and fault coverage). After applying different mutation and crossover operators to create a new generation, evaluation is conducted in order to choose and always keep the original number of individuals.

C. FSA approach

Finally, the third approach is based on resorting to a modelling the RSN as a finite state automation (FSA). The length of the active path is used as an output symbol, while input symbols correspond to possible configurations. The permanent faults are represented as incorrect transitions. The FSA is being built dynamically and the approach used to choose the input symbol is based on a greedy algorithm (with respect to the number of faults detected and the cost of each session). Encoding of the FSA's states is not complete on purpose, since a carefully chosen subset of all valid configurations is considered. Accordingly, the size of the automaton has an effect both on the quality of the results and on the performance of the algorithm.

IV. Comparing the proposed approaches

The effectiveness of the proposed approaches has been evaluated on a sub-set of the ITC'16 benchmark reconfigurable scan networks [6]. Full coverage is always achieved, since each of the proposed approaches is able to produce a test that detects all the modelled detectable faults. Comparison of the approaches relies on the effectiveness, which is assessed taking into account the following experimental data:

- The CPU time required by the tool to apply the algorithm (referred to as *Runtime*)
- The duration of the generated test. It consists of the number of clock cycles needed to apply configuration and test patterns.

The depth-first approach is able to provide the test almost instantaneously, even for the most complex networks (large number of configurable modules with high hierarchical depth). However, this approach is not always optimal regarding the duration of the generated test.

The evolutionary approach shows an improvement regarding the total test time varying on the method used to generate the initial population of individuals. However, this algorithm has shown to be computationally demanding, since the time required to perform evolution is measured in hours for all benchmarks used for the experimental purposes, and in case of large networks may even take more than two days to complete.

The FSA approach is flexible in terms of the states and input symbols that are being considered. Furthermore, it has been able to manage large networks with a constrained computational effort and significantly reduce duration of the test for more than half of the used benchmarks. The simplicity of the network in terms of the hierarchical depth and number of configurable elements may have an impact on the results. In that case, the depth-first method is very likely to find the global-optimum. Additionally, evolutionary and FSA approach may generate the same test as the depth-first method, with the same cost.

During the presentation, we will analyse in detail the RSN characteristics that may more significantly impact on the effectiveness of the considered approaches and identify their limitations in terms of scalability.

ACKNOWLEDGMENT

The work has been partially supported by the European Commission through the Horizon 2020 RESCUE-ITN project under the agreement No. 722325.

References

[1] S. Narayanan and M. A. Breuer, "Reconfigurable scan chains: A novel approach to reduce test application time," in *Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design*. IEEE Computer Society Press, 1993, pp. 710–715.

[2] "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Std 1687-2014*, pp. 1–283, Dec 2014.

[3] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Reconfigurable scan networks: Modeling, verification, and optimal pattern generation," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 2, p. 30, 2015.

[4] R. Cantoro, M. Montazeri, M. Sonza Reorda, F. G. Zadegan, and E. Larsson, "On the testability of IEEE 1687 networks," in *Test Symposium (ATS), 2015 IEEE 24th Asian*. IEEE, 2015, pp. 211–216

[5] MicroGP web-site. Online: http://ugp3.sourceforge.net/

[6] Tšertov, Anton et al.; Krenz-Baath, Rene, "A Suite of IEEE 1687 Benchmark Networks", IEEE International Test Conference (ITC), 2016.