On Testability Analysis Driven Generation of Synthetic Register-Transfer Level Benchmark Circuits

Josef Strnadel[1], Tomáš Pečenka[1], Lukáš Sekanina[1]

[1]*Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic*
strnadel@fit.vutbr.cz, pecenka@fit.vutbr.cz, sekanina@fit.vutbr.cz

Abstract. Use of benchmark designs has become an important part of a process of designing complex systems. However, existing register-transfer level benchmark suites are not sufficient for evaluation of new architectures and tools; synthetic benchmark circuits are an alternative. In the paper, it is demonstrated how evolutionary techniques can be used to generate synthetic benchmarks covering a wide scale of testability properties. The generation process is driven by a register-transfer level testability analysis method and generated benchmarks are stored in synthesizable VHDL source-code. Results gained by proposed method together with future research trends are discussed at the end of the paper.

Keywords. Register-transfer level, synthetic benchmark circuit, testability analysis, evolutionary algorithm

I. Introduction

One of the most difficult tasks CAD users face is the evaluation and comparison of different tools and algorithms. The efficiency of critical algorithms must be measured and compared to understand both tool behavior and progress over time. The evaluation and comparison of new technologies, architectures and *electronic design automation* (*EDA*) tools can be done using so-called *benchmark circuits* (*benchmarks*). A *benchmark set* (*suite*) is a set of benchmarks that (in the ideal case) is representative for the circuit space, or at least that part at which particular EDA tool is aimed. The type of description of a benchmark and its level of abstraction depend on the application. E.g., the evaluation of high-level synthesis algorithms requires high-level behavioral circuit descriptions, while routing algorithms can only be tested with low-level physical descriptions. Many initiatives dealing with benchmarks exist. Benchmark suites for following areas are available in [2]: gate-level test generation (ISCAS8x), high-level synthesis (HLSynth89, HLSynth9x), logic synthesis (LGSynth89, LGSynth9x), physical implementation (LayoutSynth9x, PDWorkshop9x etc.), circuit simulation (CircuitSim90), partitioning (Partitioning93) etc. However–existing benchmark suites are insufficient, since they usually consist of too few and too small circuits and they usually are not very representative for all circuit classes, e.g., for diagnostics purposes. Also, because of the proprietary nature of industrial circuits, it is almost impossible to compile sufficiently large benchmark sets of sufficiently large real-circuits. Recently, the generation of synthetic benchmarks is seen as a viable alternative–see, e.g., [4, 5, 7]. Major advantage of synthetic benchmarks is they provide full control over important characteristics, such as size, topological, diagnostic or functional parameters of particular circuit. For each circuit class, different parameters are important in general.

II. Problem Definition and Our Research Goals

As mentioned above, **the "benchmark set construction" problem is to compile sufficiently large sets of sufficiently large benchmarks with desired properties**. Our research has been focused on solving the problem in the area of *register-transfer level* (*RTL*) circuits.

Our main research goal in the area was to use our previously developed structural-analysis based RTL testability analysis method for finding the worst testable RTL design from given RTL design state-space. **Our hypothesis is the worst testable design is the design that is highly suitable to be included in a "RTL diagnosis benchmark set".** Thus, our goal was to develop an efficient method that would be able to explore the state-space and to find proper candidates to be included in the set.

III. Proposed Benchmark Generation Method

During our research, novel method utilizing an evolutionary *genetic algorithm* (*GA*) to create RTL benchmarks automatically according to user-specified requirements was developed. To be able to compare quality of two different solutions from a diagnostics point of view, a fitness function was developed. It evaluates particular solution by a real number according to its RTL testability results.

A. Inputs of the Method

As an input of proposed method, the user is supposed to specify following data: number of circuit primary inputs and outputs, number and type of in-circuit components, testability requirements (controllable and observable nodes ratios: sets with various diagnostics parameters could be needed; **in our case, both ratios should be set to their worst values**, e.g., to 0.0) and GA parameters. Following *XML* code is an example of a user-entered data stored in an input file:

```
<circuit>
        <testability con_ratio="0.0" obs_ratio="0.0"/>
        <evolution population="30" replacement="1.0" crossover="0.0" mutation="1.0" steps="10"/>
        <primary inputs="24" outputs="16"/>
        <comp type="SUB_A" width="8" quantity="20"/>
        <comp type ="ADD_B" width="16" quantity="11"/>
        <comp type ="MUL_A" width="8,16" quantity="8"/>
        <comp type ="MUX_2" width="8" quantity="13"/>
</circuit>
```

B. Circuit Representation

Each circuit is seen as a graph represented by an integer array (see Fig 1). GA operates over such arrays. In the representation, registers are not taken into account; they are post-inserted into the circuit structure before the testability analysis and synthesis are started. Each input and output belonging to the circuit structure is assigned a unique number. Because a component input can be connected to at most one output, the circuit can be represented by means of an array, in which the index is the input number and the value identifies the output connected to the input. Primary inputs are treated as outputs and primary outputs are treated as inputs of a component connected to a test-bench circuit.

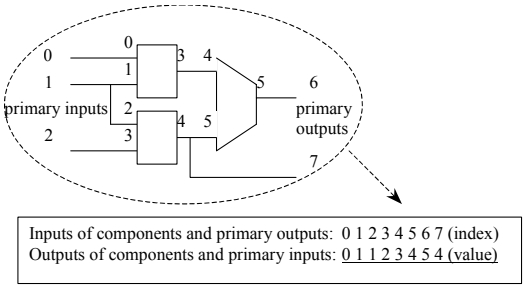


Fig 1 Circuit encoding in a chromosome

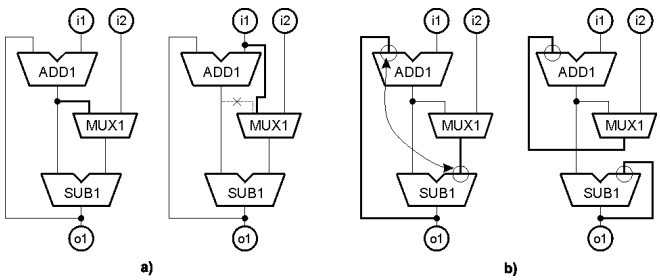


Fig 2 Illustration of used approaches to mutation

C. Principle of the Method

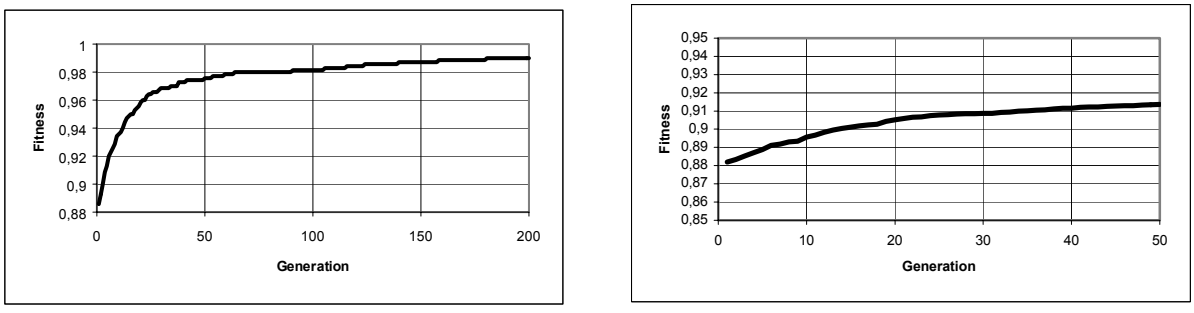
Evolutionary algorithms have become a successful design method. User has to specify requirements posed on desired-solution properties (circuit structure properties in our case) in the fitness function. Then, the evolutionary algorithm tries to meet the requirements by means of components that are available beforehand and using a population-based search (e.g., [1, 3]). However, no approach is known for evolutionary design of benchmark circuits. In our approach, simple GA operating with the representation introduced in the previous section was utilized. Initial population consisting of P individuals (candidate circuits) is generated randomly. New populations are formed using roulette wheel selection and mutation operator, n weakest individuals are replaced by mutated parents. Elitism is used and evolution runs for a given number of generations. The fittest individual is considered as the result and it is transformed to VHDL. During the evolution, only inter-connections are mutated. Used mutation principles can be summarized as follows (for illustration, see Fig 2): the input of a component on which the mutation operator will be applied is randomly selected. If the output connected to the input is connected to other input(s), then the selected input is reconnected to a randomly selected output of other component or to a primary input (see Fig 2a). In case an output of a component would become disconnected after the mutation, the output must be connected to a randomly selected input as illustrated in Fig 2b. The mutation respects the circuit data-path width. To be able to compare individuals belonging to P, fitness function is used that assigns a numeric value to each individual within P. Alike the mutation the fitness function is a crucial part of a GA since it substantially affects quality of evolved solutions (RTL benchmarks in our case). The fitness function (see formula 1), which has to be maximized here, combines three objectives x_1, x_2 and x_3. x_1 (x_2) characterize interconnectivity (variability) of in-circuit components, x_3 is calculated using tool based on method [6] and reflects the result of comparing circuit testability with user testability requirements posed on resulting circuit. Experimentally found weight system c_1=0,3, c_2=0,2 and c_3=0,5 is used.

$$fitness = x_1c_1 + x_2c_2 + x_3c_3 \tag{1}$$

IV. Experimental Results

We have performed hundreds runs of proposed GA in order to find suitable parameters of the algorithm. We arranged a set of experiments to evaluate the proposed approach. From all experiments, let us mention results only of two of them in the next.

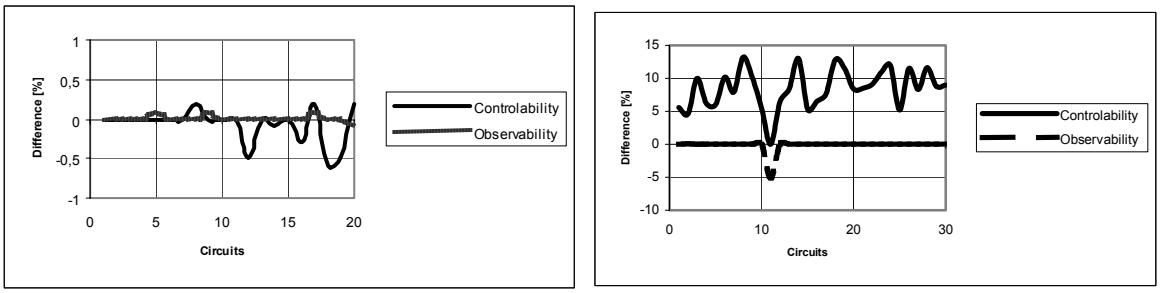
Experimenting with evolvability of generated circuits: the objective of this task was to observe how the average fitness value (gained from 20 (30) runs) increases during the evolution, i.e. how the best solution is evolved during time. This experiment was performed for a small circuit (12 in-circuit components, 20 runs) and a large circuit (250 in-circuit components, 30 runs).



(a) (b)

Fig 3 Evolution of a 12-component circuit (a) and a 250-component (b) circuit

Experimenting with meeting user requirements: the objective of this task was to check how observability and controllability of evolved benchmarks differ from values required by the user. The same experimental setup as in the previous section was used.



(a) (b)

Fig 4 The differences of required and obtained controllability and observability values for 20 evolved 12-component circuits (a) and 30 evolved 250-component circuits (b)

V. Conclusions

In the area of RTL benchmark circuits, there is a long-term need for larger sets of more complex benchmark circuits having desired diagnostics, structural and other properties. In our research we have verified that a set of RTL benchmark circuits can be evolved in an efficient way on basis of GA using testability analysis results for fitness calculation. Proposed benchmark generation method takes into account user-selected number of inputs and outputs, amount of in-circuit components of chosen types and testability requirements posed on a final circuit design; the function performed by the circuit is not considered yet. For the nearest future, we intend to develop and implement a method, which will evolve benchmark circuits fulfilling required function and still having desired testability properties. It is expected that the process of generating benchmark circuits in this direction will be significantly more complicated. We shall not also neglect the possibility of integrating proposed algorithms into existing design systems and thus offer the possibility of developing components fulfilling the required function and providing guaranteed and predefined testability properties. It is believed that utilizing evolutionary approaches will offer completely new solution to this problem. The work related to the paper was financially supported by the Grant Agency of the Czech Republic (GACR) under contract number GA102/05/P193 "Optimizing Methods in Digital Systems Diagnosis".

References

[1] Bentley, P.: *Evolutionary Design by Computers*. Morgan Kaufmann Publishers, San Francisco, 1999, 446 p., ISBN 1-55860-605-X.

[2] Computer-Aided Design Benchmarking Laboratory [2005]. Accessible from http://www.cbl.ncsu.edu/benchmarks/

[3] Garvie, M., Thompson, A.: Evolution of Self-diagnosing Hardware. In: Proceedings of the 5th Int. Conf. on Evolvable Systems: From Biology to Hardware, Trondheim, 2003, pp. 238—248.

[4] Hutton, M. D., Rose, J. S., Corneil D.: Automatic Generation of Syntetic Sequential Benchmark Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 21, No. 8, 2002, pp. 928—940.

[5] Kundarewich, P. D., Rose, J.: Synthetic circuit generation using clustering and iteration. In: *Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays*, California, 2003, pp. 245—245.

[6] Strnadel, J.: *Testability Analysis and Improvements of Register-Transfer Level Digital Circuits*. PhD thesis. Brno University of Technology, 150 p., 2004.

[7] Verplaetse, P., Stroobandt, D., Van Campenhout, J.: *Synthetic Benchmark Circuits for Timing-driven Physical Design Applications*. In: Proceedings of the International Conference on VLSI, Las Vegas, 2002, pp. 31—37.