Evolutionary Design of Synthetic RTL Benchmark Circuits

Zdeněk Kotásek, Tomáš Pečenka, Josef Strnadel, Lukáš Sekanina
Faculty of Information Technology
Brno University of Technology, Božetěchova 2, 612 66 Brno, Czech Republic
e-mail: kotasek, pecenka, strnadel, sekanina@fit.vutbr.cz

Abstract

In the paper it is demonstrated how evolutionary techniques can be used for the process of generating benchmark circuits covering a wide scale of testability properties. To calculate the value of fitness function the approach based on analytical evaluation of testability parameters is used. The solutions which cannot be synthesized by a design system are avoided from the process of developing a new generation of benchmark circuits. The output of the methodology developed and implemented is in the form of component VHDL code. The results are discussed and trends for the future research in this field are indicated.

1 Introduction

Several benchmarks suites with different levels of abstraction and circuit complexity exist [1]. However, the existing benchmark suites are not sufficient, since they usually consist of too few and too small circuits and they are not usually very representative for all circuit classes. Also, because the proprietary nature of industrial circuits, it is almost impossible to compile sufficiently large sets of sufficiently large realistic circuits.

The generation of synthetic benchmark circuits is becoming to be recognized as a viable alternative (e.g. [2], [3]). A major advantage of synthetic benchmarks is that they provide full control over important characteristic parameters, such as size, topological or functional parameters.

2 Design Method

We present a novel approach which utilizes an evolutionary algorithm to design a structure of a benchmark circuits automatically according to the requirements specified by the user. For indirect validation of the quality of generated benchmarks, register-transfer level testability analysis based on the controllability and observability measurements and on transparency properties of internal components is used to compute the value of the fitness function. The principles of testability analysis have been formulated independently of the proposed evolutionary design [4]. Requirements on the circuit function are not reflected by the procedure of developing the benchmark structure.

2.1 Problem definition

Our objective is to produce high-quality RT-level (*Register Transfer level*) benchmark circuits automatically. The user is supposed to specify the number of primary circuit inputs and outputs, the number and type of components, the requirements on testability (average controllability and observability) and parameters of the evolutionary algorithm.

The program generates a benchmark circuit according to the predefined requirements. The resulting circuit is constructed from RT-level components described by means of the structural description in VHDL. All the generated circuits are synthesizable. At the moment the user cannot specify the positions of registers. The program inserts the registers automatically in order to meet the requirements on testability and to minimize their count.

2.2 Circuit Structure Representation

A circuit consists of components and interconnections. The user has to define a set of components that will be used by the evolutionary algorithm. Interconnections are represented by a fixed-size integer array. Evolutionary algorithm operates with these arrays. Registers are not reflected in the representation; they are "inserted" into a circuit before the testability analysis and synthesis are performed.

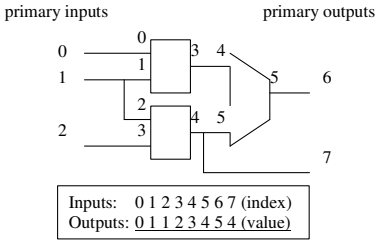

Figure 1. Circuit representation

Figure 1 gives a simple example of circuit representation. Inputs and outputs of components are uniquely numbered. Primary inputs are treated as outputs of a component and primary outputs are treated as inputs of a component connected to the test bench circuit. Because any component input can be connected to only a single component output, we can represent the circuit as the array, in which the index is the component input and the value is the identification of connected output.

2.3 Evolutionary Algorithm

An evolutionary algorithm operating with the circuit representation introduced in the previous section was developed. All evolutionary operators are realized over graph circuit representation. The evolution is left to run for a given number of generations. The fittest individual is considered as an acceptable result and is transformed to VHDL code.

The primary goal of fitness calculation is to assign a fitness value to any candidate circuit generated by the evolutionary algorithm. It is a crucial part of the proposed method since it affects substantially the quality of generated benchmarks. The fitness value, which has to be maximized here, combines two objectives. First objective of fitness function covers the circuit structure. It is analyzed for those parts that are refused by the synthesis tool in process of structure analysis. Circuits can't consist of any parts which are refused by the synthesis tool. Second part of fitness function covers testability features. Testability analysis is implemented using ADFT program developed in [4]. ADFT is able to calculate average controllability and observability of a candidate circuit. The values are compared with the values required by the user.

3 Experiments

We arranged a set of experiments to evaluate the proposed approach. All experiments were performed for a small circuit (12 components) and a large circuit (250 components). The objectives of these experiments were as follows: (1) to observe how the average fitness value (gained from several independent runs) increases during the evolution (we investigate whether the produced solution is improved over time); (2) to check how observability and controllability of the evolved benchmark circuits differ from the values required by the user; (3) to utilize the proposed evolutionary method to explore (a part of) the design space in case that controllability (observability) is changed and observability (controllability) remains a constant (see example in figure 2).

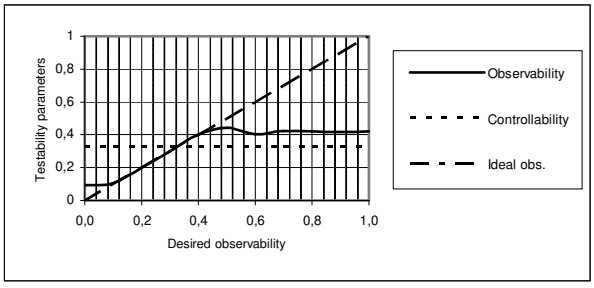

Figure 2. Relation of testability parameters for evolved circuits consisting of 30 components

4 Results

A number of circuits have been evolved with the required and predefined value of controllability and observability. One of the evolved circuits is shown in figure 3.

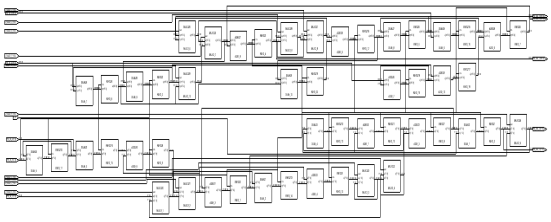

Figure 3. Example of evolved circuit (observability 20.4%, controllability 36.7%)

The circuit consists of 50 components (adders, subtractors and multiplexers) and requires 177 Virtex slices after synthesis to Xilinx Virtex 2 Pro FPGA. We required 20% observability and 33% controllability on average; the obtained results are 20.4% for observability and 36.7% for controllability.

5 Conclusion

In our research we have demonstrated that useful benchmark circuits can be effectively evolved.

The methodology developed and the software which implements the methodology take into account requirements of a user on the testability of the resulting circuit. Thus, the component has certain diagnostic properties; the aspect of function is not seen as important at the moment. For the nearest future, we intend to develop and implement the methodology which will evolve benchmark circuits fulfilling required function and still having desired testability properties. It is believed that utilizing evolutionary approaches will offer completely new solutions to this problem.

References

[1] Harlow, J.: Overview of Popular Benchmark Sets. IEEE Design & Test of Computers, vol. 7-9/2000, pp. 15–18

[2] 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

[3] Kundarewich, P. D., Rose, J.: Synthetic Circuit Generation Using Clustering and Iteration. In: Proceedings of the 2003 ACM/SIGDA, the 11th International Symposium on Field Programmable Gate Arrays, California, 2003, pp. 245–245

[4] Strnadel, J.: Normalized Testability Measures Based on RTL Digital Circuit Graph Model Analysis. In: Proceedings of The fifth International Scientific Conference Electronic Computers and Informatics, Košice, 2002, pp. 200–205