An Overview of Research Activities in Digital Circuit Diagnosis and Benchmarking

Zdeněk Kotásek, Tomáš Pečenka, Josef Strnadel, Daniel Mika, Lukáš Sekanina
Faculty of Information Technology,
Brno University of Technology, Božetěchova 2, 612 00 Brno, Czech Republic
E-mail: kotasek, pecenka, strnadel, mika, sekanina@fit.vutbr.cz

Abstract

In the paper, our research activities are described briefly. In the beginning, two different methodologies for the identification of registers to be included into the partial scan chain and principles of their implementation are described. One of them is based on the utilisation of genetic algorithms, the other one on the identification of feedback loops. Later, graphic tool for parallel *i-paths* analysis is presented. The visual representation described in the paper allows to develop a new methodology of a test controller. Benchmarks circuits are needed for verification above mentioned methodologies. At the end of this paper, it is demonstrated how evolutionary techniques can be used for the process of generating benchmark circuits.

1. Introduction

Several possibilities how to apply a test of a digital circuit exists. Scan approaches (*design for testability techniques*) are one of them. Scan approaches can be divided into two main groups: *full scan* (when all *flip-flops (FFs)* are included in the *scan layout (SL)*) and partial scan (when only part of all FFs is included in the SL).

Many approaches for the registers selection into SL exist. In Section 2, two different methodologies for the identification of registers to be included into the partial scan chain and principles of their implementation are described briefly. One of them is based on the utilisation of genetic algorithms, the other one on the identification of feedback loops. An attention is paid to the computation of time and space complexities of the developed algorithms.

I-path concept (see [1]) is used for reducing an amount of FFs selected to SL in both above-mentioned methodologies. In Section 3 graphic tool for parallel *i-paths* analysis is presented. Described visual presentation allows develop a new methodology of a test controller.

While creating new testability analysis methodologies (partial scan, parallel *i-paths* analysis etc.) benchmark circuits are needed for validating new methods. There is insufficiency of structural RTL benchmarks in actual benchmarks sets. We need to develop a tool for generating benchmark circuits at RT level.

In Section 4, 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. A number of circuits have been evolved with the required and predefined value of controllability and observability. The output of the methodology developed and implemented is in the form of component VHDL code.

2. Partial scan methodologies

We defined the goals of our research activities in the following way: (i) to develop and to implement partial scan methodologies based on distinct approaches, and (ii) to develop methodologies to compare these approaches in terms of computational complexity and the effectiveness of the methodologies. The effectiveness of the particular methodology - the number of registers selected for the scan

chain through which the test will be applied, the time needed to find the solution is considered as well.

In our research we have concentrated on different approaches for the identification of registers to be included into a partial scan chain. They can be classified as (i) the approaches based on the utilisation of genetic algorithms (GAs), and (ii) the approaches based on the identification of feedback loops.

2.1 State-space analysis

The simple analysis of the state-space of partial scan problem requires generate all possible scan configurations and evaluate each of them. Such an approach guarantees optimal solutions to be recognised although it can be considerably time consuming. To justify the development of methodologies based on a GA, we dealt with the state-space analysis first.

For the purposes of the methodology, we denoted the set of all registers in *unit under analysis (UUA)* as $REGS_{UUA}=\{R_1, R_2, \ldots, R_n\}$. The basic idea of the first stage of our research is as follows. It is true that numerous alternatives how to implement a SL in a UUA exist supposing that each such layout can consist of several parallel (multiple) scan chains in which the order of registers is not important for the result (let this case be marked as A1) or is important for the result (let this case be marked as A2). The problem of repositioning FFs in a UUA to minimise their number is solved in [2].

As the first step of our research [3], we determined exact size ($nstructs_n$ – see formula 1) of the SL state-space, i.e., how many SLs can exist within concrete UUA with n registers for both A1 and A2 case. Evaluation of $nparts_k$ [3, 4] depends on selected alternative (A1/A2).

$$nstructs_n = \sum_{k=1}^{n} \left[\binom{n}{k} \cdot nparts_k \right] \qquad (1)$$

2.2 The methodology based on a genetic algorithm

Also, SL problem (SLP) was defined: which of all unique scan chain systems (for given UUA) is characterized by the best cost/quality trade-off between diagnostic properties and acceptable costs of a final design. Because the problem was identified as NP problem, an optimising state-space exploration method based on GA was developed [3, 4]. The purpose of this method is to solve the SL problem in an acceptable time by achieving a high-quality SLP as the result at the costs of bringing a deep knowledge of the problem to the method procedure.

In the method, GA performs a multidirectional search by maintaining a population of potential solutions, i.e., potential SLs. The population of SLs undergoes a simulated evolution from one generation to another: at each generation the relatively good solutions reproduce, while the relatively bad solutions die. Each SL (possible solution) is encoded as a chromosome, which is represented by a bit-string [4]. The goodness or badness of each solution is evaluated by a fitness function, which is related to the objective function of the optimised problem. Using reproduction process, based on a crossover and mutation processes, the evolution ensures propagation of high quality features into the next generation of solutions. Producing several successive generations, the average fitness of the solutions is increasing. The algorithm stops if after a certain number of iterations no further improvements are produced. The best solution that has been produced is that one, which is hopefully close to the searched optimum. Proposed GA-based method runs with time complexity $O(n^3 \cdot log(n))$ and space complexity $O(n^2 \cdot log(n))$.

In general, there is a limitation in the chromosome length [5] when using GA. That means that the upper bound of chromosome length is the length of several thousands of bits. Applying GA to the SLP, it is suitable to process circuits with no more than 250 registers.

2.3 The methodology based on the identification of feedback loops

The goal of the methodology is to identify registers for the partial scan. The procedure is completely different from the above-described approach. It is based on the identification of feedback loops in the UUA and deriving minimal set of registers, which cover all loops.

The translation of UUA VHDL code into a structure, which reflects structural and diagnostic properties of UUA is the first step of the methodology. For this purpose, an analyser/compiler was developed which transforms the VHDL code into a special data-files (list of elements, list of ports, list of connections and list of relations between ports and connections) that are described in [6].

Then the set of PROLOG facts is generated using PROLOG facts-compiler. This set represents comprehensive result of mutual database cohesion. The PROLOG facts cover structural circuit model, including static links and excluding information about behaviour. Therefore a library of elements had to be developed, in the library the behaviour of all elements is described and loaded into the system during the analysis. Then, the identification of feedback loops follows - this step was implemented as recursive procedure in the following steps: 1) Find all successors of elements, 2) Test if there is the same element in the list of successors. If yes, direct feedback loop exists, if no, perform the algorithm recursively, 3) Apply steps 1 and 2 for all elements in the circuit. After all feedback loops were recognised in the UUA, it is necessary to find the minimal coverage of feedback loops by a subset of registers. Proposed method runs with both time and space complexity $O(n^2)$.

2.4 Combination of presented approaches

To be able to solve more complex applications using GA, which in our case means to apply GA-based method on circuits with $n >> 250$ registers, it is needed to solve the limitation [5] in the chromosome length. For these purposes, the GA-based methodology is preceded by loop analysis method and combined SLP solving method is developed.

The proposed method consists of two sequent sub-methods. The first sub-method is the loop identification method. On the basis of this method, k registers (from total number of n circuit registers) are selected to be included into scan (according to their position in feedback loops and their impact on circuit testability) and they are included into $SREG_{UUA}$ set. The $SREG_{UUA}$ set is an input to GA-based method (the second sub-method) for SLP optimisation (the second sub-method). After it is decided about chromosome bit-length (in dependence on $SREG_{UUA}$ set), GA starts. As a result of the combined method, UUA testability improving modification with an acceptable cost/quality trade-off (according to UUA limits) is recommended and it is stored to VHDL file.

2.5 Conclusion

The SLP was identified as NP problem. GA-based method for finding a feasible solution with acceptable testability value and feedback-loop analysis method performing feedback loops analysis to find minimal coverage of all feedback loops by UUA registers were presented. Computational complexities were derived for both methodologies. It can be concluded that when an unmodified GA-based method is used, it is not suitable to process RTL digital circuits with more than 250 registers. On the other hand, GA-based algorithm is able to solve such testability problems as the order of registers in the scan chains and occurrence of multiple scan chains within UUA. To fit the GA-based method to a wider class of RTL circuits, i.e. to be able to analyse circuits with more registers, we decided to combine both GA-based approach and feedback-loop analysis one.

3. Graphic tool for parallel *i-path* analysis

Many researchers have dealt with the parallel *i-path* analysis and a lot of results were published recently. The authors use different strategies (test application conflicts graphs, interval graph theory, *i-path* concept) how to prepare test scheme of RTL elements. Our research deals with the last mentioned principle. We try to develop formal tools for test controller design. For this purpose we need a tool for parallel *i-path* analysis. We are not informed about the existence of such software. Therefore, we decided to create one, which is tailored for our needs.

The principles of parallel *i-paths* analysis tool presented in this paper have its

origin in the *i-path* idea. For these purposes, formal description of digital circuit must be provided, as described in [8]. Existing *i-paths* in the circuit can be used for transport of diagnostic information. For the purposes of test application, both *i-paths* between primary inputs/outputs and internal elements and between all internal elements must be studied and analysed together with transparency properties of internal elements.

Definition 1: An ordered couple *i-path* $I_e=(I_1, I_2)$ assigned to element under test (EUT) is a couple of *i-paths*, where I_1 represents the *i-path* from circuit primary inputs to data inputs of EUT *e* and I_2 represents the *i-path* from data outputs of EUT *e* to primary outputs of a circuit under test (CUT).

3.1 Graphics form of parallel i-paths analysis

Two dimensional array (2D array) in Fig. 1 can be used as a tool to demonstrate the existence of *i-paths* in CUT in a graphic form.

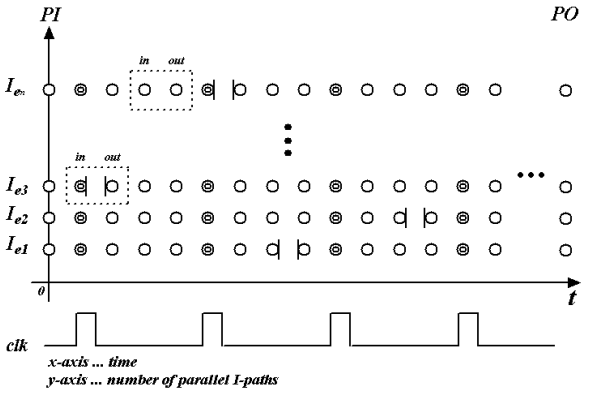
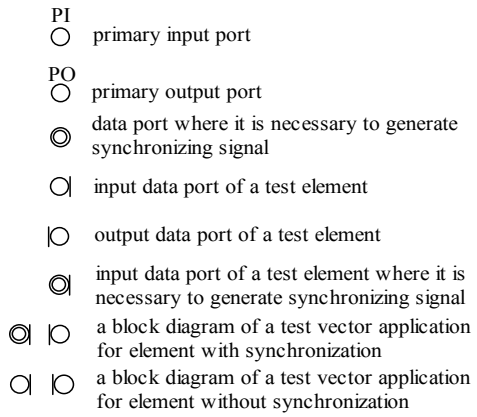


Fig. 1: Graphic form of parallel *i-path* analysis

Legend:



Clock signals are depicted as a sequence of pulses along X-axis. The ports which require clock pulse to be generated to load data(◎), are placed on the same position in a graph as clock pulses. The which do not need clock can be placed anywhere along X-axis. Between two ports of ◎ type, the sequence of ports ○ can be placed.

3.2 Test vector application in graphic form of parallel i-paths analysis

The operation of applying one input test vector consists of three steps: 1) Input test vector transport – the transport of test vector from CUT primary inputs to EUT inputs (I_1-*path*), 2) Test vector application – the propagation of response to test vector through EUT, 3) Test response transport – the transport of response to test vector from EUT outputs to CUT primary outputs (I_2-*path*).

The sequence of three above-mentioned steps we will consider as elementary indivisible unit of parallel *i-paths*. Such unit will be displayed by the port sequence of I_1-*path* directly followed by the port sequence of I_2-*path*, where $I_e=(I_1, I_2)$. Then the transport of diagnostic data through the sequence of ports represents the test vector application phase, the first port is input data port and the second port is output data port of EUT. This fact need not be explicitly brought into graph, because according to the definition of *i-paths* I_1, I_2 and the fact that I_2 follows I_1, these ports are correctly displayed in the graph (Fig. 2).

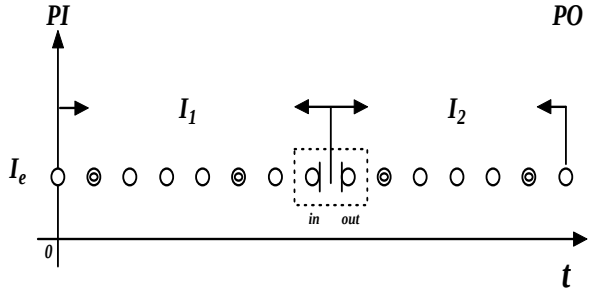


Fig. 2: *I-paths* as elementary part of analysis

From the representation, the conditions (the sequences of control and clock signals) under which I-paths are transparent for diagnostic data can be derived. It is assumed that this methodology and its implementation

will be used to develop a methodology of test controller design.

4. Evolutionary design of synthetic RTL benchmark circuits

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 generated benchmarks quality, 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 [9] have been formulated independently of the proposed evolutionary design. Requirements on the circuit function are not reflected by the procedure of developing the benchmark structure.

The user is supposed to specify the following entries: 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.

4.1 Evolutionary algorithm

An evolutionary algorithm was utilized. The initial population consisting of P individuals is generated randomly. New populations are formed using roulette-wheel selection and mutation operator. N weakest candidate circuits are replaced by the mutated ones. Elitism is ensured. 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.

Fittest of individual is calculated by fitness function. The primary goal of fitness calculation is to assign a fitness value to any candidate circuit developed by the evolutionary algorithm. It is a crucial part of the proposed method since it affects substantially the quality of generated benchmarks. The fitness function, which has to be maximized here, combines three objectives $f_1()$, $f_2()$ and $f_3()$; weight system with c_1, c_2 and c_3 weights is used:

$$fitness = c_1 \cdot f_1() + c_2 \cdot f_2() + c_3 \cdot f_3(). \qquad (2)$$

The parameter $f_1()$ specifies the level of interconnectivity of a candidate circuit. We have to avoid such solutions, which contain separated subcircuits (separated subcircuit is a subcircuit which is not connected to other parts of the benchmark circuit under development), the subcircuits that are connected to neither primary inputs nor outputs and the structures which are possibly refused by the design system as non-synthesizable. If $f_1()=1$, there are no separated subcircuits and $f_2()$ and $f_3()$ will be evaluated; otherwise, we set $f_2()=f_3()=0$ to devalue the candidate circuit. The parameter $f_2()$ shows the variability of interconnections of a circuit. If $f_2()=1$, any two inputs of a given component are not connected to the same point.

The parameter $f_3()$ is calculated using *ADFT* program developed in [9]. ADFT is able to calculate average controllability and observability of a candidate circuit.

4.2 Experiments and results

We have performed hundreds runs of the evolutionary design process in order to find suitable parameters of the evolutionary algorithm. We arranged a set of experiments to evaluate the proposed approach.

Fig. 3: Example of evolved benchmark circuit

We have verified that useful benchmark circuits can be effectively evolved. The methodology developed and the software, which implements the methodology takes into account requirements of a user on the testability of the resulting circuit while the functionality of resulting circuit is not considered yet. 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 expected that the process of generating benchmark circuits in this way will be significantly more complicated. We shall not also neglect the possibility of integrating these procedures into algorithms implemented in design systems and thus offer the possibility of developing components fulfilling the required function and still providing guaranteed and predefined testability properties. It is believed that utilizing evolutionary approaches will offer completely new solutions to this problem.

Acknowledgements

This work has been financially supported by the Grant Agency of the Czech Republic under contract No. 102/04/0737 "Modern Methods of Digital Systems Synthesis" and the Research Intent of FIT, Brno University of Technology, Czech Republic, CEZ: J22/98:262200012 "Research in the Information and Control Systems".

References

[1] Abadir, M.S., Breuer, M.A.: A Knowledge-based System for Designing Testable VLSI Chips, IEEE Design & Test of Computers, August 1985, pp. 56 – 68.

[2] Higami, Y., Kajihara, S., Kinoshita, K.: Partial Scan Design and Test Sequence Generation Based on Reduced Scan Shift Method, J. E. Testing: Theory and Applications, Vol. 7, (1995) 115–123.

[3] Strnadel, J., Kotásek, Z.: Optimising Solution of the Scan Problem at RT Level Based on a Genetic Algorithm, Proceedings of 5th IEEE Design and Diagnostics of Electronics Circuits and Systems Workshop, 2002, Brno, CZ, pp. 44–51.

[4] Strnadel, J., Kotásek, Z.: Testability Improvements Based on the Combination of Analytical and Evolutionary Approaches at RT Level, Proceedings of Euromicro Symposium on Digital System Design Architectures, Methods and Tools DSD'2002, Los Alamitos, USA, ICSP, 2002, pp. 166–173.

[5] Higuchi, T., Yao, X.: Promises and Challenges of Evolvable Hardware, IEEE Transactions on Systems, Man, and Cybernetics, Part C, No. 29, Vol. 1, 1999, pp. 87–97.

[6] Zbořil, F.: VHDL RT Level Parser / Analyser of a Source Code, Proceedings of the ECI'2000, pp. 150–155.

[7] Kotásek, Z., Mika, D., Strnadel, J.: Test Scheduling for Embedded Systems, DSD'2003 Euromicro Symposium On Digital System Design - Architectures, Methods and Tools, September 2003, Belek, Turkey, p. 463 – 467.

[8] Růžička, R.: Formal approach to the Testability Analysis of RT Level Digital Circuits, PhD thesis, FIT BUT Brno, 2002, p. 102.

[9] Strnadel, J.: Normalized Testability Measures Based on RTL Digital Circuit Graph Model Analysis. Proceedings of the 5th International Scientific Conference Electronic Computers, Košice 2002, pp. 200—205.