Testability Analysis Based on the Identification of Testable Blocks with Predefined Properties

Jaroslav Skarvada, Tomas Herrman, Zdenek Kotasek
Faculty of Information Technology,
Brno University of Technology
Bozetechova 1, 612 66 Brno, Czech Republic
skarvada@fit.vutbr.cz, herrman@fit.vutbr.cz, kotasek@fit.vutbr.cz

Abstract

In the paper, the methodology of testability analysis based on the concept of testable blocks is presented. In the methodology the power consumption during test application is also taken into account. For this purpose, power estimation tool was developed and implemented. Integration of the developed software into the professional design flow is described. Experimental results gained as a consequence of applying the methodology on both benchmark and practical designs are demonstrated. The intensions for future research are presented.

1. Introduction

The role of low-power methodologies implemented into modern design systems is increasing. The need for low power design is driven mostly by the increase in the level of integration. More logic integrated into a chip brings about the need for more power to be delivered. Since much of the power consumed by the circuit is dissipated as heat, the relationship between the test activity and the cooling capacity needs to be taken into account. This is also important for all activities connected with diagnostics and testing. For example the 10° C increase in operating temperature leads to doubling the component failure rate [1]. In diagnostic we must also beware of the chip power dissipation limit in order to avoid false error to be detected during test application [3]. High power consumption can lead to other reliability related issues such as electromigration, voltage drops on supply lines, inductive effects, etc. [1]. Strong reasons for low power testing can be also seen in increasing use of embedded systems, especially those which are powered from batteries. Through the utilization of low power design methodologies such as described in [2], it is possible to reduce significantly power consumption during normal (functional) mode of operation of the circuit, but in the test mode this is not so straightforward. As noted in many papers (e.g. [7], [3]), the chip under test usually consumes more power during testing in comparison to normal mode of operation. Greater power dissipation is mostly caused by significantly higher switching activity (when assuming CMOS technology as dominating technology for implementation of ICs that contain more than 10^5 transistors [3]). The source of higher switching activity can be generally traced in low correlation among input patterns. The correlation is used to be intentionally low in order to reduce the test application time.

The vast majority of power reduction techniques concentrate on reducing the dynamic power dissipation by minimizing the switching activity in the circuit under test. On RT level, these techniques can be divided into two main categories – Test Set Dependent (TSD) and Test Set Independent (TSI) approaches [3]. The TSI approaches depend only on the circuit structure and work regardless of the size and type of the test set. This category of techniques includes clock frequency manipulation [5], scan chains modifications/optimizations and usage of low power design methodologies. The TSD approaches are dependent on the size and type of the test set used during test application. In this category the test vector set compaction techniques as well as test vectors and scan cell reordering techniques [3] are mostly used. When higher level of circuit description is available, it is also possible to limit the maximal power consumed during the test by appropriate test schedule but of course in cost of longer test.

2. Motivation for the Research

For the purposes of RTL testability analysis, formal

model reflecting circuit structure and its diagnostic properties was defined at our department [8]. The model is based on sets and relations between them. With this model the structure of the circuit is defined as a five-tupple *UUA* (Unit Under Analysis) = (*E*, *P*, *C*, *PI*, *PO*), where *E* is the set of all circuit elements, *P* is the set of all ports, *C* is the set of all connections between ports, *PI* is the set of all primary input ports ($PI \subset P$), *PO* is the set of all primary output ports ($PO \subset P$). The software was developed which is able to transform a circuit to formal model from high level description (VHDL or Verilog). In the previous period we have demonstrated how the formal model can be used for the development of RTL testability analysis methodologies [11].

One of the methodologies developed as a part of our research is based on the identification of testable blocks (TBs) performed on the formal model. To be able to do so, the properties which must be satisfied by the testable block (TB) had to be defined first. We also had an ambition to include low power testing aspects into the methodology. As a result it was necessary to include power consumption aspects into the formal model to be able to reflect them during the process of TBs identification. We also tried to integrate our methodology into a professional design system flow. For this purpose, we used professional commercial tools.

The paper is organized as follows. In section 3, basic principles of the methodology are explained together with principles of integrating our methodology into professional design flow. Section 4 is devoted to principles of power consumption estimation. These principles were used in our methodology. In section 5, the concept of TB is defined together with the principles of their identification in RTL structures. Section 6 is devoted to the description of power consumption estimation tool which was developed and implemented as an integral part of our methodology described in details in section 7. Experimental results are discussed in section 8, while our intensions for future research are mentioned in section 9.

3. Basic Principles of the Methodology

Similar method mentioned in [12] is also based on partitioning into blocks but it does not utilize scan chain and exploits additional multiplexers. Another method from [13] is designed for at-speed testing. Our approach is different. It exploits professional third party DfT and synthesis tools. And as a result, our approach can be successfully integrated into professional design flow, as shown in Fig. 1. The resulting optimized design can then be synthesized into real ASICs. Our methods operate on RT level and the structural Verilog netlist is used as the design interchange format between our tools and third party professional software. An external test is supposed to be applied through ATE. Test vector sets to be used by ATE are generated by commercial ATPG. The generated sets are then optimized and simulated by our tools and then can be checked again with third party tools.

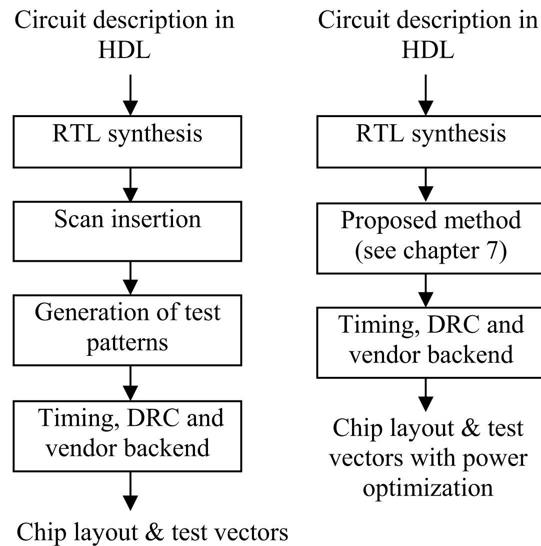

Figure. 1. Original design flow (full/partial scan) (on the left side) and integration of our method into design flow (on the right side)

In our approach the power reduction is based on several optimization processes. First, we replace scan cells design that was originally used by commercial tools that we used by cells with lower power consumption. Also the partial scan approach is used because power consumption during scan shifts is then lower in comparison with full scan approaches [3]. Then, we partition the design into independently testable units that we call *testable blocks*. All these blocks can be successfully tested through border registers that are connected together to form partial scan chain (the concept of border registers is explained later in this text). The test vectors for these blocks are then generated with commercial ATPG. Then we try to optimize the generated test vector sets. For combinational TBs we use test vectors reordering technique to find most suitable permutation of test vectors. This is one of the advantages based on the TBs concept, it allows us to use test vectors reordering techniques to save power during test application process without fault coverage decrease. For all TBs

we also try to optimize the sequence of border registers in scan chain. Lastly, we check if the power consumption during the test is not higher than the predetermined maximal limit. If so, it is possible to decrease power consumption by reorganization of test vectors and scan chain and postpone the test of TBs with high power consumption. As a result, the most power demanding TBs are not tested concurrently.

4. Power Consumption Approximation

For most of nowadays designs based on CMOS technology, it holds that most of power consumption problems exist in the area of dynamic behavior (80% of overall power consumed) [5]. It is probable that this number will slightly decrease in the future due to the threshold voltage alleviation that will lead to the increase of static part of the power consumption. Physically, the dynamic part of power consumption (P_{dp}) is composed of capacitive switching power consumption (P_{sw}) and short circuit power consumption (P_{sc}) and can be computed by equation (1) [1], where C_L is the overall capacitance of gate output, lines and connected input gates, V_{dd} is the supply power, N is the number of ($0\rightarrow1, 1\rightarrow0$) transitions, f is the frequency of clock signal, K is a constant that depends on transistors, V_T is the magnitude of threshold voltage, τ is the input rise/fall time.

$$P_{dp} = P_{sw}+P_{sc} = N f\left(\frac{1}{2}C_L V_{dd}^2 + K\,(V_{dd} - 2V_T)^3\tau\right) \quad (1)$$

The equation (1) is too complex to be used for power consumption estimation in the early stages of the design process, primarily because some parameters depend on physical properties of real chip layout. Therefore a simpler form must be used. To be able to compare designs that will be synthesized to the same technology, the input signal characteristics, transistor constants as well as supply and threshold voltages can be omitted. For further simplification of the process, it will be assumed that the input capacitances of all gates in design are equal. It is also supposed that the clock frequency does not change during the test application. When power consumptions for two or more modifications of the same design running at the same frequency are compared then it is also possible to neglect the frequency parameter. Thus, for comparisons of design modification influence on power consumption it is possible to use the *WTC* (Weighted Number of Transition Count) equation (2) [3].

$$WTC = \Sigma_{i=1..Ng}\,(N_i \times F_i) \quad (2)$$

In the equation, N_g is the number of gates in the design; F_i is the number of fan-out for gate i output. The *NTC* (Number of Transitions Count) is calculated in the same way with F_i supposed to be 1 [3].

5. Testable Block

As mentioned in section 2, a formal approach to testability analysis of RTL components was developed. From the formal model, transparency properties can be derived. For this purpose, the concept of I-path was used which was defined in [14] and further developed in [8]. To derive I-paths, the library of *I*-modes for circuit elements was developed. The set of all *I*-paths is calculated by modified Dijkstra algorithm from [9] which is used to calculate transitive closure of connections and initial *I*-paths derived from elements *I*-modes. The algorithm iteratively tries to increase all *I*-paths depth by one by checking the existence of *I*-mode of the last circuit element on this *I*-path until all *I*-paths are identified. As we discovered, this approach is unable to find all *I*-paths in the circuit. For this reason we have extended the original algorithm to count with the *inverted I-paths*. These paths are similar to classic *I*-paths with the only exception that the data at the end of the path is inverted in comparison with data loaded to the input of the path. By iteratively combining two short inverted *I*-paths together we get classic *I*-path that was omitted in the previous algorithm. This requires modification to I-modes library to be able to define whether the circuit element can operate in inverted or classic *I*-mode and how to set the mode. By this approach we are able to identify all *I*-paths in the UUA.

Our methodology is based on the identification of TBs in UUA. First, it was necessary to define precisely the properties that must be fulfilled by a TB. On the basis of these properties, TBs are recognized. Power consumption optimization in UUA with TBs identified is then performed as a part of the methodology. It is important to note that our methodology works well on circuits with a low occurrence of feedbacks (FIFOs, filters, etc.). An example of implementing these principles on a simple design is shown in Fig. 2. The figure shows simple circuit on which partitioning method was applied. One of the possibilities how to partition the circuit to 3 TBs by our algorithm is shown. Registers that are on the border of delimited zone are called *Border Registers* (BR). For example TB_2 is composed of 3 border registers (R_2, R_3, R_4), no internal registers and 1 circuit element (multiplexor2). In the following text, principles of TB concept are presented together with basic definitions.

The border register of a TB is the register that interconnects TB with other circuit logic (or other TBs). A test to a TB is always applied through these registers. The partitioning into TBs is based on the identification of border registers. In Fig. 2, the border registers for TB3 are registers 4 and 5.

Definition 1:

A set of circuit elements and their interconnection is identified as a TB if the following conditions are true:

1) Each circuit element $e \in E_{TB} \wedge e \in (MX \cup FU \cup R)$ is controllable / observable through border register or primary I/O or it is a border register.
2) Into scan chains only border registers of the TBs are included, the remaining registers of TBs are not scanable.
3) Any TB must not be overlapped with another TB (any two TBs must be disjunctive).

The symbols in the definition have the following meaning:

E_{TB} a set of circuit elements which are included in TB
MX A set of all multiplexers in the circuit
FU A set of all functional elements in the circuit
R set of all registers in the circuit

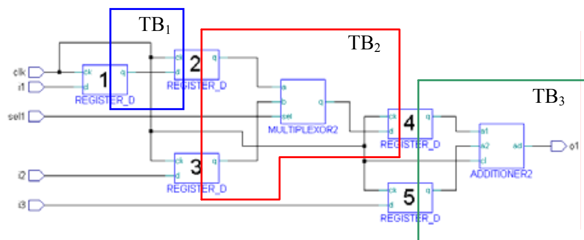

Figure 2. Example of circuit partitioning to TBs

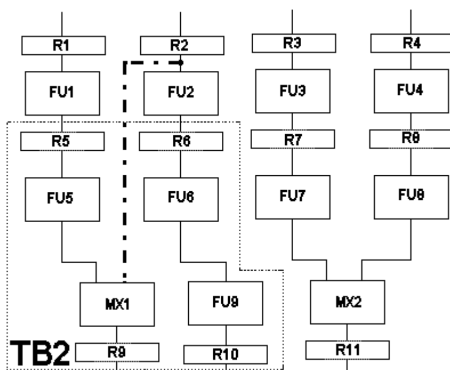

Figure 3. Example of incorrect partitioning that doesn't forms TB

In the structure in the Fig. 3 a TB consisting of R5, R6, R9, R10 border registers could be identified if the dash dotted connection did not exist in the structure. With the dash dotted connection, R2 register should be included into TB to satisfy the conditions defined by Definition 1. The complete formal definitions of TB are given in [10].

6. Power Estimation Tool

6.1. Recent Research – State of the Art

In order to provide feedback between power consumption at various stages in the design cycle, several power estimation techniques have been developed utilizing different levels of abstraction (behavioral to transistor level [5]). The most accurate power consumption estimation can be obtained on the transistor level but for most designs it is too slow. Power consumption estimation on behavioral level is very fast but not too accurate because the actual implementation of the design is not known. RT level seems to be accurate with acceptable speed of computation.

Previous RTL power estimation approaches can be classified into three categories, namely analytical techniques, characterization macro modeling and fast synthesis based estimation [4].

Analytical power modeling techniques use very little information from the functional specification. The power consumption is estimated from general parameters such as a gate count, number and types of logic used, etc. It includes power consumption estimation according to entropy of input and output signals and statistical information such as probability of logical states [5]. These methods are very fast but not too accurate and are rather useful in the early design flow.

Characterization based macro modeling methods construct "black boxes" of various macro blocks of circuit logic. Under these macro blocks various sequences of input patterns are trained. Gate level or transistor level tool is used for the estimation of power consumption of macro block. Based on this data the macro block model for the "black box" is constructed. Early methods simply use the number that denotes the maximal or average power consumption for the block, later methods use function of statistical parameters of the input signal [6].

Fast synthesis based methods use limited synthesis of RTL designs. These methods map the design to defined meta-library which consists of small number of primitives. All of them have defined parameters and known power consumption for various input stimulus.

Resulting power is obtained by gate-level simulation or probabilistic techniques over these primitives.

6.2. Our Approach to Power Estimation

Our power estimation tool operates on principles which are a mixture of the last two mentioned approaches. We exploit models for the AMI 0,5um library. For mapping the designs on various levels of abstraction to the AMI library, professional synthesis tool was used. The resulting netlist (in structural Verilog) is read by our tool together with the AMI power library that we have created. The AMI power library defines the tables of transitions for all AMI elements. Each table consists of exhaustive list of input signal levels (a_i (t)) and signal history in (t-1) with appropriate *NTC* (simulated on gate level) for the defined signal combination. For registers the internal state (s) is added and it is manipulated in the table in the same way as other input signals. Each table field can be 0, 1 or X for don't care (X are setup in optimization step, see bellow and are used for table compaction). Examples can be seen in Table 1. The tool considers only one internal control register (because no more are used in AMI library), but it can be easily extended in the future.

Table 1
Example of one element from the power library

a_1 $(t-1)$	a_2 $(t-1)$	...	a_n $(t-1)$	s $(t-1)$	a_1 (t)	a_2 (t)	...	a_n (t)	s (t)	*NTC*
0	0	...	0	0	0	0	...	0	0	10
...	...	...	...	...	...	...	...	...	...	...
1	X	...	1	X	1	1	...	1	1	5

The power library is stored in human readable format so it is possible to use custom models of various elements. When the library is read, it is stored in the memory as fast lookup table. The power estimation tool is able to compact the input netlist for the simulation purposes. It tries to group together as much combinatorial logic as possible. It groups the logic as long as there is memory available and no sequential element in the group. Then the functional table for the overall group is created. In this step the *NTC* for each element from group is multiplied by fan-out factor of this element to form *WTC*. At the end, the resulting table is constructed from the partial tables and then optimized. In the optimization step the rows of the table are scanned. When the two rows of the table have same transition count and are compatible (in signal transitions, e.g. signal values in columns t and t-1 are same in row but opposite between rows), these rows are grouped together with don't care values set in columns that do not contribute to transitions. By this approach the group of logic between registers is "black boxed" and the power can be quickly estimated by simple lookup.

The accuracy of the tool is acceptable because accurate power models are used for primitives from the power library. The tool can read the testing patterns in ASCII text format generated by commercial ATPG and sequentially apply it to the design for predefined number of clock periods. The scan chains can be also simulated. As the default, the zero delay models are used for primitives, so no glitches and spurious (hazardous) transmission are count for *NTC*. It is also possible to use non-zero delay model and count every transmission by disabling the "black boxing" behavior. It can be done by special configuration parameter. The delay model for every primitive is then read from the AMI library (or can be globally overridden by config). The tool can report overall *NTC* or *WTC* and is exploited in our methodology implementation.

7. Methodology Implementation Details

In this section, the implementation of the methodology is explained. It will also become clear how the implementation can be integrated into professional design flow. It will be demonstrated how these two methodologies differ in the implementation.

1) The starting point is the design description in HDL (VHDL or Verilog). The design can be verified by simulation and synthesized with commercial tool. The output is in the form of RTL structural representation mapped to RTL primitives from technological library (we utilize AMI 0,5µm library, but others can be used as well) which contains enough primitives (registers, scan registers, gates, multiplexers, etc) for the representation of any designs.

The next step is either 2a) step (original design flow) or 2b) (our methodology). After any step, simulation can be performed. It can be also verified whether dynamic parameters were not degraded as a result of step 1.

2a) Scan chains are inserted with commercial tool. The decision on whether BIST or ATE will be used can be accepted at this design phase. When ATE is used, then it is possible to modify the sequences of test vectors. Optionally, additional logic for compression/decompression can be used. It is also possible to implement JTAG port for the transport of test vectors and responses. Commercial ATPG is then used to generate test vectors and to calculate percentage fault coverage if full scan or partial scan is used.

2b) Our methodology is implemented in the following way: The developed RTL component design is converted into structural description in Verilog (so called Verilog Netlist), then from Verilog the formal description is constructed, based on principles described in [8]. Then the structure of the circuit is analyzed to identify *I*-paths (our “veri2parts” tool is used). During the analysis, our AMI 0,5µm I-modes library is utilized which describes *I*-modes of all primitives from the AMI library. On the formal description the partitioning into TBs is then performed (with our “tb” tool), a genetic algorithm is used for this purpose. Different parameters of the partitioning algorithm can be set-up – a preferred size of TBs can serve as an example.

Then, the identified TBs are hierarchically saved into Verilog format and for each TB the set of test vectors and responses to them are generated by commercial ATPG tool. TBs are separated by registers (see the definitions of TB earlier in this paper). Thus, the partitioning into TBs implies the registers which will be included into scan chains. The organization of the scan chains depends on the size and the number of TBs. The order of scan registers and the order of test vectors (for combinatorial only TBs) affect the power consumption and it can be converted to problem of finding proper permutation of elements. The suitable permutation can be found by our software tool based on genetic algorithm (“permfind” tool). To compute the fitness function, power consumption is evaluated by our power estimation tool (“pwrsim” tool). We also modify the design to decrease power consumption during scan shifts – this is done on our formal model. The modification is based on the replacement of implicit registers used by commercial tools, which change their outputs during scan shifts, by registers with have their outputs blocked during scan shifts.

The result of applying the methodology is: a) the UUA is partitioned into TBs, b) the scan chains are designed, c) the sequence of test vectors together with response for each TB is known, with respect to power consumption.

At this stage of the design, a Verilog file is available which describes the UUA modified by our methodology.

3) As the last step, the design can be transferred to physical level. The correct function of the design can be verified by simulation on the physical level .

The complete implementation of the methodology consists of the following software components:

vhdl2parts - software for I-paths identification and formal model construction (based on the analysis of VHDL source text)
veri2parts - software for I-paths and formal model construction identification (based on the analysis of Verilog source text)
tb - software for the identification of TBs,
pwrsim – power estimation tool.
permfind – software for finding least power demanding permutation of test vectors (for combinatorial TBs) and scan cells.

8. Experimental Results

In experiments we tried to verify:

a. Time complexity of used algorithms and its relation to the number of connections (cons) and I-paths – see Table 2
b. Effectiveness of algorithms for partitioning UUA into TBs – see Table 2.
c. Properties of TBs – see Tables 3 and 4.

Standard PC with Amd64 CPU @ 2GHz with 1GB RAM was used for the experiments.

Table 2
Results of TB partitioning algorithm

Name [-]	FUs [1]	FFs [1]	I-paths [1]	Cons [1]	I-paths search time [s]	TB partit. time [s]	Num of TBs [1]	Units outside TBs [%]
COM	45	29	1874	1217	242	3.2	5	17
ISA	75	29	8831	2988	2375	15.2	2	4
DIFFEQ	11	6	2041	213	2.2	0.81	1	0
DEC	29	7	1283	529	22.8	4.3	2	5
FIFO2	226	144	33339	11297	41200	233	6	14
S298	47	19	10204	1367	349	14.7	2	3

Legend for table 1:

Name	UUA identification
FUs	The number of functional units in the UUA (before mapping to AMI library)
FFs	The number of flip/flops in UUA
I-paths	The number of I-paths identified in UUA
Cons	The number of connections (according to definition from [8]) in UUA
I-paths search time	Time needed for identification of all I-paths in UUA
TB partit. time	Time needed for partitioning UUA into TBs
Num of TBs	The number of TBs identified in UUA
Units outside TBs	The ratio of elements (registers and functional units) which were not included into TBs

Table 3
Details about TBs identified in COM circuit – the number of test vectors generated and the number of transitions recognized during the application of these vectors
Circuit COM: partitioned to 5 blocks

Name [-]	BRs [1]	FFs [1]	FUs [1]	Nodes [1]	Vectors [1]	NTCs [1]
TB1	2	4	12	15	23	2520
TB2	4	1	6	3	8	291
TB3	2	4	12	15	23	2520
TB4	2	4	12	15	23	2520
TB5	2	4	12	15	23	2520

Table 4
Details about TBs identified in FIFO2 circuit – the number of test vectors generated and the number of transitions recognized during the application of these vectors
Circuit FIFO2: partitioned to 6 blocks

Name [-]	BRs [1]	FFs [1]	FUs [1]	Nodes [1]	Vectors [1]	NTCs [1]
TB1	28	86	222	241	139	2147059
TB2	3	3	8	7	22	1792
TB3	3	3	8	7	22	1792
TB4	3	3	8	7	22	1792
TB5	5	1	6	3	9	1080
TB6	5	1	6	3	9	1080

Legend for tables 3 and 4:

Name	TB identification
BRs	The number of border registers in TB
FFs	The number of flip/flops in TB
FUs	The number of functional units (from the AMI library) the TB is built from
Nodes	The number of nodes in TB (without interface)
Vectors	The number of test vectors needed for test (with stuck-at fault coverage better than 90%) the TB
NTCs	The number of transitions which occurred in TBs during test application

In tables 3 and 4, blocks that have equal values in columns 2-7 seem to be identical (can be freely swapped in the circuit without function change)

It can be stated that the time complexity is strongly dependent on the number of connections in UUA (Fig. 4, 5) which was an expected result. Our motivation was to gain absolute values of computational times. In Fig. 6 relation between TBs partitioning time and the number of I-paths in UUA is shown. We also verified which UUA portion was kept outside TBs and thus not tested (9th column in Table 2). For example for circuit COM, it resulted in 17% circuitry kept outside TBs.

From tables 3 and 4, the partition of COM and FIFO2 UUA into TBs is evident. FIFO2 circuit is partitioned into TBs unequally, so one block is much bigger than others and produces much more NTC when tested (because it is very complex structure that needs many test vectors) while COM2 is partitioned more equally. Tables 3 and 4 represent detailed information on partitioning of TBs.

We also verified that that the formal model can be used for the development of testability analysis algorithms. We also revealed that data needed for the construction of the model can be gained from data provided by the professional design systems.

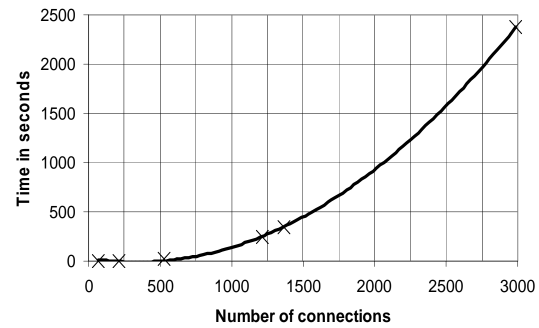

Figure. 4. Relation between I-paths computation time and the number of connections in the circuit

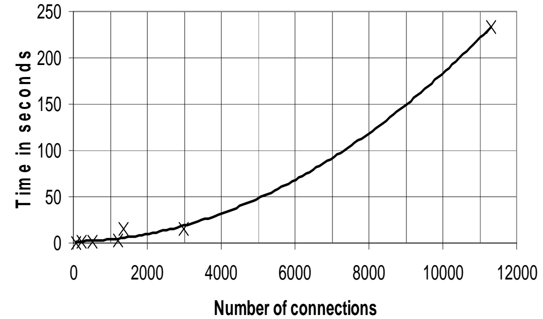

Figure. 5. Relation between TBs partitioning time and the number of connections in UUA

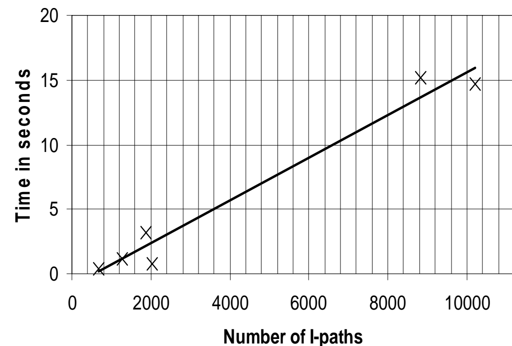

Figure. 6. Relation between TBs partitioning time and the number of I-paths in UUA

9. Conclusions and Trends for Future Research

The software we have developed and experimented with is able to:

1. analyze Verilog/VHDL description and transform it into formal model,
2. partition UUA into TBs,
3. identify border registers and include them into partial scan chain,
4. estimate power consumption of each TB during the test application.

It can be concluded that a formal model (representation) of RTL structure was developed on which testability algorithms can be implemented. It was proven that a diagnostic problem can be thus converted into a mathematical problem (theory of graphs, discrete mathematics and theory of sets). We have shown how the formal model can be utilized for the identification of TBs together with the principles of integrating the methodology into professional design tools. The formal model can be further extended in the future if a need to develop and verify testability analysis methodology appears.

For the future research we have the following intensions:

1. Harmonize and improve fitness function evaluation (used in genetic algorithm for the identification of TBs) with the goal to reflect all defined limitations. Develop the methodology how to test circuitry outside TBs.
2. Evaluate the impact of the number of scan chains used with TBs on UUA power consumption. To evaluate whether it is reasonable to construct separate scan chains (with the need of "padding" to maintain equal length of scan chains) or to allow the interleaving of border registers (belonging to different TBs) among several scan chains. To develop a methodology which will enable the border registers partitioning among different scan chains.

Acknowledgements

This work was supported by the Research Project No. MSM0021630528 - Security-Oriented Research in Information Technology and by GACR project No. 102/05/H050 - Integrated Approach to Education of PhD Students in the Area of Parallel and Distributed Systems (Grant Agency of the Czech Republic).

References

[1] Raghunathan, A., et al.: *High-Level Power Analysis and Optimization*, Boston, Kluwer Academic Publishers 1998, ISBN 0-7923-8073-8, pp. 175

[2] Schmitz, M. T., et al.: *System-Level Design Techniques for Energy-Efficient Embedded Systems*, Boston, Kluwer Academic Publishers 2004, ISBN 1-4020-7750-5, pp. 211

[3] Nicolici, N., Al-Hashimi, B. M.: *Power-Constrained Testing of VLSI Circuits*, Boston, Kluwer Academic Publishers 2003, ISBN 1-4020-7235-X, pp. 178

[4] Ravi, S., et al: Efficient RTL Power Estimation for Large Designs, In: *Proceedings of the 16th International Conference on VLSI Design*, Washington IEEE Computer Society 2003, ISBN:0-7695-1868-0, pp. 431-439

[5] Roy, K., Prasad, S. C.: *Low-Power CMOS VLSI Circuit Design*, New York, John Wiley & Sons, Inc. 2000, ISBN 0-4711-1488-X, pp. 359

[6] Bogliolo, A., Benini, L.: Robust RTL Power Macromodels, In: *IEEE Trans. VLSI Systems*, USA, 1998, vol. 6, , ISSN 1063-8210, p. 578-581

[7] Schuele, T., Stroele, A. P.: Test Scheduling for Minimal Energy Consumption under Power Constraints, In: *19th IEEE VLS Test Symposium*, USA, 2001, pp. 312-318

[8] Ruzicka, R.: *Formal approach to testability analysis of digital circuits on RT level*, VUT FIT, PhD thesis, Brno, 2001, pp. 110

[9] Ruzicka R., Skarvada J.: RTL Testability Verification System, In: *Proceedings of the Work In Progress Session of 30th Euromicro Conference*, Linz, Austria, 2004, ISBN 3-9024-5705-8, p. 101-102

[10] Herrman, T.: Testability Analysis Based on Formal Model, In: *Proceedings of the 7th International Scientific Conference ECI*, Kosice, Slovakia 2006, ISBN 80-8073-598-0, p. 243-248

[11] Hlavička, J., Kotásek, Z., Růžička, R.: Formal Approach to RTL Testability Analysis, In: *Proceedings IEEE LATW 2000*, Rio de Janeiro, BR, 2000, pp. 98-103

[12] Toshinori Hosokawa, Kenichi Kawaguchi, Mitsuyasu Ohta, Michiaki Muraoka: A Design for testability Method Using RTL Partitioning, In: *ATS, 5th Asian Test Symposium (ATS '96)*, November 20-22, 1996, Hsinchu, Taiwan, pp. 88-93

[13] Ho Fai Ko, Nicola Nicolici: RTL Scan Design for Skewed-Load At-Speed Test under Power Constraints, In: *ICCD 2006 proceedings*, pp. 6-11

[14] Abadir, M. S., Breuer, M. A.: A knowledge based system for designing testable VLSI chips, In: *IEEE Design&Test*, August 1985, pp. 56 - 68