RTL Testability Analysis Based on Circuit Partitioning and Its Link with Professional Design Tool

Jaroslav Skarvada, Tomas Herrman, Zdenek Kotasek
Faculty of Information Technology
Brno University of Technology
skarvada@fit.vutbr.cz, herrman@fit.vutbr.cz,
kotasek@fit.vutbr.cz

***Abstract*-The paper presents testability analysis method which is based on partitioning circuit under analysis (CUA) to testable blocks (TBs). A formal approach utilizing the concepts of discrete mathematics is used for this purpose. The partitioning CUA into TBs is further exploited for power consumption optimization during test application. Software tools which were developed during the research and integrated into the third party design tool are also described. Experimental results gained from applying the methodology on selected benchmarks and practical designs are demonstrated.**

***Keywords*-Testable block, power consumption estimation, test vectors generation, power consumption optimization.**

I. INTRODUCTION

The need for low power methodologies is still increasing and is mostly driven by the increase in the level of integration. One reason is that market share of mobile devices still increases, thus for the long operational time the low power design optimizations are highly welcome. Another important reason for low power optimizations are thermal requirements. Since much of the power consumed by the circuit is dissipated as heat, the relationship between the power consumption and the cooling capacity needs to be taken into account. It is worth to note that 10°C increase in operating temperature leads to doubling the component failure rate [2]. In diagnostic we must also beware of the chip power dissipation limit in order to avoid false error detection during test. High power consumption can lead to some reliability issues such as electromigration, voltage drops on supply lines, inductive effects, hot electron effects, etc. [3]. Using low power design methodologies it is possible to significantly reduce power consumption during normal (functional) mode of circuit operation, but in the test mode this is not so straightforward. As noted in various papers (e.g. [7]), the chip under test consumes more power during testing in comparison to normal mode of operation. Greater power dissipation is caused by significantly higher switching activity (when assuming CMOS technology as dominant manufacturing technology for VLSI [3]). During testing there can be generally detected more switching activity due to much lower correlation between input patterns in comparison to functional mode, so the vast majority of power reduction techniques concentrate on reducing the dynamic power dissipation by minimizing the switching activity in the circuit under test. When focusing on RTL, it is possible to divide these techniques into two main categories – Test Set Dependent (TSD) and Test Set Independent (TSI) approaches. 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, scan cell modifications and approaches based on multiple scan chains. 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 vector [1, 4, 5, 7] and scan cell reordering techniques [7, 10] are mostly used. When higher level of circuit description is available, it is also possible to lower the peak power during the test by appropriate test schedule but of course in cost of longer test. Several methods are also based on partitioning into blocks. The method from [9] does not utilize scan chain and exploits additional multiplexers.

In [10], a low-power testing methodology for the scan-based BIST is proposed. A smoother is included in the test pattern generator (TPG) to reduce average power consumption during scan testing, while a group-based greedy algorithm is employed for the scan-chain reorder in order to improve the fault coverage. The reordering algorithm is very efficient in terms of computation time, and the routing length of the reordered scan-chain is comparable to result given by commercial tools. Experimental results of ISCAS'89 benchmarks show that the fault coverage achieved by the 2-bit and 3-bit smoothers are similar to previous methods with the same test lengths. The reduction in average power consumption is 60.06% with a 2-bit smoother and 85.4% with a 3-bit smoother.

In [13], optimal solutions to the test scheduling problem for

core-based systems are presented. Given a set of tasks (test sets for the cores), a set of test resources (e.g., test buses, BIST hardware) and a test access architecture, start times for the tasks such that the total test application time is minimized are determined. A mixed-integer linear programming (MILP) model for optimal scheduling is presented and applied to a representative core-based system using an MILP solver available in the public domain. The MILP model is extended to allow optimal test set selection from a set of alternatives.

Our approach slightly differs from those mentioned above. We are tried to exploit professional DfT and third party synthesis tools and we hook our methods into their design flow. The resulting optimized designs can be then also synthesized into real ASICs. Our method operates on RT level and an external test is supposed to be applied through ATE. In our methodology, the circuit is partitioned to blocks of independent logic (with regard to test) which we call Testable Blocks (TBs). In the next step, test vector sets are generated for the partitioned circuit by third party commercial SATPG. Then, the power optimized partial scan is designed and inserted into design to interconnect TBs. All TBs can be successfully tested through border registers which are formed by partial scan chain. Optionally, it is possible to optimize generated test vector sets by our methodology. For combinatory TBs we use the test vectors reordering technique to find most suitable permutation of test vectors. This is one of advantages of usage of testable blocks that allow us to use test vectors reordering techniques to save power for subsets of original circuit. It is also possible to reduce the peak power consumption by appropriate schedule of the TBs test to avoid concurrent test of most power demanding blocks, but this will be subject of further research.

II. Motivation for The Research

In our previous research we deal with formal approaches to RTL testability analysis. At our department, an RTL testability analysis methodology was developed which utilizes such disciplines as discrete mathematics, graph theory and Petri nets. The methodology is based on the transformation of CUA netlist to structures defined by these disciplines; these structures reflect CUA structural and diagnostic properties. We have demonstrated that if this step is done, then algorithms and procedures known from the above mentioned disciplines can be used for the purposes of testability analysis. Anyway, it was evident testability analysis performed on various levels should be combined with the methodologies which take into account power constraints. It is important because power consumption is not only an important design aspect but in some applications, e.g. embedded systems, power consumption can contribute to the design quality significantly. Thus, we decided to combine testability analysis with power consumption constraints and develop a methodology covering both of these objectives.

III. Problem Definition

A. Basic concepts

Many methods to improve testability parameters of digital circuits are known. Most of them are based on the controllability and observability concepts. Controllability is seen as the ability to set values of inputs of any component from the primary inputs of the circuit, while observability is the ability to observe values of outputs of any component at the primary outputs of the circuit. If these two parameters are not good enough then it is necessary to improve them which is the goal of many testability analysis methods. It is important to note that when these methods are implemented (i.e. some additional hardware is included into CUA) then the area of chip can gently grow or dynamic parameters can become degraded.

In this paragraph, the concept of Testable Block (TB) is defined. It is also indicated how TB can be used to increase testability parameters in terms of controllability and observability of internal nodes of CUA.

TB can be seen as a segment of a digital circuit which is fully testable through its inputs and outputs – border registers or primary inputs/outputs of CUA. Such an approach can be used to reduce the number of registers included in scan chain. Border registers are the only registers which can be used as scan registers. In our methodology, TBs will be identified through evolution algorithm which will operate on a formal model of CUA. The purpose is to subdivide the circuit into number of TBs.

B. Formal Model of Circuit

Our method is based on formal model [6] of circuit on RT level which is explained in the following sections of the paper. The basic quintuple describes the overall structure of circuit. It is based on basic elements of circuit on RT level such as element, port and connection.

Definition 1:

Let $UUA = (E, P, C, PI, PO)$ be an ordered quintuple reflecting CUA structure model on RTL level. Then

- E is the set of circuit elements,
- P is the set of ports of elements,
- C is the set of connections: $C \subset (PI \cup P) \times (PO \cup P)$,
- PI is the set of primary inputs,
- PO is the set of primary outputs.

This definition is based on a traditional view on digital circuit and describes circuit structure using quintuple of sets. The set E of circuit elements is composed of three subsets: $E = (R \cup MX \cup FU)$, where R is the set of registers, FU is the set of functional units; MX is the set of multiplexers. This separation is important for testability analysis purposes. The

set of registers R represents all memory based elements which cause sequential behavior of CUA. Elements from the MX set are those responsible for data paths switching and elements from the FU set are combinational elements.

For structural analysis purposes the port is defined as interface through which diagnostic information can be transported between two circuit elements or between circuit element and primary input/output pin. There are three sets of ports defined in the CUT: PI, PO and P. The PI is set of all primary input ports (circuit input pins/circuit input interface). The PO is set of all primary output ports (circuit output pins/circuit output interface). The set of ports P is composed of three subsets: $P = (IN \cup OUT \cup CI)$ where IN is the set of input ports (excluding primary inputs), OUT is the set of output ports (excluding primary outputs) and CI is the set of control and synchronization ports.

The set of connections C defines connection between ports. It is defined as binary relation on the union of ports and primary input/output ports. It can be stated that C is reflective, symmetric and transitive.

Definition 2:

Let ψ be the function, $\psi: E \rightarrow 2^P$, which assigns elements from the set of ports to each circuit element (E is set of elements and P is set of ports from definition 1), then:

1) $\psi(e) = \{p \mid p \in P \wedge p \text{ is the port of element } e\}$
2) The function ψ is defined over all elements of set E.
3) It must hold: $e_1 \neq e_2 \Leftrightarrow \psi(e_1) \cap \psi(e_2) = \varnothing$

Thus, the function ψ creates the link between the set of elements and the set of ports and assigns the set of ports to elements. The condition 2) provides that the set of ports can be identified for each element. The purpose of condition 3) is that each port belongs to one element only.

C. Definition of the Testable Block

A Testable block (TB) can be identified in CUA as a segment of logic separated by registers from other circuit logic. These separation registers are called border registers. It is guaranteed by the TB definition that the internal circuitry of TB is testable through the TB interface. The transparency properties of elements inside the TB are utilized to transport the test data/responses from/to TB interface to/from all internal TB elements that must be tested. Only the border registers can be inserted into the scan chain. So as the consequence, the number of registers included into scan chain is decreased in comparison to full scan. The following definitions and rules define the structural properties of the TB. The definition number 4 defines the properties of border registers (the scan chain candidates).

Definition 3 – The Testable block:

Let $TB = (E_{TB}, P_{TB}, C_{TB}, PI_{TB}, PO_{TB})$ is ordered quintuple which reflects the structure of Testable block, then:

$E_{TB} \subseteq E$, $P_{TB} \subseteq P$, $P_{TB} = (IN_{TB} \cup OUT_{TB} \cup CI_{TB})$, $IN_{TB} \subseteq IN$, $OUT_{TB} \subseteq OUT, CI \subseteq CI_{TB}$,
$C_{TB} \subset ((PI_{TB} \cup PO_{TB} \cup P_{TB}) \times P) \cup (P \times (PI_{TB} \cup PO_{TB} \cup P_{TB}))$,
$PI_{TB} \subseteq PI$, $PO_{TB} \subseteq PO$

It is easy to imagine that if E_{TB} is the set of elements of TB, then it is also a subset of E set, P_{TB} is than the set of ports of elements from the set E_{TB} and also the subset of P, C_{TB} is the set of connections in TB and also the subset of C. The set C_{TB} can contain connections from and to TB interface, not only connections inside TB.

Definition 4 – Border registers:

Let $BR_{TB} \subseteq R_{TB}$ and
for $\forall r \in BR : \exists p \in \psi(r) \wedge \exists (p_1, p_2) \in C_{TB}$ it holds that
$p_1 = p \wedge p_2 \in (P \setminus P_{TB})$ or $p_1 \in (P \setminus P_{TB}) \wedge p_2 = p$

A test to a digital circuit is always applied through border registers or primary inputs/outputs. The identification of the border registers is the goal of our testability analysis methodology.

The definition 4 states that for each register r from BR_{TB}, there exists port p (belonging to the register r) and there also exists the pair (p_1, p_2) from the set of connections of TB. Then p_1 must be from the set of ports of the register r and p_2 must be a port outside the TB or symmetrically p_2 must be from the set of ports of the register r and p_1 must be a port outside the TB.

Rule 1 – An input to TB can bypass border registers only if it starts at primary input:

Let $\forall (p_1, p_2) \in C_{TB}$ where
$p_2 \in P_{TB} \wedge p_2 \notin \bigcup_{r_i \in BR_{TB}} \psi(r_i) \wedge p_2 \in IN_{TB}$, then

$p_1 \in (PI \cup IN_{TB} \cup OUT_{TB})$

$\bigcup_{r_i \in BR_{TB}} \psi(r_i)$ is the set of all ports of all TB border registers.

Rule 2 – An output from TB can bypass border registers only if it ends at primary output:

Let $\forall (p_1, p_2) \in C_{TB}$ where
$p_1 \in P_{TB} \wedge p_1 \notin \bigcup_{r_i \in BR_{TB}} \psi(r_i) \wedge (p_1 \in OUT_{TB} \vee p_1 \in IN_{TB})$

then $p_2 \in (PO \cup IN_{TB})$

$\bigcup_{r_i \in BR_{TB}} \psi(r_i)$ is the set of all ports of all TB border registers.

Following rules must be also fulfilled:

- Only border registers can be connected to the scan chain.
- All elements to be tested – subset of $(MX_{TB} \cup FU_{TB})$ – must be testable through interface of TB (through border registers or primary input/outputs). It also means that there must be applicable I-paths in the TB for test data/responses transportation from/to the TB interface.
- The overlapping of TBs is not allowed – only connections lines between TBs can be shared and every border register can be shared maximally between two TBs (in one as input and in the other as output or vice versa).

It is important to note that above given definitions and rules are used to develop and implement the algorithms for the identification of TBs. An example of simple circuit partitioning to testable blocks can be seen in figure 1.

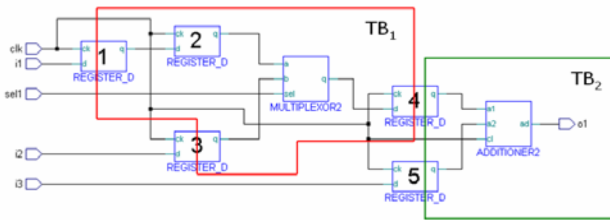

Figure 1: *Example of partitioning CUA into TBs*

D. Power Consumption Estimation

The overall power consumed by the circuit can be seen as sum of static part and dynamic part of power consumption. For most designs based on CMOS technology (excluding very low voltage designs) the static part of power can be omitted. Then the dynamic part of power can be expressed by equation (1) (derived from [2])

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

In the equation (1), the dynamic part of power consumption (P_{dp}) is composed of capacitive switching power consumption (P_{sw}) and short circuit power consumption (P_{sc}), 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 ($0 \rightarrow 1, 1 \rightarrow 0$) 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. 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. For comparisons of design modification influence on power consumption, the *NTC* (Number of Transitions Count) seems to be very important parameter. It is also possible to use the *WNTC* (Weighted Number of Transition Count) equation (2) (derived from [3]).

$$WNTC = \Sigma_{i=1..ng} (NTC_i \times F_i) \quad (2)$$

In the equation, n_g is the number of output gates in the design; NTC_i is the number of transitions count for gate i and F_i is the number of fan-out for gate i.

IV. Proposed Method

A. Method principle

The principle of proposed method can be seen in figure 2. All steps are in detail described in next subchapters. The *RTL synthesis step* and *formal model creation* steps are referenced as *preprocessing*. The dashed line illustrates manual feedback that must be performed by operator in cases when sufficient results cannot be obtained and it consists of manual tuning the genetic algorithm parameters for TB partitioning.

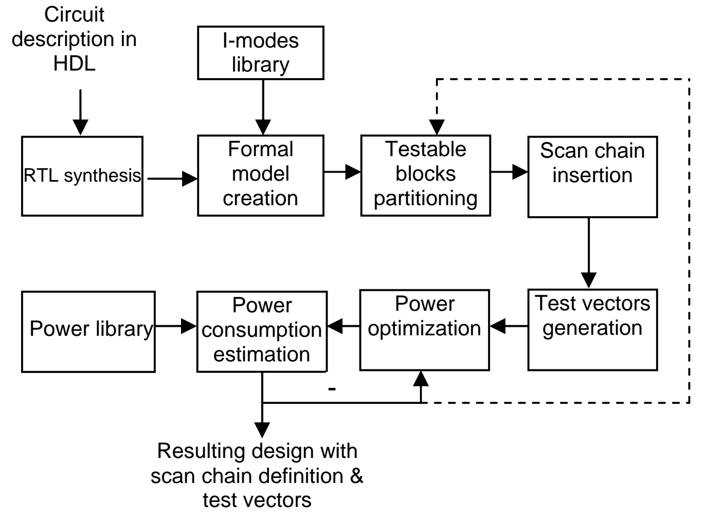

Figure 2: *Block diagram of the proposed method*

B. Preprocessing

In this step, the input design is preprocessed. The goal of this step is to convert the design into formal representation [6] that was outlined in previous chapters. Our tools can then reads the formal representation, make various evaluations on it and finally rewrite the results hierarchically into structural Verilog and other formats that commercial third party DfT tools understand.

The procedure is as follows. First, the design is synthesized into RTL by third party commercial tool. The result is structural VHDL file based on RTL primitives. This VHDL

file is converted to outlined formal model by means of our tool (*vhdl2parts* | *zb2ruz*). The *zb2ruz* tool analyzes the design and identifies all I-paths in the circuit. We recognize two I-paths categories – classical transparent and the inverting ones (can be modeled as I-path with inverter). This classification allows us to detect almost all possible data paths in the circuit. For the I-paths identification, the library of I-modes is exploited.

C. Testable Blocks Partitioning

The purpose of this step is to divide the circuit into independently testable partitions of logic – Testable blocks (TBs). TBs have good testability (preferably near to 100%) and defined interface that consists of the set of border registers and/or primary inputs/outputs that are used to apply test vectors. The mutual independence of TBs (in terms of testing) allows making various optimizations over TB test vectors set without influencing other TBs. For the partitioning *tbpart* tool is used [5]. In the partitioning process the genetic algorithm is exploited. The fact whether a particular register operates as a border register is encoded into chromosome. All identified TBs candidates are then recursively checked to satisfy all previously mentioned definitions and rules. For the TBs that passed the checks the real fitness value is calculated. For the experiments described in this paper the fitness value was evaluated according to the number of logic outside the TBs and was slightly affected by sequentional depth of found TBs (less depth blocks are preferred).

D. Test Vectors Generation

For the test vectors generation the commercial third party SATPG tool is used. For the fault modeling we are using the single stuck at fault model. Test vectors are generated for all TBs independently. This is generally no problem, because the identified TBs must have good testability (by definition). No internal scan inside TBs is allowed. The generated test vectors will be later applied to TBs through interface consisting of border registers and primary inputs/outputs.

E. Power Consumption Estimation

Power consumption during the test application is estimated with our *pwrsim* tool [13]. The tool is capable of counting the *NTC* and *WNTC* for defined design and test vectors. It operates on RT level. With this tool it is possible to do power consumption estimation based on black-boxing technique [12] or slower but more precise cycle accurate simulation. The tool is also possible to count the scan chain power consumption. For this function the external information that describes the connection of registers into scan chain is used. For the scan chain power consumption estimation there are four additional parameters defined – *PWRPer01Shift*, *PWRPer10Shift*, *PWRPer11Shift* and *PWRPer00Shift*. These parameters describe the number of *NTC* that occurs in the scan register when various combinations are shifted through it. For example the *PWRPer01Shift* represents the number of transition in scan register when 0 is shifted after 1.

F. Optimizations for Low Power Consumption

For the optimization of low power design, the tool *permfind* was developed in our department. It allows to perform two types of optimization, namely: 1) the optimization of the sequence of scan registers included into the scan chain, and 2) the optimization of the sequence of test vectors applied to each TB. The goal of both procedures is to find the best possible sequence of scan registers in scan chain and test vectors applied to TBs through scan chain, for which the lowest possible power consumption is achieved. A genetic algorithm was used to investigate the state space of this task. The influence of the optimization to overall power consumption is evaluated by our power estimation tool. As the last step, the scan chain is formed and the final test is constructed. After this phase, the sets of test vectors are ready to be applied to TBs through scan chain.

V. Experimental Results

Experiments were carried out on PC Linux AMD64 2.0GHz, 1GB RAM. The methodology was used to analyze various designs – namely ISCAS circuits, various designs downloaded from internet and synthetically generated benchmarks [8]. We have discovered that the methodology works well on circuits with a low occurrence of feedbacks (FIFOs, filters, etc.). The methodology appears as not very suitable for circuits with high occurrence of feedbacks where additional circuitry allowing breaking feedback loops must be used. For the ISCAS-89 benchmarks set our methodology seems to be practically applicable for s298 circuit which was successfully partitioned to two testable blocks within 364 seconds with 3% of logic that cannot be included into any TBs. The remaining components from ISCAS-89 benchmarks set need additional feedback breaking logic to be used. During experiments we tried to compare our methodology with optimized partial scan (in tables 1, 2 marked as PScan) and full scan (in tables 1, 2 marked as FScan) approaches used in SATPG, ATPG tools. The following factors were checked: fault coverage, test length, the extent of additional logic used and power consumption during the test application (for TBs). For this purpose, the third party commercial SATPG and testability calculator were used. For the scan chain insertion and related circuit modifications the third party commercial tool was also used. In partial scan approach we used the SATPG optimized scan cells selection strategy. The multiplexer based scan cell architecture was used and we experimented with all circuit elements with sequential behavior as possible candidates to be converted into scan type cell. We calculate the fault coverage with equation 3:

$$FC = \frac{FDT}{FFULL} \times 100 , \qquad (3)$$

where *FC* is fault coverage (in percents), *FDT* is the number of faults the test is targeted to (or possibly detects) and *FFULL* is the number of all faults including undetectable faults (tied signals, unconnected pins, etc.) that cannot be detected without circuit modifications. Test points strategy was not used in our experiments but we suppose that it may slightly improve fault coverage of our TB method. The test length is measured as the number of test cycles needed to apply the test. The amount of additional logic used is determined by equation 4:

$$AAL = \left(\frac{FSDC}{OSDC} - 1\right) \times 100, \qquad (4)$$

where *AAL* stands for the amount of additional logic (in percents), OSDC is the number of cells in the original unmodified design after synthesis to technology and *FSDC* is the number of cells in final design after all modifications and synthesis. The power consumption during the test application in *NTC* metrics is evaluated by means of our *pwrsim* tool. It is computed as the sum of power consumption during all scan cycles and all test cycles. For optimizations we used our *permfind* tool. In the experiments the test vectors reordering technique is not used. After the optimization phase the fault coverage is checked by commercial third party simulator.

Circuit FIFO2: the freely available FIFO design, partitioned to 6 blocks, 86 scan cells, 20 primary inputs, 16 primary outputs, original design synthesized to 1198 cells, power consumption – TBs method 1723676 *NTC*,

Method used	Fault coverage [%]	Num. of test cycles	Num. of scan cells	Num. of scan cycles	Added. logic [%]
SATPG	77.7	308	0	0	0.0
PScan	89.1	272	31	96	33.9
FScan	97.8	69	86	62	65.5
TBs	83.5	139	60	139	62.3

Table 1: *FIFO2 circuit results*

Circuit COM: the freely available FIFO design, partitioned to 5 blocks, 21 scan cells candidates, 12 primary inputs, 4 primary outputs, and original design synthesized to 290 cells, power consumption – TBs method 8297 *NTC*

Method used	Fault coverage [%]	Num. of test cycles	Num. of scan cells	Num. of scan cycles	Added. logic [%]
SATPG	76.1	163	0	0	0.0
PScan	84.7	123	9	43	40.3
FSscan	97.4	33	21	29	67.6
TBs	85.5	43	5	43	22.5

Table 2: *COM circuits results*

We supposed that our TB based method should theoretically have fault coverage comparable to partial scan methods, because we used partial scan as transport mechanism for test vectors/responses among testable blocks. From tables 1 and 2 it can be derived that this presumption seems to be valid (these tables were obtained from experiments with circuits that have FIFO structure). In table 1 it can be recognized that for the FIFO2 circuit our TB method has worse fault coverage than ATPG optimized partial scan method but it is still better than SATPG. On the other hand, for the COM circuit in table 2, our TB method has a better fault coverage. In both cases the full scan method seems to have highest fault coverage. The drawback of our TB approach in comparison with partial scan method is that the achieved fault coverage cannot be easily improved as with classic partial scan approach by adding more registers to scan chain, because of the logic that sits outside TBs (logic that was impossible to include to any of the TBs). This logic can only be tested by adding test points to the circuit or by utilizing registers outside the TBs (if any) into scan chain. The volume of additional logic needed for transformation of sequential elements to scan registers seems to be high because used circuits have a high number of sequential elements. We were not able to estimate the power consumption for any other methods (SATPG, Partial scan, Full scan) but TBs because in the time of writing this paper we have not the back-end for our power estimator to be able to read output of used commercial SATPG generator and scan chain selector.

VI. Conclusions

The testability analysis method which is based on partitioning circuit under analysis to testable blocks was developed. It was further exploited for power consumption optimization during test application. The software we have developed and experimented with is able to:

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

For the future research, we want to develop the methodology of partitioning TB test application process into test sessions for further power savings. We also want to evaluate the impact of the number of scan chains used with TBs on CUA power consumption and determine whether it is reasonable to construct separate scan chains or to allow the interleaving of border registers (belonging to different TBs) among several 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) and by FRVS project No. FR2472/2007/G1 – Education support for evolutionary design based on development.

REFERENCES

[1] Bellos, M.; Bakalis, D; et al.: Low power testing by test vector ordering with vector repetition, In: Proceedings of the 5th International Symposium on Quality Electronic Design, IEEE Computer Society Washington, DC, USA, 2006, ISBN 0-7695-2093-6, pp. 205-210

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

[3] 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

[4] Girard, P.; Landrault, C; et al.: Reducing power consumption during test application by test vector ordering, In: Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, 1998, pp. 296 -299

[5] Girard, P.; Landrault, C.; Reduction of power consumption during test application by test vector ordering [VLSI circuits], In: Electronics Letters, 1997, pp. 1752-1754

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

[7] Girard, P.; Guiller, L.; et al.: A Test Vector Ordering Technique for Switching Activity Reduction During Test Operation, In: Ninth Great Lakes Symposium on VLSI , IEEE Computer Society, Washington, USA, 1999, ISBN:0-7695-0104-4, pp. 24

[8] Pečenka T.; Kotásek Z.; Sekanina L.; Strnadel J.: Automatic Discovery of RTL Benchmark Circuits with Predefined Testability Properties, In: Proc. of the 2005 NASA/DoD Conference on Evolvable Hardware, Los Alamitos, US, ICSP, 2005, ISBN 0-7695-2399-4, pp. 51-58

[9] Hokawa T.; Kawaguchi, K.; Ohta, M.; Muraoka, M.: A Design for testability Method Using RTL Partitioning, In: ATS, 5th Asian Test Symposium (ATS '96), Hsinchu, Taiwan, 1996, pp. 88-93

[10] Nan-Cheng Lai, Sying-Jyan Wang and Yu-Hsuan Fu: Low Power BIST with Smoother and Scan-Chain Reorder, Proceedings of the 13th Asian Test Symposium (ATS 2004), pp. 40-45

[11] Krishnendu Chakrabarty: Test scheduling for core-based systems using mixed-integer linear programming, IEEE Transactions on computer-aided design of integrated circuits and systems, vol. 19, No. 10, October 2000, pp. 1163-1174

[12] 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

[13] Skarvada J.: RT Level Power Consumption Estimation Tool, In: Proceedings of the 13th Conference Student EEICT 2007 Volume 4, Brno 2007, ISBN 80-214-3410-3, pp. 467-471