Power Conscious RTL Test Scheduling

Jaroslav Skarvada, Zdenek Kotasek, Tomas Herrman
Brno University of Technology
Faculty of Information Technology
Bozetechova 2, 612 66, Brno, Czech Republic
skarvada@fit.vutbr.cz, kotasek@fit.vutbr.cz, herrman@fit.vutbr.cz

Abstract

In the paper, a methodology of power conscious RTL test scheduling is described. The methodology is based on the fact that circuit under analysis (CUA) is partitioned into testable blocks (TB), the information about the partitioning is the input information for the methodology. TBs are mapped into AMI platform, for each TB the sequences of test vectors are then derived, a professional tool is used for this purpose. The sequences of test vectors are then reordered with the goal to reduce power consumption during test application by reducing switching activities. The power consumption estimation is combined with the implemented platform which allows to gain more precise results. The values of TBs power consumption are then used in RTL test scheduling methodology. The goal is to find test schedule with lowest test application time and lower power consumption than the required maximal value.

1 Introduction

For some applications it is important to reduce power consumption not only for normal function but also for test application process. This holds especially for applications which are powered from batteries where the reduction in test power is important to improve battery life in portable devices employing periodic self-test, to increase reliability of testing and to reduce test-cost.

Peak power consumption during testing is an important concern. For scan designs, a high level of switching activity is created in the circuit during scan shifts, which increases power consumption considerably. In [1], the authors propose a pseudo-random BIST scheme for scan designs, which reduces the peak power consumption as well as the average power consumption as measured by the switching activity in the circuit. The method reduces the switching activity in the scan chains and the activity in the circuit under test by limiting the scan shifts to a portion of the scan chain structure using scan chain disable. Experimental results on various benchmark circuits demonstrate that the technique reduces the switching activity caused by scan shifts.

Gerstendrfer and Wunderlich [12] used the weighted switching activity (WSA) as a metric for the energy consumption and proposed design modifications for reducing power consumption. Their design modifications include NAND/NOR gating logic for masking the scan path activity during shifting, and the synthesis of additional logic for suppressing random patterns which do not contribute to increase the fault coverage.

Scan-based methodologies are often combined with BIST. In [2], a low hardware overhead test pattern generator (TPG) for scan-based BIST is presented. It can reduce switching activity in CUTs during BIST and also achieve very high fault coverage with a reasonable length of test sequence. Since the correlation between consecutive vectors applied to a circuit during BIST is significantly lower, switching activity in the circuit can be significantly higher during BIST than that during its normal operation. Excessive switching activity during test application can damage CUTs during BIST. The proposed BIST decreases the number of transitions that occur at scan inputs during scan shift operations and hence decreases switching activity during BIST. In [13] the technique for test parallelization under power constraints is presented. The technique is targeted for SOC and it uses combination of test parallelization with test scheduling. The fast greedy search algorithm is used for design space exploration process. In [14] the algorithms for testing of VLSI integrated circuits in minimum time without exceeding their power ratings during test are presented. Two formulations of the problem are given: 1) scheduling equal length tests with power constraints and 2) scheduling unequal length tests with power constraints. The test compatibility graph (TGC) model is used in this approach.

Low power/energy Built-In Self Test (BIST) strategy based on circuit partitioning is demonstrated in [3]. The goal of the proposed strategy is to minimize the average power, the peak power and the energy consumption during pseudo-random testing without modifying the fault coverage. The strategy consists in partitioning the original circuit into two structural subcircuits so that each subcircuit can be successively tested through two different BIST sessions.

In [4] 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.

Multiple scan chain has been used in DFT (design for test) architectures primarily to reduce test application time. In [5], a design technique for multiple scan chain in BIST (Built-In Self Test) to reduce average power dissipation and test application time, while maintaining the fault coverage is presented. First, the scan chain is partitioned into a set of smaller chains of similar lengths in such a way, that the total number of scan transitions in the scan chain is minimized. Then, a novel scan re-ordering algorithm in each smaller chain to further reduce the transitions is used. The solution is test-set independent and thus, can be effectively applied to large BIST circuitry.

Our methodology is a combination of circuit partitioning, test vectors reordering and test scheduling under power constraints. For the analysis we use flat RTL circuit description as the input to our methodology. Our methodology is then able to partition the circuit to blocks of logic. For these blocks the test vectors are generated and whenever possible the test vectors reordering process is utilized for power consumption reduction. When dealing with small partitions of original circuit instead of overal flat design the size of solution space for test vectors reordering problem decrease significantly. The circuit partitioning is finally utilized for power constrained test scheduling, when the tests of blocks are scheduled to not overcome predefined power dissipation limit. In this step we are able to incorporate test scheduling methods that are normaly used for SOC. Our methodology can also be used for power constrained test scheduling in situations when only flat RTL design without higher level circuit specification is available to system designer/integrator.

This paper is organized as follows. At first, the motivation for the research and definition of the problem is described. Then the concept of TB and methodology for partitioning CUT to TBs are defined. In next chapter the power consumption analysis and optimization for low power consumption are presented. In chapter 5 the ILP based model for test scheduling is introduced. Then the whole principle of the proposed methodology is described. After that the experimental results and discussion is presented. And finally there is conclusion in chapter 8.

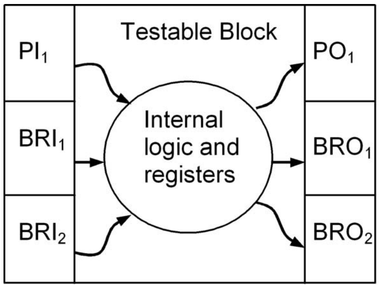

Figure 1. The testable block.

2 Motivation for the research

In our research we were motivated by the fact that in previous period we developed a methodology which allows partitioning a CUT into separate TBs which allows identifying blocks of logic which fulfill required testability parameters, transparency being one of them. To TBs test can be applied through partial scan elements which are identified as one of the TB identification methodology results. In the methodology, the TBs are then mapped into AMI platform. For each TB the set of test vectors are generated with professional tool. While most of methods which aim at power reduction during test application are based on evaluating the average Hamming distance of test vectors, our approach is closely combined with particular technology to which the design is mapped, namely AMI platform. This can be seen as the main difference between our methodology and those published previously.

3 Partitioning to TBs

In our methodology the circuit under analysis (CUT) is partitioned to blocks of logic referred to as TBs. The rules which must be satisfied by TBs are strictly defined. As a consequence, each TB has high testability parameters in terms of internal nodes observability/controllability and defined interface through which the test is applied. The interface consists of registers (we called them border registers) or primary inputs/outputs (PI/PO), see Fig. 1. The border register of a TB is the register that interconnects TB with other circuit logic (or other TBs). We recognize two types of border registers - border input register (BRI) and border output register (BRO). The BRI are used as inputs to TB, while BRO are used as outputs from TB. Additional registers can be present inside TBs, but these registers cannot be utilized as test input or output. Only border registers can be scanned.

The partitioning into TBs is based on the identification of border registers. The following rules also apply to TBs:

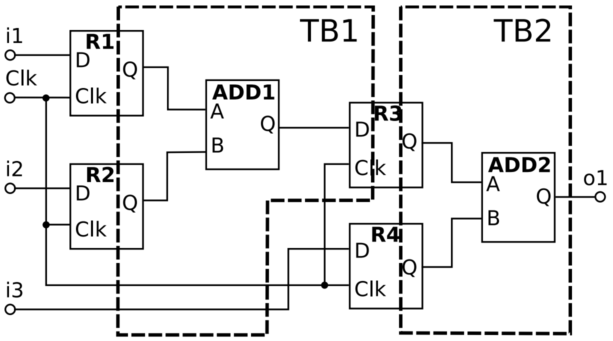

Figure 2. Example of circuit partitioning to TBs.

1. Any two TBs must not overlap. The connections between the TBs can be shared. Any border register can be shared mostly by two TBs (as BIR for one TB and as BOR for the other).
2. The border registers are the only registers that can be included into scan chains.

Fig. 2 gives an example of the implementation of these principles on a simple design which was partitioned to two TBs by our algorithm. For example $TB1$ is composed of three border registers ($R1$, $R2$, $R3$). $R1$, $R2$ are BRI for $TB1$, $R3$ is BRO for $TB1$ (and also BRI for $TB2$). The $TB1$ has no internal registers and one circuit element ($ADD1$).

The above mentioned principles was implemented into software tool called *tbpart* ([8]). Each CUA register can be classified as input, output or internal register of the TB. This means that we have 3^n options how to classify each register where n is the number of registers in the CUA. Thus the size of the search space grows exponentially. Because of the exponential complexity of the problem and because it is possible to utilize a simple representation of the problem in bit-string (chromosome), genetic algorithm was chosen. Genetic algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. Genetic algorithms are known to be quite good at finding generally good solutions in acceptable time even for difficult search spaces. The fact whether a particular register is selected as a border register is encoded into a chromosome. All the identified TBs' candidates are then recursively checked to satisfy all the previously given rules. For the TBs that passed the check, a fitness value is calculated. The fitness value depends on the number of logic outside TBs and sequentional depth of the identified TBs (blocks with the lower value of depth are preferred). So the circuit partitioning algorithm takes into account the complexity of TBs the goal is to identify TBs with possibly equal complexity. Thus, the probability that comparable number of test vectors will be generated for blocks with similar complexity is high. If TBs with different number of test vectors exist in CUA the redundant test vectors are added to TBs with lower number of test vectors to be able to use scan. For test scheduling purposes it must be also possible to deactivate some TBs. The deactivation is done by means of set of null test vectors the sets in which all test vectors have the same value (no switching occurs in connected logic).

4 Power consumption analysis and optimization

Today, the most of digital circuit designs implementations are based on the CMOS technology [10]. With this assumption taken into account the power (P_g) consumed by the gate g of this circuit can be seen as a sum of two power components – the static part of the power (P_{g_S}) and the dynamic part of the power (P_{g_D}) [17]:

$$P_g = P_{g_S} + P_{g_D}. \tag{1}$$

The P_{g_S} is power the gate consumes in steady state (no transitions) and the P_{g_D} is additional power needed for transition from one state to another ($0 \rightarrow 1$, $1 \rightarrow 0$). Actually, the P_{g_D} forms about 90 % of the P_g (from [15]). It is probable that this number will slightly decrease in the future because the actual value is highly dependant on the threshold voltage values that are constantly decreased (due to low power requirements).

Physically, the P_{g_D} is composed of capacitive switching power ($P_{g_{SW}}$) and short circuit power ($P_{g_{SC}}$) and can be computed by following formula (from [17]):

$$P_{g_D} = P_{g_{SW}} + P_{g_{SC}} \tag{2}$$

$$P_{g_D} = NTC_g f_g \left(\frac{1}{2} C_{g_L} V_{g_{DD}}^2 + K_g (V_{g_{DD}} - 2V_{g_T})^3 \tau_g \right) \tag{3}$$

In the formula (3), C_{g_L} is the overall capacitance of gate g and connections, $V_{g_{DD}}$ is the supply voltage, NTC_g is the number of ($0 \rightarrow 1$, $1 \rightarrow 0$) transitions that occurs on gate g, f_g is the frequency of clock signal, K_g is constant that depends on transistors used for gate implementation, V_{g_T} is the magnitude of threshold voltage, τ_g is the rise/fall time of input signal.

The formula (3) 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 formula 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 more precise WNTC (Weighted Number of Transition Count) equation (4) (derived from [16]) or WSA equation (5) (from [12]).

$$WNTC = \sum_{i=1}^{ng}(NTC_iF_i) \tag{4}$$

In the formula (4), $WNTC$ is weighted number of transition count, ng is the number of all gates in the design, NTC_i is the number of all transitions per gate i and F_i is fan-out factor for gate i.

$$WSA = \sum_{i=1}^{ng}(NTC_ic_i) \tag{5}$$

In the formula (5), WSA is weighted switching activity, ng is the number of all gates in the design, NTC_i is the number of all transitions per gate i and c_i is normalized capacity of gate i.

Generally more switching activity can be detected in the circuit during test application [10], that is why there is very low correlation of logical values in the circuit (in time and space), so the transitions between states lead to higher switching activity. Test vectors are often very low correlated (the common goal of test developers is to detect the most faults with the lowest possible test vectors), that means more switching activity. In the test mode, units can be activated in sequences that are unusual for normal functional mode of operation and these sequences are usually not well correlated. To wipe away the difference in power consumption in diagnostic mode and functional mode, we employ test vectors reordering technique. The reordering problem is known to be NP hard with complexity of $n!$, where n is number of elements to be reordered. When the CUT is partitioned to blocks TB_1, TB_2, ..., TB_m with n_1, $n2$, ..., n_m test vectors respectively, $\forall i : (i \geq 1 \wedge i \leq m) \Rightarrow (n_i < n_{CUT})$ the solution space can be reduced significantly.

Because of huge solution space our implementation [9] of test vectors reordering utilizes genetic algorithm. The population of genomes consists of bit strings. In each bit string ordering of test vectors is encoded. The fitness value is obtained as overall NTC from simulation over AMI primitives. The circuits used in our experiments have consistent number of fan out per node. Without fan out factors taken into account the absolute values of NTC and WNTC/WSA differs, but for comparison of quality of genomes the relative numbers (that are very near) are sufficient. Due to high number of genomes the fitness must be determined as quickly as possible, thats why we prefer the NTC metrics.

5 ILP based test scheduling

In mathematics, linear programming (LP) problems involve the optimization of a linear objective function, subject to linear equality and inequality constraints. Linear programs are problems that can be expressed in canonical matrix form:

Maximize (or miminize) c^Tx
Subject to $Ax \leq b$
Where $x > 0$

x represents the vector of variables, while c and b are vectors of coefficients and A is a matrix of coefficients. The expression c^Tx to be maximized or minimized is called the objective function. If the unknown variables are all required to be integers, then the problem is called an integer programming (IP) or integer linear programming (ILP). If only some of the unknown variables are required to be integers, then the problem is called a mixed integer programming (MILP).

In [6], Chakrabarty shows that the test scheduling problem for core based system can be transformed to the MILP programs. He used MILP for finding start times of each test set that leads to minimal test length. The goal was specified as objective function to be minimized with structural constraints taken into account.

We discovered that the ILP can be also used for TB based test scheduling. For a given CUT partitioned to combinatorial TBs the problem can be specified as follows. Let $T = \{t_1, t_2, ..., t_m\}$ represents tests for TB_1, TB_2, ..., TB_m respectively. Let each test t_i ($i = 1..m$) consists of set of generated test vectors $vi_i = \{vi_{i_1}, vi_{i_2}, ..., vi_{i_m}\}$, $Vi = \bigcup_{i=1}^{m} vi_i$ and set of appropriate response vectors $voi = \{vo_{i_1}, vo_{i_2}, ..., vo_{i_m}\}$, $Vo = \bigcup_{i=1}^{m} vo_i$. Let $gvio$ be the bijection $T \rightarrow V_i \times V_o$ that assigns to each test $t \in T$ corresponding test vectors $vi_i \in Vi$ and responses $vo_i \in Vo$. Let $L = \{l_1, l_2, ..., l_m\}$ denotes the length of each test (in test cycles) and gl be the bijection $T \rightarrow L$ that assigns to each test $t_i \in T$ corresponding test length $l_i \in L$, $\forall t_i \in T \Rightarrow (gl(t_i) = |gvio(t_i)|)$. Let ntc be the function $Vi \times Vo \rightarrow \aleph$, that assigns to each pair of test vector and appropriate response positive integer representing NTC. Let $P = \{p_1, p_2, ..., p_m\}$ denotes the average NTC per test cycle for each test and gp be the bijection $T \rightarrow P$ that assigns to each test $t_i \in T$ corresponding $p_i \in P$, $\forall t_i \in T : gp(t_i) = \frac{\sum_{\forall j \in gvio(t_i)} ntc(j)}{gl(t_i)}$. The goal is to partition the test to test sessions $S = \{s_1, s_2, ..., s_k\}$, $\forall i : i \geq 1 \wedge i \leq k \wedge s_i \in S \Rightarrow s_i \subseteq T$ with minimal test application time ($\sum_{\forall s_i \in S} max_{\forall j \in s_i}(gl(j))$ is minimal) and simultaneously not overcoming predefined chip power dissipation limit (p_{limit}) during the test application ($max_{\forall s_i \in S}(\sum_{\forall j \in s_i} gp(j) < p_{limit})$. From given statements it is possible to formulate the ILP.

At first, the matrix of unknown variables is defined. Let $X_{i,j}, i > 0 \wedge i \leq |T| \wedge j > 0 \wedge j \leq |S|$, be the 0-1 variable defined as follows: $X_{i,j} = 1$ if test i is scheduled to session j, otherwise $X_{i,j} = 0$, then the X is matrix of unknown variables for our ILP model. Now we can formulate the objective function:

> Minimize $\sum_{\forall s_i \in S} max_{\forall j \in s_i}(X_{i,j}.gl(j))$, subject to $max_{\forall s_i \in S}(\sum_{\forall j \in s_i} X_{i,j} * gp(j)) < p_{limit}$.

This non-linear objective function must be converted into a linearized form by removing max operator. For this purpose, the approach demonstrated in [11] can be used. The variables C_j, ($\forall j \in S$) must be introduced and limiting conditions must be added:

$$\forall i \in T, \forall j \in S : C_j \geq X_{i,j}.gl(i)$$

Then, the objective function can be modified into the following form:

> Minimize $\sum_{\forall j \in S} c_j$, subject to $max_{\forall s_i \in S}(\sum_{\forall j \in s_i} X_{i,j}.gp(j)) < p_{limit}$.

The ILP defined in this way can be easily converted into GNU MathProg modeling language ant then solved by means of GNU linear programming kit.

6 The methodology

First, an RTL circuit is mapped to AMI platform and partitioned to TBs. For each TB the test vectors are then generated with commercial tool. The sequences of test vectors for TBs which do not contain any additional internal registers (combinational TBs) are reordered with the goal to attain lowest possible number of transition count (NTC).

Finally, the TBs are utilized for power constrained test scheduling. The goal is to find test schedule with lowest test application time and not overcoming predefined maximal limits of NTC/time (conforming to chip power dissipation limit). The test schedule can be represented as a function $T \rightarrow S$, where $T = \{t_1, t_2, ..., t_m\}$ represents tests for TB_1, TB_2, ..., TB_m respectively and $S = \{s_1, s_2, ..., s_k\}$ represents individual test sessions. In every session s_i ($i = 1...k$), none, one, or more TBs can be tested. For each TB, the sequence of test vectors is known. The test vectors are applied through BRI and PI. Test responses are observed through BRO and PO. Border registers are all scanned. TBs which are not currently tested are deactivated. Power consumption in each test session must not be higher than maximal defined chip limit. Our model works with NTC, therefore the power consumption is related to maximal number of switchings during the application of test vectors. For the test sessions setup the ILP is used.

Table 1. SATPG results.

CUA	PI/ PO	Cells	FC in %	Test cycles	NTC
k10	20/5	560	65.50	109	1022
k20	39/10	1119	67.28	178	3134
k5	12/4	298	65.14	81	646

7 Experimental results

Experiments were performed on PC, with AMD 64bit 2GHz CPU, 1GB RAM. The circuits k10, k20, k5 [7] developed at our department were used for experiments. Details about these circuits are available in Table 1, including the information about the number of primary inputs/outputs (PI/PO) and the number of cells after mapping the component into AMI $1.2\,\mu m$ technology. The results gained with the use of professional SATPG tool are also demonstrated in the table (fault coverage, test length, and NTC for the derived sequence of test vectors), single stuck-at-fault model is used. The fault coverage is evaluated by means of a commercial tool, the formula 6 is used.

$$FC = \frac{FDT}{FFULL}.100. \qquad (6)$$

In the formula, the symbols have the following meaning: FC - fault coverage, FDT - the number of faults detected by the test, $FFULL$ - the number of all faults in CUA. The application of one test vector and the response evaluation is denoted to as one test cycle. Test length is gained as the number of all test cycles.

7.1 Partitioning into TBs

The circuits k10, k20, k5 were partitioned into TBs by means of the method described in [8] see Table 2. In the first column, the name of the circuit is given. PI/PO represents the number of primary inputs/primary outputs and cells represents the number of cells after the CUA is mapped into AMI $1,2\,\mu m$ technology. For all TBs, test vectors were generated by means of commercial ATPG tool. FC is fault coverage of the generated test while NTC/cycle represents an average number of switching during one test vector application.

Table 2. Partitioning of k5 circuit to TBs.

TB	PI/ PO	Cells	FC in %	Test cycl.	NTC/ cycle	Opt. NTC/ cycle
k5_1	5/3	17	100	6	15	8
k5_2	5/3	17	100	6	15	8
k5_3	5/2	14	100	8	21	10
k5_4	3/1	9	100	7	13	6
k5_5	3/5	17	100	8	15	6

7.2 The optimization of test vectors sequences

All identified TBs are combinational components. Thus, it is possible to reorder the sequence of test vectors with the goal to reduce the number of switchings during the test application [10]. To reorder the sequence of test vectors, the methodology published in [9] was used. The result is seen in the last column of Table 2, it provides an average number of switchings during the application of one test vector (after the reordering was performed).

It is clear that by the reordering the sequence of test vectors, the NTC/cycle value can be significantly reduced. For TB k5_5 the reduction was really considerable - 40 % of the original NTC/cycle value. For other TBs, all the values are near 50 % of the original value.

7.3 Test scheduling by means of ILP

The results of test scheduling methodology based on ILP are presented in Table 3, GNU linear programming kit was used for this purpose. The objective of the methodology was to schedule CUA test to the lowest possible number of test sessions and the power consumption in each session not exceeding a predetermined value. The maximal power consumption was modeled by means of maximal NTC/cycle (Pwr limit in Table 3). During experiments test schedules were developed for Pwr limit equal to 25 %, 50 %, 75 %, 100 % of Pwr max. The value of Pwr max was evaluated as NTC/cycle+1 gained for TB tests running concurrently in one session.

Each row in Table 3 represents a test schedule. The meanings of the symbols in the table are evident: number of test cycles, number of test sessions, the maximal value of NTC/cycle, Pwr limit in NTC/cycle and in percentage related to Pwr max.

Table 3. Results of test scheduling.

CUT	Pwr limit NTC/ cycle	Pwr limit in %	Test cycles	Test ses.	Max NTC/ cycle
k10	39	25	36	5	34
k10opt	39	25	15	2	38
k10	78	50	21	3	72
k10opt	78	50	8	1	76
k10	118	75	14	2	113
k10opt	118	75	8	1	76
k10	159	100	8	1	158
k10opt	159	100	8	1	76
k20	78	25	?	?	?
k20opt	78	25	15	2	78
k20	157	50	21	3	157
k20opt	157	50	8	1	150
k20	236	75	14	2	226
k20opt	236	75	8	1	150
k20	317	100	8	1	316
k20opt	317	100	8	1	150
k5	19	25	-	-	-
k5opt	19	25	21	3	16
k5	39	50	21	3	36
k5opt	39	50	8	1	38
k5	59	75	14	2	49
k5opt	59	75	8	1	38
k5	80	100	8	1	79
k5opt	80	100	8	1	38

The test schedule for k5 with Pwr limit equal to 39 NTC/cycle (50 %) is shown in Figure 3. The schedule consists of three test sessions. During the first session, k5_1 and k5_2 TBs are tested concurrently while k5_3, k5_4, k5_5 TBs are deactivated. The first test session consists of 6 test

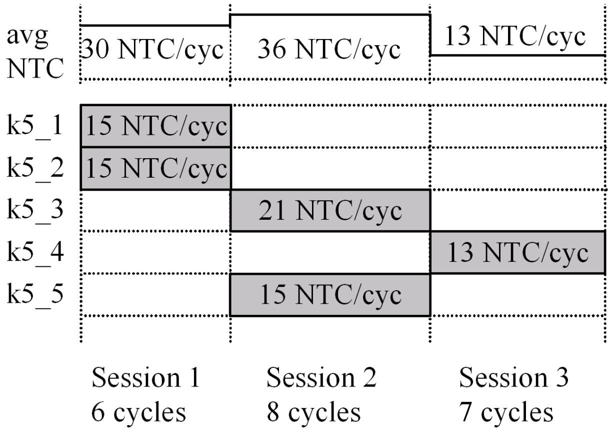

Figure 3. Example of test schedule for k5 circuit, pwr limit was set to 39 NTC/cycle.

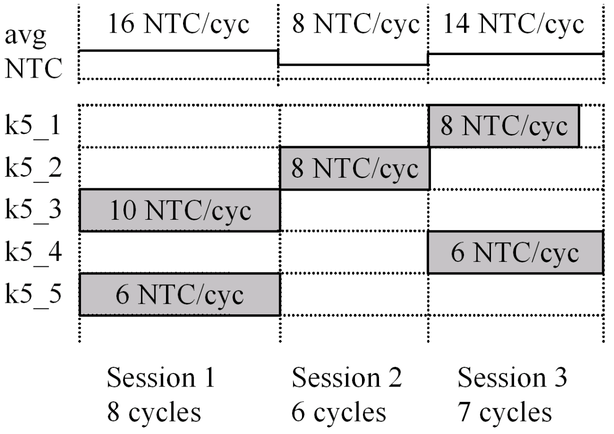

Figure 4. Example of test schedule for k5 circuit after test vectors reordering, pwr limit was set to 19 NTC/cycle.

cycles. To test k5_1 a k5_2 TBs, 12 test vectors are needed. During applying the test to k5_1 TB, the average power consumption will be 15 NTC/cycle, the same value was gained for k5_2 TB. Thus, the average power consumption for the first test session is 30 NTC/cycle.

During the second test session, k5_3 a k5_5 TBs will are tested concurrently, k5_1, k5_2, k5_4 are deactivated. The second session lasts for 8 test cycles and the average power consumption is 36 NTC/cycle. In the third session, k5_4 TB is tested, all the remaining TBs are deactivated. The session duration is 7 test cycles and the average consumption is 13 NTC/cycle.

It can be derived from the table that the length of the test is dependent on the value of Pwr limit, the lower value is required then the longer test is scheduled.

The highest number of test sessions (5) was gained for k10 component with Pwr limit equal to 39 NTC/cycle (25 %). On the contrary, for k20 with Pwr limit 78 NTC/cycle (25 %), the test schedule was not derived because GNU linear programming kit required more memory space which was not available on the system used for the evaluation.

7.4 Test Scheduling by Means of ILP for TB with Optimized Sequence of Test Vectors

Other experiments were performed for TBs with optimized sequence of test vectors. An optimized test has an opt suffix in its name (Table 3). When an optimized test set is used, then a lower number of test sessions are needed. The k10 component can serve as an example: for pwr limit equal to 39 NTC/cycle (25 %), without any optimization the test schedule consists of 5 test sessions compared with 2 test sessions after the optimization. As a result, the number of tests cycles is reduced from 36 to 15.

For other components, similar results were gained for k5 with pwr limit set to 19 NTC/cycle (25 %) without optimization the test sequence cannot be derived because k5_3 requires 21 NTC/cycle (marked by means of dashes in Table 3). The test schedule in Fig. 4 consists of 3 test sessions. During the first session, k5_3 and k5_5 are tested while k5_1, k5_2, k5_4 are deactivated. The test session lasts 8 cycles. The test application of k5_3 requires 10 NTC/cycle, while for k5_5 it is 6 NTC/cycle. An average consumption in the first session will be 16 NTC/cycle. In the second session, k5_2 is tested, the remaining TBs are deactivated. The session lasts for 6 cycles and the average consumption is 8 NTC/cycle. In the third session, k5_1 a k5_4 are tested, k5_2, k5_3 a k5_5 are deactivated. The test of k5_1 is shorter by two test cycles than k5_4, thus 2 redundant test vectors must be added to the sequence of test vectors. During the third session the average consumption is 14 NTC/cycle.

8 Conclusions

In the paper, the methodology for power constrained RTL test scheduling based on ILP was introduced. The methodology allows to partition test application into independent test sessions during which power consumption is not higher than the predetermined value. As the secondary result of partitioning test into test sessions, the minimization of the total test application was gained. The test sessions are identified by means of ILP.

Before the power constrained test scheduling is performed, TBs must be identified in the CUA. If a sufficient number of TBs cannot be identified, the test scheduling cannot be done. This fact limits the usage of the methodology to circuits without feedback loops and containing sufficient number of registers which holds especially for pipeline circuits.

If a TB is a purely combinational component, then the sequence of test vectors can be optimized and power consumption during test application reduced. The results of the methodology are then more valuable. Both approaches, i. e. with and without optimization were demonstrated in this paper.

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.

References

[1] Nadir, Z., B., Sudhakar, M., R., Irith, P.: A Low Power Pseudo-Random BIST Technique. In: 2002 IEEE International Conference on Computer Design (ICCD'02), 2002, pp. 468–473

[2] Seongmoon, W.: Generation of Low Power Dissipation and High Fault Coverage Patterns for Scan-Based BIST. In: International Test Conference 2002 (ITC'02), 2002, pp. 834–843

[3] Girard, P., Guiller, L., Landrault, C., Pravossoudovitch, S.: Circuit Partitioning for Low Power BIST Design with Minimized Peak Power Consumption. In: Eighth Asian Test Symposium (ATS'99), 1999, pp. 89–95

[4] Nan-Cheng, L., Sying-Jyan W., Yu-Hsuan F.: Low Power BIST with Smoother and Scan-Chain Reorder. In: 13th Asian Test Symposium (ATS'04), 2004, pp. 40–45

[5] Debjyoti, G., Swarup, B., Kaushik, R.: Multiple Scan Chain Design Technique for Power Reduction during Test Application in BIST. In: 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'03), 2003, pp. 191–198

[6] Chakrabarty, K.: Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming. In: Computer-Aided Design of Integrated Circuits and Systems, Volume 19, Issue 10, Oct 2000, pp. 1163–1174

[7] RTL circuits k5, k10, k20, URL: http://www.fit.vutbr.cz/research/prod/index.php?id=57&notitle=1, [May 2008]

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

[9] Skarvada, J.:RT Level Test Optimization for Low Power Consumption, In: MEMICS proceedings 2007, 2007, Czech Republic, ISBN 978-80-7355-077-6, pp. 185–192

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

[11] Williams, H. P.: Model Building in Mathematical Programming. 2nd ed., New York, Wiley, 1985

[12] Gerstendorfer, S., Wunderlich, H.-J.: Minimized power consumption for scan-based BIST. In: Proceedings of Test Conference, 1999, pp. 77–84

[13] Larsson, E., Peng, Z.: System-on-Chip Test Parallelization Under Power Constraints. In: European Test Workshop, Sweden, 2001

[14] Chou, R., Saluja, K., Agrawal, V.: Scheduling Tests for VLSI Systems under Power Constraints. In: IEEE Trans. on VLSI Systems, vol. 5, no. 6, 1997, pp. 175–185

[15] Devadas, S., Malik, S.: A Survey of Optimization Techniques Targeting Low Power VLSI Circuits. In: 32nd Design Automation Conference, USA, 1995, pp. 242–247

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

[17] Raghunathan, A., et al.: High-Level Power Analysis and Optimization, Kluwer Academic Publishers (1998), Boston, USA, ISBN 0-7923-8073-8, p. 175