

# Efficient Partial Scan Cell Gating for Low-Power Scan-Based Testing

XRYSOVALANTIS KAVOUSIANOS

University of Ioannina

and

DIMITRIS BAKALIS and DIMITRIS NIKOLOS

University of Patras

---

Gating of the outputs of a portion of the scan cells (partial gating) has been recently proposed as a method for reducing the dynamic power dissipation during scan-based testing. We present a new systematic method for selecting, under area and performance design constraints, the most suitable for gating subset of scan cells as well as the proper gating value for each one of them, aiming at the reduction of the average switching activity during testing. We show that the proposed method outperforms the corresponding already known methods, with respect to average dynamic power dissipation reduction.

Categories and Subject Descriptors: B.8.1 [**Performance and Reliability**]: Reliability, Testing and Fault-Tolerance

General Terms: Algorithms, Design, Reliability

Additional Key Words and Phrases: Low-power testing, scan-based testing, scan cell gating, partial gating

## ACM Reference Format:

Kavousianos, X., Bakalis, D., and Nikolos, D. 2009. Efficient partial scan cell gating for low-power scan-based testing. *ACM Trans. Des. Autom. Elect. Syst.*, 14, 2, Article 28 (March 2009), 15 pages, DOI = 10.1145/1497561.1497571 <http://doi.acm.org/10.1145/1497561.1497571>

---

## 1. INTRODUCTION

Advances in silicon manufacturing have provided the capability of creating large and complex systems on a single chip (SoC). This new design revolution has deteriorated traditional problems such as increased test development

---

Authors' addresses: X. Kavousianos, Department of Computer Science, University of Ioannina, Ioannina, Greece; email: [kabousia@cs.uoi.gr](mailto:kabousia@cs.uoi.gr); D. Bakalis, Department of Physics, University of Patras, Patras, Greece; email: [bakalis@physics.upatras.gr](mailto:bakalis@physics.upatras.gr); D. Nikolos, Department of Computer Engineering and Informatics, University of Patras, Patras, Greece; email: [nikolosd@ceid.upatras.gr](mailto:nikolosd@ceid.upatras.gr). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org). © 2009 ACM 1084-4309/2009/03-ART28 \$5.00 DOI 10.1145/1497561.1497571 <http://doi.acm.org/10.1145/1497561.1497571>

ACM Transactions on Design Automation of Electronic Systems, Vol. 14, No. 2, Article 28, Pub. date: March 2009.

effort and test application time, while it has revealed some others such as the increased power dissipation during testing. High power dissipation in test mode stems from several reasons as outlined in Girard [2002]:

- In scan-based testing, the large number of scan in/out operations causes high switching activity at the circuit.
- Many circuits are designed to process correlated data. On the contrary, in test mode the correlation between consecutive test vectors is low; therefore the switching activity may elevate beyond the acceptable limit.
- Concurrent testing of multiple cores in SoC environments contributes to the reduction of test application time but also results in excessive energy and power dissipation.
- The DFT circuitry which is mostly idle during normal operation is intensively used in test mode.

The elevated power dissipation during testing causes several problems such as reliability problems, permanent circuit damage, increased product costs, performance degradation, reduced autonomy of portable systems in the case of periodic testing, and decreased overall yield.

In scan-based testing, the major portion of the power and energy is dissipated during the scan in/out process as reported in Gerstendorfer and Wunderlich [2000]. Several methods have been proposed to cope with this problem. Wang and Gupta [2002] present a low transition automatic test pattern generator (ATPG). Bonhomme et al. [2006], Saxena et al. [2001] and Whetsel [2000] utilize multiple scan chains. Bonhomme et al. [2002], Dabholkar and Chakravarty [1994], Dabholkar et al. [1998] and Sinanoglu and Orailoglu [2002] deal with scan cell ordering. Bhunia et al. [2005], ElShoukry et al. [2007], Gerstendorfer and Wunderlich [2000], Parimi and Sun [2004], Sankaralingam and Touba [2002], Sharifi et al. [2005] and Zhang and Roy [2000] propose scan cell output gating during scan in/out.

All methods that gate the outputs of the scan cells during scan in/out are based on the fact that the dynamic power dissipated by the circuit during testing is mainly attributed to the large number of transitions propagating to the combinational part of the circuit while the scan chain(s) are loaded/unloaded with test vectors/responses. Even though loading/unloading of scan chain(s) is mandatory for providing stimulus and receiving the response of the circuit, the transitions in the scan chain(s) caused by the shifting of test data in and out, do not need to propagate into the combinational part of the CUT, as they do not offer anything to the testing process.

In Gerstendorfer and Wunderlich [2000] a modified scan cell was proposed to gate the outputs of all scan cells during scan in/out. By providing an extra gate at the output of every scan cell, the outputs are held at a constant value (0 or 1) during scan in/out, depending on the type of gating. The extra gate is transparent in capture mode and during normal operation. In Parimi and Sun [2004] a modified scan cell is proposed with an extra latch. The latch is active only during scan operation and ensures that the inputs of the combinational part of the CUT change only when complete vectors are shifted in. In Zhang

and Roy [2000] multiplexers have been used as gating elements to retain the combinational circuit inputs stable during shifting. Although these methods eliminate the average dynamic power dissipated by the combinational part of the CUT during scan in/out, the gating logic required introduces circuit performance degradation, due to the logic inserted on critical paths as well as high area overhead.

In order to avoid circuit performance degradation, the insertion of gating logic at the outputs of a limited number of scan cells (partial gating), which are not at the critical paths, has been proposed in ElShoukry et al. [2007], Sankaralingam and Touba [2002] and Sharifi et al. [2005]. ElShoukry et al. [2007] present an algorithm, based on random search, targeting average power reduction whereas Sankaralingam and Touba [2002] present an algorithm, based on Integer Linear Programming, targeting peak power reduction. Both take into account the area overhead. Sharifi et al. [2005] target static as well as dynamic power dissipation reduction by inserting gating logic at all scan cells which are not at the critical paths of the CUT. This method apart from the larger area overhead compared to the methods of ElShoukry et al. [2007] and Sankaralingam and Touba [2002] suffers from long computational times for the determination of the gating logic values.

In a different approach in Bhunia et al. [2005] gating transistors in the supply to ground paths are used so as, in scan mode, to turn off the first level of logic connected to scan cell outputs. The need for transistors insertion however makes this approach difficult to use with standard cell libraries that don't have power-gated cells.

The main contribution of this article is that it proposes a very efficient method for the selection of the most suitable subset of scan cells for gating along with their gating values. Contrary to the method of ElShoukry et al. [2007], which is based on random search, the proposed method is systematic. We initially propose a method for the selection of a set of lines, denoted hereafter as representative lines, consisting of the inputs and a small portion of the internal lines of the combinational part of the CUT. At each representative line a weight is assigned, which is indicative of the average switching activity at the rest circuit lines activated by a transition at that line. A very effective algorithm is then presented that minimizes the transitions propagating to the combinational part of the circuit during scan in/out. This is achieved by the efficient selection and gating of the outputs of a predetermined by the designer number of scan cells (excluding those on critical paths) which is based on an effective utilization of representative lines.

The article is organized as follows. The proposed method is given in the next section, while in Section 3 experimental results are presented. Finally Section 4 concludes the article.

## 2. PARTIAL GATING OF SCAN CELLS

In this section we, at first, present the partial gating scan architecture. We then present a method that identifies a set of representative lines as well as a method that utilizes the representative lines, in order to select and gate a subset of the scan cells.

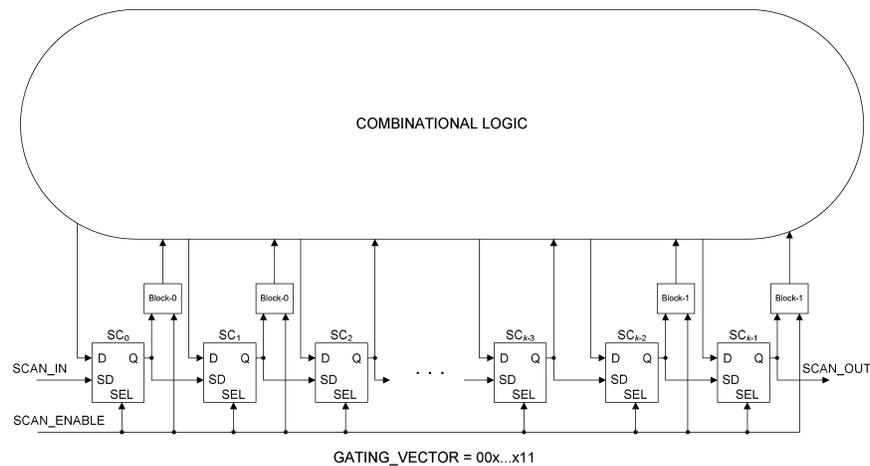


Fig. 1. Partial gating scan architecture.

## 2.1 Scan Architecture

Consider a circuit under test consisting of a combinational part and flip-flops. All flip-flops are converted to scan cells and form one or more scan chains (for simplicity we consider hereafter the single scan chain case, however the application of the proposed method to multiple scan chains is straightforward). Every scan cell output is an input and every scan cell input is an output of the combinational part. Note that the primary inputs of the CUT are also considered as parts of the scan chain (e.g., through the presence of a wrapper around the CUT). The scan chain is controlled by the *Scan-Enable* signal. When *Scan-Enable* is active, a vector is serially shifted in or out of the scan chain. When *Scan-Enable* is inactive, the circuit is either in capture mode or in normal operation.

According to the partial gating technique (Figure 1), blocking logic (Block-0 or Block-1) is inserted between the outputs of some of the scan cells and the inputs of the combinational part in order to hold the outputs of those scan cells at a constant logic value (0 for Block-0 and 1 for Block-1) during scan shift in/out thus preventing unnecessary transition activity at the combinational part. Blocking logic can be constructed in various ways (as reported in Gerstendorfer and Wunderlich [2000], Sankaralingam and Toubia [2002], and ElShoukry et al. [2007]) and it is transparent during the capture cycle and during normal operation.

The objective of the partial gating technique is to select the proper subset of the scan cells to gate as well as the appropriate gating value (0 or 1) for each one of the gated scan cells, according to various restrictions and goals. The outcome of the partial gating technique (i.e., the selected scan cells and the gating value for each one of them) can be represented by a logic vector, denoted as Gating-Vector. Every bit of the Gating-Vector corresponds to a specific scan cell and is either assigned a specified logic value (0 or 1) or it is left unspecified (x). If a bit of the Gating-Vector is specified, then a Block-0 or a Block-1 blocking logic circuit is inserted at the output of the corresponding scan cell, as it is shown in

Figure 1, to enforce the application of the 0 or 1 logic value respectively at the corresponding input of the combinational part during scan in/out. On the other hand, if a bit of the Gating-Vector is unspecified, then no gating logic is inserted at the output of the corresponding scan cell. For example, the Gating-Vector that corresponds to the circuit in Figure 1 is  $00x \dots x11$ .

Thus the target of every partial gating technique is to derive the Gating-Vector that achieves the maximum power reduction and does not violate area and performance design constraints.

Although this paper considers stuck-at fault testing, the proposed method can be also applied to transition delay fault testing. The scan architecture of Figure 1 works well with the launch-off-capture scheme (also called broad-side test) which is the most widely accepted solution in industry. For the launch-off-shift scheme (also called skewed-load test) the scan architecture needs a small modification. Specifically, instead of using the Scan-Enable signal in the blocking logic, an extra signal has to be utilized that must be deasserted one clock cycle earlier than the de-assertion of the Scan-Enable signal (both signals are asserted concurrently).

## 2.2 Representative Lines

An overview of the proposed method for the selection of the representative lines and the assignment of their weights follows. In the beginning, the combinational part of the CUT is extracted. Each line of the circuit is assigned an initial weight that is indicative of the average switching activity expected on this line. Then, the set of internal lines is gradually reduced until a predefined number of them are left, which, along with the inputs of the circuit, form the set of representative lines. Note that the term “circuit inputs” refers to the inputs of the combinational part and consists of both the primary inputs and the scan cell outputs of the CUT. When a line is dropped, its switching activity is estimated using the remaining lines that are representative-candidates. To this end the weight of each dropped line is shared to the lines in its fan-in logic cone, which have not been dropped yet. At the end of the line-dropping process the weight of each remaining line, which is considered as a representative line, is indicative of the switching activity at the dropped lines in its fanout cone. A large weight on a representative line implies that there is a strong possibility a transition at this line to activate a large number of transitions at the lines in its fanout cone.

The outline of the weight calculation and representative lines selection algorithm is given in Figure 2. The first step is to assign initial weights to all circuit lines. Let  $P(l)$  denote the signal probability at line  $l$ . The signal probability of the inputs of the circuit can be either set to  $1/2$  or even better calculated from the set of test vectors. In the first case the calculation of signal probabilities has the advantage of being test set independent, whereas in the second case it is more accurate and may lead to a somewhat better estimation of the transitions in the internal lines (in this article the experimental results have been derived considering the signal probability of each input equal to  $1/2$ ). The signal probability  $P(l)$  of each internal line can be estimated by using a signal

---

```

Calculate initial weights.
Drop Fan-out Branches.
Drop outputs of single-input gates.

While  $|CL| > RepLines$  do
  Mark the gate output  $g$  with the smaller weight.
  Propagate the weight backwards and drop line  $g$ .
End loop

```

---

Fig. 2. Weight calculation and representative lines selection algorithm.

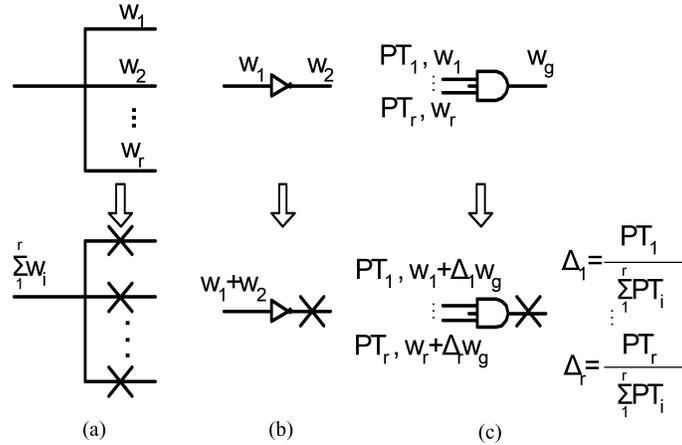


Fig. 3. Line dropping procedures.

probability propagation algorithm. In this paper we have implemented the Full Range Cutting Algorithm as described in Bardell et al. [1987], which is rather simple and provides fairly accurate signal probability estimations. Under the temporal independence assumption, the transition probability at line  $l$  is defined as  $PT(l) = 2P(l)(1 - P(l))$ . The transition probability of a line is indicative of the average number of transitions that appear at the line in a time interval. Thus we consider the transition probability of line  $l$  as its initial weight  $w_l$ . Note that if the capacitance of each line is known, then it can be taken into account for the calculation of the weights of the lines. In this case the weight of line  $l$  is equal to  $w_l = PT(l) \cdot C_l$ , where  $C_l$  is the capacitance of line  $l$ .

The second step is to drop all fanout branches, by summing their weights and assigning the result to the fanout stem, as shown in Figure 3(a). This is justified by the fact that every transition at a fanout stem implies one transition at each fanout branch. Similarly, the outputs of all single-input gates are dropped as shown in Figure 3(b) and their weights are added to the corresponding gate inputs, since a transition at the output implies a transition at the input too. If the input of the gate is a fanout branch, which has been already dropped, then the weight of the output is added directly to the fanout stem.

Let  $RepLines$  be a user-defined parameter specifying the number of representative lines. Let  $CL$  be the set of lines of the circuit. The While-End loop in Figure 2 iterates until  $RepLines$  lines are left in set  $CL$ . At each step, the

output of the gate with the smaller weight is selected and marked for dropping. A line with a small weight is a good choice because the small weight indicates a small number of transitions and thus the error caused by not considering this line as a representative candidate is also small. Moreover, the error induced by sharing a small weight to other lines is also small. The weight of the marked gate output is shared among the inputs of the gate proportionally to the transition probability of each input as shown in Figure 3(c). Transitions at the output of a gate are caused by transitions at the inputs of the gate. A gate input with large transition probability is usually expected to cause more transitions at the output of the gate than another input with smaller transition probability. Therefore, we assume that the contribution of each gate input line to the average number of transitions at the gate output line, is equal to its transition probability divided by the sum of the transition probabilities of all gate inputs. For example, in the extreme case where a gate input receives a constant logic value, that is, its transition probability is 0, its contribution to the average number of transitions at the gate output line is equal to 0 according to Figure 3(c), which is justified by the fact that all transitions at the output are caused by transitions at the remaining inputs of the gate.

Let us assume that the output  $G_{OUT}$  of a gate  $G$  is marked for dropping. If the inputs  $G_{I1}, G_{I2}, \dots, G_{In}$  of gate  $G$  are not dropped, then they share the weight of line  $G_{OUT}$  as it is shown in Figure 3. If some of them have been dropped in a previous step, then the weight share of  $G_{OUT}$  propagated to these inputs must be further propagated backwards in their fan in logic cones until it is shared among lines which are not dropped. One way to do that is to apply recursively the line dropping procedures of Figure 3 for  $G_{I1}, G_{I2}, \dots, G_{In}$ . However this must be carefully done in order to avoid visiting multiple times the common circuit lines in their fan in logic cones. Therefore, the following procedure (*Weight Back Propagation Procedure*), which guarantees that every line will be visited at most once is adopted: The circuit is partitioned into logic levels, where a gate belongs to level  $k$  only if it is driven by gates belonging to levels  $k - 1, k - 2, \dots$  etc. A line resides at level  $k$  if it is the output of a gate belonging to level  $k$  (the gates of the first level are driven from the inputs, while the gates of the last level drive the outputs). The algorithm begins from the output of the gate that must be dropped ( $G_{OUT}$ ) and propagates its weight backwards to the inputs of this gate as shown in Figure 3. Let us assume that  $G_{OUT}$  resides at level  $k$  and thus the inputs of the gate  $G_{I1}, G_{I2}, \dots, G_{In}$  reside at level  $k - 1, k - 2, \dots$  etc. Every line among  $G_{I1}, G_{I2}, \dots, G_{In}$  that has been dropped in a previous step is marked. Then the procedure continues with level  $k - 1$  considering only the marked lines of level  $k - 1$ . Every marked line of level  $k - 1$  propagates its weight (which was propagated to this line in the previous step) backwards to the level  $k - 2$  in the same way (some lines of level  $k - 2$  may be marked at this iteration). The back propagation procedure continues in the same way with level  $k - 2, k - 3$  etc and stops either at level 1, or at level  $L$  if no lines are marked at this level. At the end of the Weight Back Propagation Procedure, the marked gate output line is dropped since its weight was shared to the closest not yet dropped lines.

The Weight Back Propagation Procedure in the worst case requires only one pass of the fan-in logic cone of the marked gate output line. This case occurs

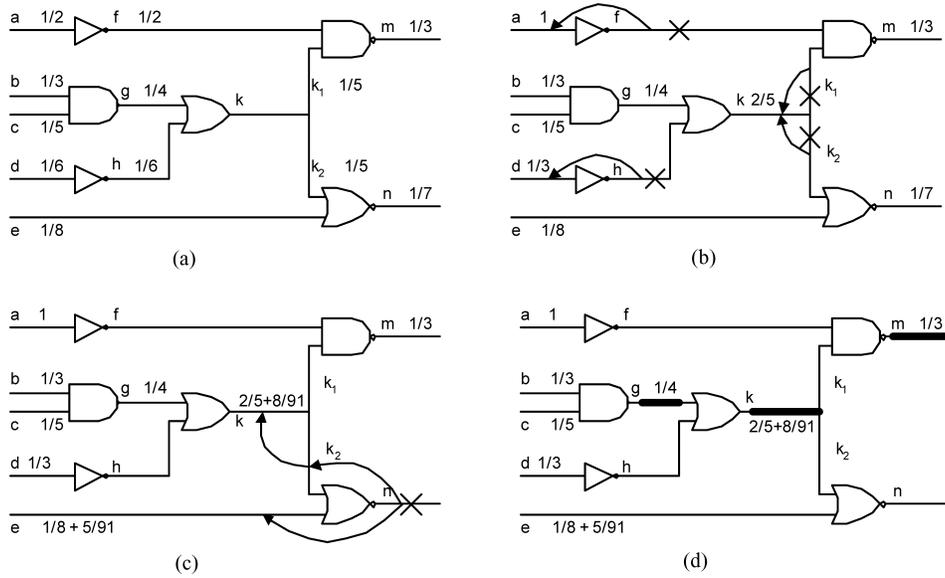


Fig. 4. The application of the proposed algorithm on an example circuit.

only when the marked output line is near the outputs of the circuit and in the same time all lines in its fan-in cones (other than the primary inputs), are already dropped. Then the weight of the marked output line has to propagate all the way back to the primary inputs. Note however that this is a very rare situation, which if it appears at all, it appears only during the last iterations of the process.

*Example:* Consider a sequential circuit whose combinational part is shown in Figure 4(a). Assume that RepLines is set equal to 8. Thus, 3 internal lines have to be finally selected, which along with the 5 inputs of the circuit will constitute the set of representative lines. The transition probability of each line has been calculated and is considered as the initial weight of the line (the values appear at the corresponding lines of Figure 4(a)). Figure 4(b) shows the dropping of the fanout branches (lines  $k_1$  and  $k_2$ ) and the outputs of the single-input gates (lines  $f$  and  $h$ ). Next, the gate output  $n$  with the smallest weight is selected. The weight of line  $n$  is shared among lines  $e$  and  $k_2$  according to the formula shown in Figure 3(c) as:

$$\begin{aligned} \text{line } e &: 1/7 \times (1/8)/(1/5 + 1/8) = 5/91 \\ \text{line } k_2 &: 1/7 \times (1/5)/(1/5 + 1/8) = 8/91 \end{aligned}$$

Since line  $k_2$  has already been dropped, the weight is back propagated and added to the weight of line  $k$  as shown in Figure 4(c). The bold lines, shown in Figure 4(d), denote the internal lines that are finally selected.

According to the line dropping procedures of Figure 3, the sum of the initial weights of the lines of the circuit, which is indicative of the average switching activity expected at these lines, is equal to the sum of weights of the representative lines. The weight of each representative line is indicative of the average

number of transitions at the circuit lines in its fanout cone stimulated by a transition at that line.

The computational time required for the selection of representative lines is negligible, even for the largest benchmarks.

### 2.3 Proposed Algorithm for Partial Gating

The objective of the proposed algorithm is to reduce the switching activity by selecting a subset of scan cells, which are not at the critical paths, and gating their outputs during loading/unloading of the scan chains. In order to minimize the hardware overhead imposed and in the same time to maximize the power reduction achieved, the proper scan cells must be selected and the suitable gating has to be applied to each one of them (output of the scan cell constantly set to either 0 or 1 during scan in/out). The smaller is the portion of scan cells that are allowed (by the design constraints) to be gated, the more critical is their selection and gating, as it will be shown in Section 3.

The main idea of the proposed algorithm is to reduce the number of transitions occurring at the representative lines and, hence, at their fanout cones during scan in/out. A transition at a representative line with a large weight is expected to stimulate transitions at more circuit lines than a transition at a representative line with a small weight. Therefore, if we maximize the number of representative lines with large weights that remain stable during scan in/out, then the switching activity at the circuit lines is expected to be minimized.

Applying partial gating in a specific CUT, or equivalently the corresponding Gating-Vector (see Section 2.1) transforms the vector sequence applied to the combinational part of the CUT during scan in/out into one with reduced switching activity. The Gating-Vector must comply with the following area and performance design constraints:

*Area overhead design constraints:* The number of specified bits of the Gating-Vector (selected cells for gating) should not exceed a given threshold.

*Performance design constraints:* Scan cells on critical paths should not be gated and thus the corresponding bits in the Gating-Vector should remain unspecified.

In our approach, we systematically derive the Gating-Vector that achieves the best switching activity savings according to the given area and/or performance constraints.

The proposed algorithm maximizes the number of representative lines with large weights which are set concurrently to constant logic values by the Gating-Vector while in the same time restricts the number of gated scan cells below the given threshold. Obviously, it may not be possible to set concurrently all representative lines at the selected logic values, especially when the number of scan cells allowed to be gated is small. Therefore, a greedy approach is adopted where the lines with the largest weights (greatest switching activity impact on the rest circuit) are considered first.

The outline of the proposed partial gating algorithm is presented in Figure 5. Initially, the representative lines are selected. Then, a number of vectors activating both logic values 0/1 at every representative line are generated by an

- 
1. For each representative line generate  $N$  vectors, complying to the performance design constraints, to activate the 0,1 logic values.
  2. While the number of specified bits in the Gating-Vector is below the threshold set by the designer do:
    - a. Choose the representative line with the maximum weight.
    - b. Among the vectors generated for this line select the one with the maximum Gain.
    - c. Merge the selected vector with the Gating-Vector.
    - d. Remove the representative line and the corresponding vectors.
    - e. Remove all generated vectors of the remaining representative lines that are not further compatible with the Gating-Vector.
  3. Gate all scan cells which correspond to logic 0/1 in Gating-Vector.
- 

Fig. 5. Outline of the partial gating algorithm.

ATPG tool. Each activating vector consists of specified bits (0/1) as well as unspecified bits (x) and if it is applied on the combinational part of the CUT it activates either a 0 or a 1 logic value on the corresponding representative line. The scan cells corresponding to the specified bits of these activation vectors are candidates for gating and the logic values of these bits determine the type of the gating that must be used. Since scan cells that belong to critical paths should not be gated, all activation vectors setting these scan cells to a specified value (0 or 1) are discarded.

The algorithm then generates the Gating-Vector by merging activation vectors generated for the representative lines. Initially, the Gating-Vector is set equal to the all x vector. Then the algorithm iterates and at each iteration, the representative line with the maximum weight among the representative lines under consideration is selected. For every activation vector generated for this line a gain value is calculated as the number of the x values of the Gating-Vector that will remain unspecified if this activation vector is merged with the Gating-Vector. By merging we mean that all specified bits of the selected activation vector are copied into the Gating-Vector (at each iteration of the algorithm all candidate activation vectors are compatible with the Gating-Vector). The activation vector generated for this line with the maximum gain is selected. If two or more activation vectors have equal maximum gains, then the activation vector that is compatible with most of the activation vectors of the remaining representative lines is selected. When an activation vector is selected, it is merged with the Gating-Vector and the activation vectors generated for the remaining lines that are not further compatible with the Gating-Vector are discarded. The algorithm terminates when the number of specified bits of the Gating-Vector reaches the predefined limit. Finally, the proper blocking logic is inserted at the outputs of the scan cells corresponding to the specified logic values of the Gating-Vector.

The vectors generated at step 1 (Figure 5) need only to activate logic values 0, 1 at the representative lines which usually rely near the inputs of the circuit (see Section 2.1). Consequently, the activation vectors have a small number of specified bits. Therefore, the proposed algorithm covers a large portion of the representative lines, which depends on the maximum number of specified bits allowed in the Gating-Vector.

### 3. EVALUATION—COMPARISONS

In order to evaluate the effectiveness of the proposed method, we performed a series of experiments on the largest ISCAS'89 benchmark circuits. Compacted test sets produced by a commercial ATPG tool were used. All test sets achieve complete fault coverage of non-redundant single stuck-at faults and, unless otherwise stated, the x values were replaced randomly with logic values 0 and 1 by the ATPG tool in order to increase the possibility of detecting un-modeled faults. All power dissipation estimations were done using the Synopsys Prime Power estimation tool and a CMOS standard cell library (180nm, 6-metal layer, 1.8V) assuming typical process parameters. All necessary procedures for the evaluation of the proposed method have been implemented in C language.

We assume that each benchmark circuit has a single full scan chain. Note that the proposed method can also be applied exactly in the same way on circuits with multiple scan chains. In all experiments we present the reduction of the dynamic power dissipation at the combinational part of the circuit throughout the whole testing phase taking into account the power in the capture cycles as well. The power dissipation of the scan chain is not affected by the proposed method, thus we do not consider it in the experiments. In all the experiments performed, the computational time for the selection of the scan cells and the appropriate value for their gating even for the larger benchmarks s38417 and s38584 was only a few seconds.

At first, we studied the effect of the percentage of gated scan cells on the average dynamic power reduction achieved. We ran 45 different experiments for the s9234 benchmark circuit by generating 45 different Gating-Vectors with 10%, 20%, . . . , 90% randomly selected and randomly specified bits (5 different Gating-Vectors for each percentage). We also generated 9 Gating-Vectors using the proposed method for the same percentages of specified bits, using as representative lines the outputs of the scan cells (inputs of the combinational part) and 1.0% of the internal lines. The results are shown in Figure 6. The X-axis presents the percentage of gated scan cells, or equivalently the percentage of specified bits in the Gating-Vector, while the Y-axis presents the reduction of the average dynamic power dissipation in the combinational part of the CUT, compared to the case where no gating is used. The lines corresponding to the random selection of gated scan cells are labeled *Random(Max.)*, *Random(Avg.)* and *Random(Min.)* and present the maximum, average and minimum reduction achieved over all 5 experiments for each selected percentage of gated scan cells. The line labeled *Proposed* presents the reduction achieved by the proposed method. The label at each data-point of the Proposed line shows the value of the ratio of the reduction percentage of the proposed method to the average reduction percentage (*Random(Avg.)* curve) of the random selection. The reduction achieved by the proposed method is far better than the reduction achieved by the random selection, especially when the number of scan cells allowed to be gated is low. This is due to the efficient selection of scan cells and the type of gating for each one of them. However, as the number of gated scan cells increases, and especially beyond 80%, the reduction of the proposed method converges with the random selection (for 90% of scan cells gated the difference is very

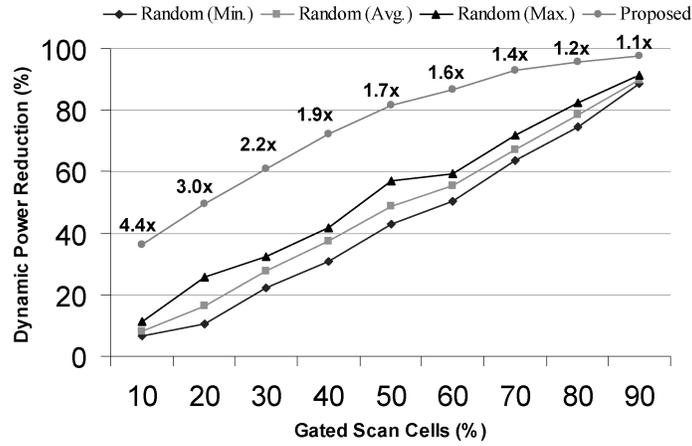


Fig. 6. Proposed method against the random gating of scan cells.

Table I. Average Dynamic Power Dissipation Reduction of the Proposed Method

Circuit	Gated Scan Cells Percentage (%)								
	10	20	30	40	50	60	70	80	90
s5378	20.8%	37.2%	50.1%	63.6%	77.9%	85.8%	90.6%	95.7%	97.9%
s9234	40.5%	53.9%	63.9%	74.6%	82.5%	88.2%	92.6%	95.7%	97.3%
s13207	40.4%	55.1%	66.9%	76.5%	86.3%	92.3%	96.8%	98.8%	99.7%
s15850	40.8%	55.3%	64.7%	75.0%	82.5%	89.6%	93.5%	96.7%	98.1%
s38417	49.4%	61.3%	71.4%	78.6%	84.7%	90.6%	94.3%	97.5%	99.9%
s38584	47.9%	62.6%	71.5%	78.8%	84.4%	89.0%	92.9%	96.0%	98.0%
Average	40.0%	54.2%	64.7%	74.5%	83.1%	89.2%	93.5%	96.7%	98.5%

small). This is expected when the number of gated scan cells is very large, because then the number of transitions propagating to the combinational part of the circuit is very small, independently of the selected scan cells for gating or the type of gating. However, the latter case is impractical since a large number of gated scan cells introduces noticeable area overhead.

The set of representative lines consists of the inputs and a small subset of the internal lines of the circuit's combinational part. As the number of the representative internal lines increases, the execution time of the partial gating algorithm outlined in Figure 5 increases as well, due to the growth of the vector's volume, which must be generated and processed. Therefore, for large circuits, it is rather preferable to use a small number of representative lines. We verified experimentally that by selecting 0.2% of the internal lines as representative, good results can be obtained for any number of gated scan cells.

In Table I we present the dynamic power dissipation reduction achieved by the proposed method for the largest ISCAS 89 benchmarks. The first column presents the name of the benchmark circuit while columns 2-10 present the average dynamic power dissipation reduction percentages of the proposed method when 10%, 20%, ..., 90% of the scan cells are selected for gating respectively. In all experiments 0.2% of the internal lines of the combinational circuit were considered. Results indicate that the effectiveness of the proposed method is

Table II. Comparing the Proposed Method with the one in ElShoukry et al. [2007]

Circuit	50% Gated Scan Cells		80% Gated Scan Cells	
	[ElShoukry et al. 2007]	Proposed	[ElShoukry et al. 2007]	Proposed
s5378	67.5%	77.9%	87.8%	95.7%
s9234	61.8%	82.5%	89.7%	95.7%
s13207	65.2%	86.3%	89.4%	98.8%
s15850	62.6%	82.5%	80.5%	96.7%
s38417	56.3%	84.7%	85.4%	97.5%
s38584	53.1%	84.4%	83.8%	96.0%
Average	61.1%	83.1%	86.1%	96.7%

very high. For example, when half (50%) of the scan cell outputs are selected for gating, the proposed method achieves an 83% overall average reduction of the dynamic power dissipated by the combinational part of the CUT.

We now compare the proposed method against the method presented in ElShoukry et al. [2007] which also targets average power dissipation reduction using partial gating. ElShoukry et al. [2007] first present a cost function that, for a specific Gating-Vector, is equal to the number of gates guaranteed not to switch during scan in/out weighted by their fanouts. The cost function is a measure of the effectiveness of the specific Gating-Vector regarding power dissipation. Then, a search procedure is proposed that evaluates the cost functions of a pre-specified number of randomly generated Gating-Vectors and selects among them the one with the best cost function. However, due to the random nature of this procedure, it cannot guarantee to find the best Gating-Vector even if the cost functions of a very large number of Gating-Vectors are taken into account. The proposed method, at first, identifies among the lines of the CUT, those that if held at a constant logic value during scan in/out will maximize the power reduction achieved. Then, a procedure is given that systematically derives the Gating-Vector that maximizes the number of representative lines that are held at a constant value during scan in/out thus maximizing the power reduction achieved.

We present, in Table II, average dynamic power reduction percentages for the two different percentages of gated scan cells that were reported in ElShoukry et al. [2007], specifically 50% and 80%. In all cases the proposed method outperforms the other method. On average, an additional 22% and 10% power saving is achieved for 50% and 80% gated scan cells respectively. It is obvious from Figure 6 that for smaller percentages of gated scan cells the proposed method results in even larger average power reductions compared to the randomly selected gated scan cells method and hence to the method presented in ElShoukry et al. [2007].

Finally, we examine the effectiveness of the proposed method on power-aware test sets. In particular, we present the average dynamic power reduction achieved by the proposed method when the unspecified (x) bits of the test set are filled a) with the zero logic value, b) with the one logic value and c) according to the adjacent fill technique proposed in Butler et al. [2004]. Table III presents the results for 50% and 80% gated scan cells. It is obvious that in all cases the reduction in average dynamic power dissipation is very high. Therefore, we

Table III. Average Dynamic Power Reduction for Power-Aware Test Sets

Circuit	50% Gated Scan Cells			80% Gated Scan Cells		
	Zero Fill	One Fill	Adjacent Fill	Zero Fill	One Fill	Adjacent Fill
s5378	75.0%	75.0%	75.4%	94.8%	93.3%	93.8%
s9234	81.1%	83.0%	82.5%	94.2%	95.0%	94.6%
s13207	81.4%	82.5%	80.2%	96.6%	97.3%	97.0%
s15850	75.5%	80.3%	77.4%	94.3%	95.8%	94.8%
s38417	82.1%	83.9%	82.6%	96.5%	97.1%	96.6%
s38584	81.0%	83.0%	81.0%	95.0%	95.6%	95.0%
Average	79.4%	81.3%	79.8%	95.2%	95.7%	95.3%

conclude that the proposed method is very efficient independently of the test set used.

#### 4. CONCLUSION

A transition at a line of a circuit may cause a significantly larger number of transitions than a transition at another line of the circuit. Such lines are called representative in this article. A method for identifying the representative lines of a circuit and assigning a weight to each one of them, indicative of the switching activity in its fanout cone, was given. Based on the representative lines, a method for selecting and gating the most suitable, under area and performance design constraints, subset of scan cells for average dynamic power dissipation reduction during scan-based testing was proposed. It was experimentally shown that the dynamic power reduction achieved by applying this method is significantly larger than that achieved by the corresponding already known methods.

The proposed method does not insert logic on critical paths, hence it does not introduce any circuit performance degradation while a trade-off can be made between the power dissipation reduction and the area overhead, which is proportional to the number of scan cells selected for gating. Furthermore, it can be combined with methods based on Xs filling for reducing further either the average power dissipation during shift or the peak power dissipation during capture cycles (e.g., Remersaro et al. [2006]). Finally, it can be also combined with various test data compression methods that utilize the Xs, such as Kavousianos et al. [2007].

#### REFERENCES

- BARDELL, P. H., McANNEY, W. H., AND SAVIR, J. 1987. *Built-In Test for VLSI: Pseudorandom Techniques*. John Wiley and Sons, 193–202.
- BHUNIA, S., MAHMOUDI, H., GHOSH, D., MUKHOPADHYAY, S., AND ROY, K. 2005. Low-power scan design using first-level supply gating. *IEEE Trans. VLSI Syst.* 13, 3, 384–395.
- BONHOMME, Y., GIRARD, P., LANDRAULT, C., AND PRAVOSSOUDOVITCH, S. 2002. Power driven chaining of flip-flops in scan architectures. In *Proceedings of International Test Conference*, 796–803.
- BONHOMME, Y., GIRARD, P., GUILLER, L., LANDRAULT, C., PRAVOSSOUDOVITCH, S., AND VIRAZEL, A. 2006. A gated clock scheme for low power testing of logic cores. *J. Electron. Test.: Theor. Appl.* 22, 89–99.
- BUTLER, K., SAXENA, J., FRYARS, T., HETHERINGTON, G., JAIN, A., AND LEWIS, J. 2004. Minimizing power consumption in scan testing: Pattern generation and DFT Techniques. In *Proceedings of the International Test Conference*, 355–364.
- DABHOLKAR, V. AND CHAKRAVARTY, S. 1994. Two techniques for minimizing power dissipation in scan circuits during test application. In *Proceedings of IEEE Asian Test Symposium*. 324–329.

- DABHOLKAR, V., CHAKRAVARTY, S., POMERANZ, I., AND REDDY, S. 1998. Techniques for minimizing power dissipation in scan and combinational circuits during test application. *IEEE Trans. Comput. Aid.-Des. Integr. Circ. Syst.* 17, 12, 1325–1333.
- ELSHOUKRY, M., TEHRANIPOOR, M., AND RAVIKUMAR, C. 2007. A critical-path-aware partial gating approach for test power reduction. *ACM Trans. Des. Automat. Electron. Syst.* 12, 2, 242–247.
- GERSTENDORFER, S. AND WUNDERLICH, H. 2000. Minimized power consumption for scan-based BIST. *J. Electron. Test.: Theor. Appl.* 16, 3, 203–212.
- GIRARD, P., LANDRAULT, C., PRAVOSSODOVITCH, S., AND SEVERAC, D. 1998. Reducing power consumption during test application by test vector ordering. In *Proceedings of the IEEE International Symposium on Circuits and System.* 296–299.
- GIRARD, P. 2002. Survey of low-power testing of VLSI Circuits, *IEEE Des. Test Comput.* 82–92.
- KAVOUSIANOS, X., KALLIGEROS, E., AND NIKOLOS, D. 2007. Multilevel Huffman Coding: An efficient test-data compression method for IP cores. *IEEE Trans. Comput. Aid.-Des. Integr. Circ. Syst.* 26, 6, 1070–1083.
- PARIMI, N. AND SUN, X. 2004. Toggle-masking for test-per-scan VLSI circuits. In *Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems.* 332–338.
- REMERASAR, S., LIN, X., ZHANG, Z., REDDY, S., POMERANZ, I., AND RAJSKI, J. 2006. Preferred fill: A scalable method to reduce capture power for scan based designs. In *Proceedings of the International Test Conference.* 32.2.1–32.2.10.
- SANKARALINGAM, R. AND TOUBA, N. 2002. Inserting test points to control peak power during scan testing. In *Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems.* 138–146.
- SAXENA, J., BUTLER, K., AND WHETSEL, L. 2001. An analysis of power reduction techniques in scan testing. In *Proceedings of the International Test Conference.* 670–677.
- SHARIFI, S., JAFFARI, J., HOSSEINABADY, M., AFZALI-KUSHA, A., AND NAVABI, Z. 2005. Simultaneous reduction of dynamic and static power in scan structures. In *Proceedings of the Design Automation and Test in Europe Conference.* 846–851.
- SINANOGLU, O. AND ORAILOGLU, A. 2002. Scan power reduction through test data transition frequency analysis. In *Proceedings of the International Test Conference.* 844–850.
- WANG, S. AND GUPTA, S. 2002. An automatic test pattern generator for minimizing switching activity during scan testing activity. *IEEE Trans. Comput. Aid.-Des. Integr. Circ. Syst.* 21, 8, 954–968.
- WHETSEL, L. 2000. Adapting scan architectures for low power operation. In *Proceedings of the International Test Conference.* 863–872.
- ZHANG, X. AND ROY, K. 2000. Power reduction in test-per-scan BIST. In *Proceedings of the 6th International OnLine Test Workshop.* 133–138.

Received July 2007; revised February 2008, November 2008; accepted December 2008