

Low Power BIST with Smoother and Scan-Chain Reorder

Nan-Cheng Lai, Sying-Jyan Wang and Yu-Hsuan Fu

*Institute of Computer Science
National Chung-Hsing University
Taichung 402, Taiwan, ROC
{lnc,sjwang,s9156023}@cs.nchu.edu.tw*

Abstract

In this paper, we propose a low-power testing methodology for the scan-based BIST. 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. These results are much better than those achieved by previous methods.

1. Introduction

Pre-designed intellectual property (IP) cores are commonly used in system-on-chip (SOC) designs. Because of the large number of IP cores and the limited pin count, the long test time required becomes a great challenge for SOC testing. In order to solve this problem, testing multiple cores concurrently was proposed for test time reduction. The built-in-self-test (BIST) is an attractive alternative to SOC testing. This technique can conduct at-speed testing, and it alleviates the burden of external testers. In pseudo-random BIST, the existence of random-pattern-resistant faults often requires an unacceptably long test sequence in order to attain high fault coverage for a CUT. Several techniques have been proposed to address this problem, including Markov Source BIST [1],[2] and reseeding [3].

Another serious problem associated with the pseudo-random BIST is the higher power consumption [4]. The power constraints defined under normal operations are usually much lower than the power consumed in the test mode [5]. Several techniques have been proposed, including test scheduling algorithms under power constraints [6], low-power built-in-self-test (BIST) [7],[8], and minimizing power during scan testing [9],[10].

In this paper, we present an approach to reduce the average power consumption during the scan-shift operations without degrading the normal-mode performance. The patterns generated by the pseudo-random pattern generator are smoothed by the smoother, and then the smoothed patterns are shifted into the scan chain. Since there are fewer bit-transitions in the smoothed patterns, the power consumed in the BIST process will be reduced.

The disadvantage of the smoothed patterns is that, given the same number of test vectors, the fault coverage is usually lower. A possible solution to the loss of fault coverage is to reorder scan cells. However, scan-chain reorder may introduce two serious problems. One is the timing closure and routing congestion due to long routing paths, and another is the difficulty to find an optimal scan cell order as the ordering problem is NP-hard (i.e., permutation) [9],[10]. The physical routing problem can be solved by using a cluster-based routing [10]. In order to reduce the computation time, we propose a group-based heuristic whose time complexity is $O(mCN)$, where m is the constraint, C is the number of test cubes, and N is the number of scan cells. Experimental results show that the final fault coverage is comparable to that achieved by the LFSR or Markov source BIST with phase 1 parameter [2]. Moreover, for larger circuits, the lengths of the scan chains are even shorter than those obtained by the commercial tool "Silicon Ensemble".

2. Preliminaries

Since the amount of leakage current is usually small, the major source of power dissipation in CMOS technology is the dynamic power dissipation. The dynamic power dissipation appears only when a CMOS gate switches from one stable state to another, and the component due to charging and discharging of load capacitance is usually the dominant factor in dynamic power dissipation. Thus the dynamic power consumption is:

$$P_d = \frac{1}{2} \times V_{DD}^2 \times f_p \times \sum_{i=1}^N E[T_i] \times C_i, \quad (1)$$

where V_{DD} is the power supply voltage, f_p is the clock

frequency, N is the number of CMOS gates, $E[T_i]$ is the expected number of transitions per cycle in gate i , and C_i is the load capacitance of gate i . We can estimate the total power consumption by using the weighted switching activity (WSA) [7],[8], which is a simplified form of Eq. (1). The weighted switching activity in each gate is given by the number of signal transitions times one plus the number of fan-outs at the gate, as defined in Eq. (2).

$$\text{WSA} = \# \text{transitions} \times (1 + \# \text{fan-outs}) \quad (2)$$

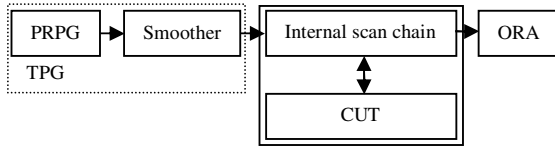


Figure 1. Test-per-scan architecture with smoother.

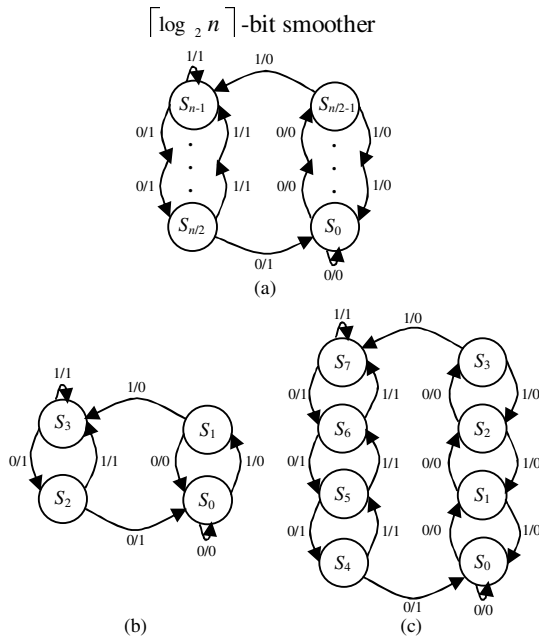


Figure 2. (a) The general smoother, (b) 2-bit smoother and (c) 3-bit smoother.

3. Test Architecture

3.1 Test Pattern Generator

The Linear Feedback Shift Register (LFSR) is often used as both a PRPG and an output response analyzer (ORA). In the proposed BIST approach, we modify the test pattern generator (TPG) by adding a smoother. The scan-based BIST architecture is shown in Figure 1.

The pseudo-random patterns generated by the PRPG are fed to a smoother, which is used to filter out excessive signal transitions. If we apply the smoothed patterns into the internal scan chain, the switching activities will be dramatically reduced during scan testing. Since the power consumption is reduced, more IP cores can be tested concurrently to reduce the test

time under given power constraints.

3.2 Smoother

The general finite state machine (FSM) model for the smoother, as shown in Fig 2(a), is similar to a predictor [11] used for branch prediction. The state transition graph (STG) G of the FSM is a directed graph $G = (V, E)$, in which a vertex in V represents a state while an edge in E represents a state transition. The input and output associated with a state transition is given on the corresponding edge. For an n -state smoother, one half of the n states are responsible for output 1, and the other states are responsible for output 0. From the STG, we can see that the smoother transforms one bit sequence into another with fewer number of transitions. The STGs for the 2-bit ($n=4$) and 3-bit ($n=8$) smoothers are shown in Fig 2(b),(c), respectively.

3.3 Transition Probability in Smoothed Patterns

The probability of '0' and '1' are equal in a maximal-length sequence produced by an LFSR [12]. Hence, the probability of a bit-transition is 0.5 in each clock. The probability of signal transition in a smoothed sequence can be analyzed through the corresponding STG. The STG can be represented by a Markov chain model, which is used to calculate the state probabilities.

The *local state transition probability* associated with a state transition from S_i to S_j , denoted as lp_{ij} , is the probability that a primary input will bring the machine to state S_j given that the current state is S_i . If the distribution of input vectors fed to the circuit is known, each lp_{ij} can be calculated accordingly. Otherwise, uniform input pattern (i.e., each legal input vector appears with the same frequency) is assumed. The *local state transition probability matrix* is defined as $LP = \{lp_{ij}\}^{m \times m}$, where m is the number of states in the FSM. When an FSM is in the steady-state, the probability that the FSM stays at S_i is denoted as sp_i . The *steady-state probability* is denoted as $SP = \{sp_i\}^{1 \times m}$. Since in the steady-state we may assume that sp_i does not change with time, the following equations hold:

$$SP = SP \times LP \text{ or } (LP - I)^T \times SP^T = 0, \quad (3)$$

where I is the identify matrix. The above equation has only $m-1$ linearly independent equations. In order to obtain SP , one additional equation is required:

$$\sum_{i=1}^m sp_i = 1 \quad (4)$$

The probability for state transition from S_i to S_j , denoted as tp_{ij} , is just $sp_i \times lp_{ij}$. The *state transition probability matrix* is defined as $TP = \{tp_{ij}\}^{m \times m}$. For example, the derivation of state transitions probability for the 2-bit smoother is shown in Figure 3. First, the *local state transition probability matrix* for the 2-bit smoother is shown in Figure 3(a). The probability of

$$OS_{cluster} = \frac{s}{c^2}, 2 \leq OS_{cluster} \leq 3, \quad (6)$$

where s is the number of scan cells in CUT, and c is a positive even integer. Because the routing length excluding $OS_{cluster}$ will be lengthier. By using Eq. (6), we can determine the optimal number of clusters (c^2) in the beginning of a design flow. Once the number of clusters is determined, the positions of scan cells within each cluster are known after cell placement. The initial scan-chain is constructed according to the cluster order illustrated in Figure 5, while scan cells in a cluster are connected in an order that minimizes the routing length within the cluster.

We propose a group-based greedy algorithm that can find an optimized scan cell order efficiently under given routing constraints. The basic idea is to find a scan-chain order so that the bit-transitions in test cubes will be minimized. Since there are fewer bit-transitions in the smoothed patterns, the probability that the test cubes corresponding to the reordered scan-chain are covered by the smoothed patterns should be higher.

The following metrics will be used in the proposed algorithm.

Definition 1: The compatible number (CN) is the number of test cubes that match the given set of smoothed vectors.

For example, let the set of test cubes be $\{(1, x, 0), (x, 0, 1), (0, 1, x)\}$ and the set of smoother vectors be $\{(1, 1, 0), (0, 0, 1), (1, 1, 1)\}$. For this example, the compatible number is 2 since the first two test cubes match the first two smoothed vectors.

Definition 2: The distance between two vectors V_1 and V_2 , which is denoted as $D(V_1, V_2)$, is the number of positions in which there are conflicts between the two vectors.

For example, let $V_1=(1, 0, x, x)$ and $V_2=(x, 1, 0, 1)$. The distance $D(V_1, V_2)$ is 1 since they are in conflict in the second position.

The inputs of the proposed algorithm include a set of test cubes (TC) that are generated from an ATPG program, and a set of smoothed vectors (SV) that are outputs of the smoother. The initial scan-chain order and clusters are known. An example of the proposed algorithm is illustrated in Figure 6, in which a scan-chain with 12 scan cells is partitioned into two clusters. A set of 4 test cubes is illustrated in Figure 6(a), and a set of smoothed vectors is shown in Figure 6(b). Our goal is to find a permutation of the column vectors in Figure 6(a) for each cluster such that a maximum number of test cubes can match the given set of smoothed vectors. In the following discussion, we shall refer to the column vector corresponding to the j -th scan cell in cluster i as TC_{ij} . The k -th row in TC_{ij} will be referred to as $TC_{ij}[k]$.

In our group-based heuristic, when the given group-size is GS , we will arrange GS consecutive columns as a group so that the bit-transitions between

adjacent columns (i.e., the distance between them) are minimized. We use an auxiliary data structure, which is referred to as the *predictor vector* (PV), for the scan-chain reorder. The predictor vector is a column vector whose length is equal to the number of test cubes in TC. In other words, the length of PV is the same as each TC_{ij} . Assume that we start to process a group in cluster i . The predictor vector PV is initially set to be an all-x vector (i.e., each position is an 'x'). Among all the scan cells in cluster i whose positions in the scan chain have not yet been determined, we try to find a column vector TC_{ij} whose distance to PV is minimum. Once such a column j is selected, the new PV is updated as follow. Assume that $TC_{ij}[k]$ is a specified bit (i.e., $TC_{ij}[k] \neq x$), we make the assignment $PV[k] \leftarrow TC_{ij}[k]$; otherwise, the position is left unchanged. This process is repeated until a group of GS consecutive scan cells have been determined, and we start with the next group again. Note that, a group may span two clusters.

Cluster 1						Cluster 2					
SC	SC	SC	SC	SC	SC	SC	SC	SC	SC	SC	SC
1	2	3	4	5	6	7	8	9	10	11	12
1	x	0	1	x	1	x	x	0	0	x	1
0	x	x	x	1	0	x	x	1	0	x	0
x	x	1	x	0	x	x	x	1	1	x	x
x	1	0	0	x	1	x	x	0	x	x	x
(a)						1	1	0	0	0	0
1	1	1	0	0	0	1	1	1	1	1	1
1	1	0	0	0	1	0	0	0	0	0	1
1	1	1	1	1	0	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	0	0	0
(b)						SC	SC	SC	SC	SC	SC
1	2	6	4	5	3	7	8	9	11	10	12
1	x	1	1	x	0	x	x	0	x	0	1
0	x	0	x	1	x	x	x	1	x	0	0
x	x	x	x	0	1	x	x	1	x	1	x
x	1	1	0	x	0	x	x	0	x	x	x
PV[1]	1	1	1	1	0	x	x	0	x	0	1
PV[2]	0	0	0	x	1	x	x	1	x	0	0
PV[3]	x	x	x	x	0	1	x	x	1	x	1
PV[4]	x	1	1	0	0	0	x	x	0	x	x
(c)											

Figure 6. An example of the group-based algorithm under various routing constraints : (a) test cubes, (b) smoothed vectors, (c) new scan cell order with group size 3.

The execution procedures for group size $GS=3$ are outlined in Figure 6(c). We will use Figure 6(c) as an example.

- (1) Initially, $PV = (x, x, x, x)^T$. Since its distance to any column vector is 0, we simply select the first available column, namely TC_{11} . Thus, the first scan cell in cluster 1 is SC1, and PV is updated to $(1, 0, x, x)^T$.
- (2) There are three columns whose distances to PV are minimum: $D(TC_{12}, PV) = D(TC_{14}, PV) = D(TC_{16}, PV) = 0$. We select column TC_{12} since it is the first column encountered. Therefore, we put scan cell SC2 next to SC1 and PV is updated to $(1, 0, x, 1)^T$.
- (3) Now, the distance from TC_{16} to PV is 0, and it is the minimum among all candidates. So the next scan cell in the scan-chain is SC6 and PV is updated to $(1, 0, x, 1)^T$. Note that, up to this point, this group has three cells.

(4) Next, $D(TC_{14}, PV)=1$ is the minimum, thus it is the starting column in the next group. As we start the next group, PV is initialized to $(x, x, x, x)^T$ again. This process goes on until all scan cells have been processed.

The proposed algorithm for a given group size is given as follows. Let CL_i be the set of scan cells in cluster i whose position in the scan-chain have not been decided.

Algorithm GBR(GS, TC, SV): Group-Based Reordering

Input: GS : group size;
 TC : set of test cubes;
 SV : set of smoothed vectors;
Output: Optimized scan-chain and the cost of the scan-chain;

```

 $PV \leftarrow (x, x, \dots, x)^T$ ;
 $g \leftarrow 0$ ; // count the number of scan cells in a group
for ( $i = 1$ ;  $i \leq c^2$ ;  $i++$ ) {
  //  $c^2$  is the number of clusters in the design
  while ( $CL_i \neq \emptyset$ ) { // check whether cluster  $i$  is empty
    find  $j$  such that  $D(TC_{ij}, PV) = \min_{SCK \in CL_i} D(TC_{ik}, PV)$ ;
    append scan cell  $j$  to the end of scan-chain;
    remove scan cell  $j$  from  $CL_i$ ;
     $g \leftarrow g + 1$ ;
    if ( $g > GS$ ) {
       $PV \leftarrow (x, x, \dots, x)^T$ ;
       $g \leftarrow 0$ ;
    }
    for ( $k=1$ ;  $k \leq |TC|$ ;  $k++$ )
      if ( $TC_{ij}[k] \neq 'x'$ )  $PV[k] \leftarrow TC_{ij}[k]$ ;
  }
}
 $new\_cost \leftarrow$  compatible number ( $CN$ ) by using the new scan-chain order;
return  $new\_cost$  and  $new\_scan\_chain\_order$ ;

```

The complexity of this algorithm can be analyzed as follows. Let the total number of scan cells be N , the number of test cubes be C , and the number of scan cells per cluster be s . Each scan cell is processed exactly once, and during the process the associated PV is compared with all the remaining column vectors in the same cluster. In order to calculate the distance between two column vectors, we need to compare C positions. Therefore, the overall complexity is $O(s \times C \times N)$. As we discuss in the beginning of this section, the optimal cluster partition corresponds to $2 \leq s \leq 3$. In other words, s can be neglected in the optimal case, and the complexity is $O(CN)$.

For various circuits under test, the optimal group size GS may be different. In general, we need to try different group sizes to find out the best scan-chain order. The complete heuristic procedure to search for the best scan-chain order with a given group constraint m is outlined as follows.

Procedure: Find the best scan-chain order under group constraint m

```

 $cost \leftarrow 0$ ;  $i \leftarrow 1$ ;
for ( $i=1$ ;  $i \leq m$ ;  $i++$ ) {
  call  $GBR(i, TC, SV)$ ;
  if ( $new\_cost > cost$ ) {
     $candidate\_group \leftarrow i$ ;
     $scan\_chain\_order \leftarrow new\_scan\_chain\_order$ ;
     $cost \leftarrow new\_cost$ ;
  }
}

```

Since the reordering algorithm will be executed m times, the overall complexity will be $O(mCN)$.

Table I . Statistics of ISCAS'89 benchmarks.

Circuit	#scan cells	LFSR	TL (test length)
S5378	214	60	65536
S9234	247	78	72848
			131072
S13207	700	80	40677
S15850	611	110	37767
S38417	1664	124	81984
S38584	1464	140	82055

Table II . Summary of average cells per cluster.

Circuit	Cluster	Cells/ cluster	WL (S.E.) (mm)	WL 2-bit smoother (mm)	% (red)	WL 3-bit smoother (mm)	% (red)
S5378	100	2.14	14.57	15.78	-7.67	15.77	-7.61
S9234	100	2.47	22.68	23.21	-2.28	23.75	-4.51
S13207	256	2.73	45.22	44.74	1.06	45.49	-0.59
S15850	256	2.39	53.4	51.97	2.68	50.98	4.53
S38417	576	2.89	136.94	123.3	9.96	125.72	8.19
S38584	576	2.54	187.89	177.82	5.36	178.14	5.19

Table III . Comparison with similar approach [2].

circuit	Test length	TE% [2]	TE% of LFSR	2-bit smoother		3-bit smoother	
				TE% w/o reorder	TE% w/ reorder	TE% w/o reorder	TE% w/ reorder
S9234	72848	98.517	95.02	93.46	95.47	83.39	92
S13207	40677	97.879	98.9	92.73	97.84	83.64	87.72
S15850	37767	98.306	96.41	93.47	97.69	90.62	92.79
S38417	81984	95.422	97.2	94.79	95.96	93.5	94.47
S38584	82055	98.359	99.76	95.29	99.23	90.35	95.82
Average	-	97.697	97.46	93.95	97.24	88.3	92.56

5. Experimental results

We have conducted experiments on some ISCAS'89 benchmarks shown in Table I . The scan chains routing was implemented with the TSMC 0.25 μ m technology. The lengths of scan chains and LFSRs are shown in the second and third columns of Table I . An LFSR's length is usually larger than the maximum number of specified bits in all test cubes plus 20 for the ease of reseeding [13]. In order to compare the proposed method with previous results, we use the same test lengths used in [2],[7]. Furthermore, the group constraint (m) used in the experiment is always 10, and the power consumption is estimated by the weighted switch activities. The industrial routing tool "Silicon Ensemble" is employed to estimate the effect of wire length. The average cells per cluster defined in Eq. (6) are summarized from the above experimental results, as shown in Table II . The wire lengths (WL) under the

Table IV. Comparison with similar approach [7].

circuit	Test length	FC% of LFSR	2-bit smoother			3-bit smoother			FC% [7]	AP% reduction [7]
			FC% w/o reorder	FC% w/ reorder	AP% reduction	FC% w/o reorder	FC% w/ reorder	AP% reduction		
S5378	65536	98.96	97.91	98.56	58.51	90.26	95.5	84.75	97.08	44
S9234	131072	89.67	87.91	91.4	59.37	77.49	86.78	85.53	91.63	55.7
S13207	40677	97.42	91.25	96.37	62.67	82.17	86.24	87.39	-	-
S15850	37767	93.06	90.12	94.43	58.21	87.27	89.44	84.96	-	-
S38417	81984	96.67	94.26	95.42	60.73	92.97	93.93	84.93	-	-
S38584	82055	95.7	91.05	95	60.87	86.14	91.58	84.81	-	-
Average	-	95.25	92.08	95.2	60.06	86.05	90.58	85.4	94.36	49.85

Table V. Hardware overhead of the 2-bit and 3-bit smoother.

2-bit smoother	3-bit smoother	LT-RTPG[7] with $k = 3$
25.5	47.5	10.5

optimal cluster are better than the ones optimized by “*Silicon Ensemble*” shown in column 5 for S13207, S15850, S38417 and S38584. Therefore, the approach is applicable to larger circuits with many scan cells. The comparison with other similar approaches is shown in Table III and Table IV. These tables show that the fault coverage (FC), test efficiency (TE), and average power (AP) reduction estimated by Eq. (1). In Table III, the first column gives the circuit names, and the second column shows the test length (i.e., number of test vectors). The TEs of the Markov source BIST with phase 1 parameter [2] and LFSR are shown in the third and forth columns, respectively. Columns five and six show the test efficiency with the scan cell order optimized by “*Silicon Ensemble*” and the scan cell order with the 2-bit smoother under optimized cluster, respectively. In addition, the test efficiency of scan cell order with the 3-bit smoother is shown in the seventh and eighth columns as well. The average test efficiency with the 2-bit smoother shown in the column six is similar to the average test efficiency of Markov source BIST and LFSR. With the 3-bit smoother, the test efficiency is lower as shown in the last column, because the smoothed vectors have much less transitions (i.e., with many more aliases). In Table IV, the first column shows the circuit names, and the second column shows the test lengths. The fault coverage, rather than test efficiency of the LFSR, is shown in the third column in order to be compared with the approach in [7]. The fault coverage of the scan cell order optimized by “*Silicon Ensemble*” and the scan cell order with the 2-bit smoother under optimized cluster is shown in the forth and fifth columns, respectively. The average power consumption is shown in the next column. In the same way as described above, the experimental results with the 3-bit smoother are shown in the seventh to ninth columns. The fault coverage and the reduction of average power consumption of the LT-RTPG with $k = 3$ [7] are shown in last two columns. The fault coverage and average power consumption shown in the fifth and sixth columns are better than the ones [7] shown in the last two columns. Assume that D flip-flop is equivalent to eight 2-input NAND gates and an inverter. The size of an n -input NAND (NOR) gate is $0.5n$, an inverter is 0.5, and an n -input XOR gate is

$2.5(n-1)$ [1]. As shown in Table V, the hardware overhead for our approach is only slightly larger than LT-RTPG [7].

6. Reference

- [1] N.Z. Basturkmen, S.M. Reddy and I. Pomeranz, “Pseudo random patterns using Markov sources for scan BIST,” in *Proc. Int. Test Conf.*, pp. 1013-1021, Oct. 2002.
- [2] I. Polian, B. Becker and S.M. Reddy, “Evolutionary optimization of Markov sources for pseudo random scan BIST,” in *Proc. of European Design and Test Conf.*, pp. 1184 – 1185, 2003.
- [3] E. Kalligeros, X. Kavousianos, D. Bakalis, and D. Nikolos, “A New reseeding technique for LFSR-based test pattern generation,” in *Proc. Int On-Line Testing Workshop*, pp. 80-86, 2001.
- [4] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, “Test power: a big issue in large SOC designs,” in *Proc. IEEE Int. Symp. On Electronic Design, Test and Applications*, pp. 447-449, Jan. 2002.
- [5] Y. Zorian, “A distributed BIST control scheme for complex VLSI devices,” in *Proc. VLSI Test Symp.*, pp. 4-9, Apr. 1993.
- [6] R.M. Chou, K.K. Saluja, and V.D. Agrawal, “Scheduling tests for VLSI systems under power constraints,” *IEEE Trans. on VLSI Systems*, vol. 5, no. 2, pp. 175-185, Jun. 1997.
- [7] W. Seongmoon and S.K. Gupta, “LT-RTPG: a new test-per-scan BIST TPG for low heat dissipation,” in *Proc. Int. Test Conf.*, pp. 85-94, Oct. 1999.
- [8] N.Z. Basturkmen, S.M. Reddy, and I. Pomeranz, “A low power pseudo-random BIST technique,” in *Proc. Computer Design: VLSI in Computers and Processors*, pp. 468-473, Sept. 2002.
- [9] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, “Power driven chaining of flip-flops in scan architectures,” in *Proc. Int. Test Conf.*, pp. 796-803, Oct. 2002.
- [10] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, “Efficient scan chain design for power minimization during scan testing under routing constraint,” in *Proc. Int. Test Conf.*, pp. 488-493, Oct. 2003.
- [11] J.K.L. Lee and A.J. Smith, “Branch prediction strategies and branch-target buffer design,” *IEEE Computer*, Vol. 17, No. 1, Jan. 1984.
- [12] M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing & Testable Design*, Wiley-IEEE Press, Jan. 1993.
- [13] B. Könenmann, “LFSR-coded test patterns for scan designs,” in *Proc. European Test Conf.*, pp. 237-242, 1991.