Scan-Cell Reordering for Minimizing Scan-Shift Power Based on Nonspecified Test Cubes

YU-ZE WU and MANGO C.-T. CHAO
National Chiao Tung University

This article presents several scan-cell reordering techniques to reduce the signal transitions during the test mode while preserving the don't-care bits in the test patterns for a later optimization. Combined with a pattern-filling technique, the proposed scan-cell reordering techniques can utilize both high response correlations and pattern correlations to simultaneously minimize scan-out and scan-in transitions. Those scan-shift transitions can be further reduced by selectively using the inverse connections between scan cells. In addition, the trade-off between routing overhead and power consumption can also be controlled by the proposed scan-cell reordering techniques. A series of experiments are conducted to demonstrate the effectiveness of each of the proposed techniques individually.

Categories and Subject Descriptors: B.7.3 [**Integrated Circuits**]: Reliability and Testing—*Testability*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Scan testing, DFT, low-power testing

ACM Reference Format:
Wu, Y.-Z. and Chao, M. C.-T. 2010. Scan-cell reordering for minimizing scan-shift power based on nonspecified test cubes. ACM Trans. Des. Autom. Electron. Syst. 16, 1, Article 10 (November 2010), 29 pages. DOI = 10.1145/1870109.1870119.
http://doi.acm.org/10.1145/1870109.1870119.

1. INTRODUCTION

By enhancing circuit's controllability and observability, scan design has been a widely used DFT technique to achieve high fault coverage for a complex circuit [Bushnell et al. 2000]. However, with the scan design, the Circuit-Under-Test (*CUT*) consumes much more power in its test mode than that in its functional mode Zorian [1993] due to the following reasons. First, when

Authors' address: Y.-Z. Wu and M. C.-T. Chao (corresponding author), Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan; email:
mango@faculty.nctu.edu.tw.
Permission to make digital or hard copies 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 the Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2010 ACM 1084-4309/2010/11-ART10 $10.00 DOI: 10.1145/1870109.1870119.
http://doi.acm.org/10.1145/1870109.1870119.

using the scan design to shift in test patterns and shift out test responses, a large number of signal transitions may occur along the scan paths, which induce even more signal transitions on the CUT and hence consume higher power. Also, the clock-gating logics, which has been a popular design technique to reduce the power consumption by selectively updating only part of the flip-flops, are forced to turn off during the scan-shift cycles. Therefore, all the flip-flops are updated simultaneously in the test mode, which leads to higher power consumption as well.

This excessive power consumption during the scan-based testing may result in physical damage or reliability degradation to the CUT, and in turn decreases the yield and product lifetime [Girard 2002]. As the number of scan cells keeps on growing in modern designs, this increasing power consumption has become one of the biggest barriers to effective scan-based testing.

A common practice to lower the power consumption during scan-based testing is to reduce the number of scan cell's signal transitions, which can be classified into the following three types: (1) capture transitions, generated by the same scan cell's value difference between the scan-in pattern and the corresponding captured response; (2) scan-out transitions, generated by two adjacent scan cells' value difference between their scan-out response; and (3) scan-in transitions, generated by two adjacent scan cells' value difference between the scan-in patterns. The first transition type is associated with the capture power and the last two types are associated with the scan-shift power.

In order to reduce the capture transitions, specialized ATPG techniques [Chandra et al. 2008; Sankaralingam et al. 2002; Remersaro et al. 2006; Wen et al. 2005] are proposed to generate test-pattern vectors which have a minimal hamming distance with their corresponding test-response vectors. Because the don't-care bits in their test cubes are fully specified for minimizing the capture transitions, the preceding ATPGs preclude the possibility for further test compaction or compression, and hence may result in a larger test set.

Methods are proposed to utilize the don't-care bits to minimize the scan-in transitions for a given test set [Mrugalski et al. 2007; Li et al. 2008; Lin et al. 2006; Sankaralingam et al. 2000; Sinanoglu et al. 2003]. Sankaralingam et al. [2000] proposed a don't-care-filling technique, named MT-fill, guaranteeing that the scan-in transitions generated by its filled patterns are minimum for the given test set and scan-cell ordering. The methods in Sankaralingam et al. [2000], Mrugalski et al. [2007], Lin et al. [2006] reduced the test power as well as the test data volume based on built-in decompression hardware. Sinanoglu et al. [2003] added Xor gates or inverters along the scan paths to minimize the scan-in transitions. Li et al. [2008] proposed a don't-care-filling technique which can simultaneously reduce the scan-in, scan-out, and capture transitions.

Another technique to reduce the scan-shift power is to partition the scan cells into multiple groups and activate only one group at a time during the scan-shift cycles [Bonhomme et al. 2001; Huang et al. 2001; Rosinger et al. 2004; Sankaralingam et al. 2001; Saxena et al. 2001; Whetsel 2000]. It can limit the concurrent transitions in a small portion of the CUT. The partition

methods require special control architectures to the scan designs, such as gated clocks [Bonhomme et al. 2001], central control unit for each group's clock signal [Rosinger et al. 2004; Whetsel 2000], or specialized scan cells along with multiphase generator [Huang et al. 2001]. Sankaralingam et al. [2001] further minimize the capture power by only capturing responses for certain selected groups of scan cells. It requires a customized ATPG and discards a significant portion of responses.

Methods in Bonhomme et al. [2002], Dabholkar et al. [1998], Sinanoglu et al. [1998] change the order of scan cells along the scan paths to minimize both scan-in and scan-out transitions based on given test patterns and responses. This scan-cell-reordering technique saves the scan-shift power, but sacrifices the opportunity of optimizing the wire length of scan paths during the APR stage [Makar 1998; Hirech et al. 1998]. Methods in Bonhomme et al. [2003, 2004] further consider the routing overhead during the reordering process such that the imposed routing overhead can be limited. However, one serious disadvantage in the scan-chain-reordering techniques [Bonhomme et al. 2002, 2003, 2004; Dabholkar et al. 1998] is that the exact test patterns and responses need to be obtained in advance. As the result, no don't-care bits can be utilized for a further reduction to scan-in transitions or test data volume, such as Sankaralingam et al. [2000], Mrugalski et al. [2007], Lin et al. [2006], Sinanoglu et al. [2003], Li et al. [2008]. Sinanoglu et al. [1998] can reorder the scan cells based on the test set with don't-care bits. However, Sinanoglu et al. [1998] relies on the controllability measures to approximate the response correlation between scan cells, which may not be able to reflect the reality. Also, Sinanoglu et al. [1998] did not consider the impact of any don't-care-filling technique.

In this article, we attempt to develop a scan-cell-reordering scheme which can minimize the scan-out transitions while preserving the don't-care bits in the test cubes for a later optimization of scan-in transitions using MT-fill [Sankaralingam et al. 2000]. To achieve this goal, we first need to predict the correlation between the response values before specifying don't-care bits. This response correlation is an index to the possible scan-out transitions between scan cells and can be used as a guidance to the reordering process (Section 4). Second, we consider the impact of scan-cell reordering on the result of MT-fill and simultaneously optimize the scan-in and scan-out transitions (Section 6). Next, we selectively inverse some connections between scan cells such that a low response correlation (or pattern correlation) between two scan cells can be turned into a high correlation, which in turn reduces the probability that scan-shift transitions occur along the scan paths (Section 7). Last, we consider the routing overhead of scan paths during the scan-cell reordering process, and thus the trade-off between scan-shift power and routing overhead can be properly controlled (Section 8). In addition, we propose a pattern reordering scheme to minimize the signal transitions resulted from the value difference between the first bit of a scan-in pattern and the last bit of its previous scan-out response after the scan-cell reordering scheme is applied (Section 5). All the proposed methods are validated through large ISCAS and ITC benchmark circuits.

2. MOTIVATION

During the scan-based testing, the total power consumption of the CUT is highly correlated with the total number of signal transitions on the scan cells [Sankaralingam et al. 2000]. In this article, we use the number of signal transitions occurring on scan cells to represent the power of the whole CUT. The proposed scan-cell-reordering scheme focuses on reducing the total scan-shift power, that is, reducing the total scan-shift transitions. The capture power is not considered in the proposed scheme since the number of capture transitions generated for a test pattern depends only on the filling of the test pattern. Changing the scan-cell ordering does not change the hamming distance between the test-pattern vector and its corresponding test-response vector.

From the discussions in Section 1, the scan-in transitions can be minimized by properly filling the don't-care bits of a test set once the scan-cell order in the scan paths is given [Sankaralingam et al. 2000]. This reduction could be more significant as the percentage of don't-care bits increases. Therefore, our scan-cell reordering scheme attempts to first minimize the scan-out transition count without specifying the don't-care bits, leaving the don't-care bits for a later minimization of scan-in transition, such as MT-fill [Sankaralingam et al. 2000]. However, before specifying the don't-care bits, the value of some responses cannot be known, implying that no explicit information for estimating the possible number of scan-out transitions can be used during the scan-cell reordering process.

We first use a simple experiment (reported in Table I) to show that certain pairs of scan cells tend to have the same response value in most cases of the random don't-care filling. Thus, even without knowing the exact test responses, the reordering scheme can still avoid the possible scan-out transitions by connecting those correlated pairs of scan cells next to each other. We first define this tendency between two scan cells as the *response correlation*, which is the probability that the two scan cells have the same response value by a random fill of don't-care bits. Please note that the similar concept of response correlation has already been used in previous works [Bonhomme et al. 2002; Chen et al. 2003; Dabholkar et al. 1998; Sinanoglu et al. 1998], but the method and assumptions for obtaining this response correlation are different from work to work.

In the experiment, we use a commercial tool [Synopsys 2010] to generate stuck-at-fault patterns with don't-care bits. By randomly filling the don't-care bits and simulating the corresponding responses for 1-million times, the statistic of the response correlation between any two scan cells can then be collected. Table I lists the range of response correlations (Columns 1 and 4), the number of scan-cell pairs whose sampled response correlation falls in the range (Columns 2 and 5), and its corresponding percentage to the total scan-cell pairs (Columns 3 and 6), for the largest ISCAS benchmark circuit s38584. The don't-care bit percentage of this test set is 78.01%. As the results show, while majority of the scan-cell pairs have a response correlation around 0.5, still 21595 scan-cell pairs (2%) have a response correlation higher than 0.75. Those 21595 scan-cell pairs could form a fair-sized solution space when reordering

Table I. Response Correlation of ISCAS Benchmark s38584

Correlation	# of cell pairs	Distribution (%)	Correlation	# of cell pairs	Distribution (%)
0.95 - 1	32	0.003	0.45 - 0.50	476,539	45.220
0.90 - 0.95	758	0.072	0.40 - 0.45	34,963	3.319
0.85 - 0.90	2,549	0.242	0.35 - 0.40	12,957	1.230
0.80 - 0.85	6,531	0.620	0.30 - 0.35	9,260	0.879
0.75 - 0.80	11,725	1.113	0.25 - 0.30	6,910	0.656
0.70 - 0.75	17,097	1.623	0.20 - 0.25	5,109	0.485
0.65 - 0.70	17,518	1.663	0.15 - 0.20	3,666	0.348
0.60 - 0.65	21,848	2.074	0.10 - 0.15	1,949	0.185
0.55 - 0.60	46,804	4.443	0.05 - 0.10	748	0.071
0.50 - 0.55	376,600	35.750	0 - 0.05	0	0

Table II. Response Correlation of ITC Benchmark b17

Correlation	# of cell pairs	Distribution (%)	Correlation	# of cell pairs	Distribution (%)
0.95 - 1	52	0.052	0.45 - 0.50	19,016	18.940
0.90 - 0.95	247	0.246	0.40 - 0.45	699	0.697
0.85 - 0.90	340	0.339	0.35 - 0.40	70	0.071
0.80 - 0.85	407	0.406	0.30 - 0.35	24	0.024
0.75 - 0.80	541	0.539	0.25 - 0.30	13	0.014
0.70 - 0.75	810	0.807	0.20 - 0.25	2	0.002
0.65 - 0.70	1360	1.355	0.15 - 0.20	1	0.001
0.60 - 0.65	2343	2.334	0.10 - 0.15	1	0.001
0.55 - 0.60	6512	6.486	0.05 - 0.10	0	0
0.50 - 0.55	67964	67.690	0 - 0.05	0	0

the 1452 scan cells in s38584. This experimental result also indicates that, even with 78.01% of don't-care bits, the response correlations are not purely random.

The same trend can be observed on other ISCAS and ITC benchmark circuits as well. Table II shows the result of a similar experiment on the largest ITC benchmark circuit, where the don't-care bit percentage of its test set is 89.98% and 1.58% of scan-cell pairs have a response correlation higher than 0.75.

3. PROBLEM FORMULATION

Our problem definition of the scan-cell reordering for reducing scan-shift power is given as follows.

Input:

—A circuit under test with scan cells inserted, and
—ATPG test patterns with don't-care bits (X's).

Output:

—An ordering of scan cells, and
—Test patterns with all don't-care bits specified by MT-Fill based on the derived cell ordering.

Objective:

—Generate the minimum number of scan-shift transitions for the given test patterns.

In this article, the proposed scan-cell-reordering scheme only discusses the situation of one scan chain in a design. However, the concept of the proposed reordering scheme could be extended to multiple-scan-chain architectures as well.

Given a test pattern and the scan-cell order for the scan chain, we can use the *Weighted Transition Count* (WTC) [Sankaralingam et al. 2000] to calculate the number of scan-in and scan-out transitions generated during the scan-shift cycles. The WTC considers not only the value difference between the patterns or responses of two adjacent scan cells, but also the number of transitions that this value difference generates during the scan-shift cycles. Eqs. (1) and (2) define the $WTC_{in}(i)$ and $WTC_{out}(i)$ to calculate the scan-in transitions and scan-out transitions generated by the ith pattern, respectively.

$$WTC_{in}(i) = \sum_{j=0}^{s-1} PD(j) \times W_{PD}(j) \tag{1}$$

$$WTC_{out}(i) = \sum_{j=0}^{s-1} RD(j) \times W_{RD}(j) \tag{2}$$

In Eqs. (1) and (2), s denotes the total number of scan cells; $PD(j)$ ($RD(j)$) denotes the value difference between the scan-in pattern (scan-out response) of the jth cell and the $j+1$ cell; $W_{PD}(j)$ denotes the number of scan-in transitions generated by the pattern-value difference $PD(j)$ when shifting in the corresponding pattern values from the scan-chain input to the $j+1$ cell; $W_{RD}(j)$ denotes the number of scan-out transitions generated by the response-value difference $RD(j)$ when shifting out the responses from the j cell to the scan-chain output.

In the WTC calculation, $W_{PD}(j) = j$, implying that a pattern-value difference can generate more scan-in transitions if this value difference occurs closer to the scan-chain output. On the contrary, $W_{RD}(j) = s - 1 - j$, implying that a response-value difference can generate more scan-out transitions if this value difference occurs closer to the scan-chain input. Figure 1 shows an example of the WTC computation on a 6-cell scan chain, assuming that three value differences occur between cells (C_1, C_2) , (C_2, C_3), and (C_5, C_6) for both the test pattern and its response.

Eq. (3) calculates the total number of transitions, WTC_{total}, generated by a given test set with m test patterns.

$$WTC_{total} = \sum_{i=1}^{m} \left[WTC_{in}(i) + WTC_{out}(i)\right] \tag{3}$$

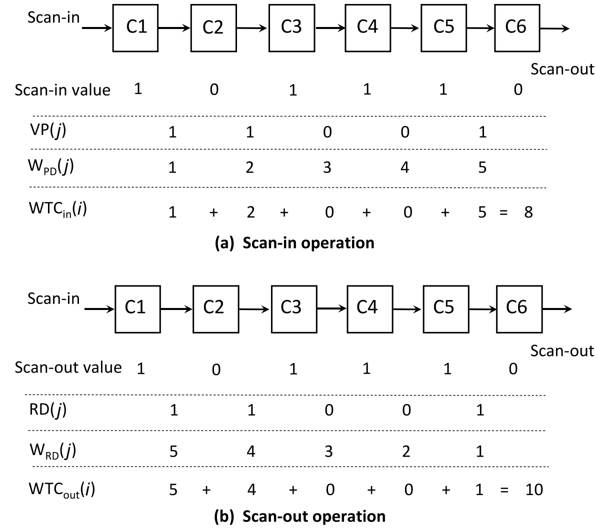
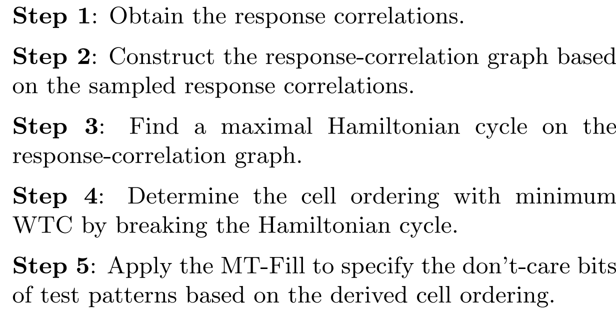


Fig. 1. Calculation of scan-in and scan-out WTC.


Step 1: Obtain the response correlations.

Step 2: Construct the response-correlation graph based on the sampled response correlations.

Step 3: Find a maximal Hamiltonian cycle on the response-correlation graph.

Step 4: Determine the cell ordering with minimum WTC by breaking the Hamiltonian cycle.

Step 5: Apply the MT-Fill to specify the don't-care bits of test patterns based on the derived cell ordering.

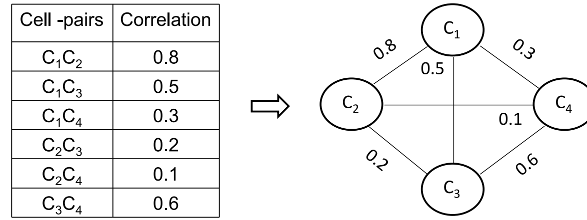
Fig. 2. Main steps of the proposed reordering scheme RORC.

4. SCAN-CELL REORDERING CONSIDERING ONLY RESPONSE CORRELATION

4.1 Detailed Steps of Reordering Scheme

We introduce a scan-cell reordering scheme, named *RORC* (ReOrdering considering Response Correlation), which first reduces the scan-out transitions by minimizing the response correlations while preserving all don't-care bits in the test patterns. Then, the scan-in transitions are further minimized by specifying the don't-care bits with MT-fill. Figure 2 shows the flow of RORC, which consists of five main steps. The detail of each step is described in the following subsections.

4.1.1 *Obtain Response Correlations.* A simulation-based method is applied to sample the response correlations between each pair of scan cells. However,



Cell -pairs	Correlation
C_1C_2	0.8
C_1C_3	0.5
C_1C_4	0.3
C_2C_3	0.2
C_2C_4	0.1
C_3C_4	0.6

Fig. 3. Construction of a response-correlation graph.

the filling of don't-care bits in RORC is not purely random since the MT-fill technique will be applied later in RORC. Therefore, in this step, we randomly generate the scan-cell ordering multiple times, specify don't-care bits using MT-fill based on each generated scan-cell ordering, and then collect the response correlations by simulating the filled patterns. The number of random-generated cell orderings used in simulation will determine the accuracy of the sampled response correlations. We use the following empirical equation to determine this number of random-generated cell orderings. We have

$$Simulation_Times = (G_Counts/50) \times P_Counts, \quad (4)$$

where *G_Counts* and *P_Counts* denote the circuit gate count and the number of given test patterns, respectively.

4.1.2 *Construct the Correlation Graph.* After obtaining the response correlations, we construct a nondirected graph, named *response-correlation graph*, in which a vertex represents a scan cell and the weight of each edge represents the response correlation between the adjacent vertices. Because any pair of scan cells could be placed next to each other, the response-correlation graph is a complete graph. Figure 3 shows an example of constructing a response-correlation graph with four scan cells.

4.1.3 *Find a Maximal Hamiltonian Cycle.* A higher response correlation between two scan cells implies a lower probability that a response-value difference occurs between the two cells. Based on this concept, the maximum Hamiltonian cycle on the response-correlation graph implies a scan-cell ordering on which the number of value differences generated between adjacent cells is statistically minimum. Finding the maximum Hamiltonian cycle is known as the *Traveling Salesman Problem* (TSP), which is NP-complete. We use a greedy TSP algorithm which orders one vertex at a time to form the cycle. The selection criterion for the new ordered vertex is to find the vertex which has the maximum weight with the previous ordered vertex. In addition, we select the first N largest edges as the initial searching points and report the best result out of these N trials, where N denotes the total number of scan cells. The time complexity of this algorithm is $Q(N^3)$.

4.1.4 *Determine Cell Ordering with Minimal WTC.* In the previous step, we obtained a maximal Hamiltonian cycle on the response-correlation graph so

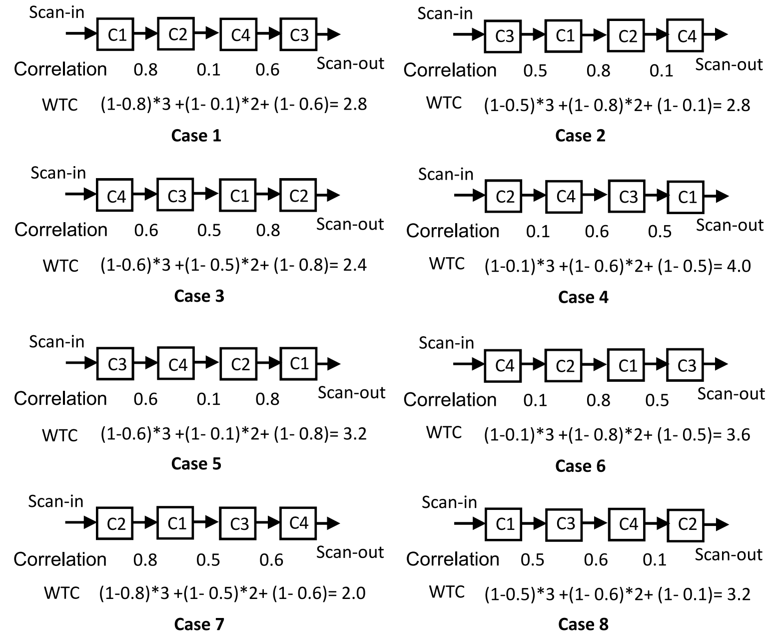


Fig. 4. Estimated WTC_{out} of different scan-chain input/output.

that the number of potential response-value differences between adjacent cells can be minimized. However, to minimize the WTC_{out}, we need to consider not only the number of response-value differences but also the positions of those value differences in the cell ordering (as discussed in Section 3). In step 4, we break the given maximal Hamiltonian cycle into a Hamiltonian path which forms the final scan-cell ordering. The breaking of the Hamiltonian cycle will affect the positions of the response-value differences and, in turn, affect the WTC_{out}. Here, we estimate the WTC_{out} generated by each possible breaking of the given Hamiltonian cycle and use the breaking with the minimum WTC_{out} to form the final cell ordering.

The estimated WTC_{out} here is obtained by replacing the $RD(j)$ in Eq. (2) with 1 minus the response correlation between cell j and $j+1$. For example, the maximal Hamiltonian cycle in Figure 3 is C_1-C_2-C_4-C_3-C_1. Figure 4 shows the estimated WTC_{out} for all eight cases of the possible cycle breaking. The final cell ordering of the scan chain is C_2-C_1-C_3-C_4.

4.1.5 *Apply MT-Fill to Specify Don't-Care Bits.* After the scan-cell ordering is decided in the previous step, we apply the MT-fill technique to fill the don't-care bits of the test patterns so that the scan-in transitions based on the scan-cell ordering can be minimized. The rule of MT-fill is that a don't-care bit is filled with the value of the first encountered specified bit when traversing from the don't-care bit toward the scan-chain output. Refer to Sankaralingam et al. [2000] for more details of MT-fill.

Table III. Statistics of the Benchmark Circuits and Their ATPG Patterns

circuit	gate count	PI	PO	# of scan cell	# of ATPG pattern	don't-care-bit percentage(%)	total faults	coverage (%)
s13207	7,951	31	121	669	108	79.65	21,190	100
s15850	9,772	14	87	597	117	75.35	23,244	100
s35932	16,065	35	320	1,728	24	37.36	57,084	100
s38417	15,106	28	106	1,636	167	78.94	61,754	100
s38584	19,253	12	278	1,452	148	78.01	71,278	100
b17	22,645	37	97	1,415	778	89.98	128,886	99.57
b20	8,875	32	22	490	539	73.37	47,040	99.56
b21	9,259	32	22	490	543	74.41	47,548	99.77
b22	14,282	32	22	735	530	75.51	70,750	99.91

4.2 Experimental Results

We conduct experiments on ten ISCAS and ITC benchmark circuits. Table III first shows the statistics of the benchmark circuits and their ATPG patterns generated by Synopsys [2010].

The following experiment compares RORC with another scan-cell reordering scheme presented in Bonhomme et al. [2002], which requires fully-specified test patterns before the reordering. Since RORC applies MT-fill to minimize the scan-in transitions, we apply MT-fill for Bonhomme et al. [2002] as well. In the following experiment of Bonhomme et al. [2002], we first randomly generate an initial scan-cell ordering and specify the don't-care bits using MT-fill according to that initial ordering. Then the reordering scheme in Bonhomme et al. [2002] is applied to obtain the final scan-cell ordering based on the filled patterns. We repeat the aforesaid steps 100 times and report the best results for Bonhomme et al. [2002]. Also, we use the same TSP algorithm in both RORC and Bonhomme et al. [2002] to make a fair comparison.

In Table IV, Columns 3, 4, and 5 list the numbers of scan-in transitions, scan-out transitions, total scan-shift transitions, respectively. Column 6 lists the peak number of scan-shift transitions at a single scan-shift cycle. Column 7 lists the runtime in seconds. The results show that RORC can outperform Bonhomme et al. [2002] with an average 43.68% and 49.50% reduction to the number of scan-in transitions and scan-out transitions, respectively. The reduction to scan-in transitions first demonstrates the advantages of preserving don't-care bits for later minimization. Also, the reduction to scan-out transitions demonstrates the effectiveness of using sampled response correlations to guide the reordering process. The reduction to peak transitions is a by product of the reduction to total scan-shift transitions. Note that the result reported for Bonhomme et al. [2002] is selected from 100 trials of random initial cell ordering. It implies that, even with MT-fill, specifying all don't-care bits before reordering will significantly decrease the opportunity in minimizing scan-shift transitions later on and, in turn, lead to a local optimum. It also implies that the optimal cell ordering obtained by RORC is hard to be achieved by randomly assigning the initial cell ordering of Bonhomme et al. [2002] for multiple times.

Note that the runtime of Bonhomme et al. [2002] listed in Table IV is the runtime for only one trial, but the reported result of Bonhomme et al. [2002] is the best result from 100 trails. Therefore, the runtime for the

Table IV. Comparisons of Generated Scan-Shift Transitions between RORC and Bonhomme et al. [2002]

circuit	method	scan-in trans.	scan-out trans.	total trans.	peak trans.	runtime (sec)
s13207	[Bonhomme et al. 2002]	3,951,373	4,188,819	8,173,642	289	4
	RORC	1,312,934	2,847,104	4,204,192	233	35
	improv.	66.77%	32.03%	48.56%	19.38%	-
s15850	[Bonhomme et al. 2002]	2,800,025	4,904,948	7,736,017	277	3
	RORC	1,497,065	2,157,662	3,685,771	211	36
	improv.	46.53%	56.01%	52.36%	23.83%	-
s35932	[Bonhomme et al. 2002]	4,543,209	4,934,478	9,524,285	525	3
	RORC	5,388,270	4,363,125	9,772,131	680	107
	improv.	-18.60%	11.58%	-2.60%	-29.52%	-
s38417	[Bonhomme et al. 2002]	29,942,845	58,416,311	88,478,584	713	41
	RORC	11,453,864	27,547,170	39,127,006	529	601
	improv.	61.75%	52.84%	55.78%	25.81%	-
s38584	[Bonhomme et al. 2002]	22,827,002	41,743,137	64,667,423	714	31
	RORC	12,489,481	27,615,042	40,223,587	694	543
	improv.	45.29%	33.85%	37.80%	2.80%	-
b17	[Bonhomme et al. 2002]	95,302,661	230,963,547	326,795,418	700	62
	RORC	24,619,742	41,550,664	66,500,101	570	3,611
	improv.	74.17%	82.01%	79.65%	18.57%	-
b20	[Bonhomme et al. 2002]	7,680,415	12,332,467	20,133,912	237	5
	RORC	4,823,088	4,662,118	9,623,386	171	138
	improv.	37.20%	62.20%	52.20%	27.85%	-
b21	[Bonhomme et al. 2002]	7,351,208	11,834,023	19,330,271	229	6
	RORC	4,546,521	4,590,188	9,266,069	205	476
	improv.	38.15%	61.21%	52.068%	10.48%	-
b22	[Bonhomme et al. 2002]	17,200,814	23,447,118	40,809,632	362	12
	RORC	9,997,996	10,844,186	21,036,957	276	154
	improv.	41.87%	53.75%	48.45%	23.76%	-
Ave. improv.		43.68%	49.50%	47.13%	13.67%	-

result of Bonhomme et al. [2002] is actually longer than that of RORC. In addition, please also note that the comparison of total scan-shift transitions shown in Table IV also represents the comparison of the average scan-shift transitions per cycle, which can be computed by dividing the total number of shift transitions by the total number of shift cycles.

Table V reports RORC's runtime distribution and memory usage for each benchmark circuit. Column 2 to 4 lists the runtime spent in the response-correlation sampling (Column 2), TSP algorithm (Column 3), and other computation (Column 4), respectively. Column 5 lists the total runtime. Column 6 lists the ratio of the runtime spent in correlation sampling over the total runtime. In average, 90% of the total runtime is spent on sampling the response correlations, which is actually the efficiency bottleneck of the proposed scan-cell reordering scheme. At last, Column 7 lists the memory usage of RORC. The largest memory usage among the benchmark circuits is 21.2M.

In Table IV, the total number of scan-shift transitions is actually slightly larger than the sum of scan-in transitions and scan-out transitions. This is because we omitted the *in-between transitions* in Table IV, which are generated by the value difference between the first bit of a scan-in pattern and the last bit of its previous scan-out response. The percentage of in-between transitions is low compared to scan-in and scan-out transitions. It can be further reduced by a pattern-reordering scheme proposed in the next section.

Table V. Runtime Distribution of RORC (in seconds) and Its Memory Usage

circuit	correlation sampling (a)	TSP	others	Total (b)	(a) / (b)	memory usage
s13207	31	1	3	35	0.89	8.4M
s15850	33	1	2	36	0.92	8.5M
s35932	82	15	10	107	0.77	20.9M
s38417	562	11	28	601	0.94	21.2M
s38584	512	10	21	543	0.94	18.2M
b17	3,505	7	99	3,611	0.97	19.1M
b20	126	0	12	138	0.91	7.9M
b21	454	1	21	476	0.95	7.9M
b22	141	0	13	154	0.92	7.7M
avg.					0.90	

Table VI. Classification of Patterns

type	first scan-in bit	last scan-out bit
TA	0	0
TB	0	1
TC	1	0
TD	1	1

5. PATTERN REORDERING FOR MINIMIZING IN-BETWEEN TRANSITIONS

5.1 Detailed Steps of Pattern Reordering

We first divide test patterns into four types, A, B, C, and D, according to the first scan-in bit of a pattern and the last scan-out bit of its response. The other bits in a pattern and its response cannot affect the number of in-between transitions. Table VI lists the classification rules for each type of patterns.

Next, we denote the A_i as the ith pattern in type A. The same suffix notation is applied to pattern type B, C, and D. The numbers of patterns of type A, B, C, and D are w, x, y, and z, respectively. If $x > y$, we reorder the patterns according to the following ordering.

$$A_1 \sim A_w B_1 C_1 B_2 C_2 \sim B_y C_y B_{y+1} \sim B_x D_1 \sim D_z \tag{5}$$

If $x < y$, then we apply the following ordering.

$$D_1 \sim D_z C_1 B_1 C_2 B_2 \sim C_x B_x C_{x+1} \sim C_y A_1 \sim A_z \tag{6}$$

The preceding two pattern orderings both attempt to alternately arrange one type-B pattern next to one type-C pattern as often as possible, such that the last bit of more responses can be the same as the first bit of their next pattern. Both aforesaid pattern orderings also consecutively arrange all type-A patterns or all type-D patterns next to each other. There is no in-between transition among such a consecutive sequence of type-A patterns or type-B patterns. Please also note that putting all consecutive type-A patterns or type-D patterns in the middle of the sequence can also result in the same total number of in-between transitions as the aforesaid two listed pattern orderings.

Table VII. Comparison of in-between Transitions between Using and Without Using Pattern Reordering

circuit	without pattern_reordering			with pattern_reordering		
	in-btwn trans.	total trans.	ratio	in-btwn transition	total trans.	ratio
s13207	44,154	4,204,192	1.050%	33,450	4,193,488	0.798%
s15850	26,268	3,680,995	0.714%	4,776	3,659,503	0.131%
s35932	20,736	9,772,131	0.212%	12,096	9,763,491	0.124%
s38417	114,520	39,115,554	0.293%	3,272	39,004,306	0.008%
s38584	92,928	40,197,451	0.231%	31,944	40,136,467	0.080%
b17	408,935	66,579,341	0.614%	16,980	66,187,386	0.026%
b20	138,180	9,623,386	1.436%	70,560	9,555,766	0.738%
b21	99,960	9,236,669	1.082%	19,600	9,156,309	0.214%
b22	188,160	21,030,342	0.895%	10,290	20,852,472	0.049%
avg.			0.915%			0.241%

Table VII compares the results with and without applying the proposed pattern reordering. Column 2, 3, and 4 list the number of in-between transitions, the number of total transitions, and the ratio of in-between transitions over the total transitions, respectively, without applying the proposed pattern reordering. Column 5, 6, and 7 list the corresponding results with applying the proposed pattern reordering. As the results show, the average ratio of in-between transitions can be reduced from 0.915% to 0.241% by applying the proposed pattern reordering. Also, the runtime of this pattern reordering is fast (less than 1 second for all benchmark circuits).

Since the percentage of in-between transitions is much lower than that of scan-in transitions or scan-out transitions (0.241% in average), we will not individually list the number of in-between transitions in later experiments so that the focus of our scan-cell reordering schemes can be on the scan-in and scan-out transitions. We will still count in-between transitions in the total number of scan-shift transitions.

6. SCAN-CELL REORDERING CONSIDERING BOTH RESPONSE AND PATTERN CORRELATIONS

As the results show in Table IV, RORC generates a lower number of total scan-shift transitions than Bonhomme et al. [2002] in all circuits but s35932. This exception may contribute to its low don't-care bit percentage of 37.36%. From our internal experiments, we found that a cell ordering will affect the results of the MT-fill more significantly when the don't-care bit percentage becomes lower. However, RORC can only reduce scan-out transitions by minimizing the response correlations between adjacent cells. It ignores the impact of the cell ordering on the number of scan-in transitions resulted from the MT-fill patterns.

In this section, we introduce another scan-cell reordering scheme, named *ROBPR* (ReOrdering considering Both Pattern and Response correlation), which can simultaneously optimize the pattern correlations and response correlations during the reordering process.

Step 1: Collect pattern and response correlations.

Step 2: Construct a directed multiple-weight graph based on the collected pattern and response correlations.

Step 3: Find the Hamiltonian path with the minimum WTC.

Step 4: Apply the MT-Fill to specify the don't-care bits based on the derived cell ordering.

Fig. 5. Main steps of the proposed reordering scheme ROBPR.

Table VIII. Different Cases of Pattern Correlations between Two Adjacent Cells

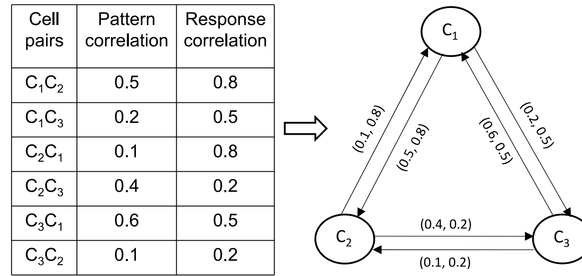
case	value of cell i	value of cell j	$PC_k(i, j)$
1	0	0	1
2	0	1	0
3	0	X	$S_0/(S_0 + S_1)$
4	1	0	0
5	1	1	1
6	1	X	$S_1/(S_0 + S_1)$
7	X	0	1
8	X	1	1
9	X	X	1

6.1 Detailed Steps of Reordering Scheme

Figure 5 shows the flow of ROBPR consisting of four main steps. The details of steps 1–3 are described in the following subsections. The detail of step 4 is the same as the step 5 in RORC and hence omitted in this section.

6.1.1 *Obtain Pattern and Response Correlations.* In order to measure the impact of a scan-cell ordering on the number of scan-in transitions, we first define the *pattern correlation* between cell i and cell j as the probability that the pattern values on these two cells are the same when the output of cell i is connected to the input of cell j. Note that this pattern correlation is dependent on the order of cells. For a test pattern k, Table VIII considers each combination of pattern values between cell i and cell j, and lists its corresponding pattern correlation after MT-fill (denoted as $PC_k(i, j)$).

In cases 1, 2, 4, and 5, both values of cell i and j are specified bits and hence their pattern correlations can be determined immediately for test pattern k. In cases 7, 8, and 9, a don't-care bit is placed prior to a specified bit and hence the don't-care bit will be filled with the same value as the specified bit. In cases 3 and 6, a specified bit is placed prior to a don't-care bit. Hence, the value of this don't-care bit cannot be derived immediately and has to be determined by its first encountered specified bit when traversing toward the scan-chain output. We use $S_0/(S_0 + S_1)$ ($S_1/(S_0 + S_1)$) to represent the probability that its first encountered specified bit is a 0 (1), where S_0 and S_1 denote the total numbers of specified 1s and 0s in the test pattern, respectively.



Cell pairs	Pattern correlation	Response correlation
C_1C_2	0.5	0.8
C_1C_3	0.2	0.5
C_2C_1	0.1	0.8
C_2C_3	0.4	0.2
C_3C_1	0.6	0.5
C_3C_2	0.1	0.2

Fig. 6. Construction of the directed graph based on pattern and response correlations.

After calculating the $PC_k(i, j)$ for each pattern k, the pattern correlation between cell i and cell j for the entire test set can be obtained by averaging the $PC_k(i, j)$ for each pattern k.

As to the response correlations, we use the same simulation-based method described in Section 6.1.1 to estimate them.

6.1.2 *Construct the Directed Correlation Graph.* The correlation graph constructed in ROBPR is a revised version of the correlation graph in Section 4.1.2. First, this correlation graph is directed. Second, an edge in this correlation graph has two weights (W_p, W_r), where W_p and W_r represent the pattern correlation and response correlation, respectively. Figure 6 shows an example of constructing such a directed correlation graph given the pattern and response correlations between three scan cells.

6.1.3 *Find the Hamiltonian Path with Minimal WTC.* Unlike RORC which finds a Hamiltonian cycle first and then breaks the Hamiltonian cycle to obtain a Hamiltonian path with minimal estimated WTC_{out}, ROBPR uses an integrated algorithm to directly obtain the Hamiltonian path with minimal estimated WTC_{total} on the correlation graph. Figure 7 shows the proposed greedy-based algorithm which also ordered one new vertex at a time to form such a Hamiltonian path.

When adding the nth nonordered vertex V_{non} for the Hamiltonian path, this algorithm uses a cost function $Cost(V_{last}, V_{non}, n)$ to measure the impact of the new-added edge (V_{last}, V_{non}) on WTC_{total}, which is defined in Eq. (3). In the definition of $Cost(V_i, V_j, n)$ in Figure 7, the $\overline{W_p}(V_i, V_j)$ ($\overline{W_r}(V_i, V_j)$) actually represents the probability that a pattern-value (response-value) difference occurs between V_i and V_j. The n in the cost function actually represents the $W_{PD}(n)$ described in the WTC Eq. (1). The $N - 1 - n$ in the cost function actually represents the $W_{RD}(n)$ described in the WTC Eq. (2).

This cost function will guide the algorithm to emphasize more on the response correlation in the beginning of the ordering process and then gradually move its emphasis to the pattern correlation in the later stage of the reordering process, which exactly reflects the WTC definition in Eqs. (1) and (2).

```
1  #define
2     W_p(V_i, V_j) : the pattern correlation of edge (V_i, V_j)
3     W_r(V_i, V_j) : the response correlation of edge (V_i, V_j)
4     ¬W_p(V_i, V_j) : 1 − W_p(V_i, V_j)
5     ¬W_r(V_i, V_j) : 1 − W_r(V_i, V_j)
6     Cost(V_i, V_j, n) : ¬W_p(V_i, V_j) × n + ¬W_r(V_i, V_j) × (N − 1 − n)
7  begin
8     N ← # of cells ; n ← 1 ;
9     Min_l ← a list of N edges having the minimum (¬W_p + ¬W_r × (N-1));
10     for each directed edge e(V_i, V_j) of Min_l
11       V_1st ← V_i, V_2nd ← V_j, V_last ← V_2nd;
12       while non-ordered V
13         cost_min ← ∞ ; n ← (n + 1) ;
14         for each non-ordered V_non
15           if (Cost(V_last, V_non, n) < cost_min)
16               cost_min ← Cost(V_last, V_non, n) ;
17               V_next ← V_non ;
18           endif
19         endfor
20         V_last ← V_next
21       endwhile
22     endfor
23 end
```

Fig. 7. The proposed algorithm for finding a Hamiltonian path with minimal WTC_{total}.

6.2 Experimental Results

We conduct experiments for ROBPR on the same benchmark circuits and test patterns as in Section 4.2. Table IX compares the results of ROBPR with the results of RORC, which considers only the response correlation during the reordering. The experimental results show that, in average, ROBPR can generate 34.87% less scan-in transitions but only 6.33% more scan-out transitions compared to RORC. This significant reduction in scan-in transitions first demonstrates the advantage of adding the pattern correlations into consideration during the ordering process. It also shows the effectiveness of the pattern-correlation estimation listed in Table VIII.

The average reduction to the total scan-shift transitions is 12.38% by ROBPR. The 7.52% reduction to the number of peak transitions is a by-product of the reduction to total scan-shift transitions as well. The overall result again demonstrates the benefit of considering pattern correlations and response correlations simultaneously during the reordering. In addition, the reported runtime of ROBPR is almost the same as RORC, even though ROBPR needs to collect additional information for pattern-correlations calculation. It is because the proposed algorithm in step 3 (Figure 7) can directly find the Hamiltonian path with minimal WTC_{total}, saving a step of breaking a Hamiltonian cycle to obtain the final ordering, such as the step 4 in RORC.

Table X further compares ROBPR with another scan-cell reordering scheme [Sinanoglu et al. 1998], which can also reorder the scan cells based on a test set with nonfilled don't-care bits. As the result shows, RORC can generate 42.69% less scan-in transitions and 68.15% less scan-out transitions compared to Sinanoglu et al. [1998]. The total number of scan-shift transitions and the number of peak transitions generated by RORC are 64.18% and 31.24% less than that generated by Sinanoglu et al. [1998], respectively. This

Table IX. Comparisons of Generated Scan-Shift Transitions between RORC and ROBPR

circuit	method	scan-in trans.	scan-out trans.	total trans.	peak trans.	runtime (sec)
s13207	RORC	1,312,934	2,847,104	4,193,488	233	40
	ROBPR	882,926	2,780,763	3,665,027	168	40
	improv.	32.75%	2.33%	12.60%	27.90%	-
s15850	RORC	1,497,065	2,157,662	3,659,503	211	43
	ROBPR	1,029,107	1,944,970	2,994,375	179	43
	improv.	31.26%	9.86%	18.18%	15.17%	-
s35932	RORC	5,388,270	4,363,125	9,763,491	680	110
	ROBPR	1,963,178	5,356,284	7,329,830	641	133
	improv.	63.57%	-22.76%	24.93%	5.74%	-
s38417	RORC	11,453,864	27,547,170	39,004,306	529	631
	ROBPR	9,599,399	29,676,522	39,396,985	521	632
	improv.	16.19%	-7.73%	-1.01%	1.51%	-
s38584	RORC	12,489,481	27,615,042	40,136,467	694	583
	ROBPR	10,064,216	27,385,766	37,493,542	580	585
	improv.	19.42%	0.83%	6.58%	16.43%	-
b17	RORC	24,619,742	41,550,664	66,187,386	570	3,464
	ROBPR	16,202,102	46,655,210	63,096,447	563	3,469
	improv.	34.19%	-12.29%	4.67%	1.23%	-
b20	RORC	4,823,088	4,662,118	9,555,766	171	144
	ROBPR	3,491,947	4,835,560	8,357,887	181	146
	improv.	27.60%	-3.72%	12.54%	-5.85%	-
b21	RORC	4,546,521	4,590,188	9,156,309	205	154
	ROBPR	2,914,102	4,960,108	7,887,930	195	158
	improv.	35.90%	-8.06%	13.85%	4.88%	-
b22	RORC	9,997,996	10,844,186	20,852,472	276	504
	ROBPR	5,603,864	11,233,009	16,878,768	261	508
	improv.	43.95%	-3.59%	19.06%	5.43%	-
Ave. improv.		34.87%	-6.33%	12.38%	7.52%	-

result first demonstrates that the traditional controllability measures used by Sinanoglu et al. [1998] may not be able to accurately predict the response correlations. Also, not considering the impact of the input-pattern filling technique in Sinanoglu et al. [1998] may fail to effectively minimize the scan-in transitions as RORC does.

7. SCAN-CELL REORDERING USING SCAN-DATA INVERSION

To reduce potential signal transitions, both RORC and ROBPR arrange the scan cells with a high response (or pattern) correlation next to each other. It is because a high correlation between two scan cells represents a high probability that their response (or pattern) values are the same. On the contrary, a low correlation between two scan cells means that their response (or pattern) values are most likely inverse to each other. In such a low-correlation case, if we can inverse the value of a cell before it propagates to the scan-in port of the other cell, this low correlation can be turned into a high correlation and become helpful for minimizing scan-shift transitions.

Table X. Comparisons of Generated Scan-Shift Transitions between Sinanoglu et al. [1998] and ROBPR

circuit	method	scan-in trans.	scan-out trans.	total trans.	peak trans.	runtime (sec)
s13207	[Sinanoglu et al. 1998]	1,274,101	10,267,563	11,541,664	344	1
	ROBPR	882,926	2,780,763	3,665,027	168	40
	improv.	30.70 %	72.92 %	68.25 %	51.16%	-
s15850	[Sinanoglu et al. 1998]	1,581,270	7,202,685	8,783,955	293	9
	ROBPR	1,029,107	1,944,970	2,994,375	179	43
	improv.	32.92 %	73.00%	65.91 %	38.91%	-
s35932	[Sinanoglu et al. 1998]	6,035,857	11,660,765	17,696,622	859	6
	ROBPR	1,963,178	5,356,284	7,329,830	641	133
	improv.	67.47 %	54.07 %	58.58 %	25.38%	-
s38417	[Sinanoglu et al. 1998]	16,645,507	90,369,301	107,014,808	812	7
	ROBPR	9,599,399	29,676,522	39,396,985	521	632
	improv.	42.33 %	67.16 %	63.19 %	35.84%	-
s38584	[Sinanoglu et al. 1998]	12,873,474	62,334,043	75,207,517	752	9
	ROBPR	10,064,216	27,385,766	37,493,542	580	585
	improv.	21.82 %	56.07%	50.15 %	22.87%	-
b17	[Sinanoglu et al. 1998]	25,829,926	360,611,858	386,441,784	765	82
	ROBPR	16,202,102	46,655,210	63,096,447	563	3,469
	improv.	37.27 %	87.06%	83.67 %	26.41%	-
b20	[Sinanoglu et al. 1998]	6,419,110	15,654,059	22,073,169	261	8
	ROBPR	3,491,947	4,835,560	8,357,887	181	146
	improv.	45.60 %	69.11 %	62.14 %	30.65%	-
b21	[Sinanoglu et al. 1998]	6,018,821	15,514,204	21,533,025	247	9
	ROBPR	2,914,102	4,960,108	7,887,930	195	158
	improv.	51.58 %	68.03%	63.37 %	21.05%	-
b22	[Sinanoglu et al. 1998]	11,801,093	33,014,920	44,816,013	367	18
	ROBPR	5,603,864	11,233,009	16,878,768	261	508
	improv.	52.51 %	65.98 %	62.34 %	28.88%	-
Ave. improv.		42.69%	68.15%	64.18%	31.24%	-

In this section, we introduce a scan-cell-reordering scheme named SIRO (Scan-data-Inversion ReOrdering). SIRO selectively applies the inversion connection between two scan cells and hence can take advantage of both high correlations and low correlations between responses and patterns. This inverse connection between adjacent cells has also been utilized in some previous works, such as Sinanoglu et al. [1998, 2003], to minimize the number of scan-shift transitions.

7.1 Detailed Steps of Reordering Scheme

Figure 8 shows the overall flow of SIRO, which consists of the following four steps.

7.1.1 *Obtain Inverse Pattern and Response Correlations.* In SIRO, when connecting a scan cell i to its next scan cell j, two types of connections can be made. One is *direct connection*, which connects the value Q of i to the scan-in port SI of j. The other type is the *inverse connection*, which connects the inverse value $\overline{Q}$ of i to the scan-in port SI of j. In RORC and ROBPR, we already discussed how to estimate the response and pattern correlations when using the direct connection. The focus here is to estimate the response correlations and pattern correlations when using the inverse connection.

Step 1: Collect the inverse and non-inverse pattern and response correlations.

Step 2: Construct a directed multiple-weight graph based on the collected inverse and non-inverse pattern and response correlations.

Step 3: Find the Hamiltonian path with the minimum WTC.

Step 4: Determine the Scan-in Patterns Based on Derived Cell Ordering.

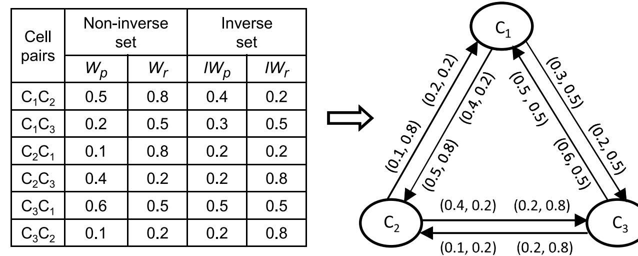
Fig. 8. Main steps of the proposed reordering scheme SIRO.

Table XI. Different Cases of Inverse Pattern Correlations between Two Adjacent Cells

case	value of cell i	value of cell j	$IPC_k(i, j)$
1	0	0	0
2	0	1	1
3	0	X	$S_1/(S_0 + S_1)$
4	1	0	1
5	1	1	0
6	1	X	$S_0/(S_0 + S_1)$
7	X	0	1
8	X	1	1
9	X	X	1

The response correlation for an inverse connection can be simply estimated by 1 minus the response correlation calculated for a direct connection. However, it is more complicated to estimate the pattern correlations for an inverse connection. This is because the MT-fill can adjust its filling of don't-care bits according to the inverse connection or the direct connection. We first define the *inverse pattern correlation* between cell i and cell j for pattern k as $IPC_k(i, j)$, which is the probability that the pattern values on these two cells are the same when cell i is inversely connected to cell j. Table XI shows the inverse pattern correlation for different combinations of pattern values between cell i and cell j after MT-fill. The derivation of Table XI is similar to Table VIII. The only difference is that, for an inversely connected cell pair, a transition is generated when the specified values of both cells are the same. The definition of S_0 and S_1 are the same as that in Table VIII.

7.1.2 *Construct the Directed Correlation Graph.* The correlation graph constructed in SIRO is a revised version of the correlation graph in ROBPR. The difference is that an edge in this correlation graph has two sets of weights: noninverse set (W_p, W_r) and inverse set (IW_p, IW_r), where W_p and W_r represent the direct pattern correlation and response correlation as calculated in ROBPR, and IW_p and IW_r represent the inverse pattern correlation and response correlation as calculated in previous step. Figure 9 shows an example of constructing such a directed correlation graph given the inverse and noninverse correlation sets between three scan cells.



Cell pairs	Non-inverse set		Inverse set	
	W_p	W_r	IW_p	IW_r
C_1C_2	0.5	0.8	0.4	0.2
C_1C_3	0.2	0.5	0.3	0.5
C_2C_1	0.1	0.8	0.2	0.2
C_2C_3	0.4	0.2	0.2	0.8
C_3C_1	0.6	0.5	0.5	0.5
C_3C_2	0.1	0.2	0.2	0.8

Fig. 9. Construction of the directed graph based on inverse and noninverse correlation sets.

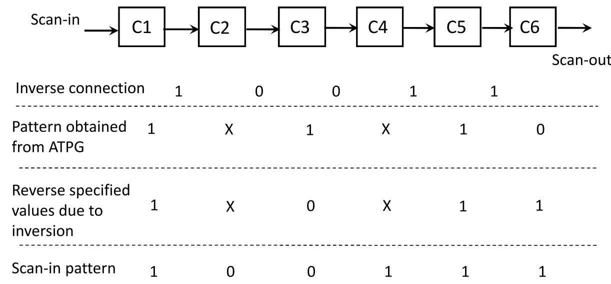


Fig. 10. An example of the MT-fill performed in SIRO.

7.1.3 *Find the Hamiltonian Path with Minimal WTC.* The algorithm in this step is similar to the algorithm in Figure 7, except two types of connections can be chosen in SIRO. When selecting the nth ordered cell, SIRO needs to evaluate both the cost function for a direct connection Cost(V_i, V_j , n) and the cost function for an inverse connection $Cost_{inv}(V_i, V_j, n)$. $Cost(V_i, V_j, n)$ is defined in ROBPR. $Cost_{inv}(V_i, V_j, n)$ is defined as follows.

$$Cost_{inv}(V_i, V_j, n) = IW_p(V_i, V_j) \times n + IW_r(V_i, V_j) \times (N - 1 - n) \quad (7)$$

The cell with the highest cost function will be selected and the selected cell is directly or inversely connected to the next cell according to the type of the highest cost function ($Cost(V_i, V_j, n)$ or $Cost_{inv}(V_i, V_j, n)$).

7.1.4 *Determine the Scan-In Patterns Based on Derived Cell Ordering.* Unlike the RORC and ROBPR which directly use the traditional MT-fill to determine patterns based on the derived cell ordering, the MT-fill in SIRO needs to apply a different filling rule to handle inverse connections. First, the value of a specified bit is inverse if an odd number of inverse connections are encountered before the specified bit. The specified bits remain the same if an even number of inverse connections are encountered. Next, the don't-care bits between the modified specified bits are filled using the traditional MT-fill. Figure 10 shows an example of the revised MT-fill to handle inverse connections.

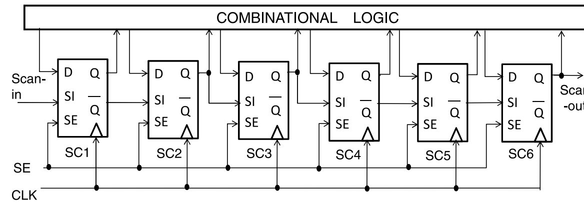


Fig. 11. Implement inverse techniques with traditional scan-chain architecture.

In Figure 10, the scan chain contains six scan cells, and three inversions occur on cell pairs (C1,C2), (C4,C5), and (C5,C6), respectively. Because C2, C3, C4, and C6 pass through odd times of inversions (1 or 3 times) during scan-in operation, the specified values are inverse before MT-fill. Then the MT-fill is applied to fill all the don't-care bits according to the modified specified bits. Figure 11 shows the inverse connections of scan cells corresponding to the example in Figure 10. In Figure 11, the differences with traditional architecture is that $\overline{Q}$ connects to SI while the inversions occur.

7.2 Experimental Results

We conduct experiments for SIRO on the same benchmark circuits and test patterns as those used in Section 4.2. Table XII compares the results of SIRO with that of ROBPR, which considers only the noninverse pattern and response correlation during the reordering process. As the results show in Table XII, SIRO in average can generate 1.23% less total scan-shift transitions with almost the same runtime compared to ROBPR. However, even though RISO can generate a smaller or at least an even number of scan-shift transitions for each circuit, this 1.23% average reduction is still less than our expectation before the experiment.

After further analysis, we found that the number of inverse connections used in each circuit is actually small (as listed in the last column of Table XII). For s35932 and b17, even no inverse connection is used by SIRO. This low usage of inverse connection means that the low correlations between scan cells in those benchmark circuits are not low enough, so that the corresponding inverse correlations cannot produce a high score for the cost function $Cost_{inv}(V_i, V_j, n)$ used in step 3's greedy algorithm. This argument is further supported by the response-correlations distribution reported in Table I, where 2.05% of response correlations are larger than 0.75 but only 1.09% of response correlations are smaller than 0.25 for s38584. This trend is even more obvious for b17 as shown in Table II, where 1.58% of response correlations are larger than 0.75 but only 0.004% of response correlations are smaller than 0.25. Table XIII lists the probability distributions of response correlations for each benchmark circuit. Overall, SIRO can further reduce the scan-shift transitions for 7 out of 9 benchmark circuits.

From the preceding experiments, we can conclude that using the inverse connections can indeed help the reduction on scan-shift transitions since the only a small number of inverse connections can achieve a 1.23% average reduction in

Table XII. Comparisons of Generated Scan-Shift Transitions between SIRO and ROBPR

circuit	method	scan-in trans.	scan-out trans.	total trans.	peak trans.	runtime (sec)	inverse times
s13207	ROBPR	882,926	2,780,763	3,665,027	170	40	-
	SIRO	884,607	2,754,677	3,641,960	170	40	4
	improv.	-0.19%	0.94%	0.63%	0.00	-	-
s15850	ROBPR	1,029,107	1,944,970	2,994,375	179	43	-
	SIRO	1,039,313	1,823,428	2,880,651	180	43	10
	improv.	-0.99%	6.25%	3.80%	-0.56%	-	-
s35932	ROBPR	1,963,178	5,356,284	7,329,830	641	133	-
	SIRO	1,963,178	5,356,284	7,329,830	641	133	0
	improv.	0%	0%	0%	0%	-	-
s38417	ROBPR	9,599,399	29,676,522	39,396,985	521	632	-
	SIRO	9,244,689	29,641,440	38,905,761	536	633	18
	improv.	3.70%	0.12%	1.25%	-2.88%	-	-
s38584	ROBPR	10,064,216	27,385,766	37,493,542	580	585	-
	SIRO	10,154,228	26,438,593	36,656,709	577	586	18
	improv.	-0.89%	3.46%	2.23%	0.52%	-	-
b17	ROBPR	16,202,102	46,655,210	63,096,447	563	3,464	-
	SIRO	16,202,102	46,655,210	63,096,447	563	3,467	0
	improv.	0.00%	0.00%	0.00%	0.00%	-	-
b20	ROBPR	3,491,947	4,835,560	8,357,887	181	146	-
	SIRO	3,095,170	5,103,108	8,224,248	176	146	2
	improv.	11.36%	-5.53%	1.60%	2.76%	-	-
b21	ROBPR	2,914,102	4,960,108	7,887,930	195	158	-
	SIRO	2,643,470	5,179,474	7,832,744	189	158	4
	improv.	9.29%	-4.42%	0.70%	3.08%	-	-
b22	ROBPR	5,603,864	11,233,009	16,878,768	261	508	-
	SIRO	5,188,530	11,457,397	16,657,687	259	509	8
	improv.	7.41%	-2.00%	1.31%	0.77%	-	-
Ave. improv.		3.30%	-0.13%	1.23%	0.41%	-	-

Table XIII. Probability Distribution of High and Low Response Correlations

circuit	Response correlation			
	$0 \sim 0.5$	$0.5 \sim 1$	$0 \sim 0.25$	$0.75 \sim 1$
s13207	30.136%	69.671%	0.6055%	3.90434%
s15850	10.9135%	89.0682%	0.7638%	3.9022%
s35932	0.8342%	99.1520%	0%	16.867%
s38417	51.4924%	48.4794%	0.1156%	0.6776%
s38584	52.3962%	47.6056%	1.0907%	2.0526%
b17	19.7490%	80.2549%	0.0038%	1.5825%
b20	12.4669%	87.5379%	0.0025%	0.8771%
b21	6.5426%	93.4586%	0.0041%	1.0550%
b22	5.116%	93.4586%	0.001%	0.7047%
avg.	21.0718%	78.7429%	0.2874%	3.5136%

the total scan-shift transitions. However, the amount of this reduction is determined by the ratio of low response or pattern correlations over the high ones, which is highly circuit dependent. The reduction could be more significant if this ratio is higher.

8. SCAN CELL REORDERING CONSIDERING BOTH POWER AND ROUTING FACTORS

All aforesaid reordering schemes, such as RORC, ROBPR, and SIRO, focus on reducing the power consumption during scan-based testing. However, these reordering schemes may result in long wire length of scan paths since the connection of scan cells is determined by cells' response or pattern correlations, not cells' physical distance. In this section, we proposed a scan-cell reordering scheme, named PRORO (Power and Routing-Overhead ReOrdering), which combines the ROBPR with routing consideration. The same idea can be applied to SIRO as well.

8.1 Detail Steps of Reordering Considering both Power and Routing Overhead

In PRORO, we reorder the scan cells after the placement is done. Based on the placement result, we use the Manhattan distance between two scan cells to approximate the wire length between the two cells. When selecting the next ordered scan cell, we incorporate this approximated wire length into the cost function and hence can limit the routing overhead. In our implementation, the placement is done by a commercial back-end tool and the position of each scan cell is obtained by parsing its DEF file.

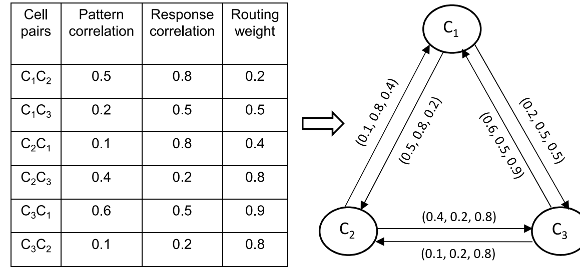
Basically, PRORO contains almost the same five steps as that of ROBPR, except some modifications to the step 2 and 3. Therefore, this subsection only shows the details of step 2 and 3. The rest of the steps all follow the steps in ROBPR.

8.1.1 *Construct a Directed Multiple-Weight Graph Based on Response/ Pattern Correlations and Routing Overhead.* As mentioned, the Manhattan distance between two cells is used to represent their routing overhead. In order to make the quantity of routing overhead compatible with the quantity of the cost function regarding scan-shift power, we normalize two cells' routing overhead (represented by the Manhattan distance) to a value between 0 to 1, which is defined as the *routing weight* between the two cells. We set the longest distance between any two cells as a routing weight of 1, and the shortest distance as a routing weight of 0.

The directed graph constructed in this section is a revised version of the directed graph introduced in ROBPR (step 2 in Section 6.1.2). An edge in the graph contains three weights (W_p, W_r, W_l), where W_p, W_r and W_l represent the pattern correlation, the response correlation, and the routing weight between the two cells, respectively. Figure 12 shows an example of constructing such a directed graph given the correlation and routing weight between three scan cells.

8.1.2 *Find the Hamiltonian Path with the Minimum WTC.* We use a similar greedy TSP algorithm as shown in Figure 7 except its cost function C_T, which is modified as follows to control the trade-off between scan-shift power and routing overhead.

$$C_T(V_i, V_j, n) = (1 - \beta) \times C_P(V_i, V_j, n) + \beta \times C_R(V_i, V_j). \tag{8}$$



Cell pairs	Pattern correlation	Response correlation	Routing weight
C_1C_2	0.5	0.8	0.2
C_1C_3	0.2	0.5	0.5
C_2C_1	0.1	0.8	0.4
C_2C_3	0.4	0.2	0.8
C_3C_1	0.6	0.5	0.9
C_3C_2	0.1	0.2	0.8

Fig. 12. Construction of the directed graph based on correlations and routing effects.

$C_R(V_i, V_j)$ represents the routing weight between cells V_i and V_j. $C_P(V_i, V_j, n)$ represents the cost function of scan-shift power when selecting the nth cell and ranges from 0 to 1 as well. The value of $C_P(V_i, V_j, n)$ is computed by the value of $Cost(V_i, V_j, n)$ divided by the maximum value of $Cost(V_i, V_j, n)$ between any two cells, where $Cost(V_i, V_j, n)$ is defined in ROBPR (see Figure 7). The parameter β in $C_T(V_i, V_j, n)$ is call the *optimization factor*, which is used to control the trade-off between scan-shift power and routing overhead. The value of β ranges from 0 to 1. If β increases, this TSP algorithm focuses more on reducing routing overhead. If β decreases, this TSP algorithm focuses more on reducing scan-shift transitions. Figure 13 shows the details of this TSP algorithm.

8.2 Experimental Results

We conduct the following experiments to compare the results of PRORO using different optimization factors with the results of ROBPR and a scan-cell reordering scheme supported by a commercial back-end tool [Cadence 2006], where ROBPR only focuses on minimizing the scan-shift transitions and Cadence's [2006] scan-cell reordering only focuses on minimizing the routing overhead of scan paths after the placement is done. In the following experiments, we first use ROBPR to obtain a scan-cell ordering and apply the APR tool in Cadence [2006] to get its placement. Then both PRORO and Cadence's [2006] scan-cell reordering are performed based on this placement of ROBPR. Cadence's [2006] scan-cell reordering is performed by using the command "scanreorder" in Cadence [2006]. A TSMC 0.18μm CMOS technology with 5 metal layers is used in the experiments.

Table XIV first lists the total number of scan-shift transitions generated by different scan-cell reordering schemes. For the convenience of result comparison, Table XIV normalizes the total number of scan-shift transitions of each reordering scheme by dividing it with the total number of scan-shift transitions of ROBPR, which is supposed to be the reordering scheme generating the least scan-shift transitions in this experiment.

Table XV lists the estimated wire length of the scan paths (in μm) generated by different scan-cell reordering schemes. This estimated wire length of scan paths is measured by the summation of the Manhattan distance between any two adjacent scan cells. Similar to Table XIV, Table XV also normalizes the

```
1  #define
2      MD(V_i, V_j) : Manhattan distance between V_i and V_j
3      MaxCost : the maximum cost of Cost(V_i, V_j, n)
4      MinCost : the minimum cost of Cost(V_i, V_j, n)
5      MaxMD : the maximum Manhattan distance between
two cells
6      MinMD : the minimum Manhattan distance between
two cells
7      C_P(V_i, V_j, n) : Cost(V_i, V_j, n)/(MaxCost − MinCost)
8      C_R(V_i, V_j) : MD(V_i, V_j)/(MaxMD − MinMD)
9      C_T(V_i, V_j, n) : (1 − β) × C_P(V_i, V_j, n) + β × C_R(V_i, V_j)
10  begin
11      N ← # of cells ; n ← 1 ;
12      Min_l ← a list of N edges having the minimum C_T;
13      for each directed edge e(V_i, V_j) of Min_l
14        V_1st ← V_i, V_2nd ← V_j, V_last ← V_2nd;
15        while non-ordered V
16          cost_min ← ∞ ; n ← (n + 1) ;
17          for each non-ordered V_non
18            if (C_T(V_last, V_non, n) < cost_min)
19                cost_min ← C_T(V_last, V_non, n) ;
20                V_next ← V_non ;
21            endif
22          endfor
23          V_last ← V_next
24        endwhile
25      endfor
26  end
```

Fig. 13. The proposed algorithm for finding a Hamiltonian path optimizing power and routing overhead in PRORO.

Table XIV. Comparisons of Scan-Shift Transitions Generated by Different Scan-Cell Reordering Schemes

circuit	method	ROBPR	PRORO			[Cadence 2006]
			$\beta = 0.25$	$\beta = 0.5$	$\beta = 0.75$	reordering
s13207	total trans.	3,665,027	3,895,618	4,074,157	4,324,338	8,490,452
	normalized	1.00	1.06	1.11	1.18	2.32
s15850	total trans.	2,994,375	3,034,897	3,533,745	3,729,978	7,013,465
	normalized	1.00	1.02	1.19	1.25	2.36
s35932	total trans.	7,329,830	7,491,731	8,165,155	9,500,240	16,994,567
	normalized	1.00	1.02	1.12	1.30	2.32
s38417	total trans.	39,396,985	40,505,086	41,866,409	43,820,365	82,459,089
	normalized	1.00	1.03	1.07	1.12	2.10
s38584	total trans.	37,493,542	37,527,256	38,305,451	39,692,049	60,049,467
	normalized	1.00	1.00	1.02	1.06	1.60
b17	total trans.	63,096,447	64,846,104	68,542,710	84,217,531	295,180,622
	normalized	1.00	1.03	1.09	1.34	4.70
b20	total trans.	8,357,887	8,913,037	9,275,721	11,098,671	16,092,439
	normalized	1.00	1.07	1.11	1.33	1.93
b21	total trans.	7,887,930	9,169,859	9,383,991	10,140,868	17,417,163
	normalized	1.00	1.16	1.19	1.29	2.21
b22	total trans.	16,878,768	18,206,365	19,318,666	22,159,697	34,902,204
	normalized	1.00	1.08	1.15	1.32	2.07
avg. normalized		1.00	1.05	1.12	1.24	2.40

Table XV. Comparisons of Scan Path's Wire Length (μm) after Global Route Generated by Different Scan-Cell Reordering Schemes

circuit	method	ROBPR	PRORO			[Cadence 2006] reordering
			β = 0.25	β = 0.5	β = 0.75	
s13207	scan wire length	23,494	20,366	20,240	17,939	8,769
	normalized	2.68	2.32	2.31	2.05	1.00
s15850	scan wire length	20,628	17,235	16,787	15,017	8,204
	normalized	2.51	2.10	2.05	1.83	1.00
s35932	scan wire length	174,595	115,435	81,122	64,720	24,551
	normalized	7.11	4.70	3.30	2.64	1.00
s38417	scan wire length	65,372	57,655	49,622	47,316	22,605
	normalized	2.89	2.55	2.20	2.09	1.00
s38584	scan wire length	91,460	64,784	56,182	54,412	21,361
	normalized	4.28	3.03	2.63	2.55	1.00
b17	scan wire length	60,729	56,566	54,437	51,740	23,657
	normalized	2.57	2.39	2.30	2.19	1.00
b20	scan wire length	20,838	19,821	18,599	18,278	8,814
	normalized	2.36	2.25	2.11	2.07	1.00
b21	scan wire length	21,032	20,502	17,915	16,812	8,371
	normalized	2.51	2.45	2.14	2.01	1.00
b22	scan wire length	36,080	31,928	30,083	28,120	13,139
	normalized	2.75	2.43	2.29	2.14	1.00
avg. normalized		3.30	2.69	2.37	2.17	1.00

Table XVI. Comparisons of Total Wire Length (μm) after Detailed Route Generated by Different Scan-Cell Reordering Schemes

circuit	method	ROBPR	PRORO			[Cadence 2006] reordering
			β = 0.25	β = 0.5	β = 0.75	
s13207	total wire length	179,304	150,990	140,076	132,945	132,240
	normalized	1.35	1.14	1.06	1.01	1.00
s15850	total wire length	166,007	148,994	140,033	144,092	132,585
	normalized	1.25	1.12	1.06	1.09	1.00
s35932	total wire length	822,755	533,494	513,903	481,645	398,855
	normalized	2.06	1.34	1.29	1.21	1.00
s38417	total wire length	493,421	429,109	432,452	409,148	397,568
	normalized	1.24	1.08	1.09	1.03	1.00
s38584	total wire length	786,806	656,511	673,550	667,683	658,329
	normalized	1.20	1.00	1.02	1.01	1.00
b17	total wire length	1,269,029	1,245,434	1,239,651	1,255,198	1,245,979
	normalized	1.02	1.00	0.99	1.01	1.00
b20	total wire length	376,651	371,546	378,538	375,255	358,385
	normalized	1.05	1.04	1.06	1.05	1.00
b21	total wire length	397,848	380,094	375,095	380,892	364,586
	normalized	1.09	1.04	1.03	1.04	1.00
b22	total wire length	579,858	566,337	562,492	550,547	549,135
	normalized	1.06	1.03	1.02	1.00	1.00
avg. normalized		1.26	1.09	1.07	1.05	1.00

wire length of scan paths of each reordering scheme by dividing it with that of Cadence's [2006] reordering scheme, which is supposed to be the reordering scheme generating the shortest wire length of scan paths in this experiment. Table XVI further lists the total wire length (including the routing for both scan paths and CUT) generated by each reordering scheme after detailed route.

As the results show in Table XIV, if only minimizing the wire length of scan paths such as tool Cadence's [2006] reordering scheme, 2.4 times the scan-shift transitions of ROBPR are generated, where ROBPR only minimizes scan-shift transitions. On the other hand, ROBPR requires 3.3 times the wire length of scan paths of tool Cadence's [2006] reordering scheme as shown in Table XV. In fact, the wire length spent on CUT's routing is much more than the wire length spent on scan paths' routing. Thus, after detailed route, the total wire length of ROBPR is 1.26 times the total wire length of Cadence's [2006] reordering scheme as shown in Table XVI.

Also, the experimental results in Tables XIV, XV, and XVI show that the trade-off between scan-shift transitions and scan path's wire length can be controlled by PRORO with different optimization factors. Using a larger optimization factor, PRORO can reduce more wire length of scan paths but generate more scan-shift transitions. When the optimization factor equals 0.5, PRORO generates 12% more scan-shift transitions compared to ROBPR but only requires 7% total wire length after detailed route, which is an acceptable level of routing overhead as long as the design is not intensively routing-congested.

Another reason to sacrifice the wire length of scan paths for the scan-shift power is that the for advanced process technologies, the violation of hold-time constraints on scan paths occurs more often than the violation of setup-time constraints. Designers even intentionally increase the wire length of some scan paths to meet the hold-time constraint instead of applying a scan-cell reordering to reduce its wire length. Therefore, the motivation of reducing wire length on scan paths may not be as strong as that in the old process technologies.

9. CONCLUSIONS

In this article, we first presented a scan-cell reordering technique which can simultaneously reduce scan-shift transitions based on the response correlations and preserve don't-care bits in the test patterns for a later minimization of scan-in transitions using MT-fill (Section 4). Second, we considered both the response correlation and pattern correlations during the cell reordering process to further reduce the scan-in transitions generated by MT-fill (Section 6). Next, we utilized the inverse connection between scan cells to turn a low correlation into a high one and developed a corresponding scan-cell reordering scheme to consider those inverse correlations (Section 7). Last, we incorporated the routing overhead of scan paths into the cost function of our scan-cell reordering and hence the trade-off between scan path's routing overhead and the number of scan-shift transitions can be controlled by a user-specified factor. In addition, a postprocess pattern-reordering scheme was also proposed to minimize the in-between transitions (Section 5). A series of experiments were conducted to compare the proposed schemes with a previous reordering scheme [Bonhomme et al. 2002] and a commercial tool's reordering scheme [Cadence 2006]. The experimental results demonstrated the effectiveness and efficiency of each of the proposed scan-cell reordering schemes.

REFERENCES

BONHOMME, Y., GIRARD, P., GUILLER, L, LANDRAULT, C., PRAVOSSOUDOVITCH, S, AND VIRAZEL, A. 2004. Design of routing-constrained low power scan chains. In *Proceedings of the Conference on Design, Automation, and Test in Europe (DATE'04).* 62-67.

BONHOMME, Y., GIRARD, P., LANDRAULT, C., AND PRAVOSSOUDOVITCH, S. 2002. Power driven chaining of flip-flops in scan architectures. In *Proceedings of the IEEE International Test Conference.* 796-803.

BONHOMME, Y., GIRARD, P., GUILLER, L., LANDRAULT, C., AND PRAVOSSOOUDOVITCH, S. 2001. A gated clock scheme for low power scan testing of logic ics or embedded cores. In *Proceedings of the IEEE Asian Test Symposium.* 253-258.

BONHOMME, Y., GIRARD, P., GUILLER, L., LANDRAULT, C., AND PRAVOSSOUDOVITCH, S. 2003. Efficient scan chain design for power minimization during scan testing under routing constraint. In *Proceedings of the IEEE International Test Conference.* 488-493.

BUSHNELL, M. AND AGRAWAL, V. 2000. Essentials of Electronic Testing. Kluwer Academic Publishers.

CHANDRA, A., AND KAPUR, R. 2008. Bounded adjacent fill for low capture power scan testing. In *Proceedings of the IEEE VLSI Test Symposium.*

CHEN, X. AND HSIAO, M.-S. 2003. Energy-Efficient logic BIST based on state correlation analysis. In *Proceedings of the IEEE VLSI Test Symposium.* 267-272.

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.*, 1325-1333.

GIRARD, P. 2002. Survey of low-power testing of vlsi circuits. *IEEE Des. Test Comput. 19*, 3, 82-92.

HIRECH, M., BEAUSANG, J., AND GU, X. 1998. A new approach to scan chain reordering using physical design information. In *Proceedings of the IEEE International Test Conference.* 348-355.

HUANG, T. AND LEE, K.-J. 2001. A token scan architecture for low power testing. In *Proceedings of the IEEE International Test Conference.* 660-669.

LI, J., XU, Q., HU, Y., AND LI, X. 2008. iFill: An impact-oriented x-filling method for shift- and capture-power reduction in at-speed scan-based testing. In *Proceedings of the ACM/IEEE Conference on Design, Automation, and Test in Europe (DATE'08).* 1184-1189.

LIN, S. P., LEE, C. L., CHEN, J. E., CHEN, J.-J., LUO, K.-L., AND WU, W.-C. 2006. A multilayer data copy scheme for low cost test with controlled scan-in power for multiple scan chain designs. In *Proceedings of the IEEE International Test Conference.*

MAKAR, S. 1998. A layout-based approach for ordering scan chain flip-flops. In *Proceedings of the IEEE International Test Conference.* 341-347.

MRUGALSKI, G., RAJSKI, J., CZYSZ, D., AND TYSZER, J. 2007. New test data decompressor for low power applications. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'07).* 539-544.

REMERSARO, S., LIN, X., ZHANG, Z., REDDY, S. M., POMERANZ, I., AND RAJSKI, J. 2006. Preferred fill: A scalable method to reduce capture power for scan based designs. In *Proceedings of the IEEE International Test Conference.*

ROSINGER, P., AL-HASHIMI, B. M., AND NICOLICI, N. 2004. Scan architecture with mutually exclusive scan segment activation for shift and capture-power reduction. *IEEE Trans. Comput.-Aid. Des. 23*, 1142-1153.

SANKARALINGAM, R. AND TOUBA, N.-A. 2002. Controlling peak power during scan testing. In *Proceedings of the IEEE VLSI Test Symposium.* 153-159.

SANKARALINGAM, R., ORUGANTI, R. R., AND TOUBA, N. A. 2000. Static compaction techniques to control scan vector power dissipation. In *Proceedings of the IEEE VLSI Test Symposium.* 35-40.

SANKARALINGAM, R., POUYA, B., AND TOUBA, N. A. 2001. Reducing power dissipation during test using scan chain disable. In *Proceedings of the IEEE VLSI Test Symposium.* 319-324.

SAXENA, J., BUTLER, K. M., AND WHETSEL, L. 2001. An analysis of power reduction techniques in scan testing. In *Proceedings of the IEEE International Test Conference.* 670-677.

SINANOGLU, O. AND ORAILOGLU, A. 2003. Modeling scan chain modifications for scan-in test power minimization. In *Proceedings of the IEEE International Test Conference.*

SINANOGLU, O., BAYRAKTAROGLU, I., AND ORAILOGLU, A. 1998. Scan power reduction through test data transition frequency analysis. *IEEE Trans. Comput.-Aid. Des.,* 1325-1333.

WEN, X., YAMASHITA, Y., KAJIHARA, K., WANG, L.-T., SALUJA, K. K., AND KINOSHITA, K. 2005. On low-capture-power test generaion for scan testing. In *Proceedings of the IEEE VLSI Test Symposium.* 265-270.

WHETSEL, L. 2000. Adapting scan architectures for low power operation. In *Proceedings of the IEEE International Test Conference.* 863-872.

ZORIAN, Y. 1993. A distributed bist control scheme for complex vlsi devices. In *Proceedings of the IEEE VLSI Test Symposium.* 4-9.

Received February 2010; revised July 2010; accepted September 2010