

Scan-Chain Reordering for Minimizing Scan-Shift Power Based on Non-Specified Test Cubes

Yu-Ze Wu
 Dept. of Electronics Engineering
 National Chiao Tung Univ.
 Hsinchu, Taiwan
 jasonwu.ee94g@nctu.edu.tw

Mango C.-T. Chao
 Dept. of Electronics Engineering
 National Chiao Tung Univ.
 Hsinchu, Taiwan
 mango@faculty.nctu.edu.tw

Abstract

This paper proposes a scan-cell reordering scheme, named ROBPR, to reduce the signal transitions during 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 scheme utilizes both response correlation and pattern correlation to simultaneously minimize scan-out and scan-in transitions. A series of experiments demonstrate the effectiveness and superiority of the proposed scheme on reducing total scan-shift transitions. The trade-off between our power-driven scan-cell reordering and a routing-driven scan-cell reordering is discussed based on experiments as well.

1. Introduction

The scan design has been a widely used DFT technique which can guarantee high fault coverage for a complex design by enhancing its controllability and observability [1]. When using the scan design to shift test data, however, a large number of signal transitions may occur along the scan paths, which induces even more signal transitions on the circuit-under-test (CUT). Therefore, with the scan design, the CUT will consume much more power in its test mode than that in its functional mode [2]. 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 [3]. 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 the 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) the capture transition – generated by the same scan cell's value difference between the scan-in pattern and the corresponding captured response, (2) the scan-out transition – generated by two adjacent scan cells' value difference between their scan-out response, and (3) the scan-in transition – generated by two adjacent scan cells' value difference between the scan-in patterns. The first tran-

sition 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, complex ATPGs [4][5][6] 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 above 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 [7][8][9][10]. [7] proposed a don't-care-filling technique, named MT-fill, guaranteeing that the scan-in transitions generated by its filled patterns are minimized for the given test set. The methods in [7][8][9] reduced the test power as well as the test data volume based on build-in de-compression hardware. [10] added Xor gates or inverters along the scan paths to minimize the scan-in transitions. However, none of [7][8][9][10] considered the scan-out transitions simultaneously.

Another concept 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 [11][12][13][14][15][16]. 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 [11], central control unit for each group's clock signal [12][13], or specialized scan cells along with multiphase generator [15]. [16] further minimizes 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 [17][18][19][20] 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 [21][22]. One important reason in making this tradeoff is that, for advanced process technologies, the violation of hold-time constraints on scan paths occurs more often than the violation of setup-time constraints. Hence, the need of minimizing wire length for scan paths is not as urgent as that of mini-

mizing test power. However, the existing scan-chain-reordering techniques [17][18][19][20] need to obtain the exact test patterns and responses 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 [7][8][9][10].

In this paper, 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 [7]. 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). Next, 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 5). Last, a comparison between our power-driven scan-cell reordering and a routing-driven scan-cell reordering is provided based on experiments (Section 6). The experimental results demonstrate the effectiveness and the superiority of the proposed reordering scheme over a previous scan-cell reordering scheme [17].

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 [7]. In this paper, we use the number of signal transitions on scan cells as the power model of the whole CUT. The proposed scan-cell-reordering scheme focuses on reducing the total scan-shift power, i.e., reducing the total scan-shift transitions. The capture power is not considered in the proposed scheme.

From the discussions in Sec. 1, the scan-in transitions can be minimized by wisely filling the don't-care bits of a test set once the scan-cell order in the scan paths are given [7]. 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 [7]. However, before specifying the don't-care bits, the value of some responses may not be obtainable, implying that no explicit information of scan-out transitions can be used during the scan-cell reordering process.

We use a simple experiment (reported in Table 1) 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 the reordering scheme can 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.

In the experiment, we use a commercial tool [23] to generate stuck-at-fault patterns with don't-care bits. We then collect the statistic of the response correlation between any two scan cells by randomly filling the don't-care bits and simulating the corresponding responses for 1-million times. Table 1 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. 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 the 1452 scan cells in s38584. This experimental result indicates that, even without specifying the 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.

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 1. Response correlation of ISCAS benchmark s38584.

3. Problem Formulation

The problem of the scan-cell reordering for scan-shift power reduction is first defined 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 paper, the proposed scan-cell-reordering scheme only discuss 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) [7] to calculate the number of scan-in and scan-out transitions. 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. Equation 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 i th pattern, respectively.

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

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

In equation 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 j th 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 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.

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

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

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, 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

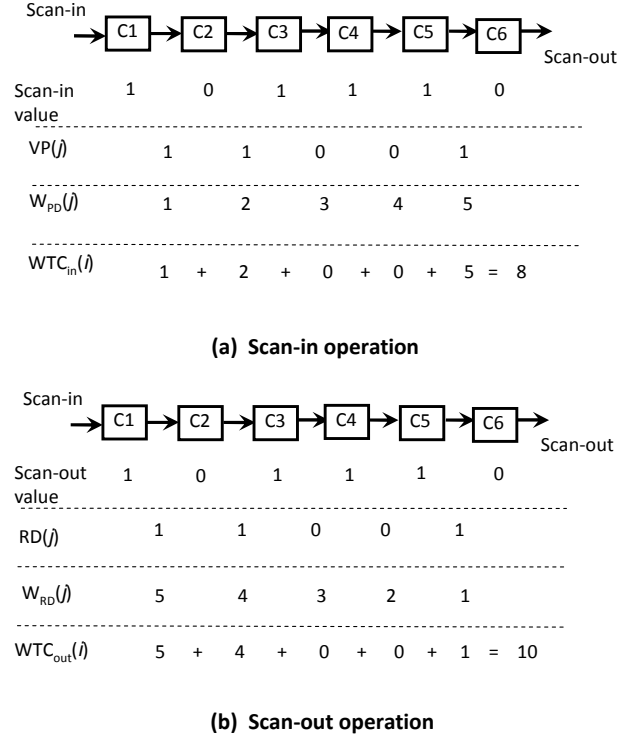


Figure 1. Calculation of scan-in and scan-out WTC.

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.

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

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 non-directed 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*

- 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

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

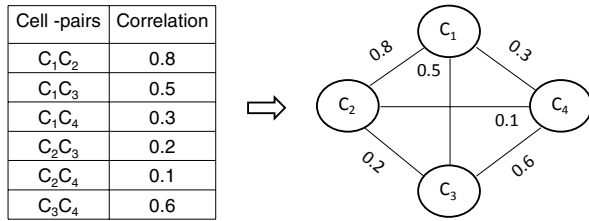


Figure 3. Construction of a response-correlation graph.

(TSP), which is NP-complete. We use a greedy TSP algorithm, which orders one vertex at a time to form the cycle. The selection criteria 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 of $Q(N^3)$.

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 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 Equation 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.

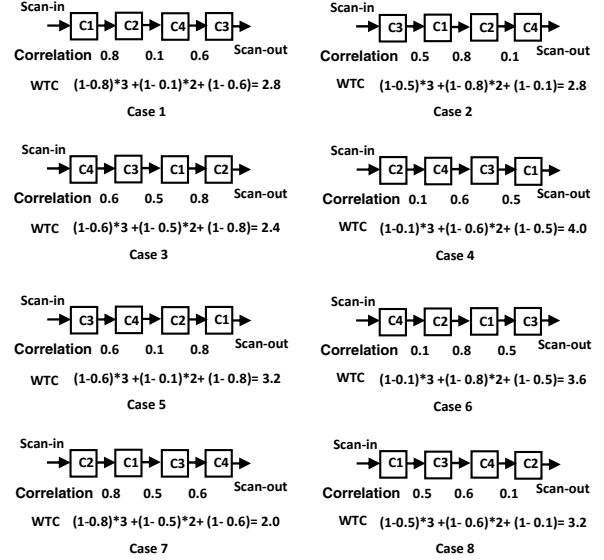


Figure 4. Estimated WTC_{out} of different scan-chain input/output.

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 [7] for more details of MT-fill.

4.2. Experimental Results

We conduct experiments on ten ISCAS and ITC benchmark circuits. Table 2 first shows the statistics of the benchmark circuits and their ATPG patterns generated by [23].

circuit	gate count	PI	PO	# of scan cell	# of patterns	don't-care -bits (%)	total faults	coverage (%)
s9234	5,597	36	39	211	141	69.43	9,920	100
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

Table 2. Statistics of the circuits and their ATPG patterns.

The following experiment compares RORC with another scan-cell reordering scheme presented in [17], 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 [17] as well. In the following experiment of [17], 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 [17] is applied to obtain the final scan-cell ordering based on the filled patterns. We repeat the above steps 100 times and report the best results for [17]. Also, we use the same TSP algorithm in both RORC and [17] to make a fair comparison.

In Table 3, 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 [17] with an average 44.29% and 45.80% 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 byproduct of the reduction to total scan-shift transitions. Note that the result reported for [17] 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.

RORC generates a lower number of total scan-shift transitions than [17] in all circuits but s35932. This exception may attribute 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 is lower. This finding further motivates us to develop a cell reordering scheme which can also consider the impact of a scan-cell ordering on the scan-in transitions generated by the MT-fill patterns.

5. Scan-cell Reordering Considering Both Response and Pattern Correlations

5.1. Detailed Steps of Reordering Scheme

- 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

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

RORC reduces scan-out transitions by minimizing the response correlations between adjacent cells. It ignores the impact of the

circuit	method	scan-in trans.	scan-out trans.	total trans.	peak trans.	runtime (sec)
s9234	[17]	633,488	623,480	1,256,968	102	100
	RORC	318,071	545,512	863,583	86	7
	improv.	49.79 %	12.51 %	31.30 %	15.69%	-
s13207	[17]	3,951,373	4,188,819	8,140,192	289	400
	RORC	1,312,934	2,847,104	4,160,038	233	45
	improv.	66.77 %	32.03 %	48.90 %	19.38%	-
s15850	[17]	2,800,025	4,904,948	7,704,973	277	300
	RORC	1,497,065	2,157,662	3,654,727	211	49
	improv.	46.53 %	56.01 %	52.57 %	23.83%	-
s35932	[17]	4,543,209	4,934,478	9,477,687	525	3,000
	RORC	5,388,270	4,363,125	9,751,395	680	120
	improv.	-18.60 %	11.58 %	-2.89 %	-29.52%	-
s38417	[17]	29,942,845	58,416,311	88,359,156	713	4,100
	RORC	11,453,864	27,547,170	39,001,034	529	666
	improv.	61.75 %	52.84 %	55.86 %	25.81%	-
s38584	[17]	22,827,002	41,743,137	64,570,139	714	3,100
	RORC	12,489,481	27,615,042	40,104,523	694	616
	improv.	45.29 %	33.85 %	37.89 %	2.80%	-
b17	[17]	95,302,661	230,963,547	326,266,208	700	6,200
	RORC	24,619,742	41,550,664	66,170,406	570	3,760
	improv.	74.17 %	82.01 %	79.72 %	18.57%	-
b20	[17]	7,680,415	12,332,467	20,012,882	237	500
	RORC	4,823,088	4,662,118	9,485,206	171	160
	improv.	37.20 %	62.20 %	52.60 %	27.85%	-
b21	[17]	7,351,208	11,834,023	19,185,231	229	600
	RORC	4,546,521	4,590,188	9,136,709	205	177
	improv.	38.15 %	61.21 %	52.38 %	10.48%	-
b22	[17]	17,200,814	23,447,118	40,647,932	362	1,200
	RORC	9,997,996	10,844,186	20,842,182	276	587
	improv.	41.87 %	53.75 %	48.73 %	23.76%	-
Ave. improv.		44.29 %	45.80 %	45.70%	13.86%	-

Table 3. Comparisons of generated scan-shift transitions between RORC and [17].

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. 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.

5.1.1 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 4 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 are 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 de-

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

Table 4. Different cases of pattern correlations between two adjacent cells.

terminated 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.

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 the Sec. 4.1.1 to estimate them.

5.1.2 Construct the Directed Correlation Graph

The correlation graph constructed in ROBPR is a revised version of the correlation graph in Sec. 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.

5.1.3 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.

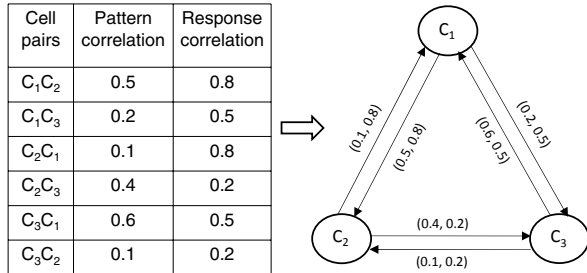


Figure 6. Construction of the directed graph based on pattern and response correlations.

```

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  $\overline{W_p}(V_i, V_j) : 1 - W_p(V_i, V_j)$ 
5  $\overline{W_r}(V_i, V_j) : 1 - W_r(V_i, V_j)$ 
6  $Cost(V_i, V_j, n) : \overline{W_p}(V_i, V_j) \times n + \overline{W_r}(V_i, V_j) \times (N - 1 - n)$ 
7 begin
8  $N \leftarrow \#$  of cells ;  $n \leftarrow 1$  ;
9  $MinL \leftarrow$  a list of  $N$  edges having the minimum  $(\overline{W_p} + \overline{W_r} \times (N-1))$ ;
10 for each directed edge  $e(V_i, V_j)$  of  $MinL$ 
11  $V_{1st} \leftarrow V_i, V_{2nd} \leftarrow V_j, V_{last} \leftarrow V_{2nd}$ ;
12 while non-ordered  $V$ 
13  $cost_{min} \leftarrow \infty$  ;  $n \leftarrow (n + 1)$  ;
14 for each non-ordered  $V_{non}$ 
15 if  $(Cost(V_{last}, V_{non}, n) < cost_{min})$ 
16  $cost_{min} \leftarrow Cost(V_{last}, V_{non}, n)$  ;
17  $V_{next} \leftarrow V_{non}$  ;
18 endif
19 endfor
20  $V_{last} \leftarrow V_{next}$ 
21 endwhile
22 endfor
23 end

```

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

When adding the n th non-ordered 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 Equation 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 equation 1. The $N - 1 - n$ in the cost function actually represents the $W_{RD}(n)$ described in the WTC equation 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 Equations 1 and 2.

5.2. Experimental Results

We conduct experiments for ROBPR on the same benchmark circuits and test patterns as in Sec. 4.2. Table 5 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 32.97% less scan-in transitions but only 3.82% 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 in ROBPR. It also shows the effectiveness of the pattern-correlation estimation listed in Table 4.

The average reduction to the total scan-shift transitions is 12.52% by ROBPR. The 8.52% reduction to the number of peak transitions is a byproduct 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 ROBPR (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 Step 4 in RORC.

circuit	method	scan-in trans.	scan-out trans.	total trans.	peak trans.	runtime (sec)
s9234	RORC	318,071	545,512	863,583	86	7
	ROBPR	239,064	507,843	746,907	75	7
	improv.	24.84 %	6.91%	13.51 %	12.79%	-
s13207	RORC	1,312,934	2,847,104	4,160,038	233	45
	ROBPR	882,926	2,780,763	3,663,689	168	45
	improv.	32.75 %	2.33 %	11.93 %	27.90%	-
s15850	RORC	1,497,065	2,157,662	3,654,727	211	49
	ROBPR	1,029,107	1,944,970	2,974,077	179	49
	improv.	31.26 %	9.86%	18.62 %	15.17%	-
s35932	RORC	5,388,270	4,363,125	9,751,395	680	120
	ROBPR	1,963,178	5,356,284	7,319,462	641	145
	improv.	63.57 %	-22.76%	24.94 %	5.74%	-
s38417	RORC	11,453,864	27,547,170	39,001,034	529	666
	ROBPR	9,599,399	29,676,522	39,275,921	521	667
	improv.	16.19 %	-7.73%	-0.70 %	1.51%	-
s38584	RORC	12,489,481	27,615,042	40,104,523	694	616
	ROBPR	10,064,216	27,385,766	37,449,982	580	618
	improv.	19.42%	0.83%	6.62 %	16.43%	-
b17	RORC	24,619,742	41,550,664	66,170,406	570	3,760
	ROBPR	16,202,102	46,655,210	62,857,312	563	3,765
	improv.	34.19 %	-12.29 %	5.01 %	1.23%	-
b20	RORC	4,823,088	4,662,118	9,485,206	171	160
	ROBPR	3,491,947	4,835,560	8,327,507	181	162
	improv.	27.60 %	-3.72 %	12.21 %	-5.85%	-
b21	RORC	4,546,521	4,590,188	9,136,709	205	177
	ROBPR	2,914,102	4,960,108	7,874,210	195	179
	improv.	35.90 %	-8.06 %	13.82 %	4.88%	-
b22	RORC	9,997,996	10,844,186	20,842,182	276	587
	ROBPR	5,603,864	11,233,009	16,836,873	261	588
	improv.	43.95 %	-3.59%	19.22 %	5.43%	-
Ave. improv.		32.97 %	-3.82 %	12.52 %	8.52%	-

Table 5. Comparisons of generated scan-shift transitions between RORC and ROBPR

6. Comparison Between Power-Driven Reordering and Routing-Driven Reordering

Although the average and peak testing power can be reduced, a major concern of the proposed scan-cell reordering scheme is its potential overhead in the total wire length. The scan-cell-reordering technique can be applied not only for the testing-power reduction but also for the wire-length minimization. Most current back-end tools support the option of the scan-cell reordering for wire-length minimization after placement. In this section, we compare our power-driven scan-cell reordering, ROBPR, with a routing-driven scan-cell reordering provided by a commercial tool [24].

The following experiment uses a TSMC 0.18 μ m CMOS technology with 5 metal layers. For the experimental results reported for ROBPR, we first obtained the scan-cell ordering by ROBPR and apply the APR tool in [24] to get its placement. For the experimental results reported for [24], we start from the same ROBPR's placement and apply the command "scanreorder" in [24] to get a routing-driven scan-cell reordering. In Table 6, Columns 3 and

4 list the total number and the peak number of scan-shift transitions, respectively, based on the scan-cell ordering of each scheme. Columns 5 and 6 list the total wire length and the wire length of all scan paths estimated by [24] based on the corresponding placement (manhattan distance).

circuit	method	total trans.	peak trans.	total wire(um)	chain wire(um)
s9234	Tool	1,363,689	113	4.03e+04	2,822
	ROBPR	746,907	75	4.44e+04	6,930
	improv.	45.23%	33.63%	-10.17 %	-145.57%
s13207	Tool	8,489,114	305	9.23e+04	8,769
	ROBPR	3,663,689	168	1.07e+05	23,494
	improv.	56.84%	44.92%	-15.93 %	-167.92%
s15850	Tool	6,993,167	274	9.37e+04	8,204
	ROBPR	2,974,077	179	1.06e+05	20,628
	improv.	57.47%	34.67%	-13.23 %	-151.44%
s35932	Tool	16,984,199	886	3.29e+05	24,551
	ROBPR	7,319,462	641	4.78e+05	174,595
	improv.	56.90%	27.65%	-45.29 %	-611.15 %
s38417	Tool	82,338,025	737	2.61e+05	22,605
	ROBPR	39,275,291	521	3.03e+05	65,372
	improv.	52.30%	29.31%	-16.09 %	-189.19%
s38584	Tool	60,005,907	727	4.57e+05	21,631
	ROBPR	37,449,982	580	5.26e+05	91,460
	improv.	37.59%	20.22%	-15.10 %	-322.82%
b17	Tool	294,941,487	658	9.01e+05	23,657
	ROBPR	62,857,312	563	9.38e+05	60,688
	improv.	78.69%	14.44%	-4.11 %	-156.53%
b20	Tool	16,062,059	211	2.68e+05	8,814
	ROBPR	8,327,507	181	2.79e+05	20,836
	improv.	48.15%	14.22%	-4.10 %	-136.40%
b21	Tool	174,034,433	238	2.81e+05	8,371
	ROBPR	7,874,210	195	2.93e+05	21,012
	improv.	95.48%	18.07%	-4.27 %	-151.01%
b22	Tool	34,860,309	321	4.23e+05	13,139
	ROBPR	16,836,873	261	4.45e+05	36,099
	improv.	51.70%	18.69%	-5.20 %	-174.75%
ave. improv.		58.04%	25.58%	-13.35 %	-220.68 %

Table 6. Comparisons of scan-shift transitions and estimated wire length after placement.

As the average results show, ROBPR can generate 58.04% less scan-shift transitions and 25.88% less peak transitions, compared to [24]. Also, ROBPR leads to a 13.35% higher estimated total wire length and a 220.68% higher estimated wire length of scan paths, compared to [24]. The reduction of the total wire length by [24] is mainly contributed from the reduction of the scan-chain wire length. However, for advanced process technologies, the violation of hold-time constraints occurs much more often than the violation of setup-time constraints on scan paths. 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.

Table 7 lists the final total wire length, the final wire length of scan paths, the number of vias in use, and the allocated area after the detail routing [24] is performed. As the results show, the average reductions to the total wire length, the scan-chain wire length, and the number of via by [24] are 8.24%, 221.24%, and 1.46%, respectively. While the reduction percentage of scan-chain wire length matches the estimated result after placement, the reduction percentage of the total wire length is significantly smaller than its estimated result. It implies that the benefit of a routing-

driven scan-cell reordering may be diluted after other back-end optimization steps are performed. However, the reduction to scan-shift transitions caused by ROBPR remains the same as long as the scan-cell ordering is kept, which is another advantage of using a power-driven scan-cell reordering

circuit	method	total wire(um)	chain wire(um)	via	area (um ²)
s9234	Tool	54,446	2,822	11,508	40,848
	ROBPR	57,394	6,930	11,547	40,848
	improv.	-5.41 %	-145.57 %	-0.34 %	-
s13207	Tool	132,928	8,769	24,706	102,057
	ROBPR	144,584	23,494	25,150	102,057
	improv.	-8.77 %	-167.92 %	-1.80%	-
s15850	Tool	132,701	8,204	26,895	102,476
	ROBPR	142,679	20,628	27,113	102,476
	improv.	-7.52 %	-151.44 %	-0.81 %	-
s35932	Tool	483,967	24,551	71,477	297,460
	ROBPR	630,677	174,595	76,956	297,460
	improv.	-30.31 %	-611.15 %	-7.67 %	-
s38417	Tool	371,595	22,605	73,538	281,164
	ROBPR	407,000	65,372	74,385	281,164
	improv.	-9.53 %	-189.19 %	-1.15%	-
s38584	Tool	619,326	21,361	90,599	284,657
	ROBPR	680,902	91,460	91,886	284,657
	improv.	-9.94 %	-328.16 %	-1.42%	-
b17	Tool	1,212,971	23,657	180,498	442,597
	ROBPR	1,241,091	60,729	181,529	442,597
	improv.	-2.32 %	-156.71 %	-0.57%	-
b20	Tool	354,303	8,814	62,276	164,191
	ROBPR	363,820	20,838	62,197	164,191
	improv.	-2.69 %	-136.42 %	0.13%	-
b21	Tool	365,384	8,371	62,504	165,522
	ROBPR	375,365	21,032	62,685	165,522
	improv.	-2.73 %	-151.25 %	-0.29 %	-
b22	Tool	563,999	13,139	92,401	246,446
	ROBPR	581,989	36,080	93,070	246,446
	improv.	-3.19 %	-174.60 %	-0.72%	-
ave. improv.		-8.24%	-221.24%	-1.46%	-

Table 7. Comparisons of scan-shift transitions and estimated wire length after detail routing.

7. Conclusions

This paper first presents a scan-cell reordering scheme which connects the scan cells with a high response correlation to reduce scan-out transitions. This reordering scheme preserves the don't-care bits during the ordering process so that a post pattern-filling technique can be applied to minimize the scan-in transitions. This paper further adds the pattern correlations into consideration and reduce even more scan-shift transitions. A set of experiments are conducted to demonstrated the effectiveness of each technique proposed in this paper. A comparison to [17] also confirms the superiority of the proposed scheme by an average 45.7% reduction to the scan-shift transitions.

References

[1] M. Bushnell and V. Agrawal. *Essentials of Electronic Testing*. Kluwer Academic Publishers, 2000.
 [2] Y. Zorian. A distributed bist control scheme for complex vlsi devices. *Proc. 11th IEEE VTS*, pages 4–9, January 1993.

[3] P. Girard. Survey of low-power testing of vlsi circuits. *IEEE Design and Test of Computers*, 19(No 3):82–92, May-June 2002.
 [4] S. Remersaro and X. L. et al. Preferred fill a scalable method to reduce capture power for scan based designs. *Proc. Int. Test Conf.*, 2006.
 [5] X. Wen and Y. Y. et al. On low-capture-power test generaion for scan testing. *Proc. VTS*, pages 265–270, 2005.
 [6] R. Sankaralingam and N. A. Toubia. Controlling peak power during scan testing. *Proc. VTS*, pages 153–159, 2002.
 [7] R. Sankaralingam and R. R. O. et al. Static compaction techniques to control scan vector power dissipation. *Proc. VTS*, pages 35–40, 2000.
 [8] G. Mrugalski and J. R. et al. New test data decompressor for low power applications. *Proc. DAC*, pages 539–544, 2007.
 [9] S. Lin and C. L. et al. A multilayer data copy scheme for low cost test with controlled scan-in power for multiple scan chain designs. *Int. Test Conf.*, 2006.
 [10] O. Sinanoglu and A. Orailoglu. Modeling scan chain modifications for scan-in test power minimization. *Int. Test Conf.*, 2003.
 [11] Y. B. et al. A gated clock scheme for low power scan testing of logic ics or embedded cores. *Proc. 10th Asian Test Symp. (ATS01)*, pages 253–258, 2001.
 [12] P. Rosinger and B. A.-H. et al. Scan architecture with mutually exclusive scan segment activation for shiftand capture-powerreduction. *IEEE Transactions on Computer-Aided Design (TCAD)*, 23(7):1142–1153, July 2004.
 [13] L. Whetsel. Adapting scan architectures for low power operation. *Proc. IEEE Int. Test Conf.*, pages 863–872, 2000.
 [14] J. Saxena and K. M. B. et al. An analysis of power reduction techniques in scan testing. *in Proc. IEEE Int. Test Conf.*, pages 670–677, 2001.
 [15] T. Huang and K.-J. Lee. A token scan architecture for low power testing. *Proc. Int. Test Conf.*, pages 660–669, 2001.
 [16] R. Sankaralingam and B. P. et al. Reducing power dissipation during test using scan chain disable. *Proc. 19th VLSI Test Symp. (VTS 01)*, pages 319–324, 2001.
 [17] Y. Bonhomme and P. et al. Power driven chaining of flip-flops in scan architectures. *Int. Test Conf.*, pages 796–803, 2002.
 [18] V. Dabholkar and S. C. et al. Techniques for reducing power dissipation during test application in full scan circuits. *IEEE Transactions on CAD*, December 1998.
 [19] Y. Bonhomme and P. G. et al. Efficient scan chain design for power minimization during scan testing under routing constraint. *IEEE Int. Test Conf.*, pages 488–493, 2003.
 [20] Y. Bonhomme and P. G. et al. Design of routing-constrained low power scan chains. *Proc. DATE*, pages 62–67, 2004.
 [21] S. Makar. A layout-based approach for ordering scan chain flip-flops. *Proc. of the IEEE Int. Test Conf.*, pages 341–347, 1998.
 [22] M. Hirech and J. B. et al. A new approach to scan chain reordering using physical design information. *IEEE Int. Test Conf.*, pages 348–355, 1998.
 [23] Synopsys tetramax atpg user guide version x-2005.09.
 [24] Cadence encounter product version 5.2.3 june 2006.