

# 行政院國家科學委員會專題研究計畫 成果報告

## 利用重新排列掃描元件減低掃描位移功率 研究成果報告(精簡版)

計畫類別：個別型  
計畫編號：NSC 97-2221-E-009-173-  
執行期間：97年08月01日至98年07月31日  
執行單位：國立交通大學電子工程學系及電子研究所

計畫主持人：趙家佐

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中華民國 98 年 10 月 20 日



針對含有未知值的測試向量，使用掃描單元重新排列最小化掃描平移功率

“Scan-Cell Reordering for Minimizing Scan-Shift Power Based on Non-Specified Test Cubes”

計畫編號：NSC97-2221-E-009-173

執行期間：98年8月1日至99年7月31日

主持人：趙家佐 交通大學電子工程系助理教授

## 一、中文摘要

關鍵詞：測試化設計；低功率可掃描式測試；重新排列掃描元件；

掃描設計已是被廣泛使用的可測試性設計技術，它能增加複雜電路的控制能力 (controllability) 與觀察能力 (observability)，使得電路在測試時可得到高度的錯誤涵蓋率 (fault coverage)。有了掃描設計，被測電路 (circuit under test) 在測試模式下會比在功能模式下產生更多的信號轉變 (signal transitions)。大量的信號轉變會導致過高的功率消耗，進而會造成電路實體上的損壞，或者是被測試電路的可靠度降低，這些現象會降低電路的良率並縮短產品的生命週期。

在這個計畫中，我們將發展一個重新排列掃描元件的技術 (scan-cell reordering)，來減少在掃描移動時所產生的信號轉變。不像之前的重新排列掃描元件技術，需要先指定測試集中不在乎位元的值，我們提出的方案能在保留不在乎位元的前提下，重新排列掃描元件來減少信號轉變。在完成重新排列掃描元件後，我們利用測試圖案填值技術 (pattern-filling technique) 來更進一步減低功率消耗。為了達到這個目標，且讓這個方案可實行，我們所提之技術須擁有下面四項功能：(1) 在未指定測試圖案 (unspecified test patterns) 的狀況下，發現並利用掃描元件間，測試反應的關係性 (response correlation)；(2) 在未指定測試圖案 (unspecified test patterns) 的狀況下，發現並利用掃描元件間，測試圖案的關係性 (response correlation)；(3) 使用掃描資料反轉來增加測試反應和測試圖案的關係性；(4) 考慮並限制重新排列掃描元件所產生的繞線資源。最後，我們會透過實驗，將所提出的重新排列掃描元件技術，與現有論文中的重新排列掃描元件技術做比較。

## 英文摘要

Key words: design-for-test, low-power scan-based testing, scan-cell reordering

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. With the scan design, however, the CUT generates a much larger number of signal transitions in its test mode than that in its functional mode. This excessive power consumption caused by the signal transitions may result in physical damage or reliability degradation to the CUT, and in turn decreases the yield and product lifetime.

In this project, we will develop a scan-cell reordering scheme to minimize the signal transitions during the scan shift. Unlike the previous scan-cell reordering techniques which need to specify all the don't-care bits in the test cubes before reordering, the proposed scheme can preserve the don't-care bits while reordering the scan cells for signal-transition minimization. Those preserved don't-cares can then be utilized by a pattern-filling technique for a further power optimization. In order to achieve this goal and make the proposed scheme practical, the proposed cell-reordering scheme are designed to have the following four capabilities: (1) Identify and utilize the response correlations between scan cells based on unspecified test patterns; (2) Identify and utilize the pattern correlations between scan cells based on unspecified test patterns; (3) Increase both response and pattern correlations using scan-value inversion; and (4) Consider and limit the routing-resource overhead caused by scan-cell reordering. Its experimental results will be compared with published scan-cell reordering schemes as well.

## 二、計畫緣由、目的、研究方法與實驗結果

### 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 [1]. However, with the scan design, the circuit-under-test (CUT) consumes much more power in its test mode than that in its functional mode [2] due to the following reasons. First, when 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 have 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 [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 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 [4][5][6][7] 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 [8][9][10][11]. [8] 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 [8][9][10] reduced the test power as well as the test data volume based on build-in decompression hardware. [11] Added Xor gates or inverters along the scan paths to minimize the scan-in transitions. However, none of [8][9][10][11] considered the scan-out transitions simultaneously.

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 [12][13][14][15][16][17]. 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 [12], central control unit for each group's clock signal [13][14], or specialized scan cells along with multiphase generator [16]. [17] 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 [18][19] 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 [22][23]. Methods in [20][21] further consider the routing overhead during the reordering process such that the imposed routing overhead can be limited. However, one serious disadvantage of the existing scan-chain-reordering techniques [18][19][20][21] 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 [8][9][10][11].

In this project, 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 [8]. 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 guidance to the reordering process. 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. 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. Last, we consider the routing overhead of scan paths during the scan-cell reordering process, and thus the tradeoff between scan-shift power and routing overhead can be properly controlled. 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. 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 [8]. In this project, 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, i.e., 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 I, 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 [8]. 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 [8]. 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.

In the experiment, we use a commercial tool [24] 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 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.

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 I  
RESPONSE CORRELATION OF ISCAS BENCHMARK S38584.

Correlation	# of cell pairs	Distribution (%)	Correlation	# of cell pairs	Distribution (%)
0.95 - 1	52	0.052	0.45 - 0.50	19016	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

TABLE II  
RESPONSE CORRELATION OF ITC BENCHMARK B17.

### 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 project, 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) [8] 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. 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.

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

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

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-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.

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

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

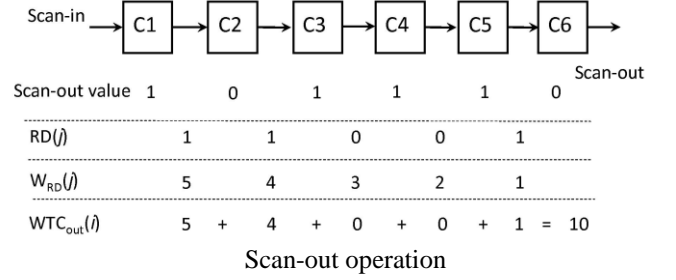
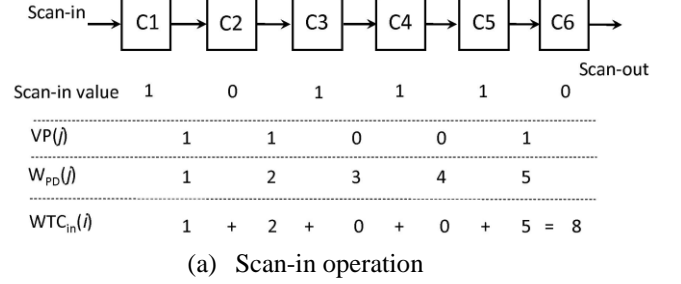


Fig. 1 Calculation of scan-in and scan-out WTC.

#### 4. SCAN-CELL REORDERING CONSIDERING ONLY RESPONSE CORRELATION

##### A. 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.

- 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.

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 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.

$$Simulation\_Times = (G\_Counts / 50) * P\_Counts, \quad (4)$$

where  $G\_Counts$  and  $P\_Counts$  denote the circuit gate count and the number of given test patterns, respectively.

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.

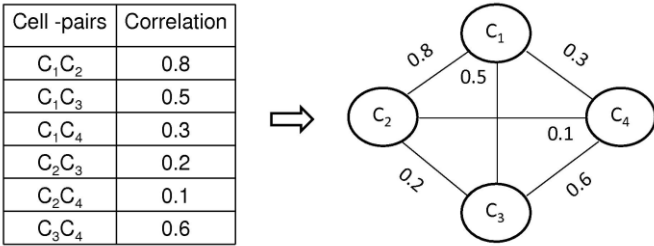


Fig. 3 Construction of a response-correlation graph.

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 criteria for the new ordered vertex are 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  $O(N^3)$ .

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. 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  $C1-C2-C4-C3-C1$ . 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  $C2-C1-C3-C4$ .

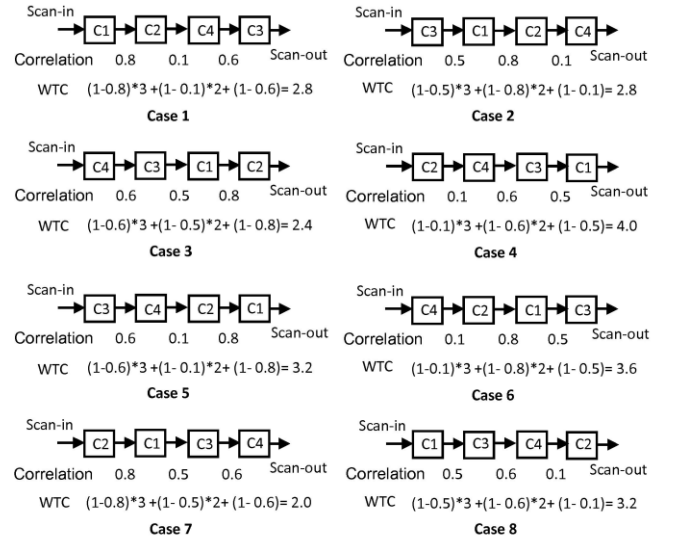


Fig. 4 Estimated  $WTC_{out}$  of different scan-chain input/out.

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 [8] for more details of MT-fill.



## B. 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 [24].

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

TABLE III  
STATISTICS OF THE BENCHMARK CIRCUITS AND THEIR ATPG PATTERNS.

The following experiment compares RORC with another scan-cell reordering scheme presented in [18], 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 [18] as well. In the following experiment of [18], 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 [18] 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 [18]. Also, we use the same TSP algorithm in both RORC and [18] 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 [18] 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 byproduct of the reduction to total scan-shift transitions. Note that the result reported for [18] 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 [18] for multiple times.

The runtime of [18] reported in Table IV is the runtime for 100 trials but the runtime of RORC is for only one trial. Thus, the runtime for one RORC is actually longer than the runtime for one [18]. Table V reports RORC's runtime distribution 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.

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 the scan-in and scan-out transition. It can be further reduced by a pattern-reordering scheme proposed in the next section.

circuit	method	scan-in trans.	scan-out trans.	total trans.	peak trans.	runtime (sec)
s13207	[17]	3,951,373	4,188,819	8,173,642	289	400
	RORC	1,312,934	2,847,104	4,204,192	233	35
	improv.	66.77 %	32.03 %	48.56 %	19.38%	-
s15850	[17]	2,800,025	4,904,948	7,736,017	277	300
	RORC	1,497,065	2,157,662	3,685,771	211	36
	improv.	46.53 %	56.01%	52.36 %	23.83%	-
s35932	[17]	4,543,209	4,934,478	9,524,285	525	3,000
	RORC	5,388,270	4,363,125	9,772,131	680	107
	improv.	-18.60 %	11.58 %	-2.60 %	-29.52%	-
s38417	[17]	29,942,845	58,416,311	88,478,584	713	4,100
	RORC	11,453,864	27,547,170	39,127,006	529	601
	improv.	61.75 %	52.84 %	55.78 %	25.81%	-
s38584	[17]	22,827,002	41,743,137	64,667,423	714	3,100
	RORC	12,489,481	27,615,042	40,223,587	694	543
	improv.	45.29 %	33.85%	37.80 %	2.80%	-
b17	[17]	95,302,661	230,963,547	326,795,418	700	6,200
	RORC	24,619,742	41,550,664	66,500,101	570	3,611
	improv.	74.17 %	82.01%	79.65 %	18.57%	-
b20	[17]	7,680,415	12,332,467	20,133,912	237	500
	RORC	4,823,088	4,662,118	9,623,386	171	138
	improv.	37.20 %	62.20 %	52.20 %	27.85%	-
b21	[17]	7,351,208	11,834,023	19,330,271	229	600
	RORC	4,546,521	4,590,188	9,266,069	205	476
	improv.	38.15 %	61.21%	52.068 %	10.48%	-
b22	[17]	17,200,814	23,447,118	40,809,632	362	1,200
	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%	-

TABLE IV  
COMPARISONS OF GENERATED SCAN-SHIFT TRANSITIONS BETWEEN RORC AND [18].

circuit	correlation sampling (a)	TSP	others	Total (b)	(a) over (b)
s9234	4	0	1	5	0.80
s13207	31	1	3	35	0.89
s15850	33	1	2	36	0.92
s35932	82	15	10	107	0.77
s38417	562	11	28	601	0.94
s38584	512	10	21	543	0.94
b17	3,505	7	99	3,611	0.97
b20	126	0	12	138	0.91
b21	454	1	21	476	0.95
b22	141	0	13	154	0.92
avg.					0.90

TABLE V  
RUNTIME DISTRIBUTION OF RORC (IN SECONDS).

## 5. PATTERN REORDERING FOR MINIMIZING IN-BETWEEN TRANSITIONS

### A. Detailed Steps of Pattern Reordering Scheme

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.

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

TABLE VI  
CLASSIFICATION OF PATTERNS.

Next, we denote the  $A_i$  as the  $i_{th}$  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:

$$A \sim A_w B_1 C_1 B_2 C_2 \sim B_y C_y B_{y+1} \sim B_x D_1 \sim D_z \quad (5)$$

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

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

The above 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 above 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.

Table VII compare 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).

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  
COMPARISON ON IN-BETWEEN TRANSITIONS BETWEEN USING AND WITHOUT USING PATTERN REORDERING.

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 RESPONSE AND PATTERN CORRELATIONS

As the results shown in Table IV, RORC generates a lower number of total scan-shift transitions than [18] 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 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.

### A. 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.

- 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.

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  $PC_k(i, j)$ ).

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 VIII  
DIFFERENT CASES OF PATTERN CORRELATIONS BETWEEN TWO ADJACENT CELLS.

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 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.

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. IV-A1 to estimate them.

2) Construct the Directed Correlation Graph: The correlation graph constructed in ROBPR is a revised version of the correlation graph. 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.

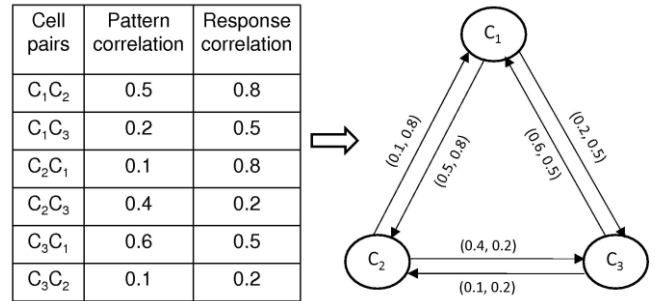


Fig. 6 Construction of the directed graph based on pattern and response correlations.

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  $n_{th}$  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  $\text{Cost}(V_i, V_j, n)$  in Figure 7, the  $W_p(V_i, V_j)$  ( $W_r(V_i, V_j)$ ) 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 WPD( $n$ ) described in the WTC equation 1. The  $N - 1 - n$  in the cost function actually represents the WRD( $n$ ) described in the WTC equation 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  $\overline{W_p}(V_i, V_j) : 1 - W_p(V_i, V_j)$ 
5  $\overline{W_r}(V_i, V_j) : 1 - W_r(V_i, V_j)$ 
6  $\text{Cost}(V_i, V_j, n) : \overline{W_p}(V_i, V_j) \times n + \overline{W_r}(V_i, V_j) \times (N - 1 - n)$ 
7 begin
8  $N \leftarrow \# \text{ of cells} ; n \leftarrow 1 ;$ 
9  $\text{Min}_l \leftarrow \text{a list of } N \text{ edges having the minimum } (\overline{W_p} + \overline{W_r}) \times (N-1);$ 
10 for each directed edge  $e(V_i, V_j)$  of  $\text{Min}_l$ 
11  $V_{1st} \leftarrow V_i, V_{2nd} \leftarrow V_j, V_{last} \leftarrow V_{2nd};$ 
12 while non-ordered  $V$ 
13  $\text{cost}_{min} \leftarrow \infty ; n \leftarrow (n + 1) ;$ 
14 for each non-ordered  $V_{non}$ 
15 if  $(\text{Cost}(V_{last}, V_{non}, n) < \text{cost}_{min})$ 
16  $\text{cost}_{min} \leftarrow \text{Cost}(V_{last}, V_{non}, n) ;$ 
17  $V_{next} \leftarrow V_{non} ;$ 
18 endif
19 endfor
20  $V_{last} \leftarrow V_{next}$ 
21 endwhile
22 endfor
23 end

```

Fig. 7 The proposed algorithm for finding a Hamilton path with minimal  $\text{WTC}_{\text{total}}$ .

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.

## B. Experimental Results

We conduct experiments for ROBPR on the same benchmark circuits and test patterns as in Sec. IV-B. 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 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 Step 3 (Figure 7) can directly find the Hamiltonian path with minimal  $\text{WTC}_{\text{total}}$ , saving a step of breaking a Hamilton cycle to obtain the final ordering, such as the Step 4 in RORC.

## 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.

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.

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.	31.26 %	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%	-

TABLE IX  
COMPARISONS OF GENERATED SCAN-SHIFT TRANSITION BETWEEN RORC AND ROBPR

### A. Detailed Steps of Reordering Scheme

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

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  $Q$  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.

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 X shows the inverse pattern correlation for different combinations of pattern values between cell  $i$  and cell  $j$  after MT-fill. The derivation of Table X 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.

<b>Step 1:</b> Collect the inverse and non-inverse pattern and response correlations.
<b>Step 2:</b> Construct a directed multiple-weight graph based on the collected inverse and non-inverse pattern and response correlations.
<b>Step 3:</b> Find the Hamiltonian path with the minimum WTC.
<b>Step 4:</b> Determine the Scan-in Patterns Based on Derived Cell Ordering.

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

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

TABLE X  
DIFFERENT CASES OF INVERSE PATTERN CORRELATIONS BETWEEN TWO ADJACENT CELLS.

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: non-inverse set ( $W_p, W_r$ ) and inverse set ( $I W_p, I W_r$ ), where  $W_p$  and  $W_r$  represent the direct pattern correlation and response correlation as calculated in ROBPR, and  $I W_p$  and  $I W_r$  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 non-inverse correlation sets between three scan cells.

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  $n$ th 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  $Cost_{inv}(V_i, V_j, n)$ ).

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 even numbers 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.

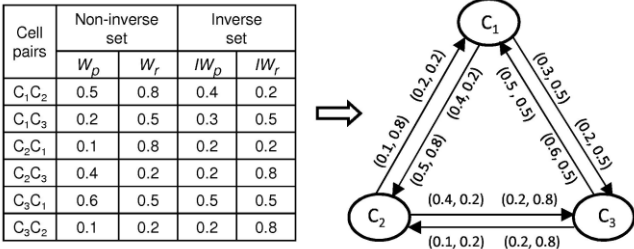


Fig. 9 Construction of the directed graph based on inverse and non-inverse correlation sets.

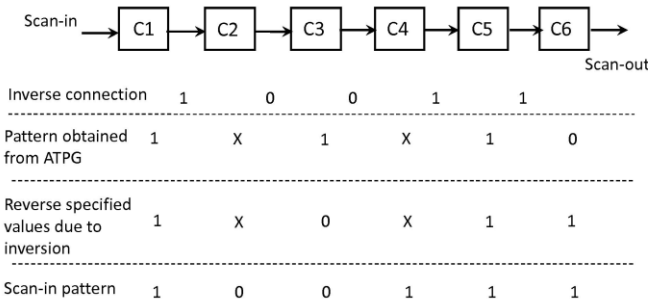


Fig. 10 An example of the MT-fill performed in SIRO.

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 are that Q connects to SI while the inversions occur.

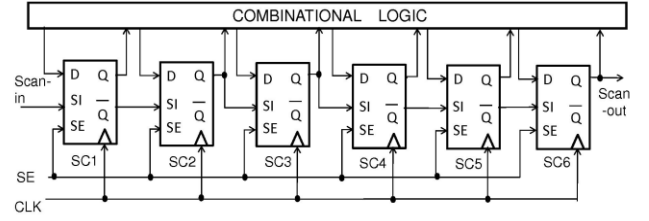


Fig. 11 Implement inverse techniques with traditional scan-chain architecture.

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 XI  
COMPARISON OF GENERATED SCAN-SHIFT TRANSITIONS BETWEEN SIRO AND ROBPR.

## B. Experimental Results

We conduct experiments for SIRO on the same benchmark circuits and test patterns as those used in Sec IV-B. Table XI compares the results of SIRO with that of ROBPR, which considers only the non-inverse pattern and response correlation during the reordering process. As the results show in Table XI, SIRO in average can generate 1.23% less total scan-shift transitions with almost the same runtime compared to ROBPR. However, even though SIRO can generate a smaller or at least 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 XI. 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  $\text{Cost}_{\text{inv}}(V_i, V_j, n)$  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 XII lists the probability distributions of response correlations for each benchmark circuit.

circuit	Response correlation			
	0 ~ 0.5	0.5 ~ 1	0 ~ 0.25	0.75 ~ 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%

TABLE XII

Probability distribution of high and low response correlations.

From the above 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 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.

## 7. SCAN CELL REORDERING CONSIDERING BOTH POWER AND ROUTING FACTORS

All above 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.

### A. Detailed Steps of Reordering Consideration 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 steps all follow the steps in ROBPR.

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.

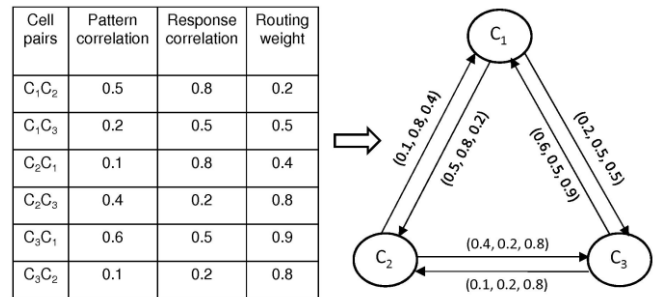


Fig. 12 Construction of the directed graph based on correlations and routing effects.

The directed graph constructed in this section is a revised version of the directed graph introduced in ROBPR (Step 2 in Sec VI-A2). 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.

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 tradeoff between scan-shift power and routing overhead:

$$C_T(V_i, V_j, n) = (1 - \beta) \times C_P(V_i, V_j, n) + \beta \times C_R(V_i, V_j) \quad (8)$$

$CR(V_i, V_j)$  represents the routing weight between cells  $V_i$  and  $V_j$ .  $CP(V_i, V_j, n)$  represents the cost function of scan-shift power when selecting the  $n$ -th cell and ranges from 0 to 1 as well. The value of  $CP(V_i, V_j, n)$  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  $\beta$  in  $C_T(V_i, V_j, n)$  is called the optimization factor, which is used to control the tradeoff between scan-shift power and routing overhead. The value of  $\beta$  ranges from 0 to 1. If  $\beta$  increases, this TSP algorithm focuses more on reducing routing overhead. If  $\beta$  decreases, this TSP algorithm focuses more on reducing scan-shift transitions. Figure 13 shows the details of the TSP algorithm.

```

1 #define
2 MD(Vi, Vj) : Manhattan distance between Vi and Vj
3 MaxCost : the maximum cost of Cost(Vi, Vj, n)
4 MinCost : the minimum cost of Cost(Vi, Vj, n)
5 MaxMD : the maximum Manhattan distance between two cells
6 MinMD : the minimum Manhattan distance between two cells
7 CP(Vi, Vj, n) : Cost(Vi, Vj, n)/(MaxCost - MinCost)
8 CR(Vi, Vj) : MD(Vi, Vj)/(MaxMD - MinMD)
9 CT(Vi, Vj, n) : (1 - β) × CP(Vi, Vj, n) + β × CR(Vi, Vj)
10 begin
11 N ← # of cells ; n ← 1 ;
12 Min_L ← a list of N edges having the minimum CT;
13 for each directed edge e(Vi, Vj) of Min_L
14 V1st ← Vi, V2nd ← Vj, Vlast ← V2nd;
15 while non-ordered V
16 costmin ← ∞ ; n ← (n + 1) ;
17 for each non-ordered Vnon
18 if (CT(Vlast, Vnon, n) < costmin)
19 costmin ← CT(Vlast, Vnon, n) ;
20 Vnext ← Vnon ;
21 endif
22 endfor
23 Vlast ← Vnext
24 endwhile
25 endfor
26 end

```

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

## B. 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 [25], where ROBPR only focuses on minimizing the scan-shift transitions and [25]’s 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 [25] to get its placement. Then both PRORO and [25]’s scan-cell reordering are performed based on this placement of ROBPR. [25]’s scan-cell reordering is performed by using the command”scan reorder” in [25]. A TSMC 0.18 $\mu$ m CMOS technology with 5 metal layers is used in the experiments.

Table XIII first lists the total number of scan-shift transitions generated by different scan-cell reordering schemes. For the convenience of result comparison, Table XIII 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 XIV lists the estimated wire length of the scan paths (in  $\mu$ m) 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 XIII, Table XIV also normalizes the wire length of scan paths of each reordering scheme by dividing it with that of [25]’s reordering scheme, which is supposed to be the reordering scheme generating the shortest wire length of scan paths in this experiment. Table XV further lists the total wire length (including the routing for both scan paths and CUT) generated by each reordering scheme after detailed route.

circuit	method	ROBPR	PRORO			[25] reordering
			$\beta=0.25$	$\beta=0.5$	$\beta=0.75$	
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 XIII  
COMPARISONS OF SCAN-SHIFT TRANSITIONS  
GENERATED BY DIFFERENT SCAN-CELL REORDERING  
SCHEMES.



circuit	method	ROBPR	PRORO			[25] reordering
			$\beta=0.25$	$\beta=0.5$	$\beta=0.75$	
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 XIV  
COMPARISONS OF SCAN PATH'S WIRE LENGTH ( $\mu\text{m}$ )  
AFTER GLOBAL ROUTE GENERATED BY DIFFERENT  
SCAN-CELL REORDERING SCHEMES.

As the results shown in Table XIII, if only minimizing the wire length of scan paths such as tool [25]'s 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 [25]'s reordering scheme as shown in Table XIV. 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 tool [25]'s reordering scheme as shown in Table XV.

circuit	method	ROBPR	PRORO			[25] reordering
			$\beta=0.25$	$\beta=0.5$	$\beta=0.75$	
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

TABLE XV  
COMPARISONS OF TOTAL WIRE LENGTH ( $\mu\text{m}$ )  
AFTER DETAILED ROUTE GENERATED BY DIFFERENT  
SCAN-CELL REORDERING SCHEMES.

Also, the experimental results in Table XIII, XIV, and XV show that the tradeoff 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.

### 三、結論

In this project, 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. 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 VI). 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. 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 post-process pattern-reordering scheme was also proposed to minimize the in-between transitions. A series of experiments were conducted to compare the proposed schemes with a previous reordering scheme [18] and a commercial tool's reordering scheme [25]. The experimental results demonstrated the effectiveness and efficiency of each of the proposed scan-cell reordering schemes

### 四、參考文獻

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. Touba. Controlling peak power during scan testing. Proc. VTS, pages 153–159, 2002.
  7. A. Chandra and R. Kapur. Bounded adjacent fill for low capture power scan testing. Proc. VTS, pages, 2008.
  8. R. Sankaralingam and R. R. O. et al. Static compaction techniques to control scan vector power dissipation. Proc. VTS, pages 35–40, 2000.
  9. G. Mrugalski and J. R. et al. New test data decompressor for low power applications. Proc. DAC, pages 539–544, 2007.
  10. 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.
  11. O. Sinanoglu and A. Orailoglu. Modeling scan chain modifications for scan-in test power minimization. Int. Test Conf., 2003.
  12. 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.
  13. P. Rosinger and B. A.-H. et al. Scan architecture with mutually exclusive scan segment activation for shiftand capture-power reduction. IEEE Transactions on Computer-Aided Design (TCAD), 23(7):1142–1153, July 2004.
  14. L. Whetsel. Adapting scan architectures for low power operation. Proc. IEEE Int. Test Conf., pages 863–872, 2000.
  15. 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.
  16. T. Huang and K.-J. Lee. A token scan architecture for low power testing. Proc. Int. Test Conf., pages 660–669, 2001.
  17. 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.
  18. Y. Bonhomme and P. et al. Power driven chaining of flip-flops in scan architectures. Int. Test Conf., pages 796–803, 2002.
  19. 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.
  20. 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.
  21. Y. Bonhomme and P. G. et al. Design of routing-constrained low power scan chains. Proc. DATE, pages 62–67, 2004.
  22. S. Makar. A layout-based approach for ordering scan chain flip-flops. Proc. of the IEEE Int. Test Conf., pages 341–347, 1998.
  23. 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.
  24. Synopsys tetramax atpg user guide version x-2005.09.
  25. Cadence encounter product version 5.2.3 June 2006.

# 行政院國家科學委員會補助國內專家學者出席國際學術會議報告

2008 年 11 月 20 日

附件三

報告人姓名	趙家佐	服務機構 及職稱	國立交通大學 電子工程學系
時間 會議 地點	10/28/2008~10/31/2008 Santa Clara, CA, USA	本會核定 補助文號	計畫編號 NSC97-2221-E-009-173
會議 名稱	2008 IEEE International Test Conference		
發表 論文 題目	Testing Methodology of Embedded DRAMs		
<p>一、參與會議經過：</p> <p>ITC08 在美國加州 Santa Clara 舉行，共有 35 個 session，4 場 tutorial，1 個 poster session。本次參加最主要的目的地是論文報告，在 Session25: EMBEDDED MEMORY TEST。其餘參與之 session 有 “DEALING WITH OUTLIERS AND VARIATION IN TODAYS ICS”， “DEFECT AVOIDANCE AND COST MODELING”， “POWER-AWARE DFT METHODS”， “SYSTEMS EFFECTS OF ERRORS AND PROTECTION METHODS”， “TESTING FOR INTERCONNECT OPENS AND CROSSTALK”， “POWER IMPACT ON COMPRESSION AND AT-SPEED SCAN”。</p>			
<p>二、與會心得</p> <p>本會議為 VLSI Testing 最重要的會議，論文受重視度高，參展廠商多，涵概研究領域新穎，台灣在 2008 年的論文數目為 3。參與人員業界學界各占一半，所以不會僅限於單純的學術交流，很適合拓展研究視野。另外 poster session 的討論也相當踴躍，應該多鼓勵學生參與。</p>			
<p>三、建議</p> <p>以後只要經費許可，應多鼓勵學生參與，特別是做 VLSI testing 領域的博士生，應該藉此機會了解測試領域最先進的研究成果，藉此刺激對研究工作的嚮往，並擴展國際學生之間交流。</p>			

四、攜回資料名稱及內容:  
攜回一 CD 論文集

五、其他

無