

國立交通大學  
電子工程學系電子研究所碩士班

碩士論文

利用掃描鏈重新排序增加橋接錯誤故障涵蓋率

架構在低功率固定錯誤樣式填充

**Scan reordering for improving bridge fault coverage  
based on low-power stuck-at-fault pattern fill**



研究生：郭淳仁

指導教授：趙家佐博士

中華民國一百年一月

利用掃描鏈重新排序增加橋接錯誤故障涵蓋率

架構在低功率固定錯誤樣式填充

# Scan reordering for improving bridge fault coverage based on low-power stuck-at-fault pattern fill

研 究 生：郭淳仁

Student : Chun-Ren Kuo

指導教授：趙家佐

Advisor : Chia-Tso Chao



國立交通大學  
電子工程學系電子研究所碩士班  
碩士論文

A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

In Partial Fulfillment of the Requirements

for the Degree of

Master

In

Electronics Engineering

September 2010

Hsinchu, Taiwan, Republic of China

中華民國 一〇一〇年一月

利用掃描鏈重新排序增加橋接錯誤故障涵蓋率

架構在低功率固定錯誤樣式填充

學生：郭淳仁

指導教授：趙家佐

國立交通大學電子工程學系電子研究所碩士班

## 摘 要

這篇論文提出掃描鏈的重新排列架構在低功率固定錯誤樣式填充的基礎上，增加橋接錯誤故障涵蓋率。為了克服功率固定錯誤樣式填充所帶來的困難，我們先從物理佈局上面抽出可能的橋接錯誤的發生點，接著利用這些發生點計算每一對PPI在橋接錯誤的可能觸發數，並用這個數字當作每一對PPI的影響力。接著我們提出掃描鏈的劃分以及掃描鏈的重新排列來減少每一條掃描鏈內所有掃描相對間的關聯性，進而提高橋接錯誤故障涵蓋率。最後我們做一些實驗來驗證方法的有效性，繞線的影響也投過實驗來做討論。

# **Scan reordering for improving bridge fault coverage based on low-power stuck-at-fault pattern fill**

**Student: Chun-Ren Kuo**

**Advisor: Dr. Chia-Tso Chao**

**Department of Electronics Engineering**

**Institute of Electronics**

**National Chiao Tung University**

## **Abstract**

This thesis proposes a scan reordering method based on low-power fill on stuck-at fault pattern to improve the bridge fault coverage. In order to overcome the drawback of low-power fill on bridge fault coverage, we first extract bridge faults from physical layout and calculate the impact of each PPI pair on bridge-fault activation to measure the correlation between two PPI pair. Next, scan partition and scan reorder are performed to generate scan order that minimizes the correlation hold by scan cells in each scan chain. Experiment shows our proposed method generates better bridge fault coverage comparing to random scan order and the wire length overhead is also provided. Finally, some future possible improvement is presented.

## 誌 謝

首先要先感謝家中對我唸碩士的支持。在研究所兩年半的期間，家人們持續地給予鼓勵與支持是我走下去的動力。接下來要感謝指導老師 趙家佐教授不辭辛苦地耐心教導我。不僅教我一個研究生該有的態度，也教我日後身為工程師應有的工作態度。在研究領域上，老師除了反覆耐心地教導我應有的背景知識，一方面也鼓勵我培養科技英文寫作的能力。在研究上遇到困難時，更耐心陪我探索研究的方向，陪我走過低潮。

能夠從研究所畢業，更要感謝研究所所有同仁的陪伴與幫助。增錦在研究上大力地支持，並且告訴我許多業界與學術界地差異，讓我增廣見聞。振安學長、政偉學長也常在給我不同領域的知識。偉勝學長在程式上面給我很大的幫助，也在興趣方面讓我大開眼界。思邦在我低潮失落的時候，一邊開導我一邊跟我分享情緒控制的技巧。弘昕的陪伴也讓我的研究所生活多采多姿。也很感謝擴安陪我採買和發洩情緒。皓宇教導我許多文件編輯的技巧。澤葉陪伴我許多時光，交流兩岸資訊。也感謝學弟們不吝分享 Tool 使用分面的知識。也要感謝建男、臆安、千慧在研究所的這段時光中陪我談心解悶。再次感謝身邊所有人的陪伴，有你們才有愉快的回憶！

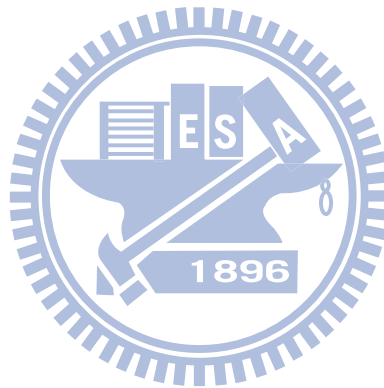
2011 年 1 月

## Contents

摘 要.....	i
Chapter 1 Introduction.....	1
Chapter 2 Preliminaries.....	4
2.1 Bridge fault model.....	4
2.2 Optimization Algorithm.....	6
Chapter 3 Overall Flow.....	9
Step1 Synthesis.....	9
Step2 Stuck-at Fault Pattern Creation.....	10
Step3 Placement & Routing.....	10
Step4 Fault List Generation.....	10
Step5 Calculate impact of each PPI pair on bridge-fault activation .....	11
Chapter 4 Scan Partition and Scan Reorder .....	13
4.1 Scan Partition .....	13
4.2 Scan Reorder .....	16
Chapter 5 Experimental Result.....	19
Chapter 6 Conclusion.....	22
Reference.....	23

## List of Tables

Table1 Database File Creation Time.....	11
Table2 Result of Single Scan Chain.....	20
Table3 Result of Ten Scan Chains.....	20
Table4 Wire length Overhead of Single Scan Chain.....	21
Table5 Wire length Overhead of Ten Scan Chains.....	21



## List of Figures

Figure 2.1 Brief bridge fault model and its representation.....	5
Figure 2.2 Seven bridge defect types.....	6
Figure 2.3 Illustration of order-based crossover.....	8
Figure 3.1 Overall flow.....	9
Figure 4.1 Main steps of scan partition.....	13
Figure 4.2 Main steps of scan reorder.....	16
Figure 4.3 Smoothing operation.....	17





# Chapter 1 Introduction

Decreasing feature size comes with variety of defects need to be take care to guarantee high yield rate. Lot of fault model have been proposed for pattern generation such as stuck-at fault, transition fault, path delay fault, stuck-open fault, bridge fault, etc [3]. However, to reduce the cost of chip testing, only testing patterns generated from stuck-at fault are used in current IC industry.

On the other hand, although scan design has been a widely used DFT technique to guarantee high coverage [2], shift test data into scan chain may results in excessive power consumption during the scan-based testing, suffering from physical damage or reliability degradation to circuit-under-test (CUT), and in turn decreasing the yield and product lifetime [1]. Furthermore, the System-on-a-Chip (SoC) revolution makes a circuit generally consumes more power in test mode than in normal mode. This extra power consumption could give rise to severe hazards in circuit reliability and may even result in instant circuit damage [7]. To consider the trend in using stuck-at fault pattern for cost concern, many works have been investigated the possibility to improve un-model fault coverage.

Methods are proposed to utilize the don't-care bits to minimize the scan-in

transitions for a given test set [4][5][6]. Low power pattern fill is one favorite technique to reduce power consumption when shifting the test data into scan chain. [4] proposes 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. Other methods like [5][6] reduces the test power as well as the test data volume based on de-compression architecture.

Research has showed that if we random fill the unknown bits of stuck-at fault pattern, it will has high possibility to detect un-model faults. Although MT-fill is good at minimize the total transition in scan testing, it will make adjacent scan cells filled with same value and thus greatly reduce the number of fault activation. In order to increase the number of fault activation, the scan reorder is needed to make high correlation scan cells separated.

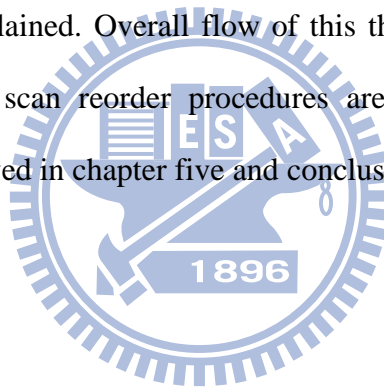
Based on the MT-fill [4] on the stuck-at fault pattern, this thesis has developed a scan reordering method to improving the bridge fault coverage while the benefit of low power is reserved. Since the low power pattern fill makes adjacent scan cell with same bit, this method make bridge sites more difficult to be both excited and propagated. To overcome this problem, we first investigate the fan-out cone overlap condition of each PPI, and further check the bridge fault that may affected by the input of PPI pair. We calculate the number of bridge faults as key element to determine the correlation between each PPI pair. When two scan cells influence many bridge faults, we consider them as high-related scan cell pair and try to place them apart.

To mimic the reality occurrence of bridge defect, ISCAS89 and ITC99 test benches are used, with synthesis, placement and routing through commercial tools with UMC 90nm technology, then deterministic bridge sites extracted from physical layout are used. And since to calculate co-effected bridge fault number of each

flip-flop pair is time-consuming, we store the result as database file to make it as one-time operation. This will greatly reduce the simulation time and the runtime will be showed in following chapter.

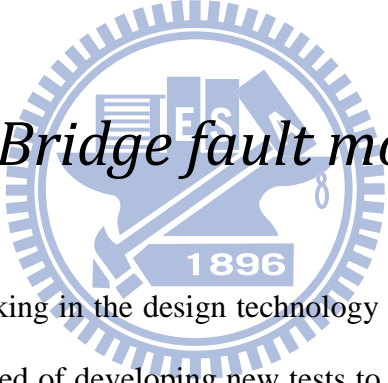
Two cost functions are proposed for evaluate both scan partition and scan ordering status. Thus the problem transforms into an optimization problem and aims to find the best scan chain order of two cost functions. We proposed algorithms for both scan partition and scan reorder. Experimental result shows that our algorithm gains good relation between cost function and bridge fault coverage.

At chapter two, we review previous works about bridge fault model, bridge fault pattern generation and technique about physical bridge site extraction. Optimization algorithm is also brief explained. Overall flow of this thesis is explained in chapter three. Scan partition and scan reorder procedures are proposed in chapter four. Experimental result is showed in chapter five and conclusion in last chapter.



## Chapter 2 Preliminaries

### 2.1 Bridge fault model



With continuous shrinking in the design technology node moving from 65nm to 45nm or even 28nm, the need of developing new tests to effectively screen defects is a must. At these complex technology nodes there is a need to screen defects such as bridging faults, open fault model based tests, etc. To detect these defects, accurate fault models are necessary in order to develop and evaluate test stimuli that will reveal the existence of the defect [8]. Many bridge fault models [9][10][21] have been proposed such as wire-AND, wire-OR, dominant bridge fault, etc. Resistive bridge fault model and its simulation have been also discussed [11][12]. Some works like [13][14][15] are proposed to treat resistive bridge fault as Byzantine Generals problem. Figure 2.1 shows a brief bridge fault model and its representation.

Since the bridge faults need to check the excitation of two nodes, it needs thousands of test condition to be tested. Papers focus on test pattern generation for

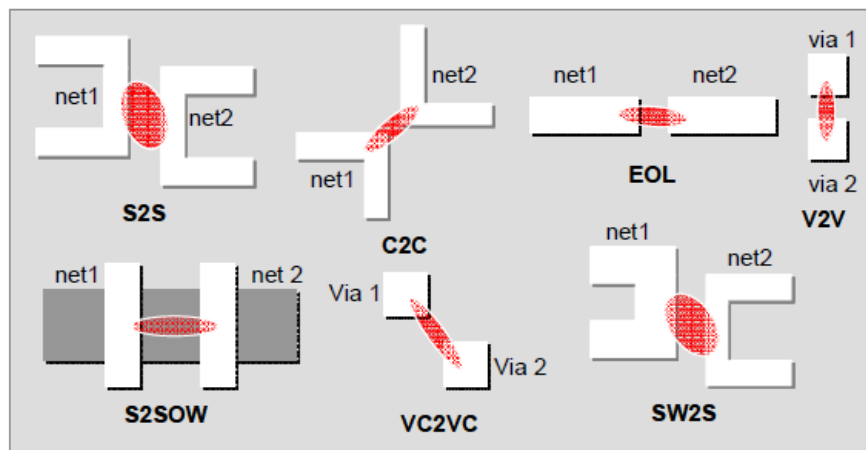
bridge faults have been proposed [11][16]. Another general method to test bridge fault is using N-detect test pattern. Works that discussed the effectiveness using N-detect test pattern for bridge faults are also undertaken [17][18][19]. Simple probabilistic metrics like Bridge Coverage Estimate (BCE) [25] are used as criterion to quantify the bridge coverage for a given test pattern set. And since the bridge fault list is normally huge when comparing with stuck-at fault list, Wu et al. [20] has also proposed an approach to reduce the bridge list by structural analysis and layout information.

Fault Model	Representation	Excitation Condition(s)
Wired-AND Bridge (a AND b)		1) (a = 0) and (b = 1) 2) (a = 1) and (b = 0)
Wired-OR Bridge (a OR b)		1) (a = 0) and (b = 1) 2) (a = 1) and (b = 0)
DOM Bridge (a DOM b)		1) (a = 0) and (b = 1) 2) (a = 1) and (b = 0)
DOM0 Bridge (a DOM0 b)		1) (a = 0) and (b = 1)
DOM1 Bridge (a DOM1 b)		1) (a = 1) and (b = 0)

**Figure 2.1: Bridge fault model and its representation**

Recent year, shrinking feature size and increased wire density increase the likelihood of occurrence of bridge related defects. DFM issue has widely been taken into consideration and merged into design flow. Since achieving high bridge defect coverage requires extraction of realistic bridge sites from the physical layout [18], works discussing extracting fault list from physical bridge defects are gradually increased [19][22][23][24]. Simulation in [19] results show that n-detect patterns have very poor bridge coverage performance and commonly used metric bridge coverage

estimate (BCE) does not relate to the true bridge fault coverage. Figure 2 shows seven bridge defect types presented in [19]. To simplify the problem, we consider only S2S in this thesis.



**Figure 2.2: Seven different bridge fault**

Due to the nature of bridge fault, bridge fault is hard to be excited and its fault coverage is relatively low when comparing to stuck-at fault. Thus, in this thesis, we try to develop a method to increase bridge fault coverage by scan reordering based on MT-fill, which can maintain low transition power in scan testing stage. Bridge fault list is generated from the physical layout for both achieving higher fault coverage and considering only the possible bridge sites. More detail will describe in next chapter.

## 2.2 Optimization Algorithm

Optimization is an area of mathematics that is concerned with finding the “best” points, curves, surfaces, and so on. And, “Best” is determined by minimizing some

measure of performance subject to equality and inequality constraints [26]. In this thesis, we create two cost functions to measure the quality of scan partition and scan reordering. The problem thus can be treated as optimization problem. There are plenty of algorithms using in wide engineering problem and EDA field, such as Divide and conquer algorithm, Greedy algorithm, Dynamic programming, Simulated annealing, etc [27]. In this thesis, we use Genetic Algorithm (GA) [29] for basic algorithm structure in our scan reordering stage.

Genetic Algorithm knows to have two issues when considering using it:

- 1) it needs to take huge computation of the cost function
- 2) it is hard to determine all the parameter about generating initial population, cross-operator to create next population, and the mutation rate.

In this thesis we choose it because (i) we create cost function other than taking fault simulation as cost function, the burden of taking heavy computation on cost function will be alleviated. And (ii) the feature that genetic algorithm will reserve the good population and reorder its permutation to find better population makes we picks genetic algorithm other than simulated annealing as our basic algorithm. Note that we only use it as basic algorithm structure. We removed mutation operation in our algorithm and take little use of the cross-operator in normal genetic algorithm. In fact, we focus on using our specific operator as main movement to achieve good solution. The algorithm detail will be presented in later chapter.

There are three famous permutation-based crossover operators, 1) Order-based crossover, 2) Partially Matched Crossover, and 3) Cycle Crossover [31]. We only use Order-based crossover in our thesis, since it maintains the order in good population which fits our purpose and may generate better populations. Figure 2.3 is an

illustration of order-based crossover.

Parent P <sub>1</sub>	A	B	C	D	E	F	G
Parent P <sub>2</sub>	C	B	G	E	F	D	A
Child C <sub>1</sub>	?	?	C	D	E	?	?
Child C <sub>2</sub>	?	?	G	E	F	?	?
Child C <sub>1</sub>	F	A	C	D	E	B	G
Child C <sub>2</sub>	C	D	G	E	F	A	B

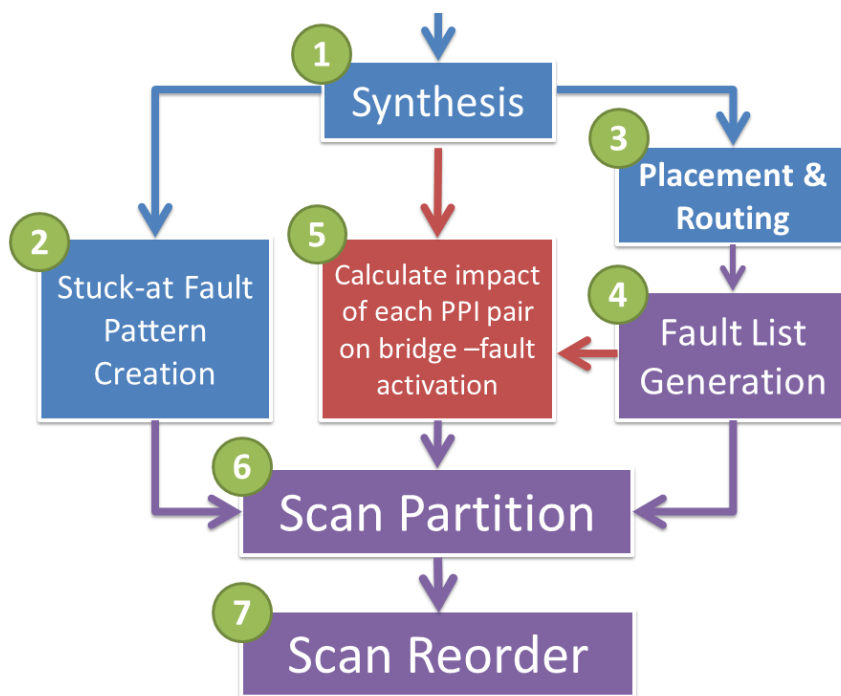
**Figure 2.3: Illustration of order-based crossover**

It takes two permutation as parents, and random choose one interval to reserve. Slots outside the interval are filled with elements in the order of another permutation. In this example, the elements filled starting from the second crossover site. This technique will reserve fragment of permutation and combine with the segment got from another permutation to find better population.



# Chapter 3 Overall Flow

In this chapter, the overall flow of this thesis will be presented and description of each step will be provided in order. Figure 3.1 shows the overall flow and its process order. There are seven steps in our overall flow; the mark number is also the process order. The steps in blue box represent a step using commercial tool. The steps in purple box stand for part of our program. And the only one step in red box emphasize it is a one-time process of each circuit for our program.



**Figure 3.1: Overall Flow**

## *Step1 Synthesis*

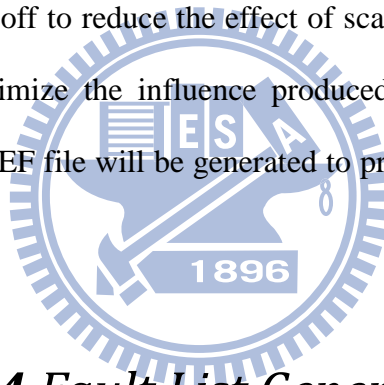
In this step, the non-scanned circuit netlist is loaded, then synthesized and replace all DFF cell with scan cell using Synopsys Design Compiler. UMC 90nm technology is used in this thesis.

## ***Step2 Stuck-at Fault Pattern Creation***

In this step, the stuck-at fault pattern generation will undertake to create the input pattern for later algorithm usage. Synopsys TetraMax ATPG is used for pattern generation. We leave all unknown bits as X (no-fill).

## ***Step3 Placement & Routing***

We then use the Cadence SOC Encounter for placement and routing. Although the tool provides optimization through scan cell reordering, this feature in placement and routing stage is turned off to reduce the effect of scan configuration. This setting is chosen in order to minimize the influence produced by placement and routing related to scan cells. The DEF file will be generated to provide layout information for later bridge site extraction.



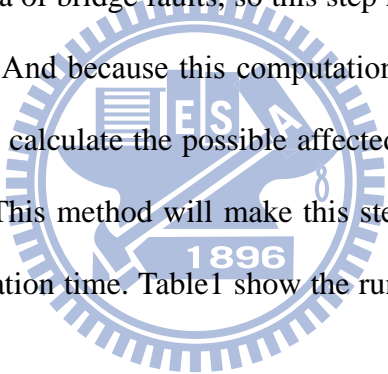
## ***Step4 Fault List Generation***

In this step, we load both DEF and LEF file to analyze the layout information and then find the local proximity for possible bridge sites. The type S2S mentioned in Figure 2.2 is considered. The parameter in determining the possible distance that bridge fault happened is process-dependent. The parameter can refer to Design rule or recommend parameter for manufacture provided by foundry. We discard all the bridge faults that related to scan cell. This is because the net of scan cell may be change after the scan reorder stage. Note that although we ignore the bridge site related to scan cell, but the scan testing is much easier than combination testing. So if the bridge defects are happened, it should be easy detected by scan testing.

## ***Step5 Calculate impact of each PPI pair on bridge-fault activation***

In this thesis, we need to have criterion the determined which PPI pair have high correlation. First, the fan-out cone of each PPI is traversed. Then each PPI pair does the computation to find how much bridge faults lie on the overlap field of these two PPI's fan-out cone. The number of bridge faults will then be the measure of this PPI pair. The higher correlation this PPI pair has, the higher bridge fault number each PPI pair holds.

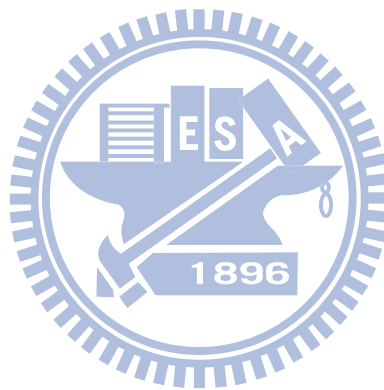
This step needs the data of bridge faults, so this step needs to be taken after *P&R* and *Fault List Generation*. And because this computation needs to check each PPI's fan-out cone and iteratively calculate the possible affected bridge faults, we store the result into a database file. This method will make this step a one-time operation and so greatly reduce the simulation time. Table1 show the runtime of each circuit for the computation in this step.



<b>Circuit Name</b>	<b>Computation Time (min)</b>
<b>s13207</b>	2.5
<b>s15850</b>	2.9
<b>s38417</b>	46
<b>s35932</b>	44
<b>s38584</b>	105
<b>b17</b>	104
<b>b20</b>	3
<b>b21</b>	3
<b>b22</b>	19

Table1 Database File Creation Time

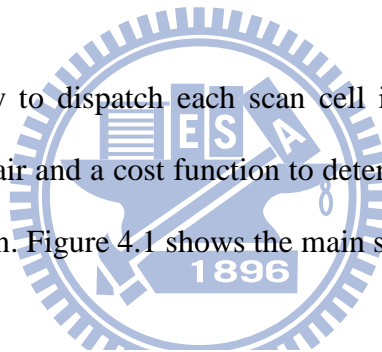
Above five steps can be treated as preparation of this thesis. Three inputs: 1) stuck-at fault pattern, 2) impact of each PPI pair, and 3) bridge fault list will be generated. The last two stages will be described in next chapter.



# Chapter 4 Scan Partition and Scan Reorder

## 4.1 Scan Partition

In this section, we try to dispatch each scan cell into scan chain. We use the impact value of each PPI pair and a cost function to determine which scan cell should be placed at which scan chain. Figure 4.1 shows the main steps for scan partition.



**Step 1:** Calculate the average impact value of each scan cell

**Step 2:** Rank all scan cells by average impact value. Place scan cell with high impact value first until each scan chain has one scan cell

**Step 3:** Calculate the impact value between each scan cell and each scan chain

**Step 4:** Choose the minimum impact value calculated in Step 3 and place that scan cell into specific scan chain

**Step 5:** Do Step 3 & Step 4 until all scan cells are placed

**Step 6:** Moving the scan cell if there has benefit

**Figure 4.1: Main steps of scan partition**

## Calculate the Average Impact Value of each scan cell

At the beginning of Scan Partition, we first calculate average impact value of each scan cell using Equation 4.1.  $Impact_{i,j}$  represents the impact value hold between scan-cell  $i$  and scan-cell  $j$ .  $N$  represents the number of impact values that one PPI holds.

$$Impact_{i,average} = \frac{\sum_N Impact_{i,j}}{N} \quad (4.1)$$

## Rank and Place Scan Cell with High Impact Value

At Step 2, we first rank all the scan cells by average impact value. The higher average impact value one scan cell has, the more important one comes first. So then scan cell with higher average impact value is chosen and place it into the empty scan chain. Do this operation until each scan chain has one scan cell in it. We need at least one scan cell in each scan chain as a seed for future partition reference base.

## Calculate the Impact Value between each scan cell and each Scan Chain

After placing the scan cell with high average impact value into the scan chain, we now deal with the scan cell left. In this operation, we first calculate the increase impact value if we place one scan cell into one specific scan chain. So we will get one impact value for each pair of one scan cell with one scan chain. Equation 4.2 shows how the impact value calculated.

$$Impact\_Value(i, j) = \sum_k Impact(i, j, k) \quad (4.2)$$

The  $Impact\_Value(i, j)$  in Equation 4.2 stands for the impact value increased if scan cell  $i$  is placed into scan chain  $j$ .  $Impact(i, j, k)$  in the right hand of Equation 4.2 represents the impact value between the scan cell  $i$  and the  $k$ -th scan cell in scan chain  $j$ . We accumulate the impact value between all the scan cell in that scan chain and candidate scan cell. Then we choose the minimum impact value and place that scan cell  $i$  into specific scan chain  $j$ .

In the Step 3 and Step 4, each time we consider all the left scan cell by calculate the impact value with all scan chains and choose the best choice to place one scan cell into scan chain.

### **Moving the scan cell if there has benefit**

The final step we take in this part is tweak of the scan partition. In order to overcome the imperfection of the straightforward partition method, we try to find some available scan cell partition change to achieve better solution. We randomly try to find available scan cell pair that will reduce total impact value in two scan chains. This operation is not ended until limit movements are already taken or not available change can be found in limit checking.

In this stage, we consider all the scan chain as a partition and attempt to find a good partition that the impact value hold by all scan cells in all scan chains is the minimum. Then we take optimization to the scan order in each scan chain in next scan reorder stage.

## 4.2 Scan Reorder

In this stage, we apply scan reorder to each scan partition generated to find the best order of least correlation between adjacent scan cells. Genetic Algorithm is used as basic algorithm structure in this stage. Figure 4.2 shows the main steps for scan reorder.

- Step 1:** Generate the initial population
- Step 2:** Rank the initial population by cost function and reserve the good population.
- Step 3:** For each good population, use Smoothing operation and Order-based crossover to generate next population.
- Step 4:** Use the population generated and do Step 2 and Step 3 again until limit loop number is achieved.

**Figure 4.2: Main steps of scan reorder**



At first, we need to generate the initial population using random order. Then all the random order is ranked by cost function listed as Equation 4.3.

$$Cost = \sum_{i=0}^{n-1} \sum_{j=i+1}^n \frac{Impact_{i,j}}{|distance(i-j)|} \quad (4.3)$$

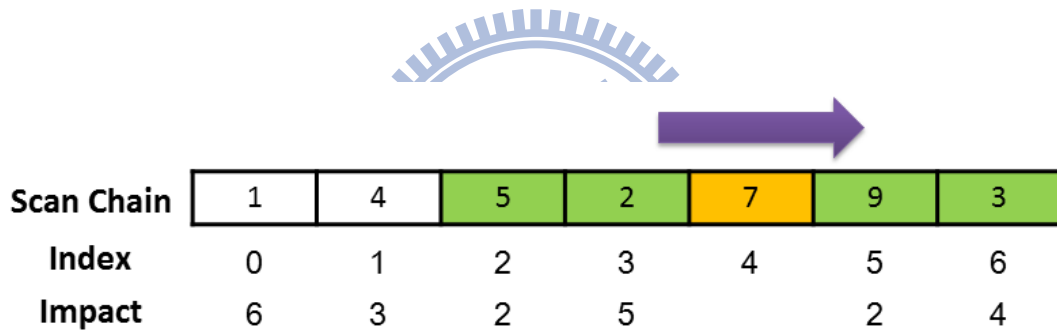
In order to evaluate the scan order, we calculate each scan cell pairs' impact value and divide with two cells' current distance difference. In this cost function, we consider the scan cell's position by the division by the distance difference. If two scan cells are far way, its impact value will be reduced and will be less important to the total cost of the scan chain.

We create this cost function not only to consider both impact value hold by each



scan cell pair and its position, but to replace fault simulation with this cost function. Since fault simulation takes greatly simulation time and genetic algorithm needs to process cost function frequently, we use this cost function to quickly evaluate the scan order.

After we calculate all the cost of population and rank them, we reserve some best scan order by directly copying into next population. For each good population, we also apply the Smoothing operation to improve it. The concept of smoothing operation is to calculate and compare the cost value came from both side of the considering scan cell. Then we move the considering scan cell to the direction of smaller cost value. Figure 4.3 shows an example to explain the smoothing operation.



**Figure 4.3: Smoothing Operation**

One scan chain with seven scan cells is given in this example and the scan cell 7 marked in the yellow box is the considered one. To consider the effect of most scan cells, we view scan cell 7 as center of circle and choose longest radius, and that is two units in this example. Scan cells located inside the circle are marked in green box. The first row below the scan chain shows the scan order index of that scan cell, and the second row shows the impact value held between that scan cell and the scan cell 7.

$$Cost_{i,j} = \frac{Impact_{i,j}}{|distance(i-j)|} \quad (4.4)$$

Equation 4.4 is used to calculate the cost value held between each scan cell pair.

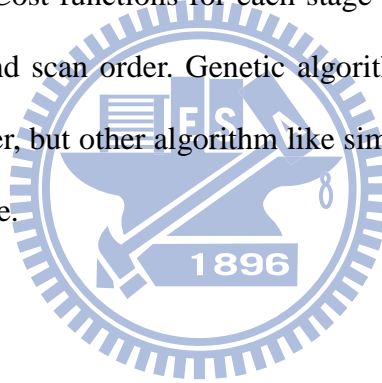
We can derive  $\frac{2}{|2-4|} + \frac{5}{|3-4|} = 6$  from the scan cells on the left side of scan cell 7 and

$\frac{2}{|5-4|} + \frac{4}{|6-4|} = 4$  from the other side. After comparing cost value from both sides, we

move the scan cell 7 to the right side (end of scan chain).

In each loop, we reserve the better scan order and generate other scan order by applying smoothing operating and order-based crossover. Then we do Step 2 and Step 3 until limit loop number is achieved. We use the scan order with lowest cost value as the final scan order.

In this section we propose two stages to generate better scan order in order to active more bridge faults. Cost functions for each stage are proposed to evaluate the quality of scan partition and scan order. Genetic algorithm are used in scan reorder stage to get better scan order, but other algorithm like simulated annealing can also be applied in scan reorder stage.



# Chapter 5 Experimental Result

ISCAS89 and ITC99 test benches are used for experiment. Test benches are first be synthesized, then taking placement and routing through commercial tools as explained in chapter three. UMC 90nm technology is used. Bridge sites are extracted from physical layout.

Nine test benches are used for simulation. The simulation results are showed in Table2 and Table3. Table2 shows the fault simulation result with single scan chain while Table3 shows the result with ten scan chains. Both tables have same structure. The first column shows the circuit name. Second column shows the bridge fault coverage without low power fill on stuck-at fault pattern (*italic*). Third column shows the best fault coverage in one hundred times random scan order simulation with low power fills (*italic*) and fourth column shows its runtime cost (**bold**). The fifth column shows the fault coverage of scan order generated by algorithm in this thesis (*italic*) and last column shows the runtime cost (**bold**).

Both tables show that the fault coverage of the scan order generated by this thesis is equal with one of the random order simulation. Single scan chain's coverage improvement is better than ten scan chains one, this result approves the effect of scan reorder method. Runtime of single scan chain is comparing high with random simulation. The reason is because the one long scan chain becomes the bottleneck of population computation in genetic algorithm. Runtime of multiple scan chain is much smaller than random

Coverage	<i>No-Fill</i>	<i>Max(100)</i>	Time(s)	<i>Work</i>	Time(s)
s13207	70.65	84.63	24	86.23	153
s15850	82.04	89.89	18	91.6	114
s38417	82.65	90.57	126	92.65	1723
s35932	92.63	93.08	130	93.06	1972
s38584	94.39	96.68	178	97.45	1277
b17	64.49	86.82	1379	88.82	1246
b20	66.78	91.72	370	93.83	82
b21	54.94	89.4	508	93.4	86
b22	65.47	90.34	548	93.61	216

Table2 Result of Single Scan Chain

Coverage	<i>No-Fill</i>	<i>Max(100)</i>	Time(s)	<i>Work</i>	Time(s)
s13207	70.65	88.62	16	88.82	18
s15850	82.04	91.64	12	91.8	13
s38417	82.65	92.56	51	92.75	286
s35932	92.63	93	47	93.06	821
s38584	94.39	97.34	124	97.42	174
b17	64.49	89.08	1428	90.23	197
b20	66.78	94.18	370	94.4	58
b21	54.94	94.19	526	94.69	52
b22	65.47	93.51	550	94.28	202

Table3 Result of Ten Scan Chains

simulation. This result shows that when circuit and pattern size become bigger, the method we proposed will save more time in determining the scan order better for bridge fault coverage.

Table 4 and Table 5 present wire length overhead of single scan chain and ten scan chains. First column named ‘Original’ stands for the wire length of default scan order that automated scanned by Synopsys Design Compiler. Second column named ‘Random’ shows the wire length of random scan order. Third column named ‘Optimize’ shows the wire length of the scan order that automatically optimized by

Cadence SOC Encounter, and Fourth column named ‘Work’ shows the wire length of the scan order generated by this thesis. Last column (*italic*) show the overhead ratio of work column over optimize column. Since this thesis has not taken wire length as one consideration, the overhead is high and it leaves wire length as an issue for future improvement.

Length Unit	Original	Random	Optimize	Work	<i>Overhead</i>
s13207	20298	78124	5423	94380	<i>17.4</i>
s15850	16539	72956	5533	87527	<i>15.8</i>
s38417	47447	355591	45159	440520	<i>9.75</i>
s35932	52013	362298	51205	363437	<i>7.1</i>
s38584	34238	222703	33784	290284	<i>8.59</i>
b17	40186	373914	25789	497184	<i>19.27</i>
b20	20864	78419	5988	102497	<i>17.11</i>
b21	21208	83240	6141	85834	<i>13.97</i>
b22	32964	160183	9264	159648	<i>17.23</i>

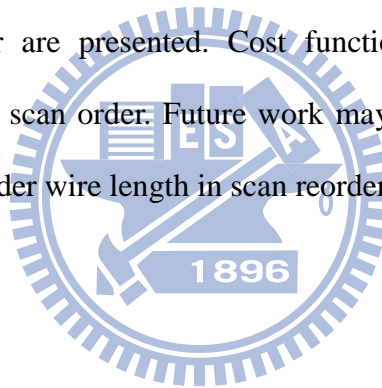
Table4 Wire length Overhead of Single Scan Chain

Length Unit	Original	Random	Optimize	Work	<i>Overhead</i>
s13207	18471	82128	5315	94924	<i>17.8</i>
s15850	15050	48177	5367	88398	<i>16.5</i>
s38417	43176	123096	44707	118718	<i>2.66</i>
s35932	46292	227406	52229	325862	<i>6.23</i>
s38584	31499	117889	33446	120658	<i>3.61</i>
b17	38176	121615	25273	389582	<i>15.4</i>
b20	19403	47275	5928	92140	<i>15.5</i>
b21	19723	53388	6445	98322	<i>15.3</i>
b22	28678	82643	9541	186838	<i>19.58</i>

Table5 Wire length Overhead of Ten Scan Chains

## Chapter 6 Conclusion

In this thesis, we demonstrate a method focus on scan reorder to improve bridge fault coverage based on low power fill on stuck-at fault pattern. Bridge fault sites extraction from layout information is used in this thesis. Algorithms for both scan partition and scan reorder are presented. Cost functions are also formulated to evaluate scan partition and scan order. Future work may focus on improve the scan partition algorithm or consider wire length in scan reorder algorithm.



# Reference

- [1] P. Girard, "Survey of Low-Power Testing of VLSI Circuits", *IEEE Design & Test of Computers*, Vol. 19, No 3, pp.82-92, May-June 2002
- [2] M.L. Bushnell and V.D. Agrawal, "Essential of Electronic Testing", Kluwer Academic Publisher, ISBN 0-7923-7991-8, 2000.
- [3] Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen, "VLSI Test Principles And Architectures", Morgan Kaufmann Publishers, 2006
- [4] Sankaralingam, R. Oruganti, R.R. Touba, N.A., "Static compaction techniques to control scan vector power dissipation," *VLSI Test Symposium, 2000. Proceedings. 18th IEEE* , vol., no., pp.35-40, 2000
- [5] Mrugalski, G. Rajski, J. Czysz, D. Tyszer, J., "New Test Data Decompressor for Low Power Applications," *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE* , vol., no., pp.539-544, 4-8 June 2007
- [6] Shih Ping Lin; Chung Len Lee; Jwu E Chen; Ji-Jan Chen; Kun-Lun Luo; Wen-Ching Wu; , "A Multilayer Data Copy Scheme for Low Cost Test with Controlled Scan-In Power for Multiple Scan Chain Designs," *Test Conference, 2006. ITC '06. IEEE International* , vol., no., pp.1-8, Oct. 2006
- [7] Girard, P.; , "Survey of low-power testing of VLSI circuits," *Design & Test of Computers, IEEE* , vol.19, no.3, pp.80-90, May/Jun 2002
- [8] M. Abramovici, M. Breuer and A. Friedman, "Digital System Testing and Testable Design", IEEE Press, 1990.
- [9] K. C. Y. Mei, "Bridging and Stuck-At Faults", *IEEE Trans. on Computers*, Vol 23, No. 7, pp. 720-727, 1974.
- [10] Emmert, J.M. Stroud, C.E. Bailey, J.R., "A new bridging fault model for more accurate fault behavior," *AUTOTESTCON Proceedings, 2000 IEEE* , vol., no.,

pp.481-485, 2000

- [11] Sar-Dessai, V.R. Walker, D.M.H., "Resistive bridge fault modeling, simulation and test generation," *Test Conference, 1999. Proceedings. International* , vol., no., pp.596-605, 1999
- [12] Polian, I. Engelke, P. Becker, B. Kundu, S. Galliere, J.-M. Renovell, M., "Resistive Bridge fault model evolution from conventional to ultra deep submicron," *VLSI Test Symposium, 2005. Proceedings. 23rd IEEE* , vol., no., pp. 343- 348, 1-5 May 2005
- [13] Lavo, D.B. Larrabee, T. Chess, B., "Beyond the byzantine generals: unexpected behavior and bridging fault diagnosis," *Test Conference, 1996. Proceedings., International* , vol., no., pp.611-619, 20-25 Oct 1996
- [14] Keshk, A. Miura, Y. Kinoshita, K., "Procedure to overcome the Byzantine General's problem for bridging faults in CMOS circuits," *Test Symposium, 1999. (ATS '99) Proceedings. Eighth Asian* , vol., no., pp.121-126, 1999
- [15] Jue Wu, Rudnick, E.M., "Bridge fault diagnosis using stuck-at fault simulation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.19, no.4, pp.489-495, Apr 2000
- [16] Shinogi, T. Kanbayashi, T. Yoshikawa, T. Tsuruoka, S. Hayashi, T., "Faulty resistance sectioning technique for resistive bridging fault ATPG systems," *Test Symposium, 2001. Proceedings. 10th Asian* , vol., no., pp.76-81, 2001
- [17] Amyeen, M.E. Venkataraman, S. Ojha, A. Sangbong Lee, "Evaluation of the quality of N-detect scan ATPG patterns on a processor," *Test Conference, 2004. Proceedings. ITC 2004. International* , vol., no., pp. 669- 678, 26-28 Oct. 2004
- [18] Tran, E.N. Kasulasrinivas, V. Chakravarty, S. , "Silicon evaluation of logic proximity bridge patterns," *VLSI Test Symposium, 2006. Proceedings. 24th IEEE* , vol., no., pp.6 pp.-85, April 30 2006-May 4 2006
- [19] Goel, S.K. Devta-Prasanna, N. Ward, M., "Comparing the effectiveness of



- deterministic bridge fault and multiple-detect stuck fault patterns for physical bridge defects: A simulation and silicon study," *Test Conference, 2009. ITC 2009. International* , vol., no., pp.1-10, 1-6 Nov. 2009
- [20] Wu, J. Greenstein, G.S. Rudnick, E.M., "A fault list reduction approach for efficient bridge fault diagnosis," *Design, Automation and Test in Europe Conference and Exhibition 1999. Proceedings* , vol., no., pp.780-781, 1999
- [21] Syal, M. Hsiao, M.S. Doreswamy, K.B. Chakravarty, S., "Efficient implication-based untestable bridge fault identifier," *VLSI Test Symposium, 2003. Proceedings. 21st* , vol., no., pp. 393- 398, 27 April-1 May 2003
- [22] Wei Zou, Wu-Tung Cheng, Reddy, S.M., "Bridge Defect Diagnosis with Physical Information," *Test Symposium, 2005. Proceedings. 14th Asian* , vol., no., pp. 248-253, 18-21 Dec. 2005
- [23] Tran, E.N. Krishna, V. Zachariah, S. Chakravarty, S., "Logic proximity bridges," *Test Conference, 2005. Proceedings. ITC 2005. IEEE International* , vol., no., pp.10 pp.-761, 8-8 Nov. 2005
- [24] Gkatziani, M. Kapur, R. Qing Su Mathew, B. Mattiuzzo, R. Tarantini, L. Hay, C. Talluto, S. Williams, T.W., "Accurately Determining Bridging Defects from Layout," *Design and Diagnostics of Electronic Circuits and Systems, 2007. DDECS '07. IEEE* , vol., no., pp.1-4, 11-13 April 2007
- [25] Benware, B. Schuermyer, C. Tamarapalli, N. Kun-Han Tsai, Ranganathan, S. Madge, R. Rajski, J. Krishnamurthy, P., "Impact of multiple-detect test patterns on product quality," *Test Conference, 2003. Proceedings. ITC 2003. International* , vol.1, no., pp. 1031- 1040, Sept. 30-Oct. 2, 2003
- [26] David G. Hull, "Optimal Control Theory for Applications", Springer 2003
- [27] "Category:Optimization algorithms", *Wikipedia: The Free Encyclopedia*, 2 Jan. 2010, [http://en.wikipedia.org/wiki/Category:Optimization\\_algorithms](http://en.wikipedia.org/wiki/Category:Optimization_algorithms)

- [28] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning”, pp. 1989. Addison-Wesley
- [29] Holland, J. H., 1975, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI.

