

國立交通大學  
電子工程學系電子研究所  
博士論文

低耗能並考慮低成本效益之系統晶片測試策略



Low Power and Low Test Data Volume Testing for  
Scan Design VLSI

研究生：林世平

指導教授：李崇仁教授

中華民國九十六年九月

低耗能並考慮低成本效益之系統晶片測試策略  
Low Power and Low Test Data Volume Testing  
for Scan Design VLSI

研究生：林世平

Student: Shih-Ping Lin

指導老師：李崇仁教授

Advisor: Prof. Chung-Len Lee

國立交通大學

電子工程學系 電子研究所

博士論文



Submitted to Department of Electronics Engineering and  
Institute of Electronics  
College of Electrical and Computer Engineering  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Doctor of Philosophy  
in  
Electronics Engineering

September 2007

Hsinchu, Taiwan, Republic of China

中華民國 九十六年 九月

# 低耗能並考慮低成本效益之系統晶片測試策略

學生：林世平

指導老師：李崇仁教授

國立交通大學

電子工程學系 電子研究所

## 摘要

本論文就關於當前測試所面臨之問題：龐大之測試資料量以及過高的測試能量，做一完整的研究探討。針對此兩個測試問題，我們提出幾個解決方案。首先，提出一個類似矩陣架構之掃描方式，並搭配一個新設計之掃描暫存器，來構成掃描矩陣。此架構藉由略過不必要的掃描暫存器，減少掃描暫存器與電路內部的變化次數，來降低測試向量輸入時造成的能量消耗。我們提出之掃描暫存器不會降低待測物的運作效能，同時能降低時脈樹之能量消耗。由實驗結果顯示此架構可大幅降低測試時產生的能量消耗。

其次，我們提出一個基於隨機存取掃描之混合式測試流程。藉由觀察傳統的測試向量壓縮方法之不足處，我們提出幾個改進的技巧，並搭配混合式測試的流程來增進測試效率：(1)改進傳統靜態壓縮的方式，使壓縮後不會增加位元翻轉次數；(2)檢視各種模組，用來預估含不確定位元之測試向量重新排列時造成

之花費(位元翻轉)，並找出最有效之模組；(3)提出向量捨去的方法，可以保證錯誤涵蓋率不會因捨去後而降低。實驗結果顯示此提出之流程可以很有效率地減少位元翻轉次數，平均來說可以達到百分之八十六之壓縮率與增加近十倍的效能。

之後我們提出一個適用於測試系統晶片之測試資料壓縮方法，採用一個能支援各種區塊大小之適應性編碼方法，同時利用勘入之記憶體與一個解碼器來做測試資料解壓縮。此方式避免了傳統會因選擇之區塊大小而影響壓縮率之問題，我們同時提出一個能降低測試能量之技巧，來取捨壓縮率與測試能量之平衡，我們也採用兩階段之混合式測試來比較測試效能之增進。實驗結果證明此方式能有效降低測試資料量與測試能量，並且隨著測試電路之複雜度增加，整體效能亦能提升。

最後，我們提出一個針對具多重掃描鏈之待測物之低功率測試資料壓縮方法，稱為多階層次資料複製。此方法利用測試項量中之不確定位元來做不同層次的複製，以達到資料壓縮，我們並系統化分析此方式可以達到之壓縮率以及降低之能量。多層次資料複製不但可以直接針對測試向量做壓縮，亦可整合至測試向量產生器中提升效能。我們在實驗中也做了詳盡的比較，證明提出之多層次資料複製具有高壓縮率與低功率之特性，同時所付出之面積成本亦非常少，我們也與習知技術比較，列出此多階層次資料複製之各項優點。

# Low Power and Low Cost Test Strategies for SOC

Student: Shih-Ping Lin

Advisor: Prof. Chung Len Lee

Department of Electronics Engineering

& Institute of Electronics

National Chiao Tung University

## Abstract



Scan design is now a necessary practice for today's ICs when considering their testing. As the size of today's ICs now becomes tremendously large, the traditional scan test becomes inefficient and troublesome due to two problems: the large test data volume which leads to unaffordable test application time and the high test power which may cause reliability problem to ICs. This dissertation makes a comprehensive study on these two test challenges.

We propose several solutions. First, we propose a scan test architecture like matrix where a new scan cell is invented to be bypassed during pattern shifting when it is not addressed. This reduces the number of transitions of scan cells and the circuit under test (CUT) hence reduces the power consumption. In addition, the scan cell

does not introduce any penalty on degrading the performance of the CUT. Moreover, we also adopt a design to reduce the power of clock tree. Experimental results show that it can achieve nearly 99% power savings for large size designs.

Next, based on Random Access Scan (RAS), we propose a cocktail scan strategy. After surveying previous works, we present several improved strategies to improve the efficiency on test compression. These are: (1) *a constrained static compaction*, which is a compaction strategy to keep the number of bit flips the same after test cubes are compacted; (2) *optimum reordering of test cubes*: which is the best ordering of test cubes and is adopted by examining several cost models to estimate the number of bit flips; (3) *test cube dropping*: a method to drop test cubes while guarantee the same fault coverage. Experimental results show that the adoption of the above strategies is very effective in reducing the number of bit flips, leading to an 86% reduction in test data and ten times of speedup in test application time.

Thirdly, we propose an encoding scheme, *Adaptive Encoding*, which is suitable for test data compression in System-on-Chip (SoC), by utilizing an embedded memory and encoder. The conventional test data encoding schemes usually suffer the drawback that the compression rate is affected by the block size, leading inefficiency in compressing test data. The proposed scheme supports variable block size encoding, thus eliminates the above drawback and improves the encoding efficiency. In addition,

we also adopt a *hybrid test* technique to further reduce the volume of test data. We also try to make consideration of making tradeoff between the test compression rate and the test power during the above process. Experimental results show that the proposed method effectively reduces the volume of test data and test power. More specifically, we can reduce the test energy by 91.60% and reduce the peak power by 15.57% at the expense of 10.82% loss in test compression.

Finally, we propose a Multilayer Data Copy (MDC) scheme, which is very suitable for designs with large number of scan chains, to obtain high test compression with low-power testing. This scheme proposes an architecture which performs two operations, *Copy* and *Shift*, to achieve high test compression rate by exploiting don't care bits of test patterns. MDC can not only be used to compress test data sets but also be incorporated into automatic test pattern generator (ATPG) to give better efficiency. Similarly, we also consider test power reduction when do test data compression. Systematic study on this scheme shows that the schem has high compression rate and low testing power but has a negligible area overhead.

# 誌 謝

僅獻上最誠摯之謝意，感謝指導教授李崇仁博士，在學生論文研究期間給予最大之空間，使學生能在 IC 測試領域盡情發展，也由於老師的指導得以完成此博士論文。承蒙老師多年的教誨與照顧，亦師亦友的態度讓學生在研究領域、生活面上皆受益良多，在此獻上誠心之敬意。

另感謝我的家人與女朋友給予我長期的支持與鼓勵，有了他們使我有足夠的動力去面臨的一切挫折，才能完成博士的成就。

最後感謝實驗室的夥伴，包含竹一、明學、文慶、永嘉、淑敏、順志、俊偉、俊良、明和、俊言、劉坪、威憲、見明等，在我研究生涯中給予許多幫助，實驗室的生活點滴也是令人難忘的，謝謝一路上有你們相陪伴。

最後感謝工研院夥伴們：崑崙、繼展等，在研究路上給予我寶貴意見，並獲得工作上的經驗。

林世平

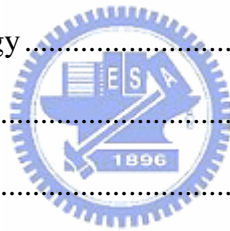
謹誌於 新竹交大

九十六年九月

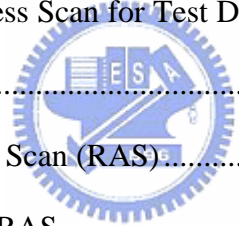


# Table of Contents

Abstract (in Chinese) .....	iii
Abstract (in English) .....	v
Acknowledge .....	viii
Table of Contents.....	ix
Table Captions .....	xiii
Figure Captions.....	xv
Chapter 1 Introduction.....	1
1.1 IC Test Methodology and Design-for-Testability .....	1
1.1.1 Test Points .....	2
1.1.2 Scan Design .....	3
1.1.3 BIST Methodology .....	5
1.2 IC Test Challenges .....	6
1.2.1 Test Cost .....	6
1.2.2 Test Power.....	8
1.3 Test Strategies for Test Power Reduction.....	10
1.3.1 Low Power DFT Techniques .....	10
1.3.2 Modifying Test Patterns for Low Power Test .....	16
1.4 Test Strategies for Test Data Reduction.....	18
1.4.1 Automatic Test Pattern Generation (ATPG) Approaches .....	18
1.4.2 Test Compression Approaches with On-Chip Circuitry .....	18
1.4.2.1 Combinational Type Decoding .....	19
1.4.2.2 Sequential Type Decoding .....	21
1.4.2.3 Bit Flipping Decoding .....	22
1.4.2.4 Code-based Schemes .....	23



1.5 Test Strategies for Simultaneous Test Power and Test Data Reduction.....	25
1.6 Classification and Term Definition for Test Compression Methods .....	27
1.6.1 Classified by Type of Decompressor .....	27
1.6.2 Classified by Application of ATPG.....	28
1.7 Overview and Organization of the Dissertation .....	29
Chapter 2 A Scan Matrix Design for Low Power Scan-Based Test.....	31
2.1 Introduction.....	31
2.2 The Proposed Scan Matrix Architecture .....	31
2.3 Evaluation and Comparison of SM with Other Scan Approaches.....	35
2.4 Experimental Results .....	39
2.5 Summary.....	43
Chapter 3 Cocktail Random Access Scan for Test Data and Power Reduction .....	44
3.1 Introduction.....	44
3.2 Review Random Access Scan (RAS).....	44
3.3 Cocktail Scan Based on RAS .....	46
3.3.1 Segmented Random Scan Test (SRST) .....	46
3.3.2 RAS Test .....	48
3.3.2.1 Test Pattern Generation for RAS Test.....	49
3.3.2.2 Cost Model for Don' t Care Bits.....	49
3.3.2.3 Test Response Abandonment .....	52
3.3.2.4 Constrained Static Compaction (CSC) .....	52
3.3.2.5 Bit-Propagation before Test Vector Dropping (BPBTVD).....	53
3.3.3 Hardware Modifications .....	54
3.3.3.1 Modified Address Register .....	54
3.3.3.2 On-Chip Scan Controller .....	55
3.4 Experimental Results .....	55



3.4.1 Experiment on Test Efficiency of the Proposed Process Compared with Other Processes.....	55
3.4.2 Experiment on Cocktail Scan .....	57
3.4.3 Hardware Overhead .....	61
3.4.4 Discussion on Some Problems on the RAS Architecture .....	62
3.5 Summary.....	63
Chapter 4 Adaptive Encoding Scheme Using Embedded Memory for Low-Cost and Low Power SoC Test .....	65
4.1 Introduction.....	65
4.2 Correlation-Related Compression Methods .....	66
4.3 Adaptive Encoding.....	67
4.3.1 Encoding Scheme .....	67
4.3.2 Decoder Machine Design and Its Operation.....	70
4.4 “Two-Phase Test” and “Test Vector Reordering” Techniques for Improvement on Test Compression.....	77
4.5 Compression Efficiency Analysis .....	79
4.6 Test Application Time Analysis .....	82
4.7 Pattern Fillings for Test Power Consideration.....	83
4.8 Experimental Results .....	85
4.9 Summary.....	90
Chapter 5 Low Power Test Compression for Multiple Scan Chain Designs .....	92
5.1 Introduction.....	92
5.2 The Proposed Multilayer Data Copy Scheme .....	93
5.3 Efficiency Analysis for MDC .....	99
5.3.1 Compression Analysis .....	99
5.3.2 Scan-In Power Reduction Analysis .....	101

5.4 Pattern Generator with MDC .....	103
5.5 Experimental Results .....	105
5.5.1 Compression Comparison.....	105
5.5.1.1 ATPG-independent MDC for Mintest test sets .....	106
5.5.1.2 ATPG-independent MDC .....	107
5.5.2 Scan-In Power Comparison.....	108
5.5.2.1 ATPG-independent MDC for Mintest test sets .....	109
5.5.2.2 ATPG-independent MDC.....	110
5.5.3 Comparison of MDCGEN with another Low Power Test Compression Method .....	111
5.5.4 MDCGEN for Large-scale Circuits .....	112
5.6 Summary.....	113
Chapter 6 Conclusions .....	115
6.1 A Scan Matrix Design for Low Power Scan-Based Test .....	115
6.2 Cocktail Random Access Scan for Test Data and Power Reduction.....	116
6.3 Adaptive Encoding Scheme Using Embedded Memory for Low-Cost and Low Power SoC Test .....	116
6.4 Low Power Test Compression for Multiple Scan Chain Designs .....	117
6.5 Future Work .....	118
Reference .....	120



# Table Captions

Table 2-1 Delay performance comparison between four different scan cells (in picoseconds) .....	36
Table 2-2 Area comparison of four scan schemes.....	37
Table 2-3 Routing overhead comparison of four scan schemes in terms of signal routings.....	38
Table 2-4 Simulated relative power consumption with respect to SFF for different transitions.....	39
Table 2-5 Overall comparison for four power-saving schemes.....	39
Table 2-6 Benchmark circuits used in our experiments.....	41
Table 2-7 Total/peak power reduction obtained for different approaches.....	42
Table 2-8 Test time and area overhead using the SMR and the small SMR compared with full scan.....	42
Table 3-1 Four different models used for estimating cost for bit flipping.....	51
Table 3-2 Experiment results on different cost models.....	51
Table 3-3 Comparison of traditional process and the proposed process on CSC and BPBTVD strategies on s5378.....	57
Table 3-4 Details on benchmark circuits.....	58
Table 3-5 Comparison on the encoding efficiency for different flows applied to MBFP.....	59
Table 3-6 Comparison the encoding efficiency for full scan with the proposed Cocktail Scan.....	59
Table 3-7 Comparison of data sizes of the proposed Cocktail Scan and previous works.....	60
Table 3-8 Power reduction obtained for RAS in terms of switching activity of SFFs.....	60
Table 3-9 Hardware overhead comparison between RAS and standard scan designs.....	62

Table 4-1	Compression comparison between different encoding methods.....	86
Table 4-2	Test speedups for BR and Adaptive Encoding under different number of scan chains, $m$ , and clock ratio, $r$ , between test clock and system clock.....	87
Table 4-3	Tradeoff between test power and test compression for different user defined values.....	88
Table 4-4	Data volume for Adaptive Encoding with different techniques.....	90
Table 5-1	Compression results for MDC (Mintest test set) and MDCGEN.....	107
Table 5-2	Compression comparison between the MDC scheme and other compression methods on Mintest test sets.....	107
Table 5-3	Data volume comparison between MDCGEN and other ATPG-dependent methods.....	108
Table 5-4	Normalized (a) average, (b) peak and (c) test energy comparisons between different compression methods.....	110
Table 5-5	Normalized (a) average, (b) peak and (c) test energy comparisons between MDCGEN and random patterns [37, 39, 45, 92-94] .....	111
Table 5-6	Comparison of data compression and test power for MDCGEN with the work FCN [62] .....	112
Table 5-7	Comparison of data compression and test power for MDCGEN on large-scale circuits.....	113
Table 5-8	Overall comparisons between the proposed scheme and other schemes...	114

# Figure Captions

Figure 1-1 An example for test points (a) Original circuit, (b) Using a CP for 0-injection and an OP for observation, (c) Using a 0/1 injection circuit, and (d) Using a multiplexer for 0/1 injection.....	2
Figure 1-2 Generic scan-based circuit.....	3
Figure 1-3 Different scan registers (a) Standard scan register, (b) Level-Sensitive Scan Design (LSSD) and (c) Random Access Scan (RAS) cell.....	4
Figure 1-4 BIST architecture.....	6
Figure 1-5 Prediction of test cost ( <i>Source: ITRS 2001</i> ) .....	7
Figure 1-6 Gate Count v.s. Test Time ( <i>Source: [4]</i> ) .....	8
Figure 1-7 An example shows that scan test application has higher transitions than functional transitions due to illegal state transitions. ....	9
Figure 1-8 Clock gating scheme [9] .....	11
Figure 1-9 MD-Scan method [11].....	11
Figure 1-10 Random access scan architecture (top) and the scan cell of it (bottom) .....	12
Figure 1-11 Scan path modification and scan cell reordering to reduce transitions...	13
Figure 1-12 Test vector inhibiting structure.....	15
Figure 1-13 LT-RTPG for low power testing.....	15
Figure 1-14 A scan cell with toggle suppression when mode = 1.....	15
Figure 1-15 Distribution of peak power during scan testing of s9234 benchmark circuit.....	17
Figure 1-16 A general test compression environment.....	19
Figure 1-17 XOR-type decoder used in SCC [37] .....	21
Figure 1-18 A generic example for dynamic reseeding schemes [46] .....	22
Figure 1-19 Example of Golomb coding for $m = 4$ .....	24
Figure 1-20 Golomb coding for a $T_{diff}$ .....	25

Figure 1-21 Dictionary-based coding.....	25
Figure 2-1 A 4x4 SM scan architecture.....	33
Figure 2-2 A simplified model to demonstrate the scan in/out operation of the 4x4 SM.....	33
Figure 2-3 Resettable circular shift register for low-power signal generator.....	34
Figure 2-4 (a) The conventional SFF; the proposed (b) positive polarity SMR, and (c) negative polarity SMR.....	35
Figure 2-5 Another implementation of SMR with smaller area.....	36
Figure 2-6 Four types of global signal connection.....	38
Figure 2-7 The layout of SMR.....	42
Figure 3-1 Random access scan architecture.....	46
Figure 3-2 The multiplexer based scan cell of RAS.....	46
Figure 3-3 Fault coverage curves for Segmented Random Scan Test.....	48
Figure 3-4 The proposed process for solving MBFP.....	49
Figure 3-5 The modified scan cell for RAS to support Test Response Abandonment.....	52
Figure 3-6 An example to explain Constraint Static Compression (CSC) and Bit-propagation Before Test Vector Dropping (BPBTVD).....	53
Figure 3-7 The proposed address shift register (3 bits) .....	54
Figure 3-8 The proposed Cocktail Scan flow.....	55
Figure 3-9 Test volume plots of each benchmark circuit for the full scan scheme and the Cocktail Scan scheme.....	61
Figure 3-10 Reducing routing area and signal skew by using hierarchical decoders rather than using a global decoder.....	63
Figure 4-1 Two 16-bit test patterns and their <i>diff</i> pattern for demonstration of RAS.....	66
Figure 4-2 Three encoding alternatives of using packet representation for the same pattern: one-packet, two-packet, and three-packet.....	68



Figure 4-3	Using <i>difference address</i> and reduced <i>data length</i> to improve encoding efficiency.....	70
Figure 4-4	The three main steps that the decoder machine does: (a) step 1: loads three head fields (b) step 2: configures test pattern (c) step 3: loads test pattern.....	71
Figure 4-5	Relationship between decoder's address and memory blocks, and implementation of the memory buffer to support updating memory block in a random access fashion.....	74
Figure 4-6	An example to demonstrate the decoding process in Step 2.....	76
Figure 4-7	Graphs to model test sequence ordering: (a) Direct edges for partially specified patterns (b) Undirected edges for fully specified patterns.....	78
Figure 4-8	Compressed data volume, $DV$ , v.s. the number of blocks to be replaced, $M$ , and the block size, $b$ .....	81
Figure 4-9	The block sizes, which give the best compression, are plotted with respect to test patterns for a test set for circuit <b>b19_1</b> for Adaptive Encoding scheme and BR scheme respectively. The number of don't care bits is also plotted for each pattern. ....	81
Figure 4-10	An example pattern with three different filling strategies which result in different number of flips and WTC's.....	85
Figure 4-11	Experimental results on test data compression, test energy and peak power (transitions) for different value of $a$ on circuit <b>b19_1</b> . The most suitable value for $a$ is 0.15 to obtain a 45.22% test compression, a 83.63% energy reduction and a 32.25% peak power reduction.....	89
Figure 5-1	(a) Proposed decoding architecture, (b) a decoding buffer with a dffs and its multilayer organization, (c) a switch box is used to support the two operations of a decoding buffer, and (d) the switching box implementation.....	95
Figure 5-2	An example to demonstrate shift and copy operations.....	96
Figure 5-3	Decoding flow of the decoder.....	97
Figure 5-4	Another example to show how to encode slices using Shift and Copy operations.....	99

Figure 5-5 Compression under two parameters:  $gs_1$  and  $p$  .....101

Figure 5-6 Plots of (a) weighted transition counts (WTC), and (b) peak transitions,  
with respect to  $gs_1$  in terms of  $p$  .....103

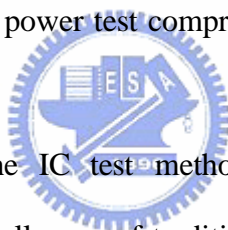
Figure 5-7 The proposed flow of an ATPG, MDCGEN, incorporated with MDC  
strategy.....104

Figure 5-8 Power profiles for each pattern of two circuits: s15850 and s35932 for  
MDCGEN and random-filling patterns.....111



# Chapter 1 Introduction

As IC designs become more complex, traditional test strategies become inefficient and several new problems have risen. Two of them are the increasing test data volume and test power due to ever increasing size of chip. First, since large test data volume increases the test time, test cost is also getting higher, sometimes even higher than the manufacturing cost. Second, since large number of non-functional test patterns are applied during scan testing, which is the most popular design-for-testability (DFT) used in the industry, large test power is observed during scan shifting and this may damage the circuit-under-test (CUT) or reduce its reliability. This thesis mainly focuses on solving those two test problems. We propose one novel low power test architecture and invent three low power test compression schemes to both reduce test data volume and test power.



This chapter first reviews the IC test methodology and design-for-testability techniques, then describes test challenges of traditional approaches by addressing the need for low power test compression which is the focus of this work. Afterward, it reviews previous works on test power reduction techniques, test data compression techniques and methods which simultaneously reduce the test power and data volume. At the end, it overviews and describes the organization of this dissertation.

## 1.1 IC Test Methodology and Design-for-Testability

Testing of an IC is an experiment in which the IC is exercised and its results are analyzed to see if the function of the IC is correct. Since during the manufacturing process of ICs, there is a certain chance that some defects, which cause function of the IC fail or out of specification, exist on the IC due to uncertainty of controlling process factors. The goal of IC test is to screen out the chips which fail completely or to meet

specifications. A simple way to test ICs is to apply function patterns which are derived from and supplied by designers. However, as the size and complexity of today's IC become so huge, it is hard to guarantee "all" functions of the IC by just exercising the IC's function patterns. This enables people to resort to adopt design-for-testability (DFT) techniques to design ICs. In the following, some of DFT techniques which are used often are briefly described:

### 1.1.1 Test Points

Two types of test points can be used, namely, control test points (CP) and observation test points (OP). The aim of the former test points is to increase the controllability and that of the latter test points is to increase the observability of the circuit. Figure 1.1 (a) shows an example circuit for this technique. *C1* is a sub-circuit connected to *C2* via a two-input NOR gate. In Figure 1.1 (b), if we replace the NOR with a three-input NOR gate, of which one input, *CP*, is controllable, we can inject a 0 to *C2* through *G\** by setting *CP* to a 1. Similarly, we can use Figure 1.1 (c) for 0/1 injection for *C2*, which is the same as Figure 1.1 (d) using a multiplexer for 0/1

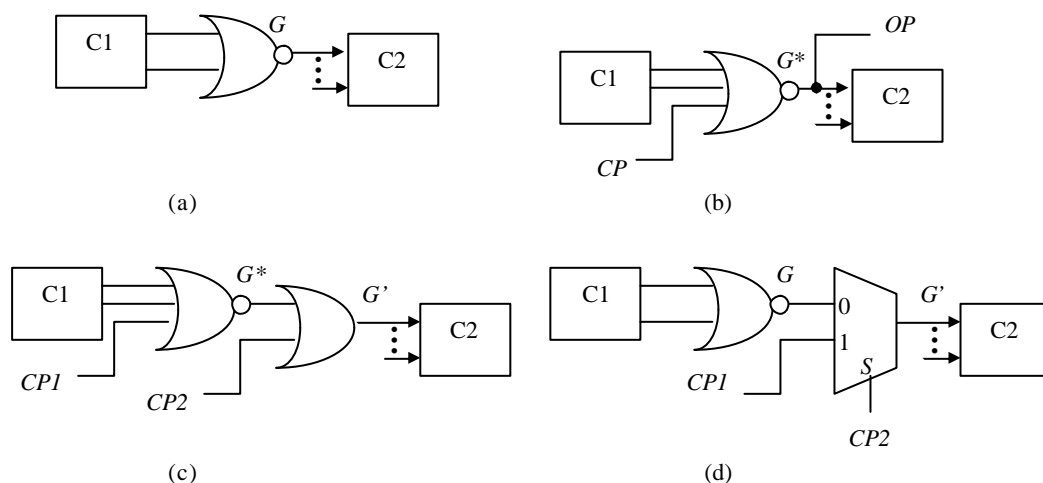


Figure 1.1 An example for test points (a) Original circuit, (b) Using a CP for 0-injection and an OP for observation, (c) Using a 0/1 injection circuit, and (d) Using a multiplexer for 0/1 injection

injection with two control points, *CPI* and *CP2*.

### 1.1.2 Scan Design

Scan design replaces register of circuits using scan register (SR). A scan register is a register with both shift and parallel-load capability and it has many different implementations. With SR, we can easily control and observe registers, which are deeply located inside circuits and are not accessible from primary I/O pins. They are used as pseudo-primary input/output during testing. A generic scan-based design is shown in Figure 2. Suppose a circuit has  $n$  primary inputs,  $m$  primary outputs and  $k$  registers, the registers are replaced by SR and connected, for example, in serial. Then an automatic test pattern generator (ATPG) is employed to produce test patterns, which have  $(n + k)$  bits for each input stimulus and  $(m + k)$  bits for each output response. Scan-based approach is a most popular technique in industry and this thesis also deals with problems of test volume and power reduction for scan-based designs.

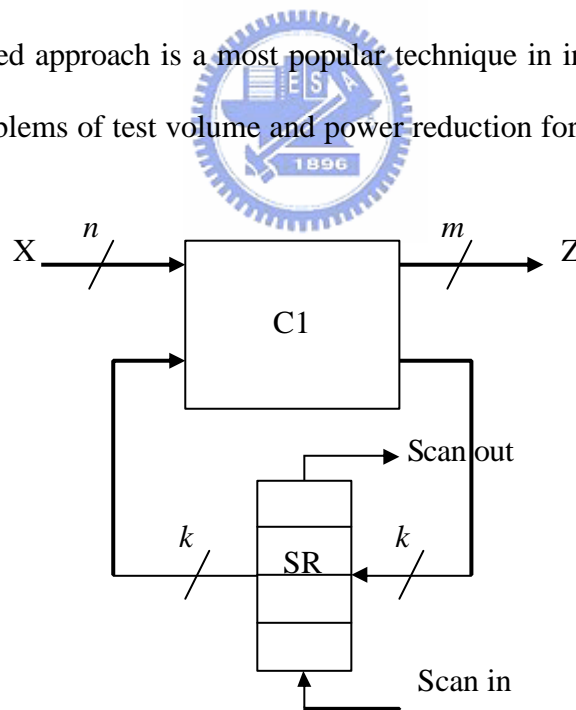


Figure 1.2 Generic scan-based circuit

Scan design has been studied for decades and various SR implementations have been proposed for various purposes. A normal SR is composed of a multiplexer and a register, which is shown in Figure 1.3 (a). Several SRs are serially connected via *SI*

and  $SO$  pins to form a scan chain. When the scan chain operates in scan shift mode, i.e.,  $TM = 1$ , test data are delivered to  $SI$ . For each test clock, data are shifted one bit from one SR to its next connected SR. Once a test pattern is completely shifted to SRs,  $TM$  changes to 0 and clock asserts to capture the responses of the circuit-under-test (CUT). IBM has proposed Level-Sensitive Scan Design (LSSD) [1], which uses a polarity-hold, hazard-free, and level-sensitive latch, as shown in Figure 1.3 (b). To obtain race-free operation, clocks  $CK1$  and  $CK3$  as well as  $CK2$  and  $CK3$  are non-overlapping. Random Access Scan (RAS) uses an addressable SR, just like memory. To be able to address an RAS cell, a decoder is embedded in the CUT to provide individual  $SE$  signal for each RAS cell. When  $SE = 1$ , a bit is shifted into the RAS cell. On the contrary, the RAS cell keeps its data. RAS cells can thus provide low power testing since, unlike like serial-scan architecture, only one scan cell is switching during test pattern shifting. We have also proposed a special scan cell for

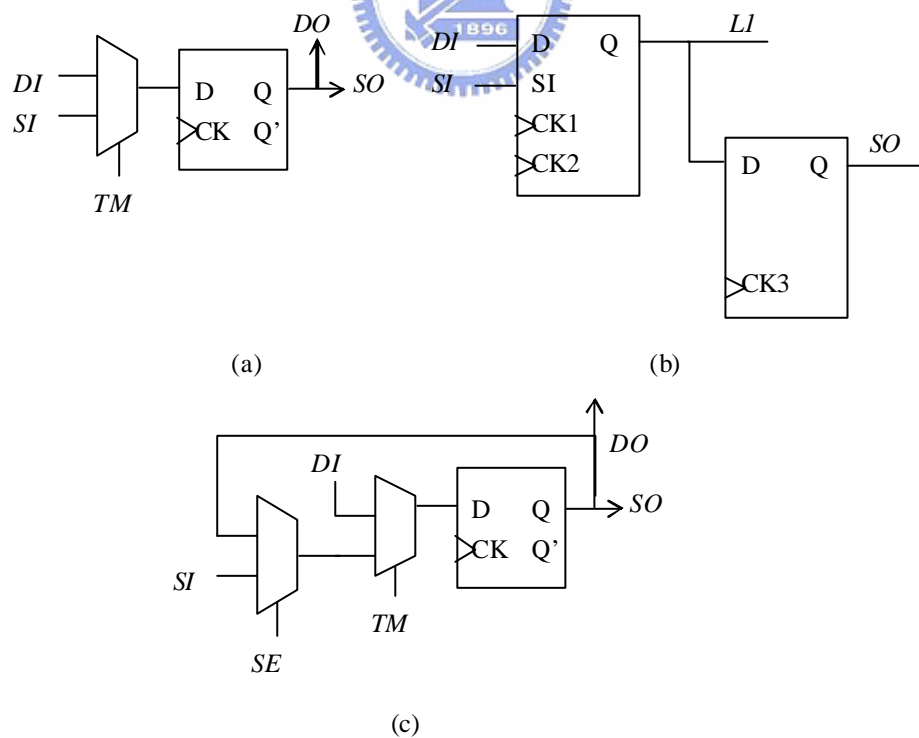


Figure 1.3 Different scan registers (a) Standard scan register, (b) Level-Sensitive Scan Design (LSSD) and (c) Random Access Scan (RAS) cell

low power testing, which will be described later in this thesis.

### 1.1.3 BIST Methodology

Built-in-self-test (BIST) is a testing methodology that a circuit has a capability to test itself. Compared with scan methodology, BIST substantially reduces test application time since it applies test patterns to CUT every test cycle. Usually a linear feedback shift register (LFSR) is used as a pattern generator and a multiple input shift register (MISR) is used as a signature analyzer to decide “pass” or “fail”. A general BIST architecture is shown in Figure 1.4. The advantages of BIST are as follows. First, using on-chip hardware, clock speed can run at-speed without the need of high speed clock generated from an expensive ATE. Second, it also increases the number of patterns applied so that the probability to detect random defects that are not modeled is increased. Moreover, BIST also reduces the number of pin counts required of the ATE. BIST applies large amount of random patterns to detect faults of CUT. However, due to hard-to-detect faults of circuit, BIST achieves lower fault coverage than deterministic test patterns. Therefore, test points or other techniques such as multiple-polynomial LFSR [3] are used to detect those hard-to-detect faults.

The BIST methodology can be applied to not only functional block but also memory. A system-on-a-chip (SoC) can thus have many blocks already made with BIST. Although all BISTed circuits can be run simultaneously to reduce test time, this will consume very large test power. Test scheduling should be used to make a tradeoff between testing time and test power. Although BIST may have area and performance overhead, it reduces the test effort of a complex system that integrates many function units and memories.

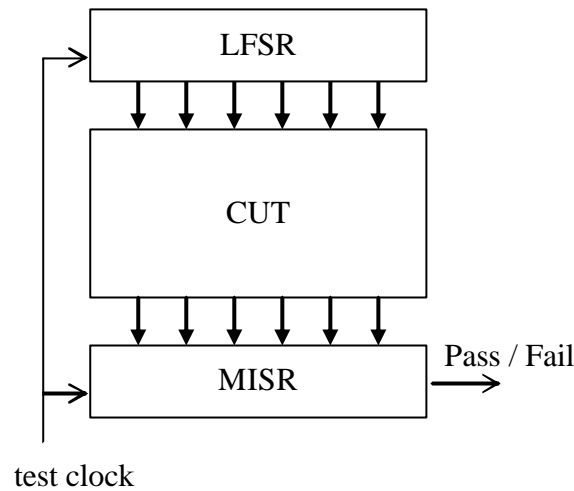


Figure 1.4 BIST architecture

## 1.2 IC Test Challenges

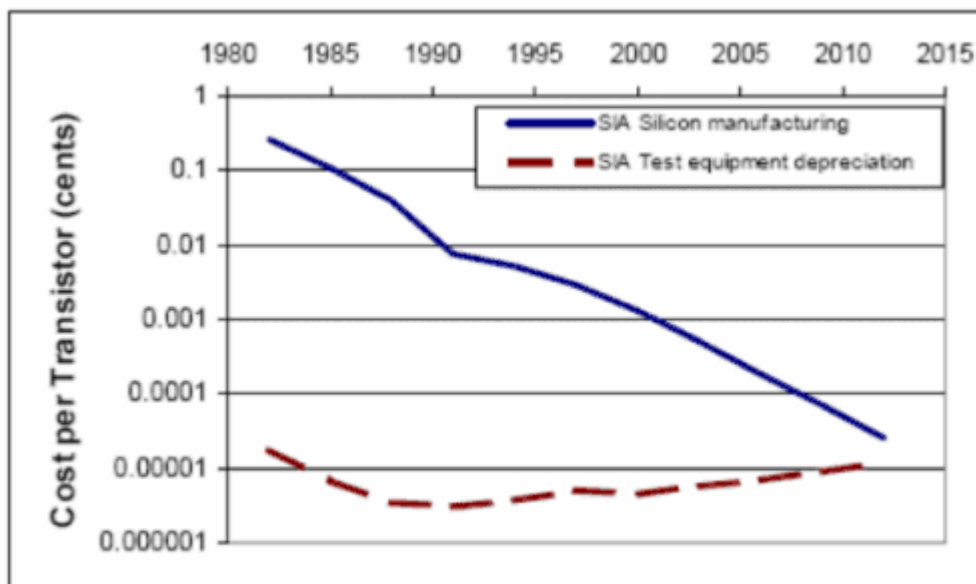
As today's technique moves to nanometer era, designs have become more complex and advanced so that more challenges are introduced. IC vendors are experiencing a decline in yield at 0.13um and the problem is only worse at 90nm, 65nm and beyond. This is due to the increasing side effects such as interconnect delay, crosstalk or process variation accompanying with smaller geometries. Besides, since the size of modern designs is increasing, the size of test set and test power are also getting larger.

### 1.2.1 Test Cost

As predicted in ITRS 2001 [4] as shown in Figure 1.5, the test cost of a transistor will equal to its manufacturing cost in about 2010. Test cost is increasing since test time is getting longer due to huge test data volume. This is explained as follows. Suppose a circuit has  $G$  gate counts and also suppose the percentage of registers of the circuit is a constant,  $p$ . Test data volume (TDV) is  $T \times G \times p$ . According to Moore's Law, circuit size is double every 18 months. Thus TDV is also doubled every 18 months. To avoid reload of test data, tester's memory have to be large enough to



contain all test data; besides, to reduce test time, the number of pins of tester should be large, too. As reported in [5], as shown in Figure 1.6, the test time significantly grows due to large test data volume with the increase of the size of design if the number of test channels of tester keeps the same. Therefore, IC tester will require larger memory and higher bandwidth to accommodate the increasing data volume and reduce test time. However, the memory of tester is very expensive and cost of a tester is proportional to the number of pins and frequency it supports. For high-end functional tester, the cost is \$8-10K/pin [6]. To reduce test cost, novel testing strategy such as test data compression method needs to be developed so that low-cost testers can be used and the problem of high test cost is alleviated.



**1997 Microprocessor Cost of Test Trend Model**

Figure 1.5 Prediction of test cost (Source: ITRS 2001)

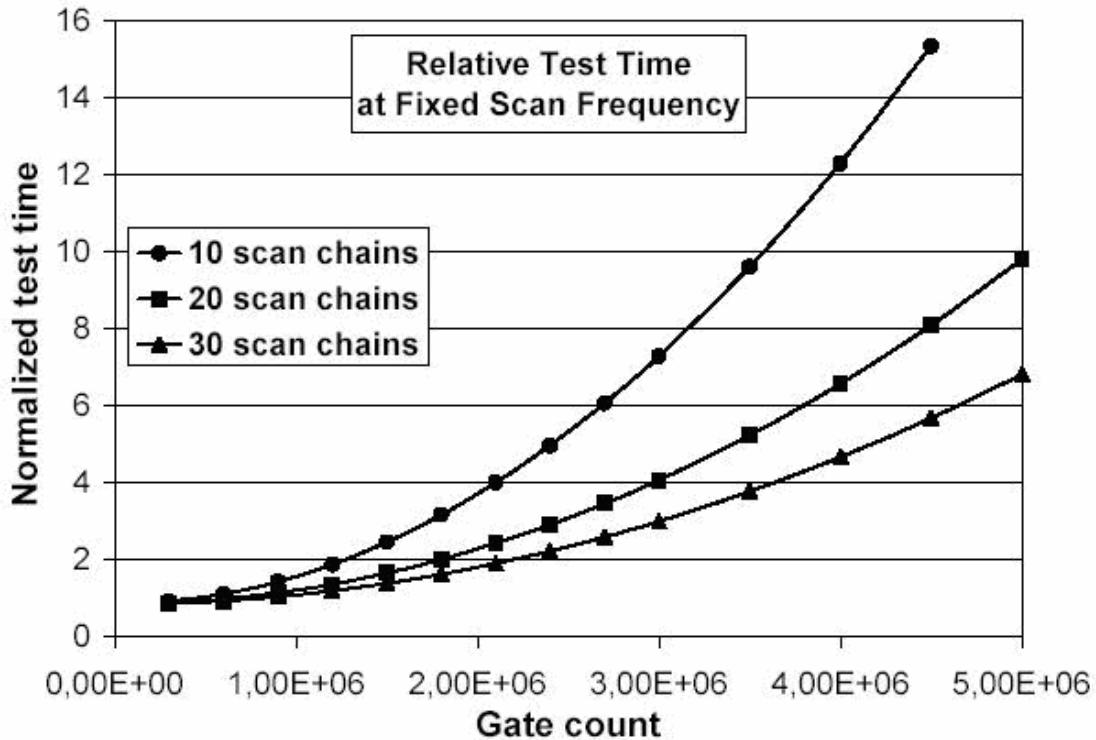


Figure 1.6 Gate Count v.s. Test Time (Source: [4])

### 1.2.2 Test Power



Power dissipation of chips is a very important fact to determine the life time of electronic devices, such as mobile phones or laptops. Although today designers have successfully invented many low power devices with very long life time, there is still a power problem during test application. The reasons are as follows:

- High Switching Activity during Pattern Shifting:** Due to low accessibility from the limited I/O pins to internal part of the CUT, DFT technique such as serial scan is usually used to reduce the test complexity by increase the controllability and observability. However, the test patterns produced from ATPG have very low correlation for the states of registers of the design. This causes higher switching activity during test application. It was reported in [7] that a VLSI chip can dissipate up to three times higher power during testing when compared to normal operation. This is demonstrated in the example in Figure 1.7,

in which the state transition of the circuit is also shown. We can see that functional transition has only 1 or 2 transitions. However, given that scan in pattern “010” and scan out pattern “100”, as scan DFT is employed, test transition causes 2 or 3 transitions. Thus test power is higher than functional power in this example (test power has been proven to be proportional to the number of transitions [8]).

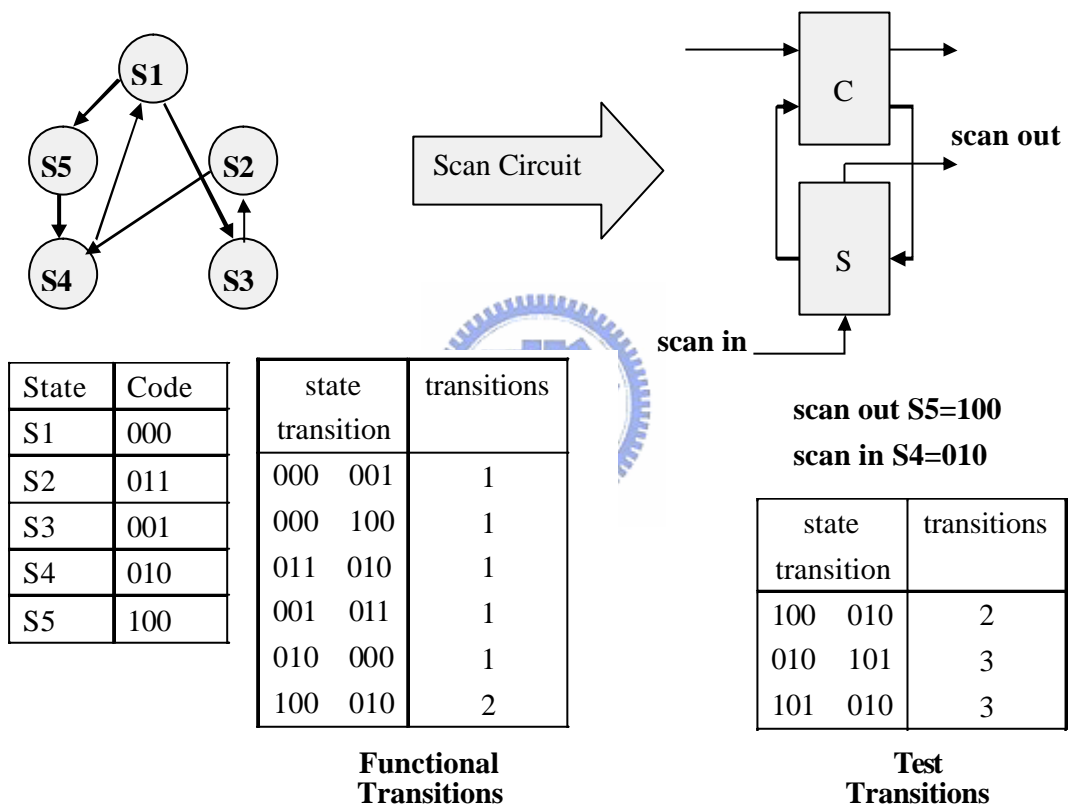


Figure 1.7 An example shows that scan test application has higher transitions than functional transitions due to illegal state transitions.

- **Concurrent Execution of Circuit Components:** For complex circuits, they control different functional unit such as memories or multifunctional execution units for power management. Also, to reduce power dissipation during functional operation, part of design may be gated or shut down; however, during test application, all parts of design simultaneously operate so that test time can be

minimized. This causes significantly high switching activity and test power, which may exceed the limitation of CUT.

Since power dissipation of VLSI circuits should be constrained by the power dissipation of functional operation [9], aforementioned high switch activity and concurrent execution during test application cause high test power and should be taken care in order to avoid reducing the reliability of CUTs. Reliability will be reduced because high test power causes high temperature and electromigration. Besides, power and ground noises are also induced by high test power, leading to yield loss. Therefore, testing power should be constrained not too high to avoid those problems.

### **1.3 Test Strategies for Test Power Reduction**

Power dissipation of circuit consists of dynamic power, leakage power, and short circuit power. The source of test power comes mostly from the dynamic switching power of circuit causing by pattern shifting. In the following, to evaluate test power at an abstract level so as to avoid complex circuit simulation, an estimation method proposed in [8], which shows that test power is proportional to the number of transitions of CUT, is used. Therefore, the basic concept for low power testing is to reduce the number of transitions (Total Power) and the peak number of transitions at any instance (Peak Power).

#### **1.3.1 Low Power DFT Techniques**

Low Power DFT techniques use additional circuits or modify of the CUT to achieve low power testing. It thus may have some design overhead such as additional area, longer routing wires or performance degradation. Clock gating technique uses two clocks with half speed of the standard scan clock [9] as shown in Figure 1.8. The

original scan clock,  $Sclk$ , is replaced with two clocks,  $Sclk1$  and  $Sclk2$ , and the scan path is also divided into two sub scan paths. With this scheme, peak power is reduced since only half of registers are active at any instance but the test time is the same as original one. The clock tree of these two clocks has to be carefully balanced to avoid clock skew between them. A similar scheme called Adapting Scan, which divided scan path into  $N$  sub scan paths and activates only one at any time, is proposed in [10]. By using several clocks with different duty cycle, MD-Scan provides a method to avoid the power supply voltage drop causing by simultaneously switching of scan cells [11]. In Figure 1.9, each scan chain has its own scan clock. Therefore, each scan chain updates at different time and avoids simultaneously switching.

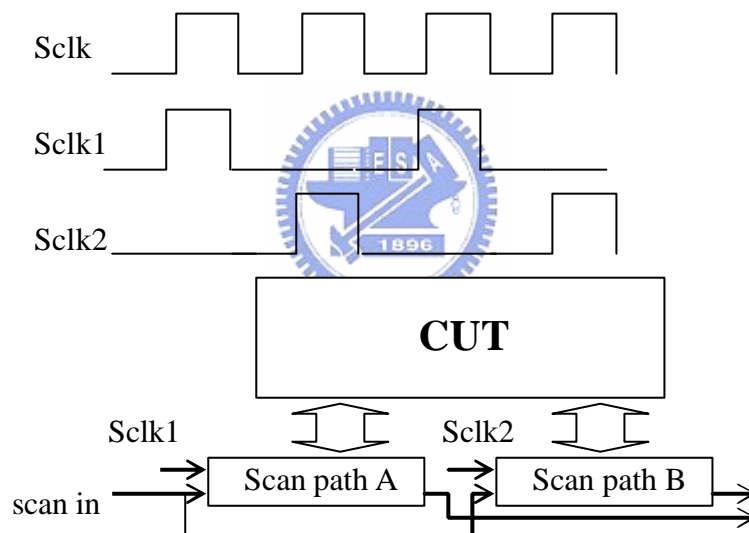


Figure 1.8 Clock gating scheme [9]

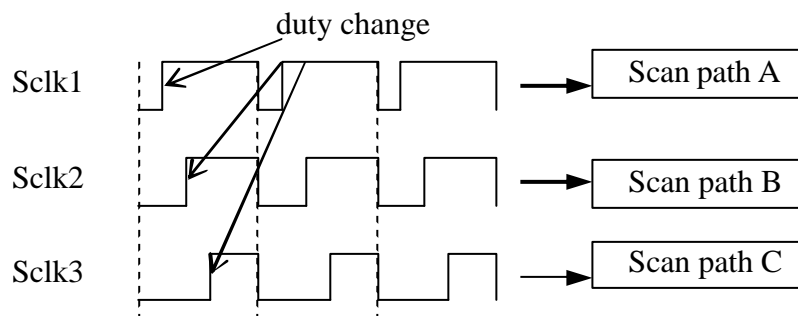


Figure 1.9 MD-Scan method [11]

Several novel scan architectures have been proposed for low power testing [2, 12-13]. Random Access Scan (RAS) [2] is suitable for low power testing since this architecture toggles one scan cell each scan clock while standard scan approach activates all scan cells simultaneously. It also provides a solution for reduction of test data and test time, which will be described later. In [12], token scan architecture with token gating cells was proposed, which greatly reduces switching activity of scan cells. The use of token gating cells also reduces the power consumption of clock tree. It successfully obtains large power reduction at the expense of area and signal routing overhead. Double Tree is presented using a binary tree scan structure with hierarchical clock control logic to reduce the shift length and switching activity of scan cells [13]. However, this method has complex clock tree routing.

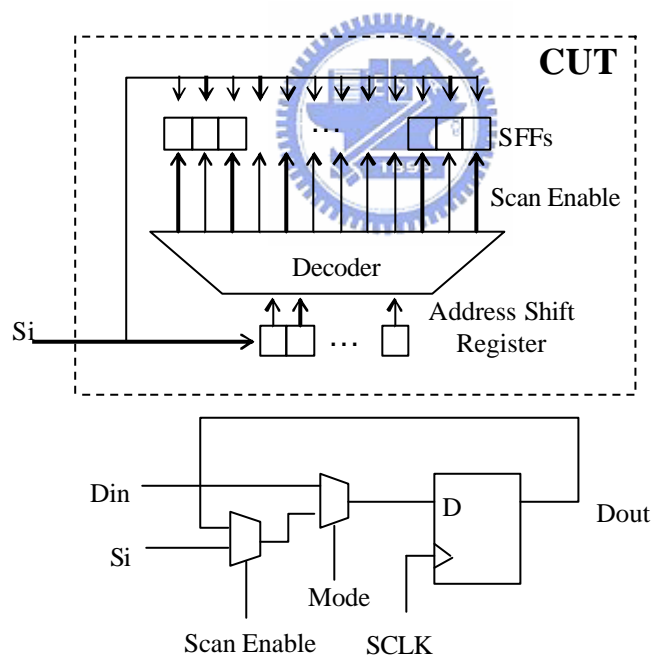


Figure 1.10 Random access scan architecture (top) and the scan cell of it (bottom)

By analyzing the transition frequency of a certain test set, low power test can be achieved by reordering scan cells and modifying the scan path [14-17]. This can be explained in Figure 1.11. For scan path modification, suppose a test stimulus “00111”

is going to shift to a scan path. The fifth and the fourth scan cell will both cause a transition. If we insert an inverter at the output of the fourth scan cell, after a test stimulus “00000” is shifted, the same test pattern is stored in scan cells as above. However, the test stimulus “00000” will not cause any transition and thus the number of transitions is reduced. For scan cell reordering method, a test set is first analyzed to calculate the transition frequency of each scan cell when shifting to the next one. In the example, the second scan cell has probability of 0.6 when shifting to the first one, that is, the probability is equivalent to 0.4 when an inverter is inserted at the output of the second cell. After reordering, the transition frequency of scan cells is further reduced. Orailoglu et. al. systematically analyze the impact on inserting XOR gates along scan path and show the transformation for XOR gate insertion. Their result shows significant reduction on testing power [15]. Virazel et. al. report that scan reordering without wire routing consideration result in complex routing of scan path and increase additional area [17]. They thus proposed a routing constrained scan cell reordering method.

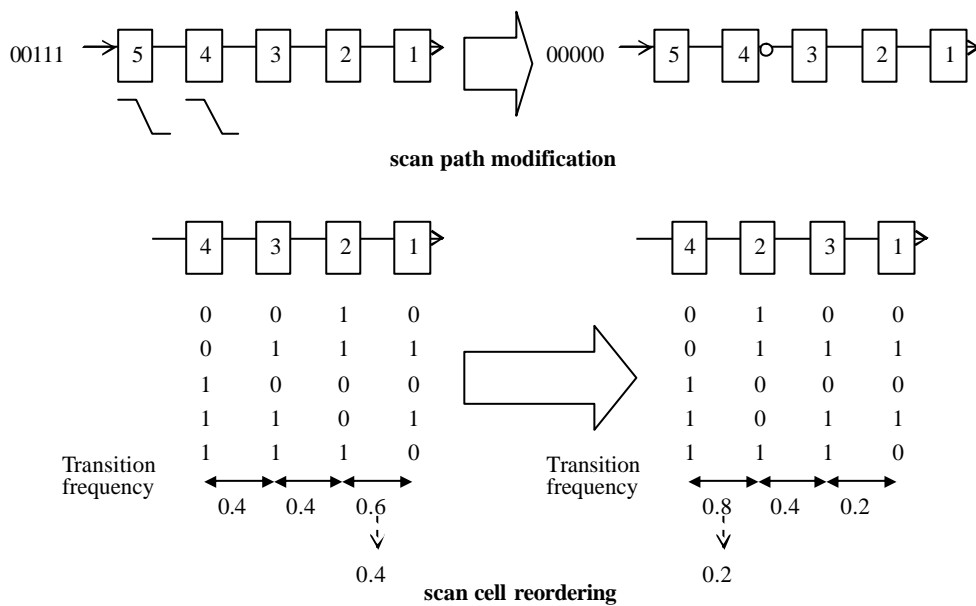


Figure 1.11 Scan path modification and scan cell reordering to reduce transitions

For BIST test methodology, the clock runs at system speed causing very large test power during pattern shift. A technique called test vector inhibiting is proposed to reduce the switching activity of the CUT [18]. The inhibiting circuitry inhibits useless patterns from shifting to the CUT, i.e. those patterns that do not detect any new faults are inhibited. To avoid producing test patterns with high switch activity from LFSR, low transition random test pattern generator (LT-RTPG) is proposed [19]. The idea is to connect some of the outputs of LFSR to an AND gate to generate low transition random test patterns. For example in Figure 1.13, suppose that the probability of being 1 of each flip-flop is 0.5, the output of the AND gate has a transition probability of 1/8. Therefore, the number of transitions of generated test pattern is reduced. A similar approach is a dual-speed LFSR (DS-LFSR) proposed by Gupta et. al. [20]. Nicolici refines those techniques by designing multiple polynomial LFSRs for low power mixed-mode BIST. The method generates “mask pattern” to reduce the number of transitions in the scan chain by AND (OR) composition. To reduce the switch activity during test, Basturkmen et. al. proposed a low power pseudo-random BIST architecture [21]. In that work, scan chains are divided into groups and scan cells of each group are active only when its group enable signal is active. Counters are used to produce group enable signals. Although disabling a subset of scan chains reduces both peak and average power, it has an adverse effect on fault coverage. Therefore, the number of test patterns may increase. Ghosh et. al. proposed a SFNC scan cell to be used in a low power BIST environment [22]. Their approach uses weighted pseudo-random pattern and achieves better reduction in test power than previous methods. However, the overhead of the proposed scan cell is very large.



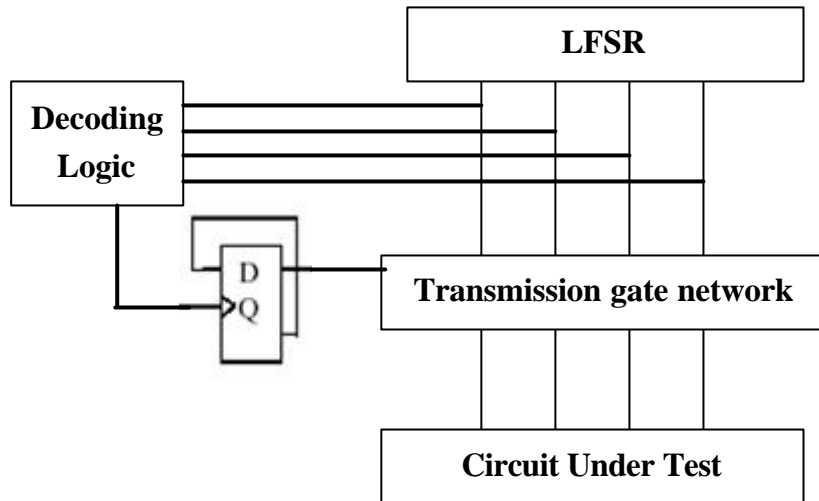


Figure 1.12 Test vector inhibiting structure

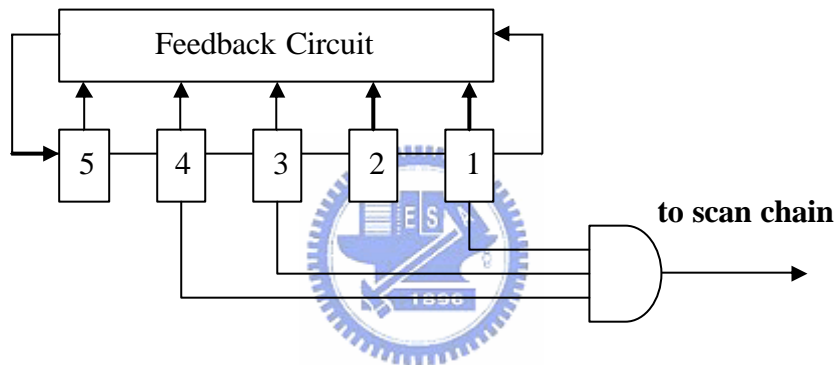


Figure 1.13 LT-RTPG for low power testing

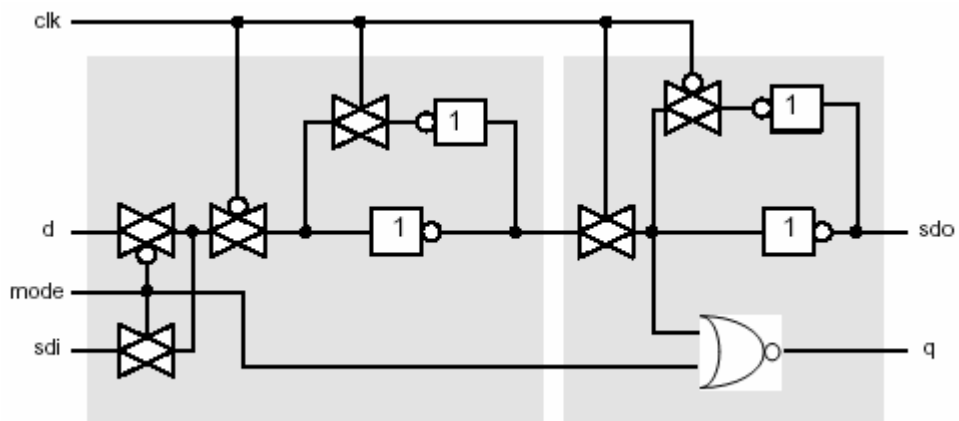


Figure 1.14 A scan cell with toggle suppression when mode = 1

Since it is reported in [22] that about 70% of power dissipation during test is consumed by the CUT, it is thus important to reduce the switch activity of the CUT. A

simple way to reduce its switch activity during pattern shifting is to make scan cell with toggle suppression. This can be achieved by adding a NOR2 gate at the q output of scan cell, as shown in Figure 1.14. When mode = 1, the CUT is not affected when pattern is shifting, but this technique adds extra delay along the functional path. The performance of the CUT is thus reduced.

### 1.3.2 Modifying Test Patterns for Low Power Test

This section introduces methods belong to another category used for low power test without circuit modification. The idea is to process test patterns before they are applied to scan chains so as to minimize switch activity.

Static compaction used in ATPG tools aims to reduce the number of test patterns. By incorporating low power techniques to this procedure can also produce patterns suitable for low power testing. A common way is to fill the unspecified bits in test cubes with a *minimum transition fill* (MT-fill) [8]. By test cube, we mean test patterns containing unspecified bits. For each string of X' s in a test cube, if the specified bits on either side of the string have the same value, then the string of X' s should be filled with that value to minimize the number of transitions. If they have opposite values, then it doesn' t matter which value the string of X' s is filled with. For example, when filling 0XX01X1X0, the first two X' s should be filled with 0' s, the third X should be filled with a 1, and the last X could be filled randomly with either 0 or 1. The resulting test cube with MT-fill is 00001111(0)0. While MT-fill minimizes power, the drawback is that it may not be as effective as R-fill for detecting additional faults. Therefore, the number of test pattern usually becomes larger. Another technique to reduce test power is to use power-aware ATPG tools [23-24]

Test power contains shift-in, shift-out and capture power. The above MT-fill technique only reduces the shift-in power but not the shift-out and the capture power.

The ATPG approach can be used to reduce the three test power but its pattern generation becomes more complicated. Although capture power dissipation has less impact on the total heat dissipation than shift power dissipation, it may nonetheless cause significant yield loss [25]. To reduce the capture power, an ATPG technique has been proposed in [26]. Low power testing is even more important for at-speed delay test to avoid IR-drop and yield loss [27-28]

Although most low power test methods aim to reduce test power as much as possible, Sankaralingam et. al. proposed a method by handling only the test patterns with test power violation [29]. As reported in [29], the peak power distribution is like a normal curve for most circuits. For example, the distribution of peak power during each clock cycle of scan testing of s9234 circuit is shown in Figure 1.15. They analyze test patterns and find out those with scan-in, scan-out and capture problem. The problem means peak power violation occurs during scan-in, scan-out or scan capture. Then they try to solve the problems by altering test patterns.

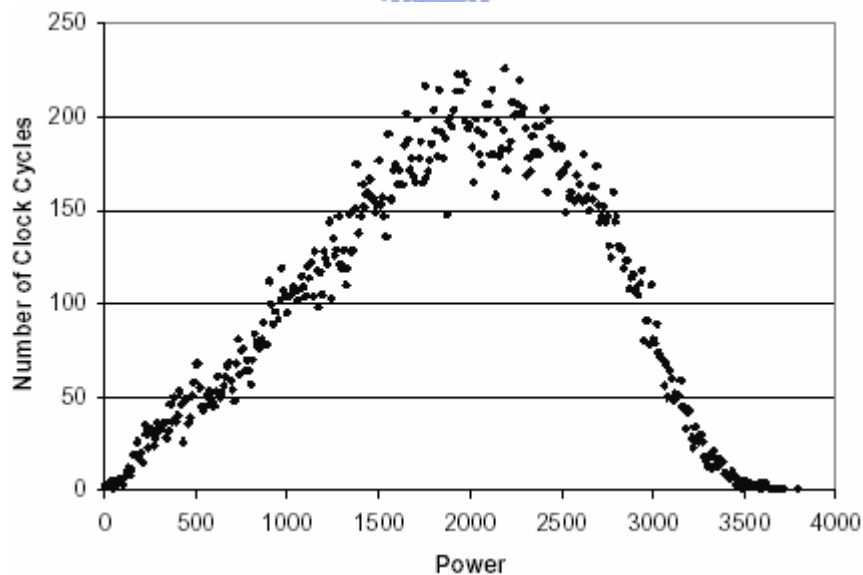


Figure 1.15 Distribution of peak power during scan testing of s9234 benchmark circuit

## 1.4 Test Strategies for Test Data Reduction

As describe above, as circuit size grows, the volume of test data also grows linearly. Large test data volume thus increase the test time and test cost. Methods to reduce test data volume are introduced below. The first category lies in ATPG tools and the second one uses on-chip circuitry, which is usually called a decoder or a decompressor, to decode compressed test set.

### 1.4.1 Automatic Test Pattern Generation (ATPG) Approaches

In order to reduce test application time, test compaction is required to achieve maximum fault coverage with a smallest possible number of test patterns [30-31]. During pattern generation of ATPG, it uses two kinds of test compaction techniques, which are static and dynamic compaction, to combine test cubes to reduce the number of test patterns. Static compaction merges compatible test patterns after ATPG while dynamic compaction uses unspecified bit in a test cube to detect other faults during ATPG. Usually, dynamic compaction is more effective to reduce the number of test patterns than static compaction.

To further reduce the number of test pattern, ATPG may also include some simulation approaches after ATPG. Post processing of test pattern can use reserve order simulation or shuffling to drop those test patterns that do not detect any additional faults [32].

### 1.4.2 Test Compression Approaches with On-Chip Circuitry

Nowadays, with only ATPG technique to reduce the volume of test data becomes insufficient for large size designs. Test compression techniques have been proven to be very effective and practically used in advanced designs. The concept of test data compression can be described in Figure 1.16. A decompressor/decoder and a

compressor/encoder are put between the CUT and the ATE. Test set stored in ATE is first compressed by certain encoding methods using software. Then, during testing, encoded data is sent to the decoder, decoded by the decoder and then sent to the CUT. Test response of the CUT is also compressed to a signature and sent to the ATE to see if any fault exists. In this thesis, we only focus on the decoder part. Readers interested in the part of output compactor can see some works in [33-34]. In the following, previous works on test data compression are classified into different categories.

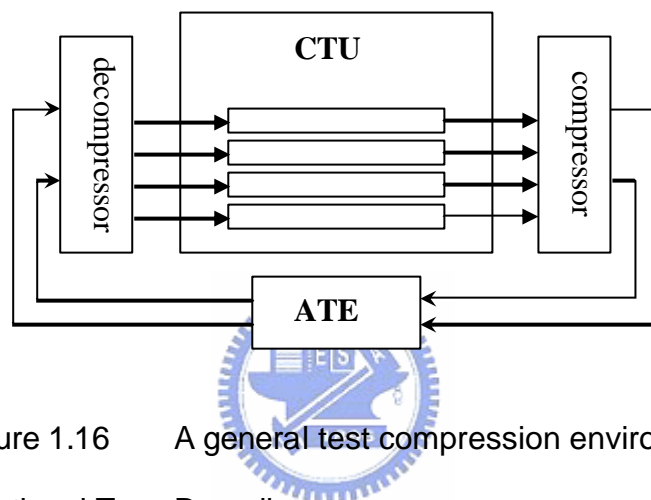


Figure 1.16 A general test compression environment

#### 1.4.2.1 Combinational Type Decoding

The first category uses decoders composed of combinational gates or interconnects to transmit data. Lee et. al. first introduces a concept of “Test Pattern Broadcasting” to reduce test data volume [35]. They proposed to use a single input to support multiple scan chains and suggested some configurations. Later, *Illinois Scan Architecture* (ISA), using two modes, i.e., the broadcast mode and the serial mode, reduces both test time and the test pins between CUT and the ATE [36]. An ISA example is shown in Figure 1.17. Since a majority of the bits in ATPG patterns are don't care bits, there are chances that these segments will have compatible vectors (not having opposite care bits in one location). In this case, all segments of a given chain are configured in broadcast mode to read the same vector. This speeds up the test vector loading time and reduces the data volume by a factor equivalent to the number of segments. If the

segments have conflict value, scan segments are configured as a single scan chain using multiplexers, which is called serial mode. Then a test pattern can be serially shifted.

In [37], *Scan Chain Concealment* (SCC) uses a decoder composed by XOR gates to drive a large number of scan chains and the authors proposed a dedicated ATPG for test compression. An example shows its implementation in Figure 1.17. For this scheme, it can be seen from the figure that the decoder has three inputs to drive five scan chains. With three inputs, only  $2^3=8$  combinations can be generated. However,  $2^5=32$  combinations may exist in the scan chains. Although unspecified bits in test patterns can be exploited, it still may fail to produce some combinations for decoding. This is especially true for test pattern with only few unspecified bits. The problem is due to the limitation of the decoder and fault coverage may decrease if this problem is not handled. In [38], the authors proposed a decoder using mapping logic to solve the need of serial load for test patterns of ISA. The mapping logic consisting of muxes, wires and inverters and is synthesized by a conflict analysis-based DFT synthesis. Tang et. al. [39] also proposed a scheme to use switch configurations to deliver test patterns and provided a method to determine the optimum switch configuration, which uses omega networks and allows CUT-independent design.



simultaneously [43-45]. A generic example in Figure 1.18 shows that  $b$  channels from the tester injects  $b$  free variables into LFSR through a combinational XOR network to load the  $n$  scan chains. The advantages of dynamic reseeding over static reseeding are that it allows a continuous flow decompression (the tester is never idle during decompression) and the size of LFSR is smaller. Rajski et al. proposed to use ring generator to improve the encoding efficiency and provide high performance [45]. They generate test cubes suitable to be encoded by the ring generator by constraining their ATPG.

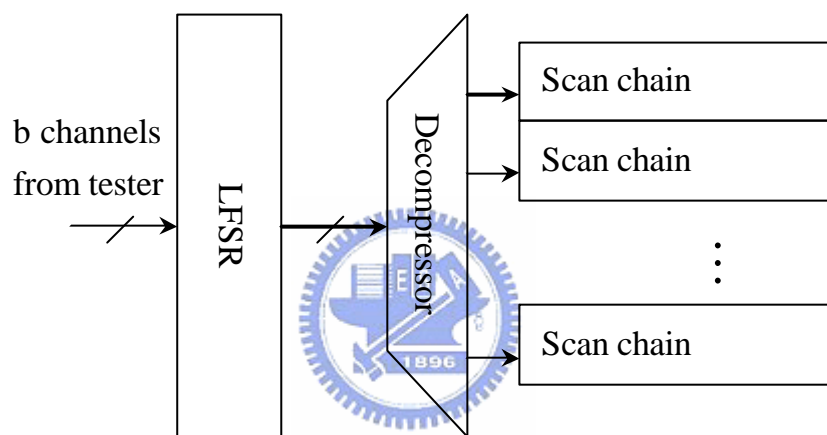


Figure 1.18 A generic example for dynamic reseeding schemes [46]

#### 1.4.2.3 Bit Flipping Decoding

One category of test data reduction techniques is to exploit test vector correlation and minimize the bit flips between consequent test patterns, since, due to the circuit structure dependency, correlation between test vectors are usually high and only small amount of different bits are needed to be flipped from one test vector to another one [47-51]. As for those works, they intend to exploit test vector correlation and minimize the bit flips between consequent test patterns. For those works, since test vector correlation by reordering test patterns is exploited to minimize bit flips between consequent test patterns, it is referred as Minimum Bit Flips Problem



(MBFP) – based scan strategies [78]. Combining test reordering and don't care bits can provide more reduction of test data volume, test time and test power. In [49], the authors proposed a scan architecture consisting of a decoder, a Decoder Shifter Register (DSR) and Decoder Output Register (DOR). Test data compression can be carried out by encoding the different bits between consequent bit slices and reordering the transition addresses. In [47], Random Access Scan (RAS) [2] scheme was used for which the test sequence was reordered to minimize bit flips to achieve high efficiency test and test power reduction. A similar method using decoder to flip bit was used in [48]. Test cubes are first statically compacted and then followed by reordering. Karimi et. al. proposed a new scan cell and suggested two decoding configurations: off-chip (to embed the decompression hardware into the ATE) and on-chip (to embed the decompression hardware into the CUT) [50]. Jas and Touba exploited the embedded hardware for test decompression [51]. For that proposed scheme, the test program, test vectors and replacement words are initially transmitted from the ATE to the internal memory of the CUT. During scan testing, an embedded processor of the CUT iteratively runs its test program to “configure” the test pattern with replacement words in the memory and then loads the “configured” pattern into scan chains. After testing, the test results are captured. The test data are continuously loaded from the ATE to the internal memory during testing. Test data in the memory are organized as blocks to be processed by the embedded processor. It has been shown that the size of block affects the compression of test data.

#### 1.4.2.4 Code-based Schemes

Code-based schemes use data compression schemes to encode the test cubes. The original data is partitioned into symbols, and then each symbol is replaced with a code word to form the compressed data. During decoding, a decoder is used to convert each code word in the compressed data back into the test patterns.

The characteristic of the schemes in this category is that the maximum compression is bounded by the entropy of data, and usually it can be predicted by analyzing test data. Ref [52] did a comprehensive study on these compression schemes, including Golomb codes [53], Huffman codes [54], VIHC codes [55], and FDR [56], etc. It also showed that the compression obtained by the VIHC scheme approaches to the entropy bound. For Golomb scheme, a “difference vector”  $T_{diff}$  (by XOR each bits of the two patterns) determined from two successively applied test patterns is compressed. In the beginning, a group size,  $m$ , is determined, and then the runs of 0s in  $T_{diff}$  are mapped to codeword by using Figure 1.19. Figure 1.20 shows an example of encoding a  $T_{diff}$ . The drawback for this scheme is that it requires separate CSRs and thereby increases hardware overhead. Huffman code is a statistical coding scheme, which codes each symbol based on each symbol’s frequency of occurrence. It assigns shorter code words to symbols that occur more frequently, and longer code words to those that occur less frequently. This strategy can minimize the average length of a code word.

Group	Run-length	Group prefix	Tail	Codeword
$A_1$	0	0	00	000
	1		01	001
	2		10	010
	3		11	011
$A_2$	4	10	00	1000
	5		01	1001
	6		10	1010
	7		11	1011
$A_3$	8	110	00	11000
	9		01	11001
	10		10	11010
	11		11	11011
...	...	...	...	...

Figure 1.19 Example of Golomb coding for  $m = 4$

$T_{diff} =$	0001	000001	1	00001	00001	0000001	001	00000001	001
0 runs	3	5	0	4	4	6	2	7	2

Encoded = 011 1001 000 1000 1000 1010 010 1011 001 (32 bits)

Figure 1.20 Golomb coding for a  $T_{diff}$

The advantage of the low bandwidth for these schemes allows the use of a low-cost ATE, but some of them have heavy synchronization overheads which harm their use to the practical industry application [57]. The decoders in this category are usually relatively small if the encoded states and the chosen group size are not too large.

Another way of coding is dictionary coding, which partitions the original data into  $n$ -bit symbols and uses a dictionary to store each unique symbol [58]. In Figure 1.21, it encodes each  $n$  bits slice using  $b$  bits code word, where  $b$  is less than  $n$  to achieve compression. A drawback of using a complete dictionary to encode test set requires very large size of dictionary so that results in too much overhead.

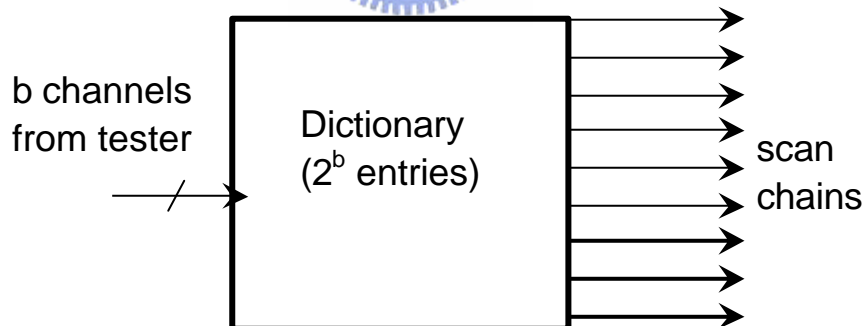


Figure 1.21 Dictionary-based coding

## 1.5 Test Strategies for Simultaneous Test Power and Test Data Reduction

Methods for simultaneous test power and test data reduction are getting more attentions in recent years since they provide a combined solution for the two test

challenges. Some extend existing test compression methods to reduce test power. For example, Golomb code has been shown to be suitable for reducing test power [59]. Later, alternating run-length (ARL) code with minimum transition filling strategy [60] for filling unspecified bits in test set shows better result than that obtains in [59]. Nourani et. al. combined run-length and Huffman code to form a Mixed RL-Huffman encoding method [61], in which run-length code is firstly applied to minimize bit-transition and test power, and then Huffman code is used to further enhance the compression.

Random Access Scan can be applied to reduce both test power and the volume of test data [2, 47], which has been described previously. Besides, new decoding architecture has been proposed, which consists of a decompression control unit (DCU), a flip configuration network (FCN), and a decoder [62]. The flip configuration network uses a combinational block composed of one XOR gate and one MUX for each internal scan chain. If it is possible, the method loads the same values as the previous scanned-in slice so as to reduce power consumption. FCN is used to flip some bits in the scanned-in slice if they are conflict with the previous one. In [63], the authors used the LFSR reseeding and hold flag shift register (HF-SR) to achieve test compression and reduce test power. The architecture shows in Figure 1.26 and its operations is explained below. Test cubes are partitioned into blocks with  $B$  bits. If the current block to be sent can be filled using the last bit of the previous block, then the HS-SR will set the MUX before the scan chain to a 1 so that the last bit of the previous block will be shifted into the scan chain  $B$  times. In this way, the number of transitions can be reduced. The method is suitable to be combined with commercial compression methods based on LFSR reseeding.

## 1.6 Classification and Term Definition for Test Compression Methods

Throughout this thesis, we classify test compression schemes proposed in literature by two different ways. Terms are defined in the following to facilitate our explanation and comparison between ours and prior works in the following chapters.

### 1.6.1 Classified by Type of Decompressor

For test compression schemes proposed in literature, they generally can be categorized into three groups: Entropy-related schemes, Correlation-related schemes and Architecture-related schemes:

- Entropy-related: The characteristic of the schemes in this category is that the maximum compression is bounded by the entropy of data, and usually it can be predicted by analyzing test data. In [52], the authors did a comprehensively study on these compression schemes, including Golomb codes [53], Huffman codes [54], VIHC codes [55], and FDR codes [56], etc. It also showed that the compression obtained by the VIHC scheme approaches the entropy bound. The advantage of the low bandwidth for these schemes allows the use of a low-cost ATE, but some of them have heavy synchronization overheads, which harm their use to the practical industry application [52]. The decoders in this category are usually relatively small if the encoded states and the chosen group size are not too large.
- Correlation-related: The characteristic of the schemes in this category is that the maximum compression depends on the correlation between test patterns [47-51, 86]. Since correlations between test vectors are usually high, due to the circuit

structure dependency, test vector correlation is exploited to minimize the bit flips or the replacement words between consequent test patterns. Furthermore, don't care bits in test patterns are exploited to further compress the patterns. However, the area overhead may be large [47, 50, 86] unless an embedded processor or memory is utilized. In addition, the synchronization overhead in some schemes of this category is also unavoidable [51].

- Architecture-related: The characteristic of the schemes in this category is that the compression capability depends on a decompression/expansion network [37, 45, 87-88]. An ATE of a higher channel bandwidth may be required to facilitate data compression and the test time reduction. However, the test patterns obtained from a commercial ATPG may not be directly applied to those schemes.

Therefore, a special and dedicated ATPG for its decoder was developed to increase data compression.



### 1.6.2 Classified by Application of ATPG

We can also classify compression methods by the relationship with ATPG. We classify them into *ATPG-independent* approach and *ATPG-dependent* approach.

- ATPG-independent: For the compression methods of this category, in the traditional design flow, they are applied after test patterns have been generated. This type of approaches usually encodes test pattern by utilizing don't care bits or makes use of regularity of test patterns to reduce test data volume. One type of these compression methods is to use codeword, for example, Golomb codes [53], Selective Huffman codes [82], VIHC codes [55], and FDR codes [56], etc to represent data block. A comprehensive study on these compression schemes was presented in [52] and the maximum achievable compression of the methods of this type is bounded by the entropy within test data [52].

Another type of compression methods is to compress test data utilizing the bit correlation of test vectors to obtain minimum bit flips between consequent test patterns [49, 51, 86] to achieve test compression. Selective Encoding compresses scan slices using slice codes, which mix of control and data codes, to reduce test data volume [91].

- ATPG-dependent: For the methods of this category, test compression procedure is incorporated during the stage of test generation. As it was reported that, test patterns for large designs have very low percentage of specified bits, and by exploiting that, high compression rate can be achieved. For example, the hybrid test [74] approach generated both random and deterministic patterns for the tested chip while used an on-chip LFSR to generate the random patterns. The Broadcast (or Illinois) approach [36] used one pin to feed multiple scan chains. In [37, 92], a combinational network was used to compress test cubes to conceal large internal scan chains seen by the ATE. Also, several efficient methods such as RIN [93], SoCBIST [94], and EDT [45], etc, were proposed to achieve test data reduction by using an on-chip circuitry to expand compressed data to test patterns. Tang et al [39] also proposed a scheme to use switch configurations to deliver test patterns and provided a method to determine the optimum switch configuration.

## 1.7 Overview and Organization of the Dissertation

In this thesis, we focus on low power testing and test data compression. The first work is to propose a new scan architecture that can reduce the shift power with some additional area overhead, called Scan Matrix. By employing a new scan cell, we reduce the test power by bypassing unnecessary transition in scan cells to provide toggle suppression. Moreover, clock is also gated for inactive scan cells. Therefore,

testing power can be reduced significantly. In the second one we apply random access scan (RAS) [2] architecture, which can reduce the number of switch activity to achieve low power testing, to improve data compression rate. By reordering the applying sequence of test patterns, we can reduce test data volume. We formulate the problem of test vector reordering as *Minimum Bit Flip Problem* and present a flow to solve it. The result shows we improve the encoding efficiency than other works. Followed by a proposed adaptive encoding scheme, we utilized the embedded memory in chip to provide space to decode test data to achieve data compression. The proposed encoding method has more flexibility since it handles various size of block without the limitation due to the specification of the CUT. A scheme for the tradeoff between shift power and compression rate is also analyzed. Finally, we proposed a new encoding scheme called multilayer data copy, which can be used for simultaneous test data and test power reduction. We note that few works in literatures addressing the low power test data compression for multiple-scan-chain designs. Commercial tools can generate highly compressed test data but have very large testing power. This work presents their problems and provides a useful solution.

The rest of the thesis is organized as follows. Chap 2 presents our solution for low power scan testing with Scan Matrix. Chap 3 shows a framework for low power test compression using RAS architecture. Followed by Chap 4, we propose an architecture for test compression utilizing embedded memory and an on-chip encoder for decoding. Thereafter, a practical low power testing architecture with high efficiency compression rate targeting multiple-scan-chain designs is revealed in Chap 5. Finally, the conclusion for this thesis is given in Chap 6.



# Chapter 2 A Scan Matrix Design for Low Power Scan-Based Test

## 2.1 Introduction

As mentioned previously, for the scan design, the circuit under test (CUT) in the test mode usually has larger switching activity, causing excessive power dissipation. In this chapter, we propose a new Scan Matrix (SM) architecture for the scan-based design to achieve low power testing. For this scheme, scan flip-flops are connected in a matrix style with its addressing controlled by two ring generators during test for pattern scanning-in. Unlike the traditional scan, for which scan-in data need to pass through a long path and many scan flip-flops switch simultaneously, it dynamically forms low-power scan paths to reduce test energy and peak power for pattern shifting. The architecture is scalable for large designs and has minimal circuit performance penalty. Experimental results show that it can achieve nearly 99% power savings for large size designs.

In the following, in Section 2.2, we present the detail structure and describe operation of this “new” SM; and in Section 2.3, we include the simulation results of the proposed circuit and discuss the overhead of the scheme. In Section 2.4, we show our experimental results as compared with other approaches. Finally, we give our summary for the proposed architecture.

## 2.2 The Proposed Scan Matrix Architecture

The proposed new Scan Matrix (SM) architecture is shown in Figure 2.1 for a 4X4 example, where SFFs are organized into a two-dimension array with roughly  $m$  (in this case,  $m = 4$ ) rows and  $m$  columns where  $m$  equals roughly to  $\sqrt{N}$ . For each row, SFFs

are connected in the fashion of one scan chain for which column and row signals are used to control scan in/out operations. For SFF control, two circular shift registers are used to address SFFs in sequence in a row by row token-like [12] fashion. The operation of SM is quite simple. Figure 2.2 shows the simplified model of the 4X4 SM structure of Figure 2.1. In the figure, each scan cell (SMR), which will be described later, has two bits, for which the left bit is the datum of pre-latch and the right one is the datum of the master latch. Assume that in the figure the two bits of all SMRs are 0s initially and the scan-in data, 1111..., are to be scanned in via  $S_i$  as shown in Figure 2.2 (a). First, the signal generators reset (Figure 2.2 (b)) and only the first signal line is “High” (Figure 2.2 (c)). Therefore, all SMRs at the first column are activated since  $SEL$  is high. However, only the first SMR of the first row obtains the scanned-in datum since the latch of the first row is triggered to be updated. At the same time, the test result in the master latch of that SMR is passed to  $S_o$ . For the next test cycle, the “High” signal shifts to the second row and the column signals are unchanged (Figure 2.2 (d)). This makes the first SMR of the second scan chain to be updated. After all SMRs of the first column are scanned, the “High” column signal advances to the second one (Figure 2.2 (g)). Once the test pattern is stored in pre-latches completely, “update cycle” updates the data of SMRs by loading the data in pre-latches to master latches. Then “capture cycle” applies the pattern to the CUT and captures the test results to master latches of SMRs. Repeating this (Figure 2.2 (b)-(l)), we can apply test patterns continuously.

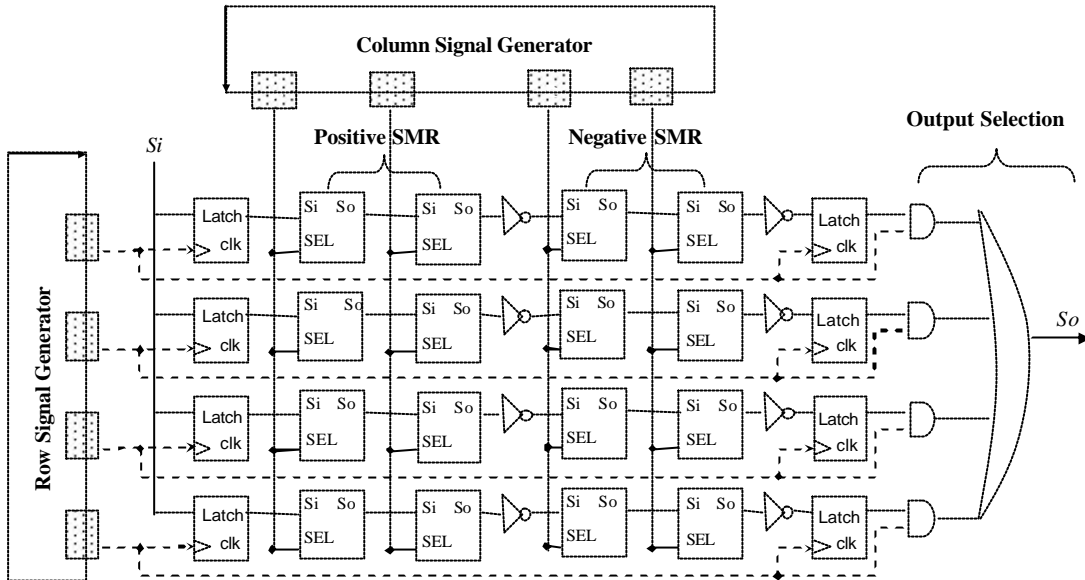


Figure 2.1 A 4x4 SM scan architecture

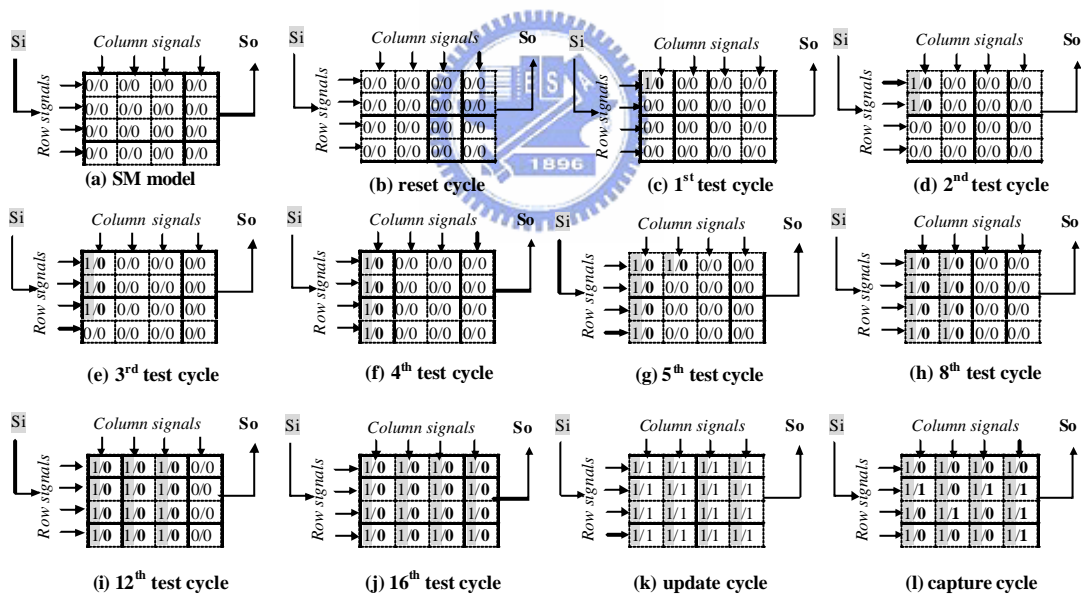


Figure 2.2 A simplified model to demonstrate the scan in/out operation of the 4x4 SM

The detail circuit of the shift registers which are used to compose the circular shift registers of Figure 2.1 is shown in Figure 2.3. It has only one half clock loading and it has less area than a conventional SFF. In the figure,  $WL_0, WL_1, WL_2, \dots, WL_m$ , are word line signals corresponding to column or row signals in Figure 2.1.

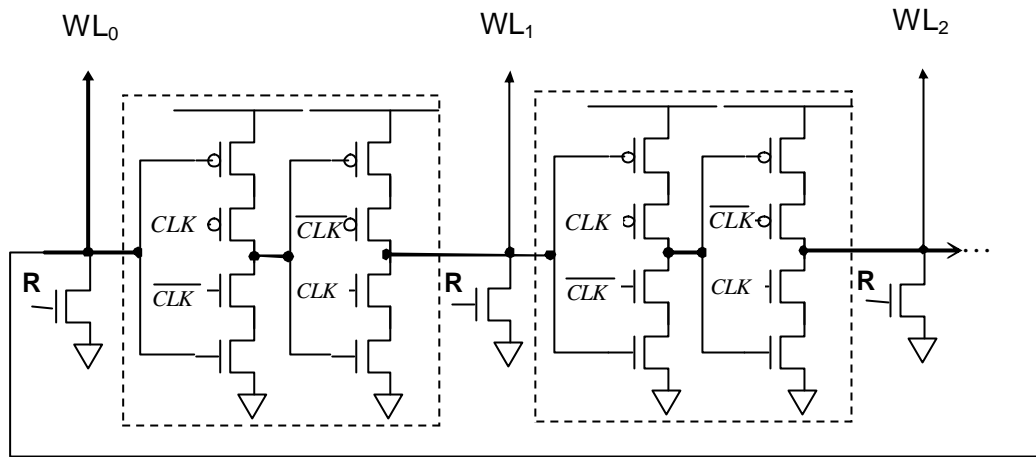


Figure 2.3 Resettable circular shift register for low-power signal generator

Figure 2.4 (b) and (c) show the Positive and Negative Polarity SMRs respectively used in our proposed architecture with a standard scan cell (Figure 2.4 (a)). Four transmission gates and two inverters are added. The two inverters together with the two transmission gates form a pre-latch for  $S_i$ . The pre-latch only needs to drive a SFF; therefore, it is designed with a minimum-sized latch to reduce the area overhead.  $SEL$  goes to high **only** when the corresponding **column signal** is high. At the same time,  $S_i$  will be stored in the pre-latch and test result in the master latch of SMR will be scanned out to  $S_{out}$ . If  $SEL$  is low,  $S_i$  is bypassed to the next SMR. Toggle suppression is capable since the scan-in datum is stored in the pre-latch without affecting the CUT. To achieve power saving during shift, the proposed new scan cell has the capacity to provide a dynamic scan path which is controlled by column signals. Instead of the traditional scan path of a scan chain, the scan-in/scan-out datum will pass through this combinational scan path, which is formed dynamically, leading lower power consumption. However, a long dynamic scan path with serial-connected transmission gates will introduce large delay and raising/falling time. Therefore, an inverting buffer is added after several serial-connected SMRs to provide signal amplification. Hence, a Positive and a Negative Polarity SMR are provided. In addition, the  $CLK$  of SMRs are disabled during the

pattern scanning-in phase to save the power. They are active only during the pattern application phase when the CUT is tested.

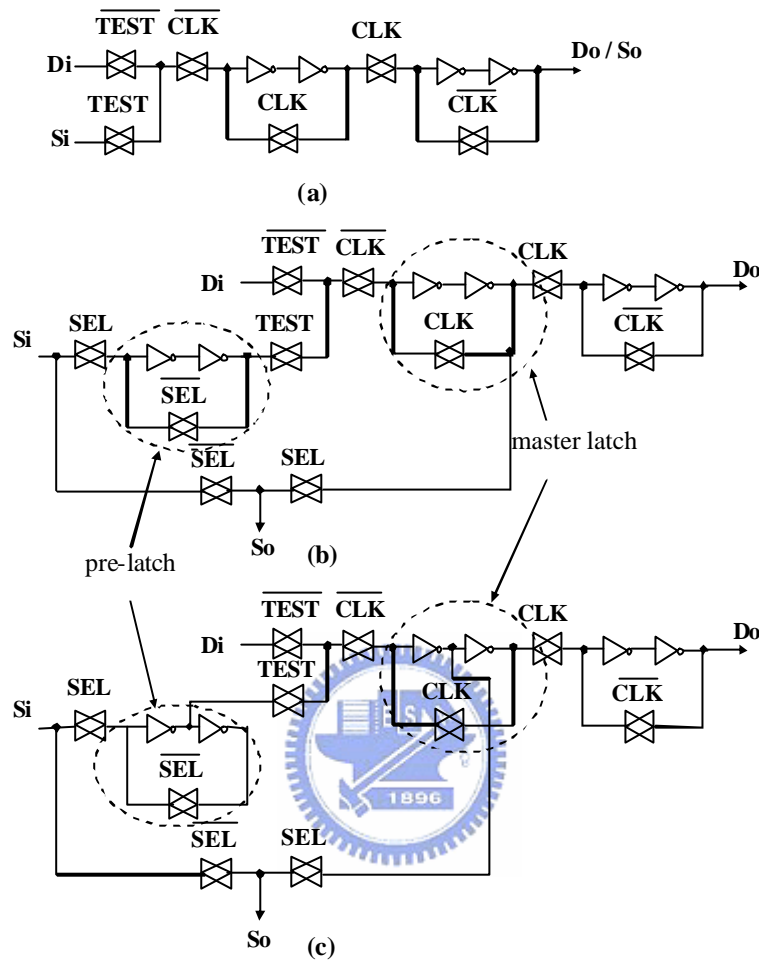


Figure 2.4 (a) The conventional SFF; the proposed (b) positive polarity SMR, and (c) negative polarity SMR

## 2.3 Evaluation and Comparison of SM with Other Scan Approaches

Different implementations of scan schemes and scan cells lead to different speed and power performance and size and routing overhead. In this section, we compare the proposed SM with three other scan schemes, namely,  $m$  scan chains, the token scan scheme [12] and random access scan (RAS) scheme [47]. The comparison is based on

circuit level simulation results with a 0.18  $\mu$  m digital CMOS standard library.

Before comparison, we introduce another type of SMR which has a smaller area but a poorer performance as shown in Figure 2.5. It is termed as SSMR and can be used as a tradeoff between area and performance.

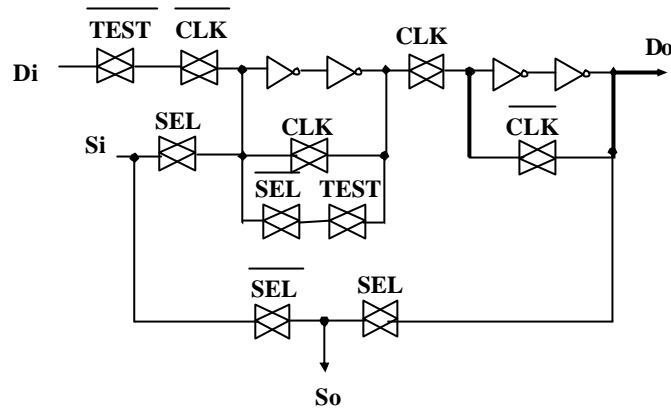


Figure 2.5 Another implementation of SMR with smaller area

Table 2.1 compiles the simulated delay performance of: the conventional SFF, the conventional SFF with a NOR2 gate to provide toggle suppression [22], the proposed SSMR and SMR respectively. From the table, it is seen that the proposed SMR provides the best performance over all other types of SFF and SSMR has a better performance than the SFF with NOR2 gate.

Table 2.1 Delay performance comparison between four different scan cells (in picoseconds)

	Setup Time	Hold Time	C to Q Delay	Total Delay
Conventional SFF	97	0	140	237
SFF with NOR2	97	0	181	278
SSMR	117.5	0	145.5	263
SMR	99	0	134.5	233.5

Table 2.2 compiles the area overhead of: conventional *m*-Scan-Chain, SM, RAS and

Token schemes respectively. From the table, we can see that Token has roughly two times area of that of a conventional SFF while a SMR is about 1.4X of the size of a conventional SFF. The additional area is for clock gating cells, signal generators, latches or decoders associated with each scheme. The size of decoder of RAS is about  $N/G$  where  $G$  is the gate count of a SFF.

Table 2.2 Area comparison of four scan schemes

	Scan Cell	Additional	Overall Rank
<i>m</i> -Scan-Chain	$N$	$\sqrt{N}$ (gating cells)	Small
SSMR	$1.3*N$	$2*\sqrt{N}$	Middle
SMR	$1.4*N$	(signal generators & latches)	Middle
RAS [3]	$1.2*N$	$N/G$ (decoders)	Middle
Token [17]	$2.2*N$	$\sqrt{N}$ (gating cells)	Large

To estimate the routing overhead of four schemes, we consider four types of connection as shown in Figure 2.6. For Type 1 connection, each wire connects two SFFs, hence there are nearly  $N$  wires. For Type 2 connection, it is a global wire connecting all SFFs. Type 3 is a variation of Type 2 where all scan cells are divided into groups and only one group is activated at any instance to avoid simultaneously charging a large wire capacitance. Though Type 3 has less power consumption, its routing length/area is larger than that of Type 2 due to its longer length and additional gating buffers. Type 4 has the largest routing area since every SFF is connected by a routing wire. We investigate the routing cost for each types of connection for the four different SFFs of Tables 2.1 and 3.2, and compile the results in Table 2.3, where the number in each entry is the number of signal wire for each connection and  $S_i$ , CLK, Test,  $S_o$ , Between SFF, and Additional represent the nodes, to which wires are connected, of an SFF. As it can be seen from the table, SM has a better routability than those of Token and RAS schemes due to having fewer routing signals (wires). Although row signals of the SM scheme should be considered, the routing overhead of this scheme is still the lowest among all other

schemes.

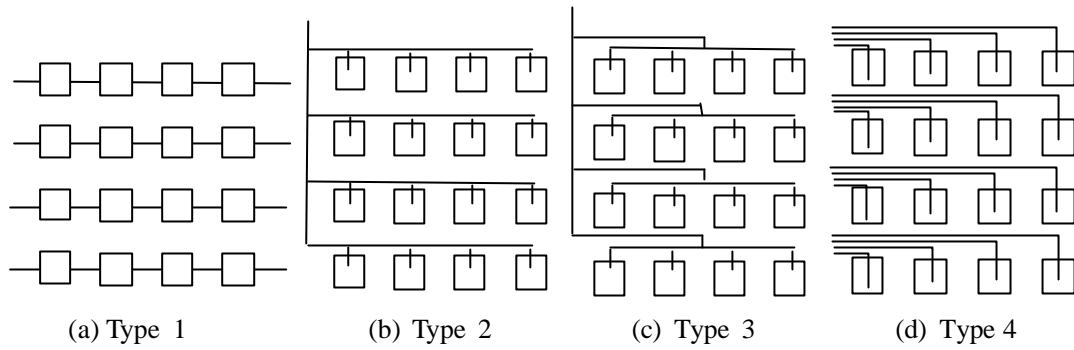


Figure 2.6 Four types of global signal connection

Table 2.3 Routing overhead comparison of four scan schemes in terms of signal routings

	Si	CLK	Test	So	Between SFF	Additional
$m$ scan chains	/	3	2	/	1 (Si-So)	/
SM	/	2	2	/	1 (Si-So)	3 (SEL)
RAS [3]	2	2	2	2	/	4 (enable)
Token [17]	3	3	2	2	1 (token)	/

The power performance of the SMR is also compared with those of the conventional SFF and the DFF of Figure 2.2. The comparison is based on the power *weighted transition counts* (WTC) model [12, 21-22] which considers the number of total SFF transitions, i.e.  $E = \sum n * c$ , where  $n$  is the toggle count and  $c$  is the normalized capacitance of a circuit node, and the clock power which may consume up to 30% of total power dissipation. Table 2.4 shows relative energy consumption of three kinds of scan cells. It is mentioned that, for this simulation, an inverting buffer is inserted every four SMRs of a scan chain. In the table, *Transition1* represents for a transition of the path from  $Si$  to  $So$  while *Transition2* represents for a transition from  $Si$  to  $Do$ . As it can be seen from the table, SMR reduces a shift transition power by 78% as compared to that of SFF.



Table 2.4 Simulated relative power consumption with respect to SFF for different transitions

	Clock	Transition1 ( <i>Si So</i> )	Transition2 ( <i>Si Do</i> )
Convectional SFF	0.3	1	1
DFF (Figure 2)	0.18	0.67	0.67
SMR	0.3	0.22	1.33

Table 2.5 compiles all above factors for the four schemes where the cell area and toggle suppression are also included. Compared with RAS, SM has a slighter larger cell area than that of RAS but has a significantly reduced routing overhead in addition to its avoiding use of decoders which may introduce large delay. Compared with the Token scheme, SM has a smaller area and lower routing overhead, too. In addition, it has advantage that its test response can be easily shifted out. Compared with the *m*-Scan-Chain scheme, SM has a larger power saving and a better clock tree routing strategy at the expense of the cell area overhead.

Table 2.5 Overall comparison for four power-saving schemes

Scheme	Scan Cell			Cell Area	Routing Overhead
	Scan Cell	Toggle Suppression	Performance Degradation		
<i>m</i> scan chains	SFF with NOR2	Yes	Large	Small	Small
<b>SM</b>	<b>SMR</b>	<b>Yes</b>	<b>No</b>	<b>Middle</b>	<b>Middle</b>
RAS [3]	SFF	No	No	Middle	Large
Token [17]	SFF with NOR2	Yes	Large	Large	Large

## 2.4 Experimental Results

To verify that the proposed scheme is power-efficient, we performed experiments by applying the scheme to ISCAS89 scan benchmark circuits using test sets generated by ATALANTA [32]. A C program was used to calculate total power reduction and peak

power reduction based on the power model described in Section 2.3. Table 2.7 shows the experimental results of our scheme as compared with other approaches where the number of test set and test efficiency for each circuit are also included. In the table, the column, “3 Scan Chains”, means that the flip-flops of each tested circuit are partitioned into the conventional 3 scan chains. The column, “ $m$  Scan Chains”, means the same as that of column, “3 Scan Chains”, i.e., the flip-flops of the tested circuit are partitioned into  $m$  scan chains where the value for  $m$  is shown in parenthesis for each circuit. We can see that for the “3 Scan Chains”, the reduction of total/peak power is very close to the theoretical value,  $(1-1/3) = 66.6\%$ . For the “ $m$  Scan Chains”, the total/peak power reduction is also very close to the theoretical value of  $m = \sqrt{N}$ . However, when comparing with the results of the proposed SM scheme with the above results, we can find that it achieves even more power reduction than those of conventional “ $m$  Scan Chains”. The improved power reduction mainly comes from the power reduction of the ring generators, SMRs and low-power scan path design of this scheme. Our experiment does not consider the switching activity in the CUT since our SMR provide toggle suppression. If the power reduction of the toggle suppression is also considered, more power saving could have been achieved. We do not compare SM with the Token method and the RAS method since it is hard to build their architectures to be compared fairly.

We then compared the test time overhead and the area overhead of the proposed architecture. In the calculation, we mapped the area overhead of each gate of circuits according to a standard library with a respective weight. For example, the weight of an INVERTER or NAND/NOR gate is 1 and those of the SFF in Figure 2.3 (a) and the SMR are 7.5 and 10.3 respectively. The results of the calculation are summarized in Table 2.8, where  $PPI$  represents for pseudo primary inputs and  $R$  and  $C$  represent for numbers of signals of row and column of the SM structure. From the table, we see that

the average test time increased is 3.14% and the average area overhead is 13.89%. It is mentioned that the area overhead is with respect to the small-area SFF in Figure 2.3 (a). However, if the calculation is based on consideration of other different implementations of SFF, for examples, the simplest edge-triggered NAND-type SFF which has a gate count 9.5, and the LSSD SFF which has a gate count 14, the area overhead is still reasonable. Furthermore, if we really want to reduce the area of designs, we can use another implementation of the smaller SMR as shown in Figure 2.5. If this modified version of the smaller SMR is used, the area overhead calculated will be even smaller which is shown in the column SSM of the table. Note that the area overhead does not include the wire routing. Finally, the detailed layout of the proposed SMR is shown in Figure 2.7. The pre-latch of SMR increases the area by about 37%, which was mentioned in Table 2.2, as compared to a standard scan cell under TSMC 0.18  $\mu$  m technology.



Table 2.6 Benchmark circuits used in our experiments

Circuit	Inputs / Outputs	Gates	Scan Cells	Test Patterns	Test Efficiency
s1423	17 / 5	657	74	68	100 %
s5378	35 / 49	2779	179	263	100 %
s9234.1	36 / 39	5597	211	371	99.307 %
s13207.1	62 / 152	7951	638	476	99.908 %
s15850.1	77 / 150	9772	534	435	99.923 %
s35932	35 / 320	16065	1728	65	100 %
s38417	28 / 106	22179	1636	901	99.987%
s38584.1	38 / 304	19253	1426	647	99.934 %

Table 2.7 Total/peak power reduction obtained for different approaches

Circuit	Full Scan	3 Scan Chains		<i>m</i> Scan Chains		SM	
	Total / Peak Power	Total / Peak Power	Total / Peak Power Red. %	Total / Peak Power	Total / Peak Power Red. %	Total / Peak Power	Total / Peak Power Red. %
s1423	39.08E4 / 71.2	13.38E4 / 27.5	65.76 / 61.37	5.57E4 / 11.7 (9)	85.75 / 83.56	2.48E4 / 3.7	93.65 / 94.80
s5378	78.81E5 / 160.7	27.20E5 / 61.0	65.48 / 62.04	6.39E5 / 16.9 (14)	91.89 / 89.48	2.84E5 / 5.2	96.40 / 96.76
s9234.1	181.06E5 / 190.3	61.27E5 / 71.3	66.15 / 62.53	14.27E5 / 18.5 (15)	92.12 / 90.27	6.18E5 / 5.5	96.58 / 97.10
s13207.1	213.34E6 / 551.4	71.47E6 / 196.9	66.49 / 64.29	8.74E6 / 30.5 (26)	95.90 / 94.46	3.76E6 / 8.8	98.23 / 98.40
s15850.1	135.85E6 / 470.2	45.38E6 / 162.4	66.59 / 65.46	6.20E6 / 27.9 (24)	95.43 / 94.06	2.68E6 / 8.2	98.02 / 98.25
s35932	207.08E6 / 1438.4	69.00E6 / 495.8	66.67 / 65.53	5.20E6 / 45.6 (42)	97.48 / 96.82	2.27E6 / 13.6	98.90 / 99.05
s38417	256.20E7 / 1376.8	85.81E7 / 469.8	66.50 / 65.87	6.48E7 / 45.0 (41)	97.47 / 96.73	2.81E7 / 13.3	98.90 / 99.03
s38584.1	144.71E7 / 1198.8	48.36E7 / 415.8	66.58 / 65.31	3.97E7 / 42.4 (38)	97.25 / 96.46	1.70E7 / 12.4	98.82 / 98.96
Avg.			66.28 / 64.05		94.16 / 92.73		97.44 / 97.80

Table 2.8 Test time and area overhead using the SMR and the small SMR compared with full scan

Circuit	Full Scan		SM						SSM	
	PPI	Gates Counts	R	C	PPI	%Test Time Over.	Gates Counts	%Area Over.	Gates Counts	%Area Over.
s1423	74	1379	9	9	81	8.64	1761.8	21.72	1687.8	18.29
s5378	179	4241	14	13	182	1.65	4934.1	14.04	4755.1	10.81
s9234.1	211	7872.5	15	15	225	6.22	8780	10.33	8569	8.13
s13207.1	638	13549	26	25	650	1.85	15758	14.02	15120	10.39
s15850.1	534	14941.5	24	23	552	3.26	16898.1	11.57	16364.1	8.69
s35932	1728	31617	42	42	1764	2.04	37309.2	15.25	35581.2	11.14
s38417	1636	36639	41	40	1640	0.24	41732.5	12.21	40096.5	8.62
s38584.1	1426	34016.5	38	38	1444	1.25	38631.7	11.95	37205.7	8.57
Avg.						3.14		13.89		10.58

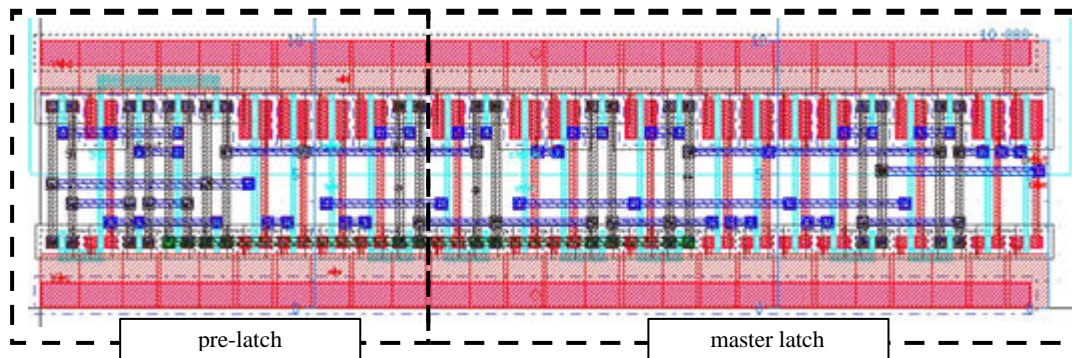


Figure 2.7 The layout of SMR

## 2.5 Summary

In this chapter, we have proposed a new “Scan Matrix” scan design architecture to save power for scan test. In the scheme, the scan flip-flops are connected in a matrix configuration and two ring generators are used to address scan cells during pattern scanning in. It utilizes elegantly designed scan cells, scan path and the ring generators to achieve toggle suppression to prevent useless switching activity in the CUT during the scan testing. As a result, it achieves great power saving not only on scan flip-flop switching power but also switching power in CUT during pattern shifting. Another nice feature of the scheme is that it suffers no performance penalty. Experiments on ISCAS89 benchmark circuits have shown that the scheme achieves significant power savings, for some large circuits 99% saving, both in energy consumption and peak power. Hence, this scheme is a good solution to the low power scan test.



# Chapter 3 Cocktail Random Access Scan for Test Data and Power Reduction

## 3.1 Introduction

In this chapter, we develop a hybrid test strategy, *Cocktail Scan*, based on RAS toward high efficiency test. This hybrid method, different from the LFSR-based hybrid BIST, adopts a two-phase approach to perform scan test for which all test data are supplied from the ATE. However, for test patterns, instead of supplying the long inefficient pseudo-random patterns generated by LFSRs, at the first phase, we supply a set of carefully-chosen efficient seed patterns to perform a segmented random pattern scan test to test the DUT to achieve considerable high fault coverage, and then at the second phase, supply deterministic patterns to detect remaining faults. At the second phase, patterns are applied in the RAS fashion and they are reordered and compressed with the proposed strategies to reduce data volume, the number of bit flips, and consequently test energy. Furthermore, due to adopting several strategies: *Test Response Abundant*, *Constrained Static Compaction*, and *Bit Propagation Before Test Vector Dropping*, which are very effective in reducing bit flipping and test data volume, we further achieve reduction on test application time and power. Experimental results show that our proposed scheme exhibits much improvement than the traditional full scan method and also significantly outperforms previous MBFP-based works in test application efficiency.

## 3.2 Review Random Access Scan (RAS)

RAS was first proposed in [2]. Recently it was applied with several techniques for test compression and test power reduction application [47]. For the applied RAS

architecture in [47], it had an address decoder and *Address Shift Registers* (ASRs) as shown in Figure 3.1. The address decoder generated scan enable signal for each scan cell. The clock source of ASR was from address clock (ACLK) and scan cells were controlled by system clock (SCLK). If there were  $N$  scan flip-flops (SFFs),  $\lceil \log_2 N \rceil$  address bits were used to address any specified SFF, where  $\lceil \log_2 N \rceil$  was the address width of the ASRs. Therefore, to update one bit of RAS, a total of (address width+1) bits and (address width+2) cycles were needed as the ASR was used in a serial manner. The basic scan cell with two multiplexers of RAS is also shown in Figure 3.2. If one SFF was addressed, the new datum was scanned into it after SCLK was activated for one cycle while the control sign *Mode* was set to high. After a test pattern was loaded completely, *Mode* was changed to low to allow SFFs to capture their test responses. Output results were observed from a multiple-input signature register (MISR). Then, the next test pattern was applied by shifting next (address width+1) bits (bit-flipping datum) in the same manner. If the current test response differed from the next test pattern by  $n$  bits, total data volume to flip to next pattern required  $n*(\text{address width}+1)$  bits. This strategy implies that the bits needed to be flipped should be kept as few as possible to achieve test data reduction. Also in [47], to solve MBFP, test vector reordering was formulated into an *Asymmetric Traveling Salesman Problem* (ATSP) and a heuristic algorithm, *Lin-Kernighan Heuristic* (LKH), was used to solve the ATSP. Together with using other skills, nearly 3 times of speedup in the test time, 60% reduction in the test data volume and over 99% reduction in the power consumption were achieved.

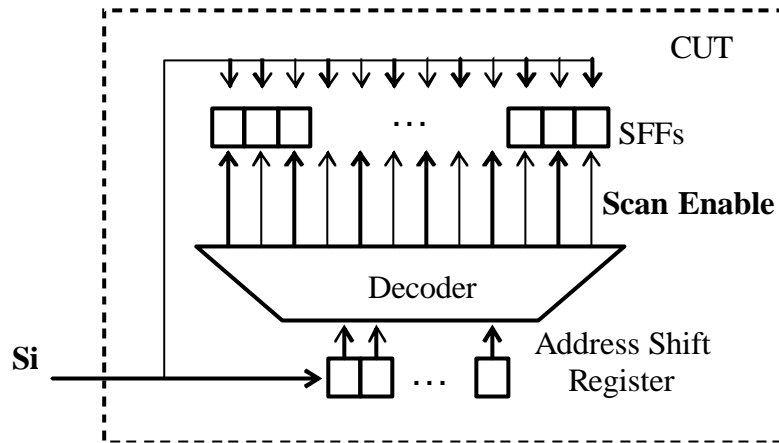


Figure 3.1 Random access scan architecture

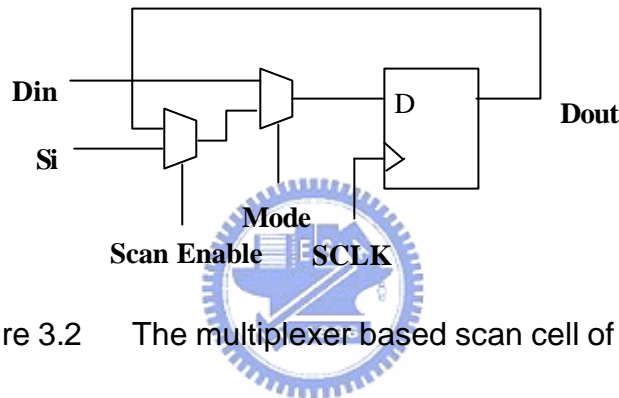


Figure 3.2 The multiplexer based scan cell of RAS

### 3.3 Cocktail Scan Based on RAS

The proposed scan test method, *Cocktail Scan*, is a hybrid method combining the segmented random scan test as the first phase and the deterministic scan test via the RAS fashion as the second phase.

#### 3.3.1 Segmented Random Scan Test (SRST)

At the first phase of the *Cocktail Scan*, the DUT is tested by a set of segmented random patterns. However, the segmented random patterns are divided into several segments and, for each segment of random patterns, it has its own seed pattern which is selected carefully to boost the fault coverage. First, one seed pattern is applied to the DUT and the output response of this seed pattern is captured to SFFs. Then the



output response serves as the input pattern of the next clock cycle. This process is repeated for several clock cycles until the fault coverage of the DUT under this set of segmented random patterns does not increase anymore. Then a next seed pattern is applied and the above process is repeated. Since different seed patterns will boost the fault coverage into a higher value, this scheme of segmented random pattern testing can get fault coverage higher than that for which only a single seed is used. The advantage of this method is that only a few seed patterns need to be stored and applied by an ATE yet relative high fault coverage can be achieved.

Some experiments had been done on several ISCAS89 circuits to verify the efficiency of the above scheme. Figure 3.3 show the results of the experiments. In the figure,  $n$  is the total number of seed patterns and  $L$  is the number of segmented random patterns for each seed pattern. For example, for s5378, if four seed patterns were used separately, as shown in Figure 3.3(a), the maximum fault coverage reached was only 52%. However, from Figure 3.3(b), if four seed patterns were used altogether and each seed pattern generated 16 segmented random patterns to test the circuit, the fault coverage could easily reach 72%. Figure 3.3(c) and (d) show fault coverages for another two circuits, s9234 and s13207, and the same results can be seen.

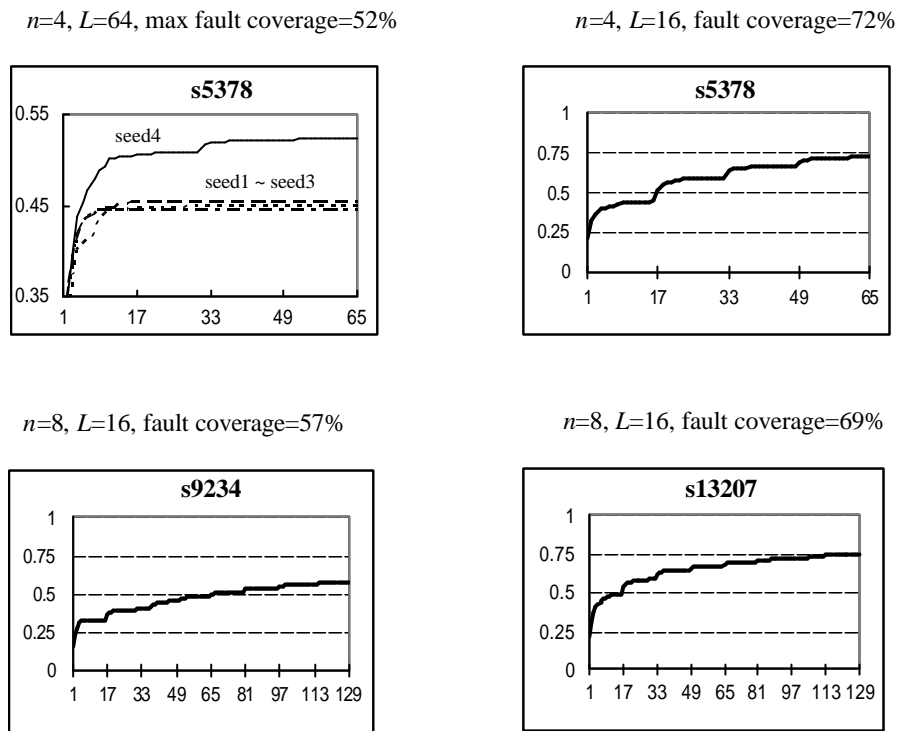


Figure 3.3 Fault coverage curves for Segmented Random Scan Test



### 3.3.2 RAS Test

After the first phase *SRST* mode, the testing process goes into the second *RAS* mode where deterministic patterns are applied in the *RAS* fashion. For this phase, the test volume and the number of bit flips are minimized. Several strategies are used to achieve the goal. Figure 3.4 shows the proposed process of this phase where the strategies used are listed. In this process, contrary to the ordinary process of first compacting patterns statistically, reordering the test vectors and then doing vector dropping [48] to minimize bit flips in *RAS* [47], it first reorders the test patterns, then uses a constrained static compression to reduce the number of test patterns and finally uses a bit-propagation vector dropping technique to maintain the fault coverage of reduced test patterns. In the following, the detailed strategies of the above process are discussed with some related considerations.

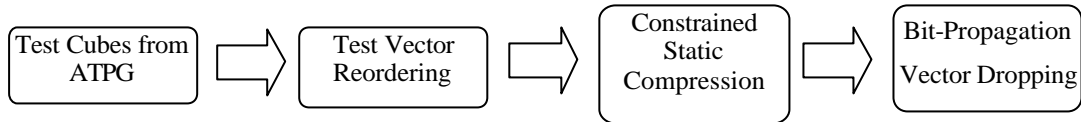


Figure 3.4 The proposed process for solving MBFP

### 3.3.2.1 Test Pattern Generation for RAS Test

Traditionally, an ATPG program applies either static or dynamic compaction, or both, during the pattern generation to reduce the number of test patterns. Consequently, the percentage of unspecified bits in test pattern will be reduced significantly and test compression for using RAS will be reduced. Therefore, while generating test patterns, on one hand, we do not want to generate too many test patterns to increase the number of patterns; on the other hand, we do not want to generate the test patterns of low percentage of unspecified bits so as to decrease the flexibility of reordering them for data reduction. Hence, we propose a modified ATPG program by considering the percentage of unspecified bits during the test generation step. That is: the percentage of unspecified bits for each generated test pattern is checked when two test cubes are to be merged during test compaction. A threshold of percentage of unspecified bits for each pattern can be defined as an index for compaction. In this way, a reasonable number of test patterns with high percentage of don't care bits can be produced.

### 3.3.2.2 Cost Model for Don't Care Bits

In order to facilitate the evaluation of the proposed approach, a cost model is first discussed. This is because as test vectors are reordered to decrease the total bit flips, if they have don't care bits in their patterns, the costs of flipping bits cannot be simply determined. Different cost models may lead to different results so that the real cost of bit flipping cannot be reflected.

Table 3.1 summarizes four cost models used in different works [47-48, 79], where different costs for flipping bits with don't care are treated differently. The cost model *Normal* [47] assigns a 1 to flip a don't care bit to a specified bit. Cost models *Optimistic* and *Pessimistic* [48, 79] assign an optimistic value, i.e., 0 and a pessimistic value, i.e., 1 to flip a don't care bit respectively. *Estimated* [79] assigns a cost of between 0 and 1 depending on the probability of 1 appearing at that bit location for all other patterns. For example, given four patterns, 010, x11, 1x0, 1xx; for  $p[1]$ , there are two 1s and a 0 in the first bit position, the probability of occurrence of 1 is  $2/3$ , similarly,  $p[2]$  is  $2/2 = 1$  and  $p[3]$  is  $1/3$  respectively. To investigate those four cost models, an experiment was done by applying them to four benchmark circuits to estimate their cost of bit flippings and compare with the real costs of bit flippings. In the experiment, the total cost of flipping  $v1$  to  $v2$  is:

$$Cost = \sum_{k=1}^m f(v1[k] \rightarrow v2[k]) \quad (1)$$

where  $f$  is the cost function of the above cost models and  $v[k]$  is the  $k$ th bit of the test cube  $v$  which has  $m$  bits. Table 3.2 shows the results where estimated costs and actual real costs are listed. In the table, we can see that cost estimated by different cost models can differ by two times. For example, *Optimistic* [48] can have an estimated cost to have more than two times of bit flips than that estimated by *Estimated* [79]. In the table, the results obtained by *Estimated* and *Normal* cost models are comparable to each other, while that obtained by *Estimated* always shows less error as compared to the actual costs. Hence in our work of estimating cost of bit flipping, the *Estimated* cost model was used.

Table 3.1 Four different models used for estimating cost for bit flipping

	1 0 X	1 0 X	1 0	0 1	1 0	X X	X X	1 0
<i>Normal</i>	0	0	1	1	0	0	1	1
<i>Optimistic</i>	0	0	1	1	0	0	0	0
<i>Pessimistic</i>	0	0	1	1	1	1	1	1
<i>Estimated</i>	0	0	1	1	0	0	1-p[k] p[k]	

Table 3.2 Experiment results on different cost models

<b>s1423</b>	<i>Normal</i>	<i>Optimistic</i>	<i>Pessimistic</i>	<i>Estimated</i>
Cost Estimated	605	0	915	349.78
Actual Cost	440	848	514	<b>409</b>
Error%	37.5%	100%	78%	<b>14.48%</b>

<b>s5378</b>	<i>Normal</i>	<i>Optimistic</i>	<i>Pessimistic</i>	<i>Estimated</i>
Cost Estimated	1645	0	2313	1090.39
Actual Cost	1270	3303	1326	<b>1213</b>
Error%	29.52%	100%	74.43%	<b>10.11%</b>

<b>b04</b>	<i>Normal</i>	<i>Optimistic</i>	<i>Pessimistic</i>	<i>Estimated</i>
Cost Estimated	518	2	734	304.79
Actual Cost	<b>375</b>	676	441	394
Error%	38.13%	99.7%	66.44%	<b>22.64%</b>

<b>b13</b>	<i>Normal</i>	<i>Optimistic</i>	<i>Pessimistic</i>	<i>Estimated</i>
Cost Estimated	285	0	414	177.53
Actual Cost	<b>225</b>	440	242	227
Error%	26.66%	100%	70.07%	<b>21.78%</b>

### 3.3.2.3 Test Response Abandonment

In the RAS architecture, after test patterns are applied to the CUT, test responses are stored back into SFFs to be compressed into a signature in a MISR [47]. In our approach, test responses are dropped (i.e., SFFs do not capture test responses in the second phase) since correlation between test vectors are usually high and can be employed to reduce bit flippings. To implement this technique, the scan cell of RAS should scan out its test response from  $Din$  rather than from  $Dout$  and the signature is observed through certain output compactor, which connects  $Di$ . Figure 3.5 depicts the modified RAS cell. More details about output compactor are described in Section 3.4.4. With this technique, only input patterns are used to be compressed and bit-flipped. The later experimental results show that this is a good strategy.

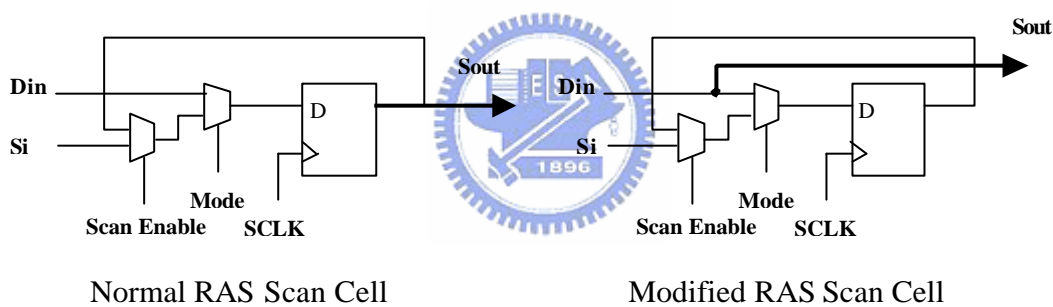


Figure 3.5 The modified scan cell for RAS to support Test Response Abandonment

### 3.3.2.4 Constrained Static Compaction (CSC)

Traditional static compaction compresses test vectors produced by the ATPG tool as much as possible. This may increase bit flips greatly since don't care bits have been assigned to specified values after this step and increases the chance of flipping for every bit. In this work, we proposed a *Constrained Static Compaction (CSC)* approach which guarantees that no additional bit flips will be introduced. For this procedure, a test vector will be compacted with those compatible test vectors that just

follow it. It does not compact two compatible test vectors if there exists any incompatible test vector in between these two compatible vectors. Figure 3.6 shows an example to demonstrate this: For the shown five reordered test vectors, CSC will only merge the second vector, 1x00x, with the third vector, 110xx, but not with the last vector, 1xxx0 since there is an incompatible vector, x11x1, in between them.

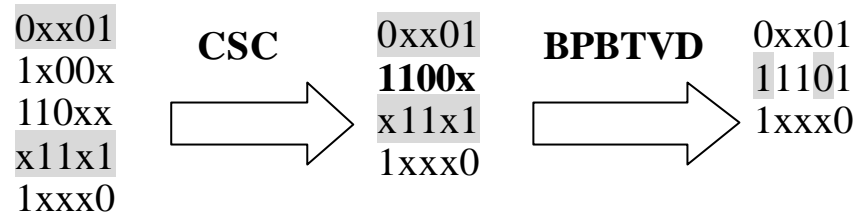


Figure 3.6 An example to explain Constraint Static Compression (CSC) and Bit-propagation Before Test Vector Dropping (BPBTVD)

### 3.3.2.5 Bit-Propagation before Test Vector Dropping (BPBTVD)

After test vectors are reordered and compacted by CSC, the number of test vectors can be further reduced by *Test Vector Dropping* (TVD) [48]. At the same time, don't care bits of a vector are padded by its prior test vector. It is very possible that, after this process, some test vectors may not detect any faults and become redundant since the faults originally detected by them have been detected by some other prior vectors. However, directly dropping those test vectors may decrease fault coverage because in the padding, the don't care bits may be padded to different values since their prior test vectors have changed. The vector to be dropped must propagate its bits to the vector which just follows it. Figure 3.6 demonstrates an example to show this. In the figure, suppose the second test vector, 1100x, does not detect any fault and is to be dropped. After its dropping, the don't care bits of the third vector, x11x1, will be padded to the first vector, making the vector 01101. However, if the second vector is not dropped, the third vector will be padded to become 11101, i.e., the don't care bits should be padded to second vector which is to be dropped. So to maintain the original fault

coverage of the test set, the bits of the second vector should be propagated to the third vector before its dropping. In this case shown in Figure 3.6, the first bit, 1, and the fourth bit, 0, should be propagated to the third vector, x11x1, to make it 11101, instead of 01101. This proposed strategy is called the *Bit-Propagation Before Test Vector Dropping* (BPBTVD).

### 3.3.3 Hardware Modifications

In order to facilitate the proposed clock scan based on RAS scheme, some hardware modifications need to be done:

#### 3.3.3.1 Modified Address Register

Since the RAS architecture does not support serial scan, scanning in seed patterns is costly. It will take  $m \cdot (\text{address width} + 1)$  clock cycles to scan in an  $m$ -bit seed pattern. In order to facilitate the SRST, the address shift registers (ASRs) of the RAS circuit need to be modified as shown in Figure 3.7. The modified ASRs has two modes, i.e., when  $Mode = 1$ , which is the SRST mode, ASRs act as a counter; when  $Mode = 0$ , which is the conventional RAS mode, ASRs act as a shift register. With this modified ASRs, an  $m$ -bit seed pattern needs only  $m$  clock cycles to be scanned in.

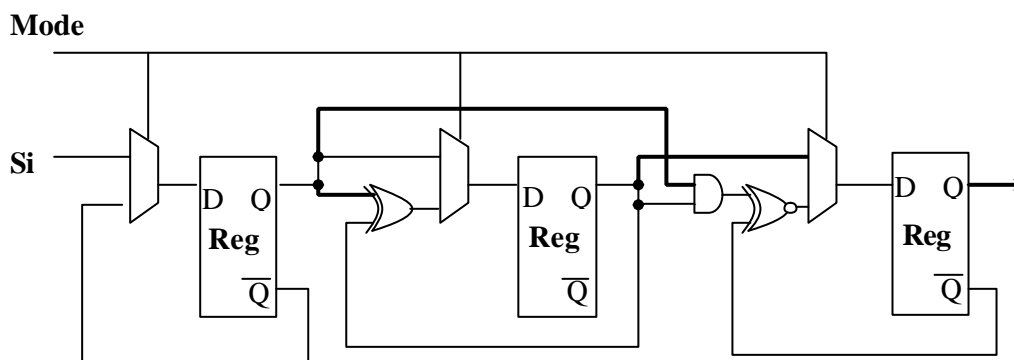


Figure 3.7 The proposed address shift register (3 bits)



### 3.3.3.2 On-Chip Scan Controller

An on-chip scan controller is also needed to switch between the SRST mode and the RAS mode. The controller mainly contains several counters to monitor the number of scan-in bits, test length applied and number of scan-in seed patterns. The controller also controls ACLK and SCLK to correctly scan or capture test responses. The proposed test flow shows in Figure 3.8.

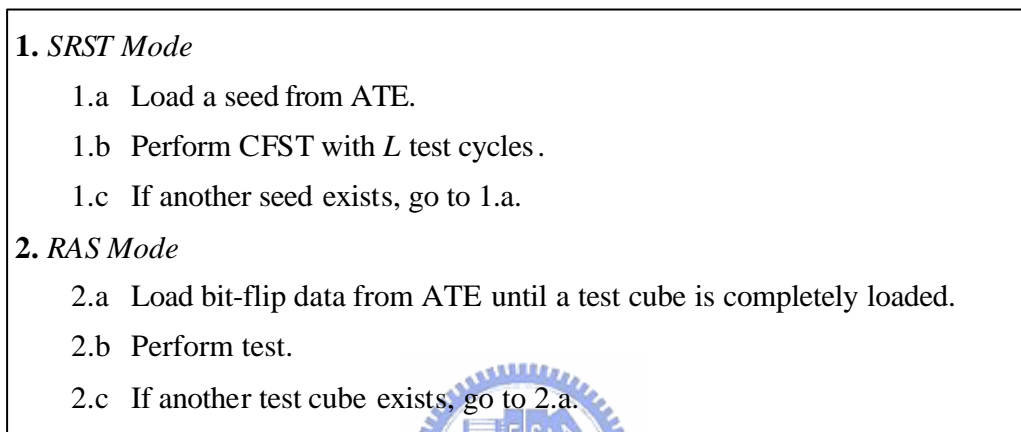


Figure 3.8 The proposed Cocktail Scan flow

## 3.4 Experimental Results

Experiments had been done to verify the proposed scheme. Two experiments' data are given:

### 3.4.1 Experiment on Test Efficiency of the Proposed Process Compared with Other Processes

First, Table 3.3 shows the experimental results of applying the CSC and BPBTVD strategies in the test vector and bit flipping reduction process to a benchmark circuit s5378. The results are compared with those applying the traditional process of first applying SC, LKH (*Lin-Kernighan Heuristic*) reordering, and then TVD [48]. The original number of test vectors was 1126 with fault coverage of 99.12%. After the


traditional process was applied, 272 vectors were obtained but the number of bit flipping was 4720. For our proposed process applied, 640 vectors were obtained first after CSC but a far less number of bit flipping, 1173, was obtained after BPBTVD. In the RAS architecture, the number of bit flipping is the most important factor since it determines the test data volume, test time and test power. Also, it is noticed that for the traditional process, after SC, the bit flips increased significantly and LKH only improved 6% of bit flips, but for our proposed process, LKH reduced bit flips significantly and CSC did not increase bit flips which were further reduced by BPBTVD. For our proposed process, the results of TVD are also listed. Although reducing more bit flips, it suffered a slight decrease on the fault coverage.

A more comprehensive experiment on larger benchmark circuits for test efficiency of the proposed process compared with other processes is shown below. Table 3.4 lists the benchmark circuits on which the experiment was performed with their associated information, where circuit name, number of inputs/outputs, number of gates, number of test patterns, which includes number of fully-specified (FS) test patterns (i.e., no don't care bits) and number of partially specified (PS) test vectors (i.e., with don't care bits), number of scan cells, and test efficiency are included respectively.

Table 3.5 lists the experimental results of applying the traditional process [48], a RAS process which is similar as that of [47] and our proposed process on the partially specified test vectors of above circuits. For the RAS process experiment, test responses were stored back to SFFs to be compacted with input vectors, while for the proposed process experiment, test responses were abandoned. The skills such as *Hamming Distance Reduction*, etc., were not implemented in both the RAS experiment and the proposed process experiment since we only wanted to compare the result of the strategy of *Test Response Abandonment* with that of the general practice of compacting input vectors with the output responses. In the table, the final

reduced number of test vectors, the number of bit flips, the data volume for storing test vectors, and test application cycles are listed. For the traditional process experiment, *Optimistic* cost model was used similar to that used in [48]. The results show that our proposed process is a great improvement on the number of bit flips and consequently on the data storage volume (in average, by 65.07% reduction over those of [48] and by 74.30% over those of [47]) and test application time (in average, by 2.96 times reduction over those of [48] and 4.56 times reduction over those of [47]), for all circuits. In the above experiments, ATALANTA [32] was used as the test generator for generating the test patterns. Since ATALANTA cannot produce complete PS test vectors for larger circuits, we only applied the obtained PS test vectors to the last five circuits in Table 3.5.

Table 3.3 Comparison of traditional process and the proposed process on CSC and BPBTVD strategies on s5378



s5378	Traditional [48]				Proposed				
	Original	SC	LKH	TVD	Original	LKH	CSC	TVD	BPBTVD
Vectors	1126	305	305	272	1126	1126	845	640	640
Flips	3505	5286	4972	4720	3505	1213	1213	1124	1173
FC	99.12	99.12	99.12	99.12	99.12	99.12	99.12	99.10	99.12

### 3.4.2 Experiment on Cocktail Scan

Experiment on the overall proposed *Cocktail Scan* scheme on the above benchmark circuits and another circuit, s35932, was done and results are compared with the traditional full scan method in Table 3.6. In the table, test vectors of each circuit in each scheme gave the same fault coverage. Also, for the proposed scheme, the number of seed patterns, the number of segmented random patterns for each SRST seed pattern, the number of final test vectors after applying our proposed bit flipping reduction process are included. It can be seen that for all circuits, our proposed

*Cocktail Scan* scheme had a large improvement on the data storage volume (over 86%) reduction and the test application time (over 10 times) reduction in average. We also compared our results with some previous works as in Table 3.7. In the table, the best results for each circuit are listed in boldface. It can be seen that for most of circuits, our proposed scheme shows better results.

In Table 3.8, we show the power reduction using RAS scheme in terms of switching activity of SFFs. In addition, the test data volumes for each circuit are plotted in terms of their circuit size for both the conventional full scan method and the *Cocktail Scan* scheme in Figure 3.9. For the full scan method, the test data volume increases greatly with the size of the circuit while for the *Cocktail Scan* scheme the test data volume increases only slightly. This implies that the *Cocktail Scan* scheme has a potential to be used in large size circuits in saving the test data storage, test application time, and hence, test power.

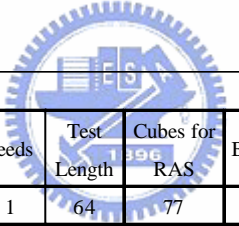
Table 3.4 Details on benchmark circuits

Circuit	Inputs / Outputs	Scan Cells	Gates	Vectors		Test Efficiency
				FS	PS	
s1423	17 / 5	74	657	0	374	100%
s5378	35 / 49	179	2279	0	1126	100%
s9234.1	36 / 39	211	5597	143	595	99.307%
s13207.1	62 / 152	638	7951	175	893	99.908%
s15850.1	77 / 150	534	9772	200	442	99.923%
s38417	28 / 106	1636	22179	699	1506	99.987%
s38584.1	38 / 304	1426	19253	460	673	99.934%

Table 3.5 Comparison on the encoding efficiency for different flows applied to MBFP

Circuit	Traditional [48]				Using Test Response [47]				Proposed							
	Test	Bit	Data	Test	Test	Bit	Data	Test	Test	Bit	Data	Test	Volume	Volume	Speed	Speed
	Cubes	Flips	Volume	Cycles	Cubes	Flips	Volume	Cycles	Cubes	Flips	Volume	Cycles	Red <sup>1</sup>	Red <sup>2</sup>	Up <sup>1</sup>	Up <sup>2</sup>
s1423	89	1167	8169	8258	374	811	5677	6051	171	364	2548	2719	68.81%	55.12%	3.04	2.23
s5378	265	4833	43497	43762	1126	3536	31824	32950	640	1173	10557	11197	75.73%	66.83%	3.91	2.94
s9234.1	239	2635	23715	23954	595	5446	49014	49609	322	814	7326	7648	69.11%	85.05%	3.13	6.49
s13207.1	253	3269	35959	36212	893	4894	53834	54727	379	990	10890	11269	69.72%	79.77%	3.21	4.86
s15850.1	113	1677	18447	18560	442	2328	26050	26492	310	709	7799	8109	57.72%	70.06%	2.29	3.27
s38417	1005	5135	61620	62625	1506	24524	294288	295794	994	2922	35064	36058	43.10%	88.09%	1.74	8.20
s38584.1	157	3393	40716	40873	673	3923	47076	47749	389	974	11688	12077	71.29%	75.17%	3.38	3.95
Avg.													65.07%	74.30%	2.96	4.56

Table 3.6 Comparison the encoding efficiency for full scan with the proposed Cocktail Scan



Circuit	Full Scan			Cocktail Scan								
	Test Vectors	Data Volume	Test Cycles	Seeds	Test Length	Cubes for RAS	Bit Flips	Data Volume	Test Cycles	Volume Red	Speed Up	
s1423	68	5032	5100	1	64	77	212	1558	1834	69.04%	2.78	
s5378	263	47077	47340	4	16	377	793	7853	8710	83.32%	5.44	
s9234.1	371	78281	78652	8	16	490	1071	11327	12526	85.53%	6.28	
s13207.1	476	303688	304164	4	16	597	1343	17325	18732	94.30%	16.24	
s15850.1	435	232290	232725	8	16	660	1180	17252	18560	92.57%	12.54	
s35932	65	112320	112385	4	16	659	1410	23832	25306	78.78%	4.44	
s38417	901	1474036	1474937	16	32	1238	3655	70036	74203	95.25%	19.88	
s38584.1	647	922622	923269	20	32	1237	2951	63932	67523	93.07%	13.67	
Avg.										86.48%	10.16	

Table 3.7 Comparison of data sizes of the proposed Cocktail Scan and previous works

Circuit	Golomb [53]	FDR [56]	EFDR [80]	VIHC [55]	Alternating Run-Length Codes [81]	Selective Huffman [82]	9 Code [83]	Proposed
s5378	14941	12352	11426	11450	11687	10666	11497	<b>7853</b>
s9234	21482	22150	21364	20697	21600	17987	19283	<b>11327</b>
s13207	33205	30892	29901	27258	32710	37996	29240	<b>17325</b>
s15850	28638	26021	24635	24713	26329	26175	25867	<b>17252</b>
s38417	117951	93405	64906	76767	64906	67542	<b>64906</b>	70036
s38584	85217	77849	73868	75062	77451	71478	68691	<b>63932</b>
<b>Avg.</b>	322483	286355	255578	287693	263607	250226	248219	<b>211557</b>

Table 3.8 Power reduction obtained for RAS in terms of switching activity of SFFs

Circuit	Total switching activity			Peak switching activity		
	Full Scan	RAS	RAS Red%	Full Scan	RAS	RAS Red%
s1423	167K	4.45K	97.33	49	1	97.96
s5378	3.23M	12.2K	99.62	107	1	99.06
s9234.1	8.2M	28K	99.66	127	1	99.21
s13207.1	97M	42K	99.95	360	1	99.72
s15850.1	61.4M	69.5K	99.88	310	1	99.68
s35932	90.6M	112K	99.87	920	1	99.89
s38417	1115M	841K	99.92	886	1	99.89
s38584.1	657M	915K	99.90	771	1	99.87
<b>Avg.</b>			99.51			99.41

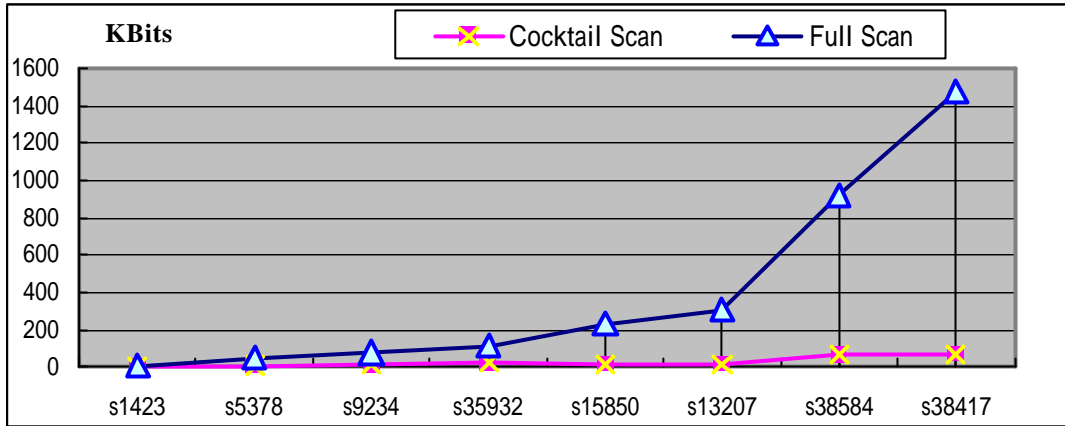


Figure 3.9 Test volume plots of each benchmark circuit for the full scan scheme and the Cocktail Scan scheme

### 3.4.3 Hardware Overhead

For RAS, the scan cell (such as in Figure 3.2) is slightly larger than the traditional scan cell. The area overhead of RAS can be expressed as

$$\frac{7 \times N_{ff}}{N_g + 10 \times N_{ff}} \quad (2)$$

where  $N_{ff}$  is the number of scan cells and  $N_g$  is the number of gate counts. In our proposed scheme, the Address Register is modified and, in addition, the address decoder is also considered for the area cost overhead. The size of our decoder was obtained by synthesizing it using Synopsys commercial tools with a TSMC cell library. In Table 3.9, the hardware overhead of this scheme is listed to be compared with that of the standard scan design for each benchmark circuit. It is seen that an approximate 18 percent increase on the overhead is needed for this scheme as compared to the standard scan design.

Table 3.9 Hardware overhead comparison between RAS and standard scan designs

Circuit	PPI	Standard Scan Gate Counts	RAS Gate Counts	Hardware Overhead%
s1423	74	1379	1764	21.8
s5378	179	4241	5099.5	16.8
s9234.1	211	7872.5	8843	10.9
s13207.1	638	13549	16710	18.9
s15850.1	534	14941.5	17738.5	15.7
s35932	1728	31617	41377	23.6
s38417	1636	36639	44441	17.5
s38584.1	1426	34016.5	42719.5	20.3
Avg.				18.2

### 3.4.4 Discussion on Some Problems on the RAS Architecture

As mentioned previously, in order to observe the output response, a MISR needs to be used, and this will increase the area and the test power. To reduce this overhead, a combinational compactor such as X-Compactor [84] needs to be adopted. The area overhead of the comparator reported in [84] is very small, e.g., a compactor of 10 outputs has a size of 232 2-input XOR gates. For our experimental circuits, the maximum number of scan cells is 1728, which is equal to 3712 2-input XOR gates. Also, for each scan cell, the routing of scan enable signals from the decoder will cause the routing congestion problem. To solve this problem, hierarchical decoders can be used to replace the single decoder. The concept is to distribute several local decoders evenly in the layout and an example is shown in Figure 3.10. Together with careful layout planning and physical location of scan cells, this problem can be solved. In a recent breakthrough of RAS architecture [85], it has been reported that further reduction of area can be achieved by using a memory-like architecture with sense amplifiers for output reading. The area overhead attained is no more than 4% as



compared to the multiple scan architecture. This makes the application of RAS more practical.

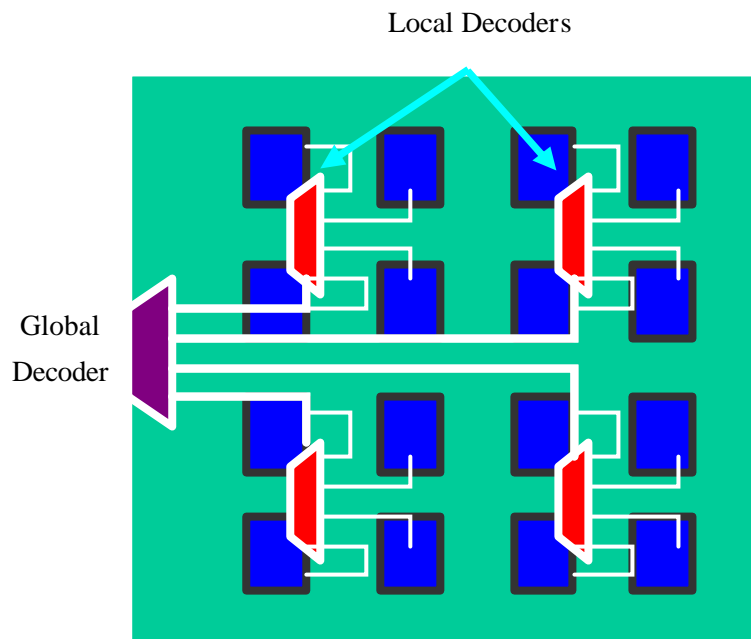


Figure 3.10 Reducing routing area and signal skew by using hierarchical decoders rather than using a global decoder

### 3.5 Summary

In this chapter, we have proposed and demonstrated a *Cocktail Scan* testing scheme to save the test vector storage, the test application time, and consequently the testing power. The scheme takes two phases to generate and apply test patterns for the DUT. In the first phase, a few number of seed patterns are applied to the DUT to generate segmented random scan patterns to test the circuit. For this phase testing, the test application is simple and a large number of faults can be detected. In the second phase, deterministic patterns are generated for the remaining faults and are applied to the circuit by the RAS fashion. For this set of test patterns, a proposed process utilizing several improved strategies, namely, Test Response Abandonment, CSC and

BPBTVD, is used to reduce the test vector volume and the number of bit flipping. Experimental results show that the process is very effective in reducing the number of bit flipping, which leads to an 86% reduction in test data and 10 times of speedup in test application time. Overall, experimental results also show that, for the *Cocktail Scan* scheme, the test data volume for this scheme increases only slowly with the size of the tested circuit, making it effective to be applied to large size circuits.



# Chapter 4 Adaptive Encoding Scheme Using Embedded Memory for Low-Cost and Low Power SoC Test

## 4.1 Introduction


In this chapter, we propose an Adaptive Encoding scheme, which handles test data in variable block size, in contrast to the fixed size as in the conventional Correlation-related schemes (Section 1.6.1), to achieve higher test data compression. The scheme utilizes a special decoder machine to deal with blocks which could be of flexible size to obtain better compression capacity than previous works. Techniques of two-phase test and test vector reordering to be incorporated with the scheme are investigated to further improve test efficiency. In addition, a constrained minimum transition fill strategy to fill patterns is adopted to help make tradeoff between test compression and test power. Experimental results show that significant reduction for test volume, test time and test power is achieved for this proposed scheme.

This chapter is organized as follows: In Section 4.2, the compression principle of Correlation-related test compression method is first reviewed briefly to bring out the motivation to develop Adaptive Encoding. Then, in Section 4.3, the proposed encoding scheme and the implementation of the decoder machine are described. In Section 4.4, a two-phase test and test vector reordering techniques which can be incorporated with the scheme to improve its efficiency are presented. In Section 4.5 and 4.6, the encoding efficiency and test time for the encoding scheme are investigated theoretically respectively. In Section 4.7, the Constrained MTF strategy, which can be incorporated in the scheme to achieve simultaneous test data and power reduction, is presented. Experiment results of the scheme are given in Section 4.8 and

finally, summary for this work is given in Section 4.9.

## 4.2 Correlation-Related Compression Methods

Due to the circuit structure dependency, correlation between test vectors are usually high and only small amount of different bits need to be changed from one test vector to another one. Random-Access-Scheme (RAS) follows this concept by flipping only different bits between two consecutive test patterns [47, 86] to achieve efficient test application. Suppose there are two test patterns, T1 and T2, which may contain don't care bits, as shown in Figure 4.1. DIFF pattern is defined as  $T1 \oplus T2$ . For the RAS scheme, any arbitrary scan flip-flip bit can be changed alone. For this example, only three bits need to be changed when T2 is loaded after T1. Thus test pattern, T2, can be eliminated from storing in the memory of the ATE and a specially designed circuit is needed to handle flipping of the selected bits.



T1:	1110	0110	1100	0010
T2:	1100	01x0	11x0	x001
Diff:	0010	0000	0000	0011

Figure 4.1 Two 16-bit test patterns and their *diff* pattern for demonstration of RAS

Unlike RAS, which has large hardware overhead, the authors in [51] exploited the embedded hardware for test decompression. For that proposed scheme, the test program, test vectors and replacement words are initially transmitted from the ATE to the internal memory of the CUT. During scan testing, an embedded processor of the CUT iteratively runs the test program to “configure” test patterns with replacement words in the memory and then loads the configured patterns into scan chains. After

testing, the test results are captured and the test data are continuously loaded from the ATE to the internal memory during testing. Test data in the memory are organized as blocks to be processed by the embedded processor. It was shown that the size of block affects the compression of test data [51].

In the above, to obtain the maximum compression, simulation must be performed to select the best size for blocks. However, even if we know the best block size for a design, it can not be adopted arbitrarily since some design specification will constrain the size of blocks. This is especially true for an SoC, which may have cores of different sizes. Hence, the advantage of achieving high compression ratio of the Correlation-related methods is somewhat restricted. In the next section, we will propose an Adaptive Encoding scheme to eliminate the above problem.

## 4.3 Adaptive Encoding



### 4.3.1 Encoding Scheme

If the size of blocks is fixed and data are processed in blocks, the scheme is called block-replacement (BR) scheme [51]. However, if the size of the blocks is variable for different patterns, a more efficient compression is usually resulted. Hence, instead of treating a test pattern in blocks, we use “packet” to represent the change of bits in the test pattern. A packet is a word, which is divided into three fields: *address*, *data length* and *data*, with variable length of bits. *address* contains the information of the location, for the replacing scan test pattern, of the starting bit which differs from the corresponding bit of the to-be-replaced test pattern. *data length* indicates the number of bits of the test pattern which is to be replaced starting from the location of *address*. *data* is the part of the DIFF pattern (in Figure 4.1) used to replace the portion of the to-be-replaced test pattern starting from the location of *address* and used to make the

replacing pattern. Figure 4.2 shows three different ways to form packets for the example of Figure 4.1, where the to-be-replaced test pattern T1 is to be replaced by the replacing test pattern T2. The figure shows cases of using one-packet, two-packet and three-packet to represent T2. We take the two-packet case for explanation: For the replacing test pattern T2, the *address* of the starting bit of the two packets is the 3rd bit (0010) and 15th bit (1110) respectively. The *data length* is 1 (0000) and 2 (0001) respectively, and the *data* is “1” and “11” respectively. Different packet representation will lead to different compression results. For this example, two-packet representation has the best compression result. The principle to form packets is to achieve as less data volume as possible.

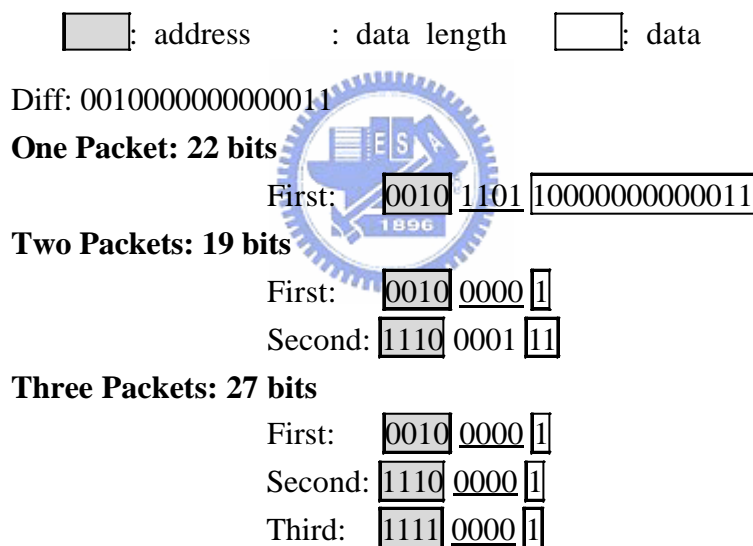
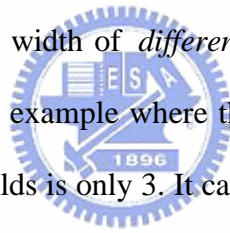


Figure 4.2 Three encoding alternatives of using packet representation for the same pattern: one-packet, two-packet, and three-packet

The width of *address* and *data length* fields is  $\lceil \log_2 N \rceil$  bits for N-bit patterns; however, it can be reduced. For *address*, since packets are processed in an increasing order of *address*, the *address* field, instead of containing the starting location of differing bit(s), can contain the number of 0s of the DIFF pattern. This improves the

encoding efficiency. The new format for *address* is called *difference address* and its width is determined by the maximum *difference address*, say  $D$ , among all packets of each pattern. Consequently, the width of *difference address* becomes  $\lceil \log_2 D \rceil$  bits, which is shorter than the width of *address*,  $\lceil \log_2 N \rceil$ . In the same manner, the width of *data length* field can also be reduced to  $\lceil \log_2 K \rceil$  bits according to the maximum width of *data*,  $K$ , among all packets of each pattern. To support these two fields, *difference data* and *data length*, with variable width, two additional header fields are added to the compressed data to indicate their widths for each pattern. The width of these header fields only needs  $\lceil \log_2(\log_2 N) \rceil$  bits, which are negligible as compared to other fields. Before one pattern is decoded, these two header fields are first loaded into the decoder to configure the width of *difference address* and the width of *data length*. Figure 4.3 shows another example where the length of the original pattern is 32 and the width of the header fields is only 3. It can be seen that the original width of *address* is reduced from 5 to 4 bits and, for *data length*, from 5 to 2 bits. The final compressed data only has 24 bits; thus, it has a reduction of  $(32-24)/32 = 25\%$ .



Diff: 01100000 00000000 11010000 00000000

*difference address*: 1, 13

width of *difference address* =  $\lceil \log_2 13 \rceil = 4$  bits

max length of *data* = 4

width of *data length* =  $\lceil \log_2 4 \rceil = 2$  bits

**Improved:** 18 bits

First: 0001 01 11

Second: 1101 11 1101

**Final compressed data:** 24 bits

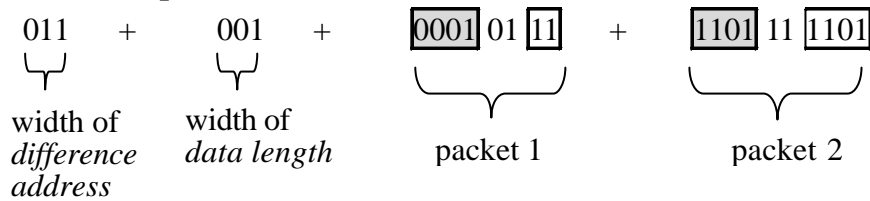


Figure 4.3 Using *difference address* and reduced *data length* to improve encoding efficiency

### 4.3.2 Decoder Machine Design and Its Operation

To decode a test pattern, a decoder machine needs to be developed. The test architecture for Adaptive Encoding is shown in Figure 4.4 (a). In our implementation, we only require one scan-in signal, which facilitates the usage of a low-cost ATE. The encoded test data serially shifts through this scan-in pin to the decoder. The scan clock, *Sclk*, for scan chains is controlled by the decoder. Figure 4.4 (b) shows that a test pattern is stored in the embedded memory and the decoder machine configures it to the next pattern and loads it onto scan chains. The main actions that this decoder machine performs for each test pattern are:

- Step 1: Loads three header fields, the number of packets, the width of *difference address* and the width of *data length* to the decoder machine;
- Step 2: Loads and uses packets to configure the next test pattern by



communicating with the memory;

- Step 3: Loads the test pattern in the memory to scan chains.

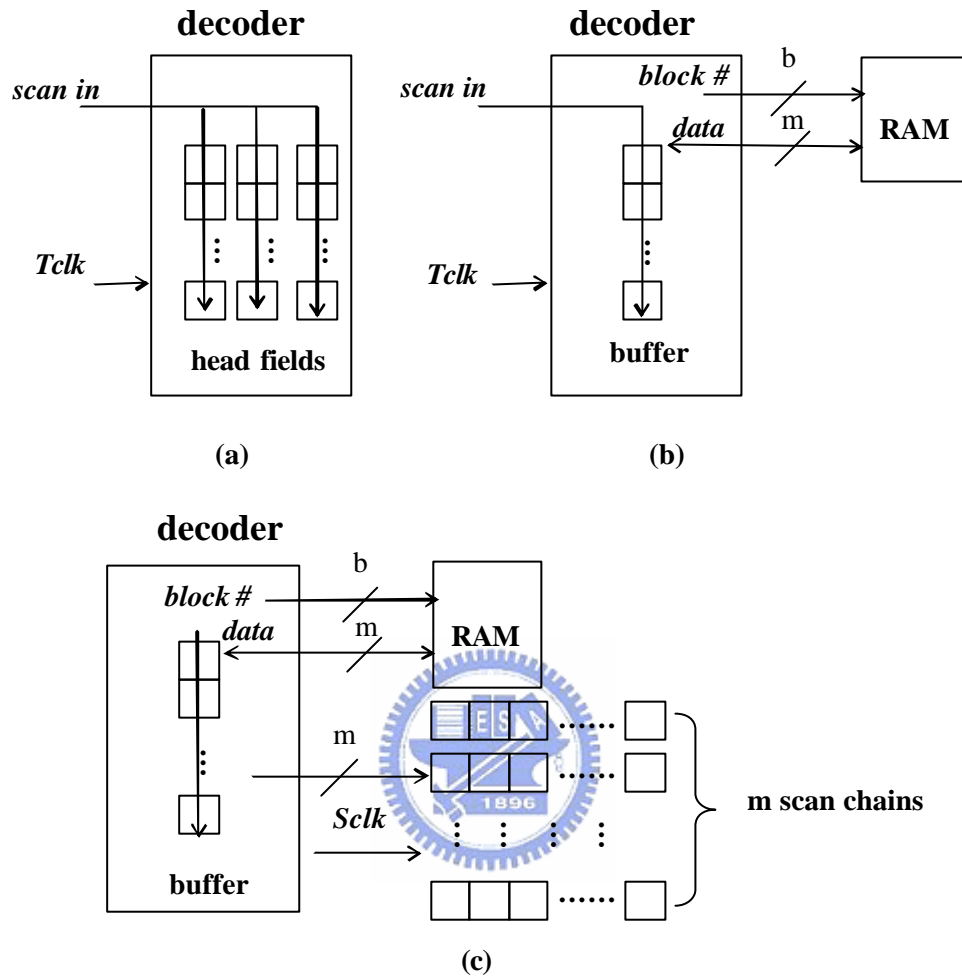


Figure 4.4 The three main steps that the decoder machine does: (a) step 1: loads three head fields (b) step 2: configures test pattern (c) step 3: loads test pattern

In Step 1, in addition to the two header fields mentioned in Section 4.3.1, there is also a header field, which indicates how many packets are to be loaded into the decoder machine for each test pattern. In Step 2, encoded packets are sent to the decoder machine to be configured into blocks in the memory. To calculate the actual address of the *difference address*, an adder of a length of  $\lceil \log_2 N \rceil$  bits is used to

accumulate the address for each packet. The adder is initialized to zero before each pattern is decoded. Once the actual address is calculated for each packet, decoder machine fetches a block from the memory to modify it. For this action, there are three sub-steps. First, the decoder machine gets the address from the adder and then the address is divided in two parts, *block number*, say  $b$ , and *offset*, say  $k$ . *block number* is the physical block number in the memory and *offset* is the starting location where the block is to be replaced from. After *block number* is sent to the memory, the selected block will be read and then updated with data in the buffer, as will be described later.

Figure 4.5 shows the detail architecture for the decoder machine. How input data is decoded is explained: In the figure, at the beginning of Step 2, the machine resets all flip flops of the buffer, sets *load* to a 1 and sends *offset* to the offset decoder. After that, *data* is loaded into a specific flip flop which is selected by the offset decoder while other flip flops behave like shift registers. Once the buffer is full or ready, the decoder sets *load* to a 0 and then selects a target block from the memory. Thereafter, the selected block loaded from memory is configured by XOR gates with the DIFF pattern in the buffer. Finally, the decoder machine writes the modified block back to the memory. The decoder machine runs at the system clock to perform those actions, which takes two system cycles for changing and writing a block within one test clock. Therefore, the ATE can continuously send packets to the decoder machine without the need of additional memory space to store replacement words. If *data* is too large to be filled in the buffer, the decoder increases the address counter and goes on to select the next block from memory to be modified without interruption.

In Figure 4.5, the relationship between address and the memory organization is also shown, where the address from the adder's output is  $\lceil \log_2 N \rceil$  bits, and the buffer has  $m$  flip-flops, which is equal to the block size of the memory. The number of scan

chains can be different from  $m$ , but here we assume it is  $m$  for simplicity. Two fields, difference address and data, of a packet are shown to explain how the decoder recognizes them; in fact, they both come from the scan-in pin. With *offset* and an offset decoder, the machine achieves random access for the selected block and a variable size of encoded block.

After all packets for one test pattern are decoded, the decoding process goes into the final step. The machine loads the configured test pattern in the memory to scan chains and shifts the test result out at the same time. Also in this step, ATE should stop sending data because the decoder is busy in loading a test pattern. In our scheme, since the time when Step 2 will finish can be known in advance (we can do that by analyzing packets), the synchronization problem is avoided by inserting a vector repeat filling instruction [90] at the time when Step 3 starts. Although vector repeat instructions in the ATE do not come for free, their numbers needed equal to the number of test patterns and are negligible.



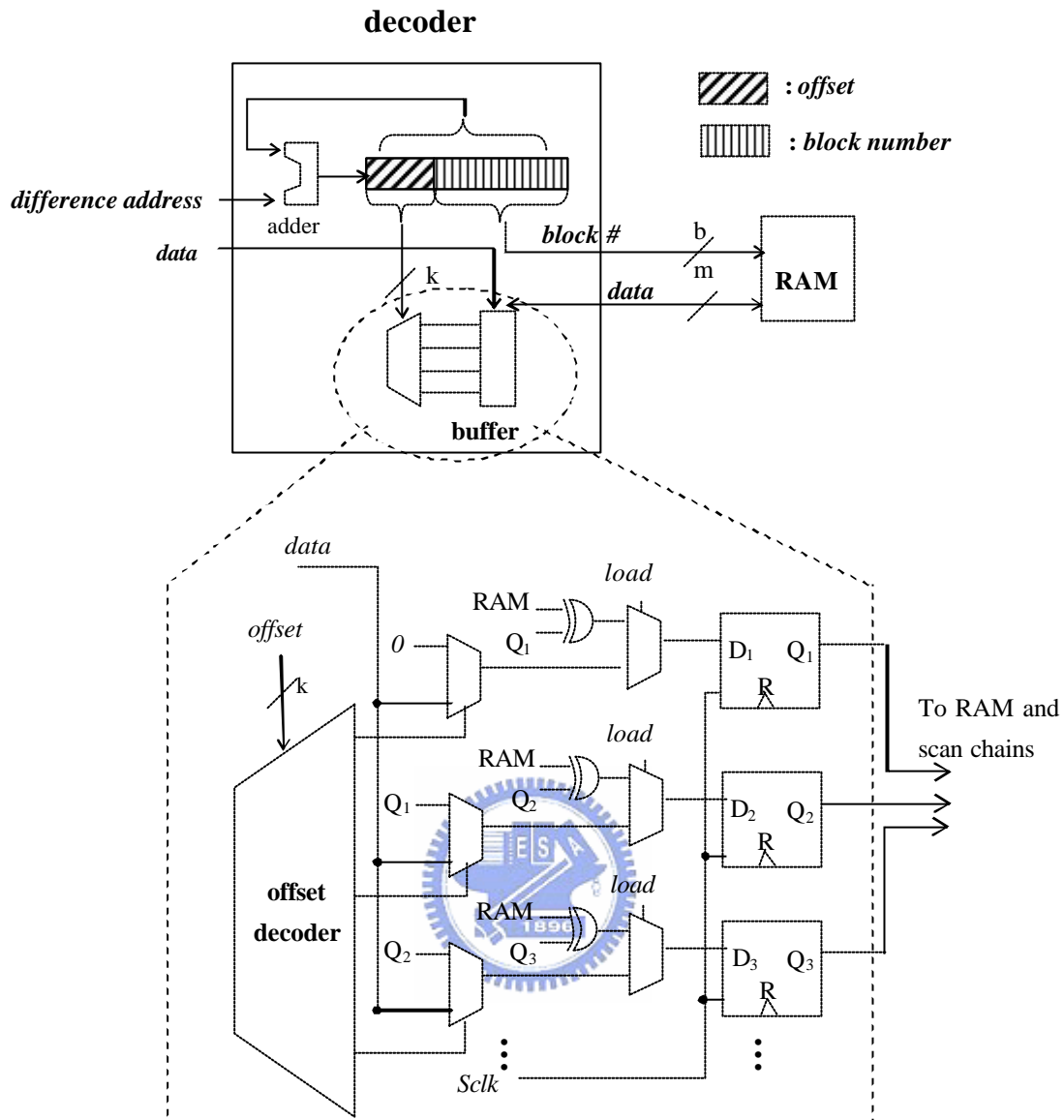


Figure 4.5 Relationship between decoder's address and memory blocks, and implementation of the memory buffer to support updating memory block in a random access fashion

Among the three steps to decode and shift a pattern, the test time in Step 2 dominates among all times of the steps; therefore, we describe this step in more details by using an example as in Figure 4.6, where a DIFF pattern is decoded into two packets. In Figure 4.6(a), after three cycles, three bits for *difference address* are loaded and then the decoder calculates the address for the first to-be-flipped bit. Once

the address is obtained, the block is fetched to be modified (Figure 4.6(b)). In Figure 4.6(c), after two bits of *data* for the first block are loaded, the decoder modifies and writes the block back to the memory. In Figure 4.6(d), the decoder automatically fetches the next block to be modified since the loading of *data* is not done yet. After the next block is also modified, the block is updated in Figure 4.6(e). Next, for the second packet, the decoder will calculate the new address for the first to-be-flipped bit (in Figure 4.6(f)). Figure 4.6(g) and Figure 4.6(h) are similar as the first packet.



: difference address   : data length     : data  
 \_\_\_: block number     : offset

DIFF: 0010 1000 0000 1000

First Packet: 010 10 101

Second Packet: 111 00 1

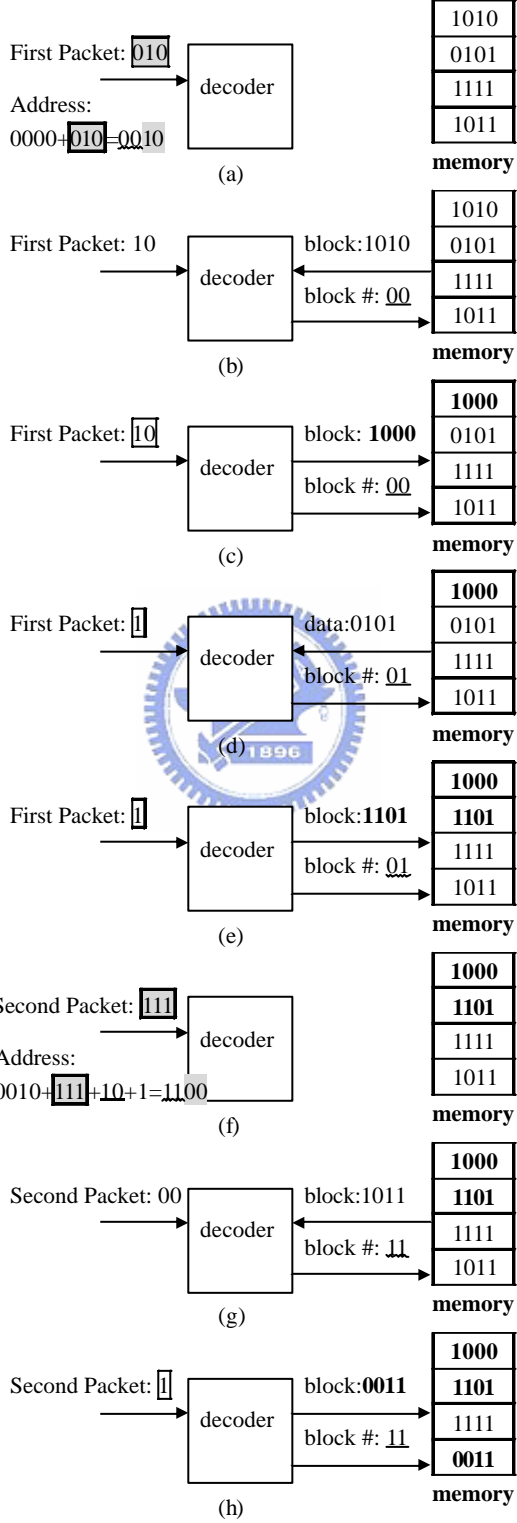


Figure 4.6 An example to demonstrate the decoding process in Step 2

## 4.4 “Two-Phase Test” and “Test Vector Reordering” Techniques for Improvement on Test Compression

The data compression of Correlation-related compression method is affected by the total number of bit flips between test patterns. The proposed scheme, when incorporated with the techniques of “two phase test” and “pattern re-ordering” [86], can be further improved its test compression efficiency. The “two-phase test” consists of generating test patterns in two phases, namely, it generates test patterns randomly in the first phase and then generates patterns deterministically, which aim to test specific faults, in the second phase. For the first phase, patterns are randomly generated to detect easy-to-detect faults. The way to generate a random pattern using our decoder machine is to randomly generate  $m$  bit first. Then those  $m$  bits are loaded to the decoding buffer and those  $m$  bits are shifted to scan chains from the buffer continuously until scan chains are full. For one scan chain, data of all scan cells are the same. For example, if a decoding buffer has, i.e., is connected with, four scan chains and each scan chain have eight scan cells, and the generated four bits are 1101, the scan cells of the first, second and fourth scan chains are all bit 1 while the third scan chain is bit 0. The decoding buffer needs eight cycles to fill the four scan chains. Therefore, each random pattern needs  $m$  bits as a seed and they are repeatedly shifted into a CUT. This saves the scan-in power since the same bit is shifted into a scan chain.

In the above, although the bit dependence between scan cells limits the fault coverage, patterns generated deterministically in the second phase increase the fault coverage.

Sequences of test patterns of different orders results in different number of bit flips. In the second phase, in addition to the aim to increase the fault coverage, a test vector

reordering technique is used to reduce the number of bit flips. The problem of finding a good order of test patterns to reduce the number of bit flips can be formulated as a Minimum Bit Flip Problem (MBFP) [86]. A process is proposed to solve this problem as follows:

Suppose  $N$  test patterns are to be reordered, a graph is built with  $N$  nodes, which represent the  $N$  test patterns. Each edge between two nodes of the directed graph represents an applied order of patterns. For example, in Figure 4.7 (a), edge  $E1$  represents that node  $A$  (pattern 1x00) is applied prior to node  $B$  (pattern 1111) while edge  $E2$  represents the reverse order. Each edge is associated with a cost which indicates the number of bit flips while patterns change from one to the other. For test patterns without don't care bits, the numbers of bit flips are the same for the two edges. For this case, only undirected edges are needed, which is shown in Figure 4.7 (b). However, test compression methods often compress test patterns with don't care bits and thus the number of bit flips is dependent on the applying order of patterns and directed edges are used in the graph. For all patterns to be ordered, a complete graph can be built with nodes connected with each other with all the edges. The test vector reordering problem can then be formulated as an Asymmetric Traveling Salesman Problem (ATSP) and a heuristic algorithm is used to solve it.

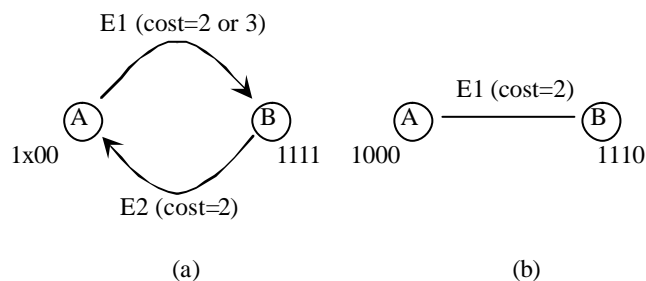


Figure 4.7 Graphs to model test sequence ordering: (a) Direct edges for partially specified patterns (b) Undirected edges for fully specified patterns



In Figure 4.7, for patterns with don't care bits, a cost model is used to help decide the cost of bit flip between two nodes. For example, the cost of edge  $E2$  in Figure 4.7(a) is 2 since the third and fourth bit of node  $B$  need to be changed from 1 to 0. However, as for  $E1$ , the cost may be 2 if the "x" of  $A$  is mapped to 1 or may be 3 if it is mapped to 0. Four cost models may be used [86]. Different cost model will lead to different results in solving the ATSP problem. Here we use an improved Estimated cost model of [86] to model the costs of edges.

The idea of Estimated cost model is to associate each corresponding bits of patterns a probability for which the bits are logic "1". For example, for four patterns, 010, x11, 1x0, 1xx; for the first bit, there are two 1s and a 0 at the first bit position, the probability of occurrence of 1 is  $2/3$  ("x" is not counted), similarly, for the second bit is  $2/2 = 1$  and for the third bit is  $1/3$  respectively. When we assign values to "x"s of the above patterns to obtain their DIFF pattern, we will assign the first bit of the pattern x11 to be "1" since the probability is  $2/3$ . Similarly, the second bits of the patterns 1x0 and 1xx are assigned to be "1" and the third bit of the pattern 1xx is assigned to be "0". In this way, all don't care bits are assigned and a DIFF pattern can be calculated. Then, the improved estimated cost model calculates the encoded data volume of the DIFF pattern and assigns the cost to edges of the modeled graph.

## 4.5 Compression Efficiency Analysis

Generally speaking, all Correlation-related compression methods can be treated as block based replacement schemes. Each block has a header field, which indicates the position of the block in a test pattern. Suppose a test pattern of  $N$  bits has a block size

$b$ , then it has  $\left\lceil \frac{N}{b} \right\rceil$  blocks and each block needs a header field of  $\left\lceil \log_2 \frac{N}{b} \right\rceil$  bits to

represent. To replace one block,  $\left(\left\lceil \log_2 \frac{N}{b} \right\rceil + b\right)$  bits are required. Given  $M$  blocks to

be replaced in a test pattern, the data volume  $DV$  will be  $M \times \left(\left\lceil \log_2 \frac{N}{b} \right\rceil + b\right)$  bits,

where  $M = \left\lceil \frac{N}{b} \right\rceil$ . The data compression obtained for a test pattern then is  $1 - \frac{DV}{N}$ . In

the worst case, all blocks are to be replaced, hence:

$$DV = M \times \left(\left\lceil \log_2 \frac{N}{b} \right\rceil + b\right) = M \times b + M \times \left\lceil \log_2 \frac{N}{b} \right\rceil = N + M \times \left\lceil \log_2 \frac{N}{b} \right\rceil$$

In this situation, the header field, which is the overhead for each block, will dominate  $DV$ . For a test pattern, different values of  $b$  and  $M$  result in different  $DV$  and compression rate. Figure 4.8, where  $DV$  is plotted with respect to  $M$ , the number of blocks needed to be replaced, and  $b$  for a test pattern of  $N=2048$  bits, explains this. In the figure, when most blocks are replaced, the overhead introduced by the header field harms greatly data compression. Therefore,  $b$  should be chosen larger to reduce  $DV$ . On the contrary, if  $M$  is small,  $b$  has the most impact on  $DV$ . For this case,  $b$  should be chosen smaller to reduce  $DV$ .

As observed in Figure 4.8,  $DV$ , is highly dependent on  $b$ . Moreover, it is known that as test patterns are generated by an ATPG program, the percentage of don't care bits rises while they are generated [45]. When the number of don't care bits is small, it tends to have more blocks to be replaced. For this case, a large block size is preferred to reduce the effect caused by the header bits. On the contrary, if there are many don't care bits, fewer blocks need to be changed and a smaller block size is preferred. This scheme has a merit as compared to the RAS scheme which can be viewed as a BR scheme of 1-bit, and other BR schemes of fixed-sizes which can not exploit the advantage of the taking various distribution of don't care bits into consideration to increase compression rate.

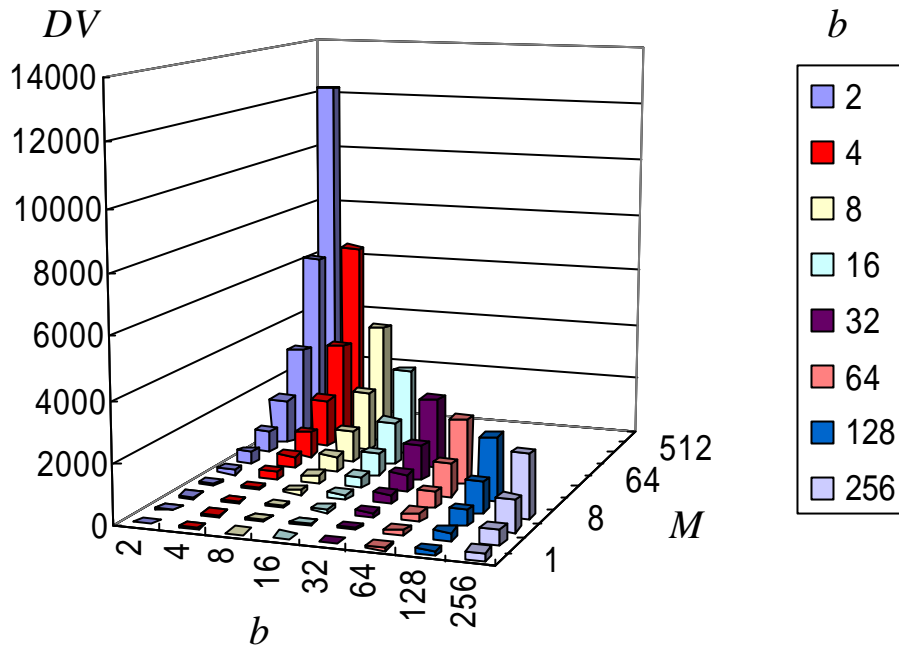


Figure 4.8 Compressed data volume,  $DV$ , v.s. the number of blocks to be replaced,  $M$ , and the block size,  $b$

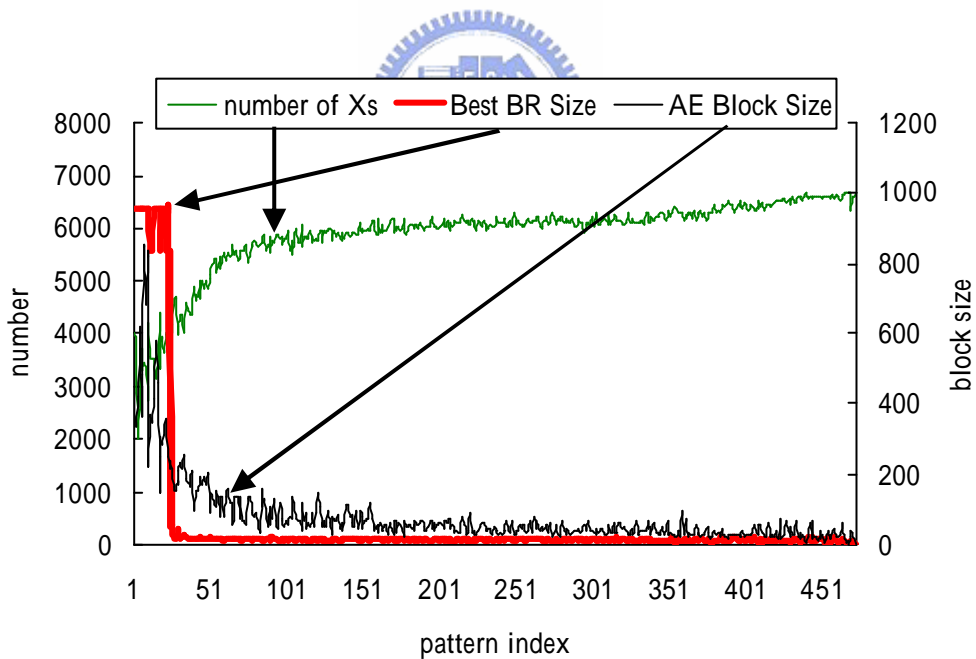


Figure 4.9 The block sizes, which give the best compression, are plotted with respect to test patterns for a test set for circuit **b19\_1** for Adaptive Encoding scheme and BR scheme respectively. The number of don't care bits is also plotted for each pattern

Figure 4.9 shows a real case study for a test set of 473 test patterns for a circuit **b19\_1**, which has 6642 scan cells. We plot the distributions of the best block size, for which the optimum bit compression was obtained, with respect to patterns for the Adaptive Encoding scheme and the BR scheme respectively. In the figure, the number of don't cares bits for each pattern is also plotted. We can see that for the Adaptive Encoding scheme, the optimum size (*data length*) initially is high and decreases gradually. However, for the BR scheme, the best block size of BR changes sharply at about the 30<sup>th</sup> pattern. So once it is fixed initially at, say 1024, to achieve for the maximum compression for the first 30 patterns, the compression efficiency drops rapidly for the patterns after 30 patterns. However, it is not so for the Adaptive Encoding scheme since its *data length* can always be adjustable and better data compression can be obtained.



## 4.6 Test Application Time Analysis

The test application time is also an important issue for a compression scheme. In this section the test application time of the proposed Adaptive Encoding scheme is analyzed.

When the test application time is considered, the processes to shift test data and to apply a test pattern should be discussed. For the scheme, Step 1 and Step 2 shift test data into the decoder machine. Since only a scan-in pin is used, the test time to shift the encoded data is directly proportional to the number of bits shifted. Once a test pattern is configured in the memory, the decoder loads the pattern to scan chains at the system speed of the CUT in Step 3. Suppose that the frequency of test clock of an

ATE is  $f$  and the system clock of the CUT is  $f'$ , and  $r = \frac{f'}{f}$ . For a CUT, the

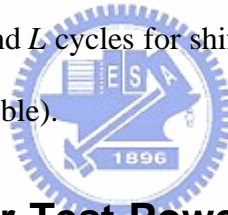
maximum shift length,  $L$ , is  $\left\lceil \frac{N}{b} \right\rceil$ , as obtained from the previous section. Therefore,

the time to shift a test pattern from the memory to scan chains is  $\left\lceil \frac{L}{r} \right\rceil$ . So the test

application time (ATE cycles) for a test pattern is  $DV + \left\lceil \frac{L}{r} \right\rceil$  and the total test time is

$\sum_i (DV_i + \left\lceil \frac{L}{r} \right\rceil)$ . When  $b$  is large,  $L$  becomes smaller; therefore, the time to load a test

pattern in Step 3 is negligible as compared to the shift time of Step 2. Hence, only the shift time needs to be considered. As the test data volume is reduced, the test time is also reduced. Hence the test application time of the Adaptive Encoding scheme is also efficiently short. For the two phase test, for applying random patterns of the first phase, only  $b$  bits loading time and  $L$  cycles for shift length are needed (the additional header bits shifting is also negligible).



## 4.7 Pattern Fillings for Test Power Consideration

Test power is also an important problem to be considered during test pattern compression. To reduce the test power, don't care bits of test patterns are filled with "0" and "1" to reduce the number of transitions when shifting patterns. However, the strategy to get low power test is somewhat contrary to the strategy to obtain test compression. Some traditional approaches used minimum transition filling (MTF) to reduce test power [8]. However, MTF is not directly applicable for the proposed Adaptive Encoding since it may significantly increase the number of bit flips, consequently, decrease the test compression efficiency. To make a tradeoff between the two strategies, we propose a modified strategy, Constrained MTF, for which only part of unspecified bits of the pattern is to be filled with MTF, to be incorporated with

the encoding scheme.

In developing Constrained MTF, we use the Weighted Transition Count (WTC) [60] to estimate for power cost. The concept of WTC is to give more costs for bits at the front positions of test patterns since they pass more scan cells and cause more transitions. We give higher priority to apply the MTF strategy to the former bits than to the latter bits of a pattern. To determine which bits are to be applied MTF, a condition is used to for judgment :

$$\alpha \times (\text{reduced WTC}) - (1 - \alpha) \times (\text{increased bit flips}) > 0$$

where  $\alpha$  is a user defined value between 0 and 1. With this condition, the proposed Constrain MTF is: For each scan chain, we start to check each don't care bit to see whether the above condition is met under a given  $\alpha$ . If the number of reduced WTC of a don't care bit applied with MTF is larger than the number of increased bit flips, the condition is satisfied and the Constrained MTF is applied. If the condition is not satisfied, MTF is not applied for the bit. This measurement guides us to make a tradeoff between the increased number of bit flips and the amount of reduced WTC when a bit is to be filled with MTF. If we want lower test power, we should choose a larger  $\alpha$  so that more bits satisfy the condition. However, this means that compared with the original achievable test compression with Adaptive Encoding, more number of bit flips are added. If we choose  $\alpha$  to be smaller, fewer don't care bits are filled with MTF. For this case, we obtain little test power gain. We note that when we make  $\alpha$  be 1, the proposed MTF becomes the traditional MTF; and when  $\alpha$  be 0, it equals to the Adaptive Encoding scheme without any MTF strategy.

In Figure 4.10, an example is shown with a test pattern applied with Constrained MTF along with no filling and the traditional MTF strategy respectively. For the Constrained MTF, the number of flips is 9, only one more than that of the no-filling

case. Its WTC is 29, which is much less than that of the without-filling case but only a moderate increase as compared to that of the traditional MTF case.

	Resulted pattern in memory	↑ higher compression	# Flips	#WTC	↓ lower power
Adaptive Encoding, no filling ( $\alpha = 0$ )	xx10xxxx111xx1x1 + 0000000000000000 = 0010000011100101		6	82	
Vector with Constrained MTF ( $\alpha = 0.05$ )	xx10xxxx11111111 + 0000000000000000 = 0010000011111111		9	29	
Vector with traditional MTF ( $\alpha = 1$ )	1110111111111111 + 0000000000000000 = 1110111111111111		15	7	

Figure 4.10 An example pattern with three different filling strategies which result in different number of flips and WTC's

## 4.8 Experimental Results

This Adaptive Encoding scheme had been implemented in C++ and applied to some largest ISCAS89/ITC99 scan circuits and locally designed circuits, which had even higher complexity. The test patterns for the circuits were obtained by SYNTTEST tools. The results obtained are then compared with those obtained by applying some other methods [47, 51, 53-55] as shown in Table 4.1, where  $m$  is the optimal group size for Golomb [53] or the optimal block size for BR [51]. As for Selective Huffman [54], we choose the block size as 12 bits and the number of encoded states is 16. In the table, for the VIHC method, the compression/reduction is listed for the group size of 16 and for the unlimited group size respectively. The column “Max Red.” is its theoretical encoding upper bound that is practically not realizable. It should be mentioned that for the Entropy-related methods, larger group size implies larger area of the decoder and larger synchronization overhead between the decoder and the ATE. In the table, it can be seen that our proposed method outperforms the Correlation-related methods, RAS and BR, for all the circuits, and furthermore, it can

get compression ratios, which are comparable to the theoretical compression upper bounds of the VIHC method, especially for the larger size circuits: leon1, leon2, rca and fft. Also, Adaptive Encoding is very efficient for test compression especially in very large-scale designs.

Table 4.1 Compression comparison between different encoding methods

Circuit	Patterns	SFFs	Entropy-Related Methods					Correlation-Related Methods				
			Golomb [53]		VIHC [55]		Selective Huffman [54] Red.	RAS [47] Red.	BR [51]		Adaptive Encoding Red.	
			$m$	Red.	$m=16$ Red.	Max. Red.			$m$	Red.		
s35932	36	1728	4	36.56%	43.81%	51.97%	<b>54.36%</b>	-91.39%	32	33.37%	47.17%	
s38417	125	1636	8	57.10%	<b>58.01%</b>	60.39%	56.19%	-3.99%	8	18.91%	43.01%	
s38584	142	1426	8	62.20%	63.00%	66.83%	<b>65.57%</b>	11.99%	8	34.47%	54.71%	
b18_1	444	3320	16	69.42%	74.26%	79.41%	71.54%	33.03%	16	63.62%	<b>75.01%</b>	
b19_1	473	6642	16	70.41%	75.46%	85.59%	71.84%	30.93%	16	63.40%	<b>75.73%</b>	
b22_1	437	735	4	51.94%	58.63%	65.05%	<b>61.29%</b>	-14.24%	8	43.57%	60.06%	
leon1	599	10405	64	89.97%	88.75%	94.19%	80.76%	81.15%	8	88.13%	<b>90.98%</b>	
leon2	3090	49455	256	97.50%	92.80%	<b>98.75%</b>	82.79%	95.94%	8	97.56%	<b>98.18%</b>	
rca	3569	20480	64	96.29%	92.14%	98.56%	82.26%	93.06%	32	97.84%	<b>98.35%</b>	
fft	790	75723	64	93.05%	91.00%	96.96%	81.89%	93.69%	16	93.85%	<b>96.09%</b>	
Avg.				72.44%	73.79%	79.77%	70.85%	32.20%		63.45%	<b>74.00%</b>	

Table 4.2 compiles the test speedups of the Adaptive Encoding method and the BR method as compared with the traditional serial scan design. Test time is calculated based on the analysis in Section 4.6. In the table, the memory size for patterns for each circuit is listed and  $m$  is the number of scan chains and  $r$  is the speed ratio between the system clock and the test clock as is defined in Section 4.6. The column “Area Over.” is the area overhead for the decoder machine, which was synthesized by Synopsys Design Compiler, and memory is not counted. For the Adaptive Encoding method, the ATE repeats filling scan data during the period when the decoder machine is shifting a pattern into scan chains. Therefore, compared to BR, it needs additional loading time overhead as analyzed in Section 4.6. Even so, the Adaptive Encoding method still obtained higher speedups. The speedups obtained for the Adaptive Encoding method is 16.46 for  $r = 5$  (the bold character cases) with only a little area



overhead of the decoder.

Table 4.2 Test speedups for BR and Adaptive Encoding under different number of scan chains,  $m$ , and clock ratio,  $r$ , between test clock and system clock

Circuit	BR	RAM	Adaptive Encoding											
			$m = 16$				$m = 256$				$m = 1024$			
			$r = 2$	$r = 5$	$r = 10$	Area Over. %	$r = 2$	$r = 5$	$r = 10$	Area Over. %	$r = 2$	$r = 5$	$r = 10$	Area Over. %
s35932	1.50	2K	1.78	<b>1.84</b>	1.87	<b>4.0</b>	-	-	-	-	-	-	-	-
s38417	1.22	2K	1.66	<b>1.72</b>	1.73	<b>4.4</b>	-	-	-	-	-	-	-	-
s38584	1.52	2K	2.07	<b>2.15</b>	2.18	<b>4.9</b>	-	-	-	-	-	-	-	-
b18_1	2.74	4K	3.55	<b>3.81</b>	3.90	<b>1.0</b>	-	-	-	-	-	-	-	-
b19_1	1.95	8K	3.65	3.92	4.01	0.5	4.09	<b>4.11</b>	4.11	<b>1.9</b>	-	-	-	-
b22_1	1.76	1K	2.32	<b>2.42</b>	2.46	<b>3.8</b>	-	-	-	-	-	-	-	-
leon1	8.41	16K	8.24	9.74	10.37	0.7	10.85	<b>10.99</b>	11.04	<b>2.7</b>	11.03	11.06	11.08	9.1
leon2	40.98	64K	20.23	32.59	40.93	0.1	49.69	52.76	53.85	0.5	53.56	<b>54.45</b>	54.75	<b>1.8</b>
rca	46.29	32K	20.92	34.43	43.87	0.3	54.07	<b>57.73</b>	59.06	<b>1.3</b>	58.72	59.75	60.10	4.4
fit	16.26	128K	14.22	19.39	22.06	0.1	24.38	25.09	25.34	0.4	25.28	<b>25.47</b>	25.53	<b>1.4</b>
Avg.	12.26													16.46

Table 4.3 shows the experimental results on the tradeoff between test compression and test power reduction for incorporating the Constrained MTF strategy with the encoding scheme. In the table, number of chains, WTC and peak transitions, and data reduction ratio for each circuit are listed for results of our Adaptive Encoding without and with the Constrained MTF of different values of  $\alpha$ 's. We can see that when circuits have higher complexity, a larger  $\alpha$ , for example, 0.1 should be chosen. Take the circuit "leon1" to be an example, with  $\alpha = 0.1$ , we reduce the test energy by  $(57.8M-4.85M)/57.8M = 91.60\%$  and reduce the peak power by  $(5156-4353)/5156 = 15.57\%$  at the expense of 10.82% loss in test compression. Another circuit example: "rca" with  $\alpha = 0.1$ , we reduce the test energy by  $(1.17G-57.2M)/1.17G = 95.11\%$  and reduce the peak power by  $(10467-9415)/10467 = 10.05\%$  at the expense of only 4.31% loss in test compression. The larger the circuit is, the more power saving can

be achieved with a little loss of test compression. When we make  $\alpha = 1$ , i.e., traditional MTF is used, we almost can not obtain any compression since too many bit flips are introduced.

Table 4.3 Tradeoff between test power and test compression for different user defined values

Circuit	Chains	Without constrained MTF		Adaptive Encoding with constrained MTF							
		WTC / Peak	Data Red. %	$\alpha=0.01$		$\alpha=0.03$		$\alpha=0.1$		$\alpha=1$	
				WTC / Peak Red. %	Data Red. %	WTC / Peak Red. %	Data Red. %	WTC / Peak Red. %	Data Red. %	WTC / Peak Red. %	Data Red. %
s35932	16	1.53M / 904	47.17	35.29 / 7.96	44.12	62.09 / 11.84	32.91	68.63 / 11.84	24.26	69.28 / 12.39	8.82
s38417	16	5.07M / 832	43.01	40.43 / 25.12	33.84	67.06 / 44.72	24.45	75.93 / 57.57	11.64	77.71 / 61.78	-1.89.
s38584	16	4.50M / 744	54.71	40.44 / 8.87	50.53	71.33 / 10.22	41.60	82.44 / 11.02	27.33	86.22 / 11.02	-2.33
b18_1	16	66.3M / 1603	75.01	69.83 / 30.44	64.91	83.41 / 42.05	49.82	88.19 / 46.29	23.12	89.65 / 46.41	-0.28
b19_1	256	21.7M / 3392	75.73	0 / 0	73.37	41.94 / 15.27	69.33	78.99 / 30.42	53.94	92.58 / 35.02	0.40
b22_1	16	3.16M / 375	60.06	0 / 0	58.47	46.52 / 5.33	44.67	68.35 / 11.20	20.98	74.37 / 11.73	-0.66
leon1	256	57.8M / 5156	90.98	0 / 0	88.34	68.51 / 10.34	85.58	91.61 / 15.57	80.16	98.86 / 17.86	-0.10
leon2	1024	1.77G / 24303	98.18	0 / 0	95.04	81.24 / 7.67	94.36	96.77 / 9.89	92.85	99.87 / 10.92	0.91
rca	256	1.17G / 10467	98.35	44.87 / 6.32	95.25	84.36 / 9.00	95.15	96.13 / 10.05	94.04	99.05 / 10.40	44.51
fft	1024	967M / 37500	96.09	45.29 / 4.12	93.28	87.69 / 7.49	92.10	95.56 / 8.88	88.63	99.33 / 9.30	0.18
Avg.			73.00	27.61 / 8.28	69.72	69.42 / 16.39	63.00	84.26 / 21.27	51.70	88.69 / 22.68	5.00

Finding a suitable  $\alpha$  not only reduces test power significantly but also obtains satisfactory test compression. We did an experiment on the circuit “b19\_1” with different  $\alpha$ ’s ranging from 0.01 to 1 and plot the results of test compression, test energy and peak power with respect to the values of  $\alpha$ ’s in Figure 4.11. From the figure, we observe that for this case, the most suitable  $\alpha$  for “b19\_1” is 0.15, for which a 45.22% test compression, a 83.63% test energy reduction and a 32.25% peak power reduction is obtained by Adaptive Encoding respectively.

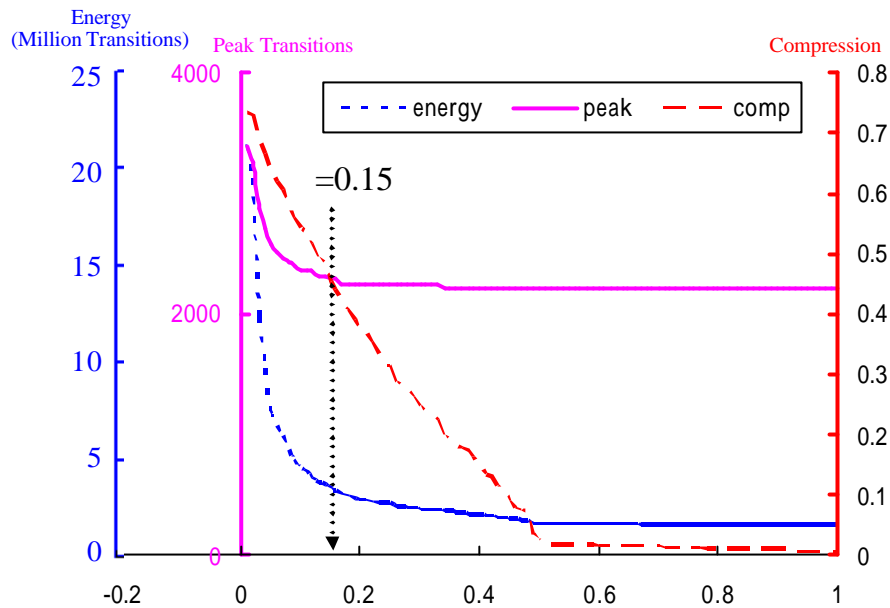


Figure 4.11 Experimental results on test data compression, test energy and peak power (transitions) for different value of  $\alpha$  on circuit **b19\_1**. The most suitable value for  $\alpha$  is 0.15 to obtain a 45.22% test compression, a 83.63% energy reduction and a 32.25% peak power reduction.

Table 4.4 shows the results of the experiments on incorporating the two phase test generation technique and the test vector reordering technique in the Adaptive Encoding scheme. In the table, the second column lists the test data volume for the Adaptive Encoding without incorporating the two techniques, the third column lists the test data volume of the Adaptive Encoding applied with the test vector reordering (TVO) technique, and the fourth column lists the test data volume of using the two phase test generation technique to generate test patterns, which gave the same fault coverage as that of the test sets of the second column, and the patterns were then applied with the TVO again to reduce their volume. For the last column, the data volumes for the random phase and the deterministic phase respectively of the two phase test generation technique are also included. For the test data volumes in second and third columns, since the test patterns were obtained for circuits from a

commercial tool SYNTEST and were very compact, applying the TVO technique obtained some reduction although the reduction was not significant. However, for the test data volume in the fourth column, the final test data volume obtained by applying the two-phase test with TVO is greatly reduced. This shows that applying the Adaptive Encoding in conjunction with the two phase test generation and the TVO techniques can greatly reduce the test data volume.

Table 4.4 Data volume for Adaptive Encoding with different techniques

Circuit	Adaptive Encoding	TVO	Two-Phase & TVO (first phase + second phase)
s35932	32864	31132	9380 (680 + 8700)
s38417	120893	117219	60890 (800 + 60090)
s38584	91630	85453	53635 (2550 + 51085)
b18_1	368226	352604	247195 (14080 + 233115)
b19_1	767955	743934	518219 (17700 + 500519)
leon1	561574	494816	335176 (22400 + 312776)

## 4.9 Summary

In this work, we have proposed an Adaptive Encoding scheme to encode the test data (patterns) to save the test data volume and the test application time. The scheme handles, instead of the data themselves, the difference between two consecutive test patterns by using packets, making the test data be encoded in variable sizes to achieve better data compression. A decoder machine is proposed to make the scheme possible and to decode the encoded data. Constrained MTF, a filling strategy, is also proposed to be adopted with the scheme to simultaneously achieve test compression and test power reduction. In addition, the scheme can incorporate the techniques to generate test patterns in two phases and to reorder the test vectors to further achieve the test data reduction. Experimental results have shown that the proposed encoding scheme

is very effective in reducing the volume of test data and test power, and in speeding up test application time for large-scale designs.



# Chapter 5 Low Power Test Compression for Multiple Scan Chain Designs

## 5.1 Introduction

The random-like filling strategy pursuing high compression for today's popular test compression schemes introduces large test power. To achieve high compression in conjunction with reducing test power for multiple-scan-chain designs is even harder and very few works were dedicated to solve this problem. This chapter proposes and demonstrates a Multilayer Data Copy (MDC) scheme for test compression as well as test power reduction for multiple-scan-chain designs. The scheme utilizes a decoding buffer, which supports fast loading using previous loaded data, to achieve test data compression and test power reduction at the same time. The scheme can be applied ATPG-independently or to be incorporated in an ATPG to generate highly compressible and power efficient test sets. Experiment results on benchmarks show that test sets generated by the scheme had large compression and power saving with only a small area design overhead.

In Section 5.2, the proposed encoding scheme, MDC, and the architecture of the decoder are first described. In Section 5.3, a complete analysis of achievable volume and power reduction for the proposed scheme with respect to different organizations of the decoding architecture is included. In Section 5.4, the proposed *ATPG-dependent* tool, which considers simultaneous test data and power reduction for multiple-scan-chain designs, is described. Experiment results on many benchmark and large-scale circuits are shown in Section 5.5 to compare and evaluate the proposed scheme with other test compression methods. Finally, a conclusion is given in Section 5.6.

## 5.2 The Proposed Multilayer Data Copy Scheme

The proposed Multilayer Data Copy (MDC) scheme is shown in Figure 5.1(a). A decoder, which has a decoding buffer to drive multiple scan chains, is used to decode compressed data. The decoding buffer is composed of a set of D flip-flops (DFFs) implicitly configured in a multilayer architecture of  $L$  layers by way of a switching box. Take a decoding buffer of  $a$  DFFs, which drive  $a$  scan chains of a CUT, as an example. If the DFFs are able to be configured into three layers, i.e.,  $L=3$ , then for layer one,  $L_{v1}$ , configuration,  $a$  DFFs are configured into one group. For layer two,  $L_{v2}$ ,  $a$  DFFs are implicitly grouped into  $m$  groups of which each group has  $b$  DFFs so that  $m \times b = a$ . For layer three,  $L_{v3}$ , configuration, the DFFs of each group of  $L_{v2}$  are further grouped into  $n$  groups of which each group has  $c$  DFFs and  $n \times c = b$ . Figure 5.1(b) shows conceptually such a multilayer structure. More layers can be continually constructed as necessary. The decoding buffer has two modes of operation: *Copy* or *Shift*. For the *Shift* mode, the DFFs act as shift registers and data is loaded into DFFs serially bit by bit from the *in* pin. For the *Copy* mode of each layer, data of DFFs of each group is “copied” into the DFFs of the next group in “block”. It is noted that during the decoding buffer is acting, the current layer is also changing with it operations. Only the last layer needs *Shift* mode while other layers need only *Copy* mode. The switching box shown in Figure 5.1(c) is used to support these operations during the decoding process and its implementation is shown in Figure 5.1(d) for one of the DFFs. For the *Copy* mode, data loading is fast and this reduces test time as well as test volume. Once  $a$  number of DFFs of the buffer are loaded with  $a$  bits of test patterns, we say one *slice* is ready and the decoder will shift the *slice* to the  $a$  scan chains of the CUT.

In Figure 5.2, there is a test cube of the length of 16 bits to be loaded into a scan design, which has 8 scan chains. For this case, the shift length for shifting the test cube

is two. Now, suppose that a three-layer buffer organization, that is:  $a=8$ ,  $b=4$  and  $c=2$ , is used. As the loading starts, for the first step, the first bit “0” is loaded into the first DFF of the buffer. At the second step, the following 1 is loaded. At the third step, *Copy* operation is applied to  $Lv_3$  since the following 3<sup>rd</sup> and 4<sup>th</sup> bits are XX, which are compatible with the first two loaded bits, 01. After the *Copy* operation is done, the two don’t care bits become the same as the two prior bits, 01. At the fourth step, *Copy* operation is again applied to  $Lv_2$  since the following four bits, 0XX1, are compatible with the previously loaded four bits, 0101. After this, we obtain a set of test data of 01010101 in DFFs. Now we say that a *slice* of test data is resident in the buffers and this slice will be loaded to eight scan chains. Since the next 8-bit slice is also compatible with the current slice, a *Copy* operation is applied to  $Lv_1$  to load the next 8-bit slice into the buffers. In this way, test data can be fast loaded into a CUT.





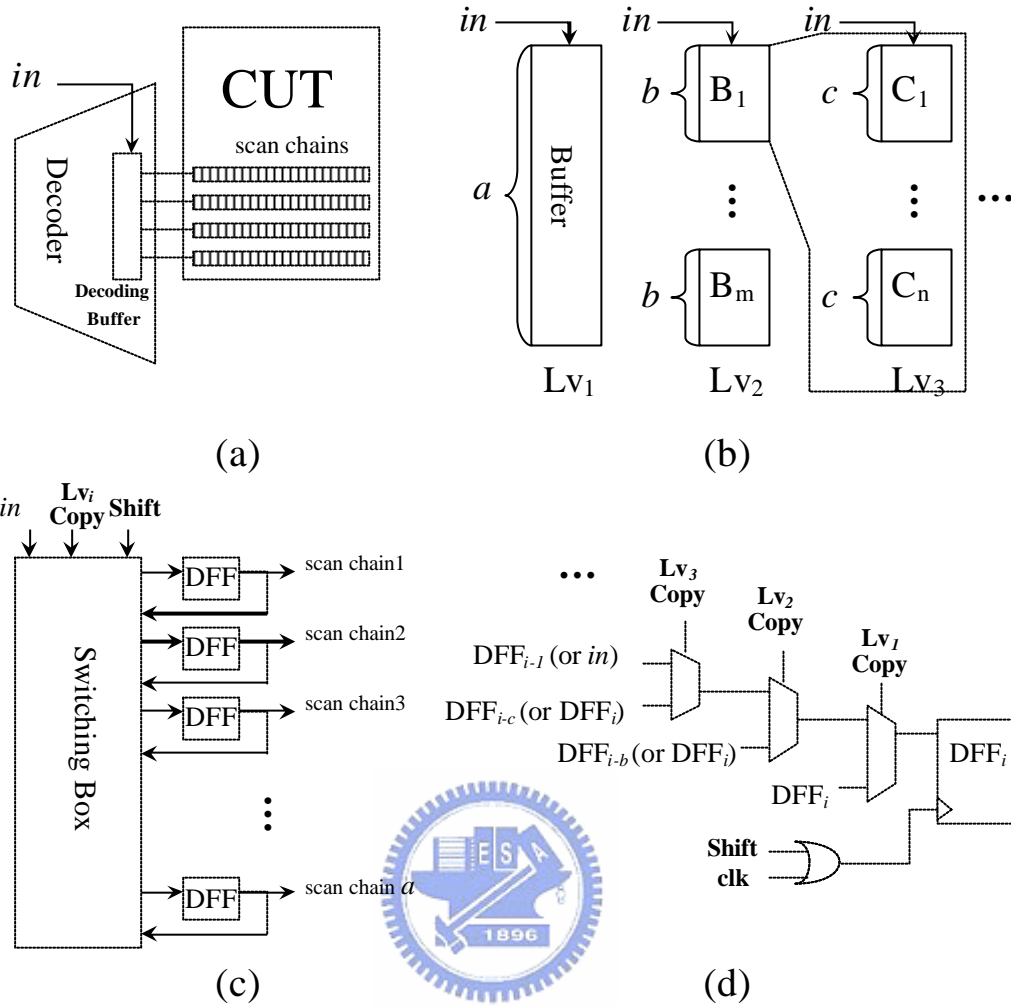


Figure 5.1 (a) Proposed decoding architecture, (b) a decoding buffer with a dffs and its multilayer organization, (c) a switch box is used to support the two operations of a decoding buffer, and (d) the switching box implementation

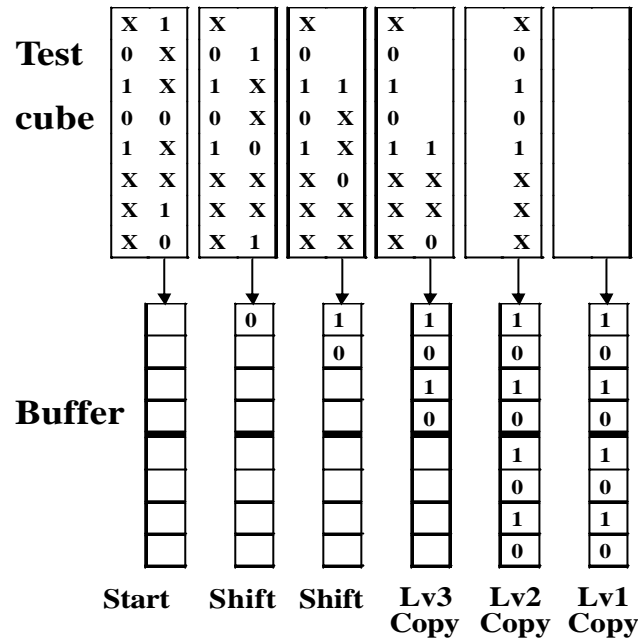


Figure 5.2 An example to demonstrate shift and copy operations

Encoding process for MDC is very simple. Starting from layer one, at each layer, the test data is checked if the *Copy* operation is applicable. If yes, a control bit “1” is added to the encoded data; otherwise, a “0” is added to the data to indicate that there is no compatible data and current layer advances to the next layer. The above action is repeated until the last layer is reached while no *Copy* is applicable. At this moment, the *Shift* operation is entered and the original raw test data of  $k$  bits are appended to the encoded data, where  $k$  is the number of bits of the last layer. After that, the current layer may change; and the above encoding process then repeats iteratively. The example in Figure 5.2 is used again to demonstrate this encoding process:

In the beginning, we check if *Copy* operation can be applied to current layer,  $Lv_1$ . Since for this case, the buffer does not have any data, the *Copy* operation can not be applied to the first layer. Thus we enter  $Lv_2$  and then  $Lv_3$  and *Copy* operation still cannot be applied for them. Therefore, three 0s are added to the encoded data. Following the two times of the *Shift* operation ( $k=2$ ), the decoder loads the first two bits, 01, into the buffer. After that, three times of the *Copy* operation are applied and the test cube is

completely loaded with all don't care bits filled. The final encoded data is **00001111**, where the first three control bits "000" mean no *Copy* applicable, the forth bit "0" and the fifth bit "1" are raw data and the last three control bits are for *Copy* operation. Compared to the original 16-bit test data, a test data reduction of 50% is obtained.

The decoding flow of our decoder is shown in Figure 5.3, where data is loaded from *in*. For this decoding flow, three notes are given: (1) *Lv* means "layer", (2) After a *Copy* or *Shift* operation, the current layer may change to other layer; that is why the task "get current *Lv*" is included in the figure. For hardware implementation, only a counter is needed to trace the number of bits shifted into the buffer for the current slice and another counter to record the current layer, and, (3) *k* is the number of DFFs of the last layer, which is two for the example in Figure 5.2.

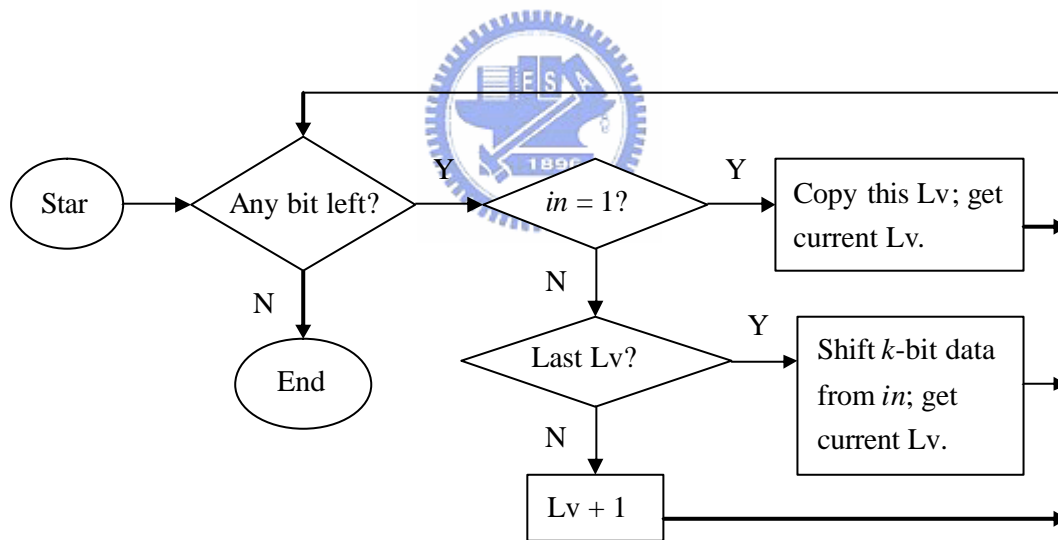


Figure 5.3 Decoding flow of the decoder

Another example is shown in the following to explain the encoding and decoding process more clearly. In Figure 5.4, a new slice having 8 bits, 1X010100, is to be shifted into the decoder's buffer which already has previous data 10101010. The checking operations according to the MDC encoding procedure for each bit are shown in the figure. In the figure, the encoded data after each checking operation are also

shown where bits in *italic* are control bits and others are data bits. For the first bit, 0, it is first checked if *Copy* for  $L_{V1}$  is applicable. Since the answer is no, the layer advances to  $L_{V2}$  and then  $L_{V3}$ . Finally, only *Shift* is applied, therefore, control bits **000** are added. Once the *Shift* operation is done, two data bits, 00, are shifted into the buffer as shown in Figure 5.4(b). Then, the decoder checks for  $L_{V3}$  and finds  $L_{V3}$  copy is not applicable, therefore, still applies *Shift* operation (Figure 5.4(c)) to shift two data bits, 10. For the 5<sup>th</sup> bit “0” in Figure 5.4(e), the decoder first checks  $L_{V2}$  then  $L_{V3}$  and applies  $L_{V3}$  *Copy* operation since the following two bits “10” are compatible with the last-shifted two bits, 10, of the buffer. Finally, for the next two last bits, the decoder checks  $L_{V3}$  and applies *Shift* operations. The final encoded data is **00000010010X1** which consists of three *Shift* and one  $L_{V3}$  *Copy* operations.

It is noted that when one slice is ready, the decoder shifts the slice into scan chains by asserting the clock of scan flip-flops. Therefore, unlike [36-37, 39, 45, 92-94], our decoding process does not load scan chains at every test cycle. Also unlike [53, 55-56, 82], synchronization overhead does not exist in our approach since the ATE has not to be stopped during the entire decoding process.

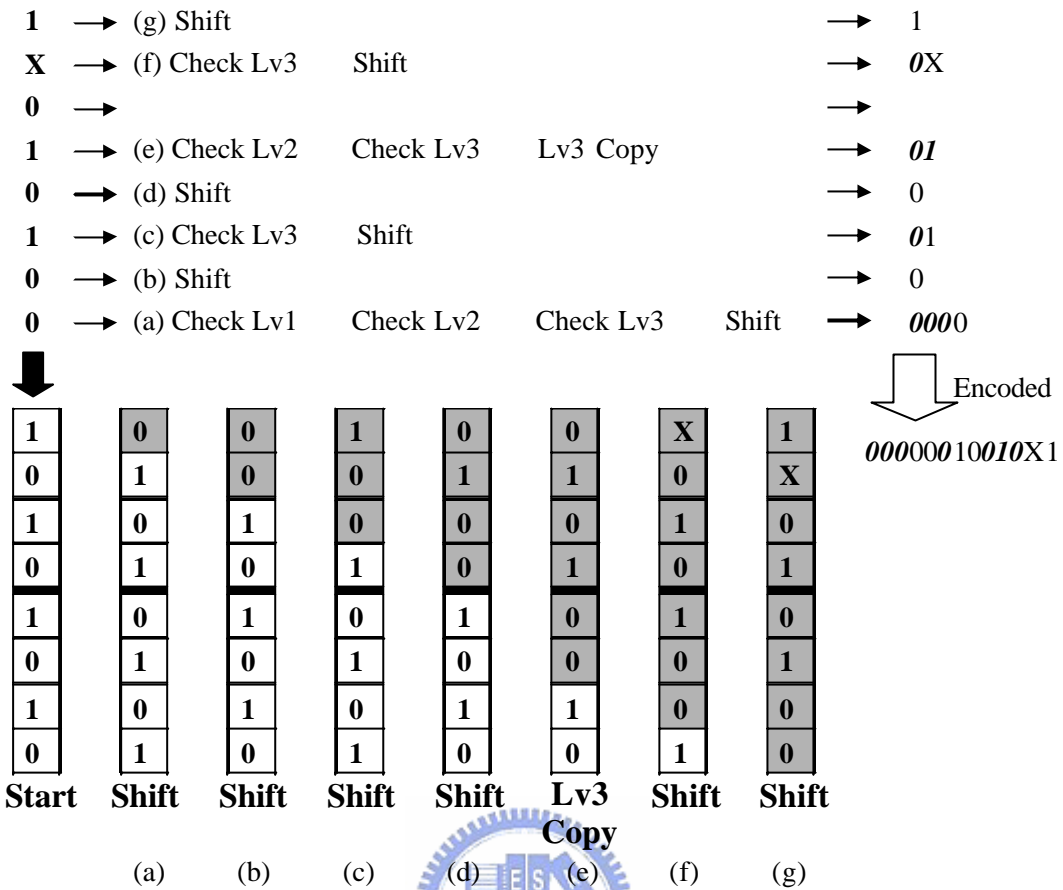


Figure 5.4 Another example to show how to encode slices using Shift and Copy operations

## 5.3 Efficiency Analysis for MDC

### 5.3.1 Compression Analysis

The compression of MDC relies on the *Copy* operation to quickly load data to the buffer. The lower layer the *Copy* operation can be applied, the larger gain can be obtained. However, increasing the group size at layer does not necessarily obtain a high compression rate since the probability that a larger size group is compatible with another group decreases. Therefore, the relation between the size of groups, the number of layers and the achievable compression should be analyzed.

Before that, several terms are defined:

- $N$  : the total number of bits in test sets
- $L$ : the total number of layers
- $lv_i$ : the  $i$ th layer, where  $1 \leq i \leq L$
- $gs_i$ : the group size (number of bits) at layer  $lv_i$
- $n_i$ : the number of groups that cannot be applied *Copy* at layer  $lv_i$
- $p$ : the specified bit density of test set

The total data volume,  $DV$ , can be derived as follows. For the first layer, it has  $\frac{N}{gs_1}$  groups and needs  $\frac{N}{gs_1}$  bits to present whether the *Copy* operation is applicable or not for each group. Given  $n_1$  groups are not copied at layer one, as for the second layer, it thus has  $n_1 \times \frac{gs_1}{gs_2}$  groups and therefore needs  $n_1 \times \frac{gs_1}{gs_2}$  bits. Finally,  $DV$  is:

$$DV = \frac{N}{gs_1} + n_1 \times \frac{gs_1}{gs_2} + n_2 \times \frac{gs_2}{gs_3} + \dots + n_L \times gs_L$$

In the formula, only  $n_i$  is to be determined. To determine  $n_i$ , we start from analyzing the probability that a group is compatible with its succeeding group, i.e., the probability that its succeeding group can be applied *Copy* operation. Given a specified bit density  $p$ , the probability that two bits are incompatible is when the first bit is a 1 and the second bit is a 0 and vice versa. Therefore, it is  $\frac{p}{2} \times \frac{p}{2} + \frac{p}{2} \times \frac{p}{2} = \frac{p^2}{2}$ . So the probability that two bits are compatible is  $(1 - \frac{p^2}{2})$ . The probability that a group having  $gs_i$  bits can be applied *Copy* operation is  $(1 - \frac{p^2}{2})^{gs_i}$ . Thus  $n_i$  is given by:

$$n_i = [1 - (1 - \frac{p^2}{2})^{gs_i}] \times \frac{N}{gs_i}, i = 1$$

$$n_i = [1 - (1 - \frac{p^2}{2})^{gs_i}] \times (\frac{gs_{i-1}}{gs_i} \times n_{i-1}), i \neq 1$$

A larger  $n_i$  results in large data volume  $DV$ .  $n_i$  depends mostly on  $gs_i$  but their relationship is implicit. An experiment was then run to find their relationship: 500 random test patterns were generated for a scan design of 1024 scan cells. The compression is defined as:

$$Compression = \frac{N - DV}{N}$$

For a two-layer buffer and  $gs_2 = 4$ , Figure 5.5 shows the compression result on different  $gs_1$  and  $p$ . It can be seen that the finally obtained compression is a strong function of the bit density probability  $p$  but a weak function of  $gs_1$ . Also, for the number of times of applying *Copy* to  $lv_1$  and  $lv_2$ , it was found that for a larger  $gs_1$ , it had fewer times of *Copy* at  $lv_1$  (larger  $n_i$ ), but more times of *Copy* at  $lv_2$ , therefore resulting in similar  $DV$ .

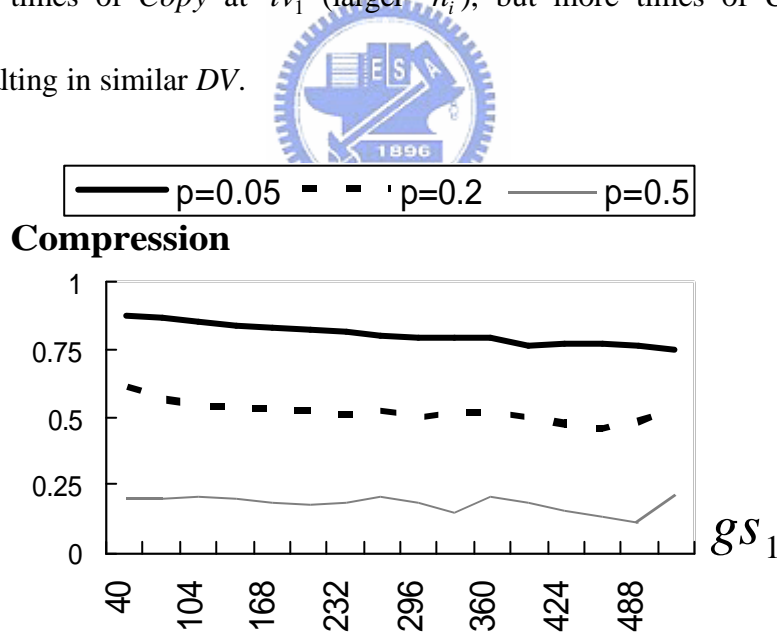


Figure 5.5 Compression under two parameters:  $gs_1$  and  $p$

### 5.3.2 Scan-In Power Reduction Analysis

In this section, it is to investigate the test power reduction of this scheme.

In order to obtain large data reduction, we like to apply more *Copy* operations at each layer. Especially, if we apply more *Copy* operations at layer one, we not only achieve data volume compression but also obtain scan-in power reduction because no transition between the coping slice and the copied slide is involved. However, as shown in Section 5.3.1, to increase the probability of applying the *Copy* operation at layer one, the group size can not be too large. Again an experiment was done to investigate the relationship between the group size and the achieved power reduction using the above randomly-generated test set. The results are shown in Figure 5.6, where 6(a) is the plot of **WTC** (weighted transition counts: the total number of transitions during scan test) [81] with  $gs_1$  in terms of  $p$  and 6(b) is the plot of peak transitions with  $gs_1$  in terms of  $p$ . From Figure 5.6(a), it can be seen that a larger number of scan chains reduces the total energy (WTC). However, from Figure 5.6(b), it is seen that for a larger group size of the first layer, for which less *Copy* can be applied, a higher peak power is resulted. Hence, to simultaneously reduce *DV* and the peak power, a moderate group size for the first layer should be chosen.

Also, from Figure 5.6(b), we can see that, for the MDC scheme, it can reduce test peak power for small  $p$  (usually smaller than 5% for real designs). This is a good advantage over those of the conventional *ATPG-dependent* LFSR-based [45, 94], Xor-based decompression network [37, 92] methods and [39] etc, where don't care bits of test patterns are essentially filled with random-like fillings, resulting in large test power [63, 96-97]. For this type of filling, it was reported that average transitions are usually about  $N/2$  where  $N$  is the number of scan cells [96]. As for the MDC scheme, for instance, for a  $p=0.05$  with 40 scan chains, the amount of peak power is about 200, which is only about 30% of that of the randomly generated test set.



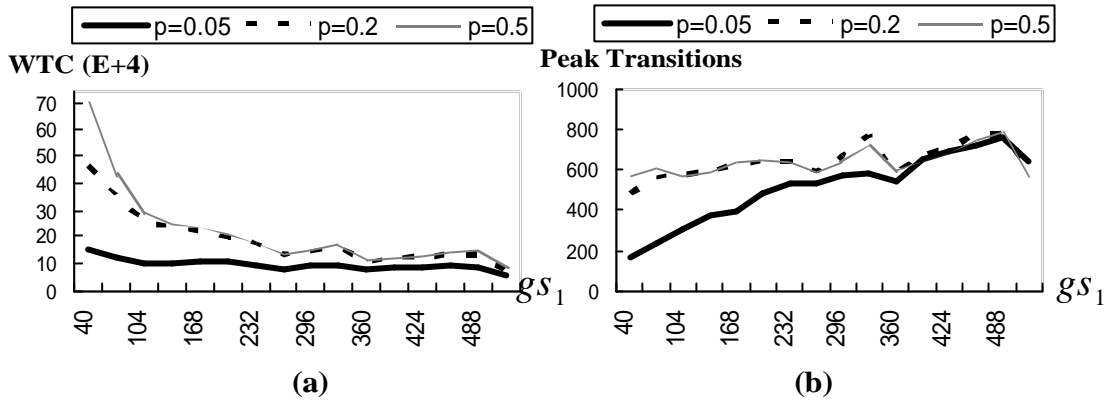


Figure 5.6 Plots of (a) weighted transition counts (WTC), and (b) peak transitions, with respect to  $gs_1$  in terms of  $p$

## 5.4 Pattern Generator with MDC

The MDC strategy can be incorporated with an ATPG to generate test patterns which are dedicated to be compressed with the MDC technique. The test patterns so obtained will provide high compression as well as test power reduction efficiency. Figure 5.7 shows such an ATPG: MDCGEN. The ATPG starts with generating a test cube and uses a test cube list (TCL) to store all generated test cubes. A test cube is generated targeting at one of remaining undetected faults. The generated cube is checked if it is compatible with cubes in the TCL. All compatible cubes will be compared and the best one is selected. That is, the numbers of *Copy* operations reduced for each compatible cube in the TCL before and after merging it with the generated test cube is recorded. Then the cubes with the least reduced number of *Copy* operations are chosen. These cubes are further checked to be selected and the cube that, after being merged with the generated cube, has the minimum resulting switching activity is selected. The selected best cube is then merged with the generated cube. After that, fault simulation is conducted and the faults detected by this merged cube are dropped from the fault list. If the generated test cube is not compatible with any cube in the TCL, it is added to the TCL. This flow

continues until all faults are tried.

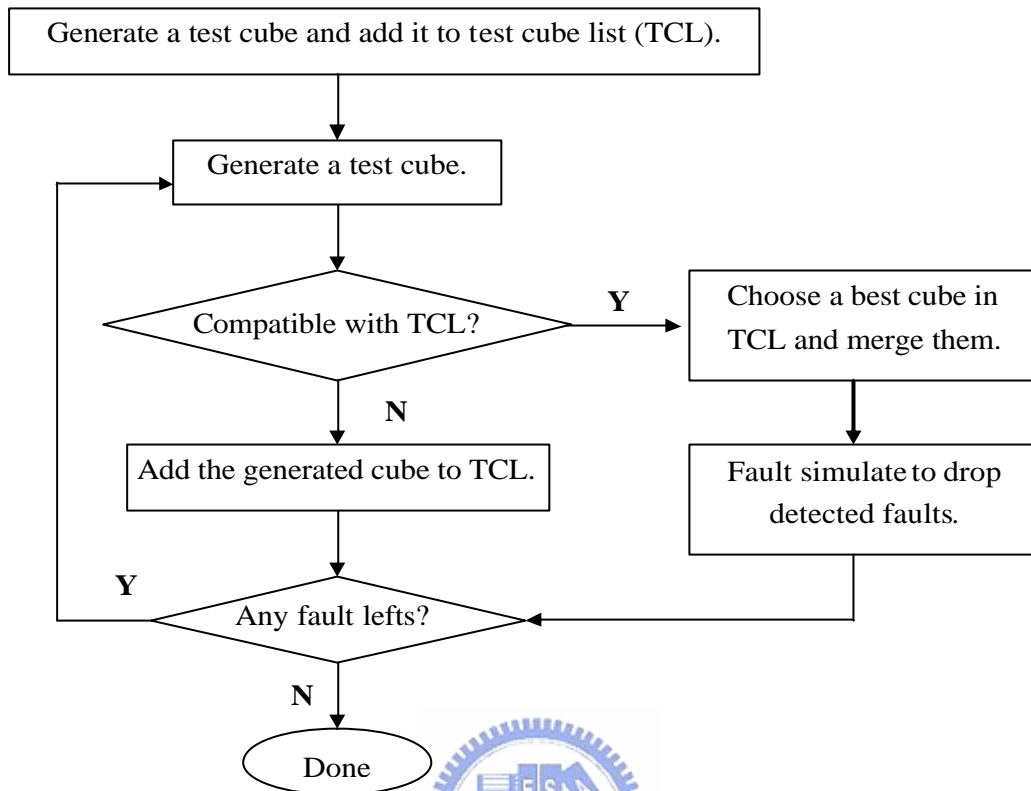


Figure 5.7 The proposed flow of an ATPG, MDCGEN, incorporated with MDC strategy

In implementing the above flow, to further reduce test data volume and test power, a special pattern generation strategy can be employed. That is: after the architecture of decoding buffer is decided, the pattern generation phase employs a random pattern generation step. For this step, a set of random bits is generated for the first slice but for all following slices the patterns are copied from the previous slices respectively. In this way, each random pattern has very high compressibility and low test power. For a scan design with  $N$  scan cells and a two-layer decoding buffer ( $a$ - $b$  architecture), the maximum encoded data volume for each random pattern is only  $a + \left\lceil \frac{N}{a} \right\rceil - 1$  bits and

the peak power for each random pattern has no more than  $a$  transitions and the average

power is even lower. This is a great saving when compared with a traditionally generated pattern, for which the average transition is  $\frac{N}{2}$  and the peak power is even higher [96].

As compared to other *ATPG-dependent* methods [37, 39, 45, 92-94], MDCGEN has the following advantages which make it efficient in generating good test patterns for compression and power reduction. First, it is not necessary to solve a set of linear equations to find the compressibility between the decoder and test cubes generated from ATPG. It only involves a procedure of compatibility checking and switching activity counting and this has a small computation overhead. Second, when some test cubes can not be compressed by decoders/decompressors, ATPGs of conventional approaches have to iteratively try or change configuration of decompressors. For MDC, it is easy for the decoder to apply any test cube and not necessary to change any configuration even with fully specified patterns. Third, the conventional *ATPG-dependent* methods fill unspecified bits only targeting test compression, and the fillings are basically of the random-like filling strategy, resulting in large test power [63, 96-97]. However, for MDC, its intrinsic nature produces low power patterns since it adopts a low-power filling mechanism for multiple scan chains. This approach thus has the advantage of simultaneously targeting test data and test power reduction for multiple-scan-chain designs without CUT modification.

## 5.5 Experimental Results

### 5.5.1 Compression Comparison

To evaluate the efficiency of the proposed scheme, we have implemented the proposed MDC technique both in the *ATPG-independent* way (denoted as MDC), i.e., Mintest test sets were obtained first and then the MDC scheme was applied to compress

the tests, and in the *ATPG-dependent* way (denoted as MDCGEN), i.e. the MDC was considered at the same time during the test generation process. The implementation was in C++ and applied to several benchmark circuits. For the MDCGEN, test sets were generated with the same fault coverage as that of a commercial tool SYNTEST [98]. The compression results are shown in Table 5.1. In the table, we present SFFs (number of scan cells), PTs (number of test patterns), buffer organization, Com (Compression), DV (compressed data volume) and Gate Counts (equivalent gate counts for hardware overhead) for each circuit. We see that, in general, the average DV of MDC and MDCGEN are quite comparable. However, as it can be seen later, MDCGEN is more efficient also on test power reduction since in an *ATPG-independent* way, test set tends to have higher specified bit density, consequently less flexibility for power reduction.

#### 5.5.1.1 ATPG-independent MDC for Mintest test sets

For the MDC, we used the Mintest test sets and compressed them using the MDC strategy for the benchmark circuits and compared the results with some previously published results as shown in Table 5.2. In the table, the compression percentages of each published method and the MDC scheme is listed and the bold numbers are the best results among all the methods. It can be seen that our MDC obtained the best compression results in four out of six circuits and in the average of the six circuits.

Table 5.1 Compression results for MDC (Mintest test set) and MDCGEN

Ckts	SFFs	MDC for Mintest				MDCGEN			
		PTs	Buffer	Com	DV	PTs	Buffer	Com	DV
s5378	214	111	35-5	56	10416	183	20-5	80	7807
s9234	247	159	20-5	55	17794	269	28-4	75	17776
s13207	700	236	50-10-5	86	22384	192	32-4	88	15596
s15850	611	126	64-16-4	73	20912	224	28-4	86	19599
s35932	1763	16	16-4	76	6736	47	40-5	94	5095
s38417	1664	99	36-9-3	62	62914	270	25-5	86	63590
s38584	1464	136	64-16-4	71	57428	260	30-5	89	41809
Avg.				68	28369			85	24467

Table 5.2 Compression comparison between the MDC scheme and other compression methods on Mintest test sets

Ckts	Golomb [53]	FDR [60]	EFDR [80]	VIHC [55]	ARL [81]	SC [82]	9 Code [83]	Mixed RL [61]	MDC
s5378	37.1	48.0	51.9	51.8	50.8	<b>55.1</b>	51.6	53.8	<b>56.2</b>
s9234	45.3	43.6	45.6	47.3	45.0	54.2	50.9	<b>55.3</b>	54.7
s13207	79.9	81.3	81.9	<b>83.5</b>	80.2	77.0	82.3	82.5	<b>86.5</b>
s15850	62.8	66.2	68.0	<b>67.9</b>	65.8	66.0	66.4	67.3	<b>72.8</b>
s35932	N/A	19.4	<b>80.3</b>	56.1	N/A	65.7	N/A	N/A	76.1
s38417	28.4	43.3	60.6	53.4	60.6	59.0	60.6	<b>64.2</b>	61.8
s38584	57.2	60.9	62.9	62.3	61.1	64.1	<b>65.5</b>	62.4	<b>71.2</b>
Avg*	51.8	57.2	61.8	57.0	60.6	62.6	62.9	<b>64.3</b>	<b>67.2</b>

\*s35932 is not included

### 5.5.1.2 ATPG-independent MDC

For the MDCGEN, the compressed results are compared with those of some published *ATPG-dependent* methods as shown in Table 5.3. In the table, Switch Configuration [39] has the best compression results. MDCGEN has slightly larger final compressed data volume than those of Unified Network but better results than those of SCC except for circuit s38584. However, it is to be mentioned that MDCGEN targets simultaneous test data and power reduction. Even so, the test volume obtained by MDCGEN is still comparable with those of SCC and Unified Network.

Table 5.3 Data volume comparison between MDCGEN and other ATPG-dependent methods

Ckts	SCC [37]	Unified Network [92]	EDT [45]	Switch Configuration [39]	MDCGEN
s5378	NA	NA	5676	NA	7807
s9234	NA	NA	9534	NA	17776
s13207	25344	19608	10585	<b>4980</b>	15596
s15850	22784	12024	9805	<b>7720</b>	19599
s35932	7128	2583	<b>NA</b>	<b>1260</b>	5095
s38417	89856	54207	31458	<b>19376</b>	63590
s38584	38796	28120	18568	<b>12888</b>	41809

### 5.5.2 Scan-In Power Comparison

In this section, it is to compare test power for the above test sets. In the text which follows, the power estimation for average power is the average number of transitions of all patterns; peak power is the maximum transition among all patterns; and total power is the total transitions during scan shift for all patterns. Total power or test energy is the same as WTC defined in [59-61, 81]. However, average power and peak power use “transition” rather than use WTC. More formally,  $n$  and  $l$  are the number of test patterns and scan chain length, respectively, and  $P_i=(b_{i1} b_{i2} \dots b_{im})$  is the  $i$ th test pattern ( $1 \leq i \leq n$ ), where  $b_{ij}$  is the  $j$ th bit, as defined in [61]. The number of transitions,  $T_i$ , for  $P_i$  is:

$$T_i = \sum_{j=1}^{m-1} (b_{ij} \oplus b_{i(j+1)})$$

Weighted transition counts,  $WTC_i$ , for  $P_i$  is:

$$WTC_i = \sum_{j=1}^{m-1} (m - j)(b_{ij} \oplus b_{i(j+1)})$$

Then,

$$\left\{ \begin{array}{l} \text{Average Power} = \frac{\sum_1^n T_i}{n} \\ \text{Peak Power} = \text{MAX}_{1 \leq i \leq n} (T_i) \\ \text{Total Power (Energy)} = \sum_1^n \text{WTC}_i \end{array} \right.$$

### 5.5.2.1 ATPG-independent MDC for Mintest test sets

On test power, we first present results of the MDC for Mintest test sets. Table 5.4 shows the results on average and peak power (average and peak transitions) during pattern scanning in and the total energy consumption, WTC, during test of the MDC strategy with those of three filling strategies, namely, fill 1/0 (used in [53, 55, 61, 80-81]), SC filling [82] and MTF. All the values are normalized with respect to the maximum values, which are of the SC filling strategy, among those methods. For the experimental data, the same number of scan chains and test patterns are used for each method. In the table, MTF represents the achievable lowest power. It is seen that SC has the largest average/peak power and energy. For MDC, it is slightly higher than that of the “fill 0/1” strategy on the scan power and is higher than MTF, which is the lower bound. Overall, MDC has best compression with only little increased test power compared with “fill 0/1”.

Table 5.4 Normalized (a) average, (b) peak and (c) test energy comparisons between different compression methods

Ckts	Normalized Average Power				Normalized Peak Power				Normalized Test Energy			
	Fill 0/1	SC	MTF	MDC	Fill 0/1	SC	MTF	MDC	Fill 0/1	SC	MTF	MDC
s5378	0.39	1	0.24	0.49	0.75	1	0.61	0.78	0.42	1	0.28	0.60
s9234	0.42	1	0.28	0.57	0.73	1	0.64	0.74	0.45	1	0.31	0.65
s13207	0.30	1	0.19	0.60	0.73	1	0.60	0.85	0.22	1	0.14	0.53
s15850	0.31	1	0.21	0.59	0.65	1	0.53	0.73	0.28	1	0.20	0.60
s35932	0.80	1	0.67	0.74	0.75	1	0.67	0.67	0.82	1	0.69	0.75
s38417	0.63	1	0.35	0.65	0.80	1	0.72	0.81	0.56	1	0.36	0.67
s38584	0.46	1	0.30	0.87	0.71	1	0.64	0.74	0.42	1	0.28	0.81
AVG.	0.48	1	0.32	0.64	0.73	1	0.63	0.76	0.45	1	0.32	0.66

### 5.5.2.2 ATPG-independent MDC

Table 5.5 shows the similar plots for the *ATPG-dependent* MDCGEN on scan-in average/peak power and the total test energy with respect to other *ATPG-dependent* methods [37, 39, 45, 92-94]. For those methods, the number of random patterns and the number of scan chains were assumed to be the same with ours and, as mentioned in Section 5.3.2, the “random-filling” strategy was used in generating test patterns. The table shows that MDCGEN is very efficient in reducing power. In average, it reduces average power, peak power and test energy to only 14%, 35% and 17% of those of the random fill. In Figure 5.8, it is also shown the detailed simulated scan-in transitions for each pattern for circuits s15850 and s35932 for MDCGEN and random patterns respectively. From that figure, it is seen that MDCGEN suppresses the scan-in power for all generated patterns for circuits.



Table 5.5 Normalized (a) average, (b) peak and (c) test energy comparisons between MDCGEN and random patterns [37, 39, 45, 92-94]

Ckts	Normalized Average Power		Normalized Peak Power		Normalized Test Energy	
	MDCGEN	Others	MDCGEN	Others	MDCGEN	Others
s5378	0.16	1	0.35	1	0.28	1
s9234	0.31	1	0.55	1	0.36	1
s13207	0.12	1	0.53	1	0.08	1
s15850	0.12	1	0.32	1	0.17	1
s35932	0.05	1	0.15	1	0.06	1
s38417	0.13	1	0.27	1	0.16	1
s38584	0.10	1	0.28	1	0.10	1
AVG.	0.14	1	0.35	1	0.17	1

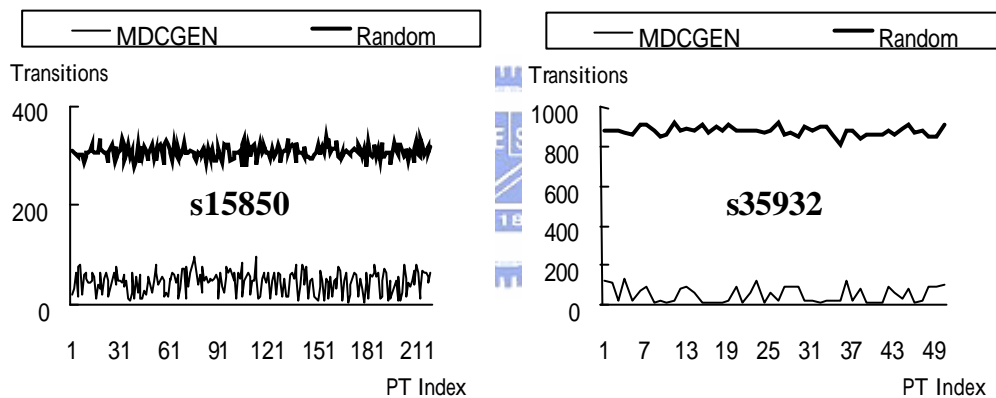


Figure 5.8 Power profiles for each pattern of two circuits: s15850 and s35932 for MDCGEN and random-filling patterns

### 5.5.3 Comparison of MDCGEN with another Low Power Test Compression Method

We also compare MDCGEN with a published work on the low power test compression method for multiple-scan-chain design [62]. The results are shown in Table 5.6, where the number of test patterns, data volume (DV) and average transitions are shown. In the table, for the data of FCN, the best compression and the lowest

average transitions of [62] are listed. It is seen that MDCGEN outperforms on the test data compression by 30% and the power reduction by 56% in average over those of FCN.

Table 5.6 Comparison of data compression and test power for MDCGEN with the work FCN [62]

Ckts	FCN [62]			MDCGEN				
	Pts	DV	Average Trans.	Pts	DV	DV Improved	Average Trans.	Power Improved
s13207	251	22092	187	192	15596	29%	44	30%
		23888	63					
s15850	148	20824	220	224	19599	6%	39	71%
		24024	137					
s35932	35	21492	638	47	5095	76%	48	90%
		25496	468					
s38417	183	77630	407	270	63590	18%	114	65%
		88152	326					
s38584	288	56296	95	260	41809	26%	72	24%
		65016	177					
Average						31%		56%

#### 5.5.4 MDCGEN for Large-scale Circuits

MDCGEN was applied to some larger circuits of higher complexity and the results are shown in Table 5.7. The SFFs, gate numbers and “Min. Pts” (the minimal number of patterns obtained from a commercial ATPG [98] with the largest compaction) for each circuit are listed. For MDCGEN, Buffer is the buffer structure used, Pts are the number of patterns, DV is the data volume, DVR is the data volume reduction. Average power, Peak power and Energy are the scan-in powers and energy for the Pts with respect to “Min. Pts” respectively. Here *DVR* is defined as:

$$DVR = \frac{Min. Pts \times SFFs}{DV}$$

Significant data volume reduction was obtained for each circuit in the table. For example, for circuit leon2, 83 times of data volume reduction was obtained. For the test power and energy reduction, MDC exhibited very well except for Peak Power of b19\_1 and Ckt1 for which we put more emphasis on DVR of MDCGEN instead of power reduction.

For the above circuits, Synopsys Design Compiler was used to evaluate the area overhead (Area % in Table 5.7) for the added decoders with decoding buffer for the MDC scheme and it was found that they all were about 1% of the original circuits.

Table 5.7 Comparison of data compression and test power for MDCGEN on large-scale circuits

Ckts	Min. Pts	SFFs	Gates	MDCGEN							
				Buffer	Pts	DV	DVR	Avg. Power	Peak Power	Test Energy	Area %
b19_1	432	6689	190K	100-20-4	636	422K	6.9 X	22%	86%	33%	1.1%
Ckt1	737	6763	235K	100-20-4	852	469K	10.6 X	14%	93%	18%	0.9%
leon1[99]	556	10460	136K	100-20-4	765	264K	22 X	5%	20%	9%	1.5%
leon2[99]	2826	49510	694K	250-50-5	3146	1.75M	83 X	6%	9%	4%	0.7%
FFT	759	75757	883K	600-60-6	835	873K	66X	10%	18%	7%	1.4%

## 5.6 Summary

In this chapter, we have proposed and demonstrated a new simple yet efficient test encoding scheme: MDC, for test compression and power reduction for multiple-scan-chain designs. The scheme adopts a simple buffer which can be flexibly organized in a multilayer structure in conjunction with a simple coding strategy to fill unspecified bits of test patterns to achieve power reduction. A layer copying mechanism, which reduces transitions between neighboring slices, makes the scheme inherently power efficient. The scheme can be incorporated into an ATPG program to generate test patterns both volume and power efficient. The scheme had been applied to some

benchmark and large size circuits to show that it achieved not only high compression rate for test patterns but also low test power. In addition, only one scan-in pin is required to support large number of scan chains. This facilitates the use of a low-cost ATE for this scheme. Also, the scheme is very flexible to be used: either in an *ATPG-independent* way or in an *ATPG-dependent* way.

Finally, Table 5.8 compiles performance aspects on the MDC scheme with other published approaches and techniques. We could conclude that the scheme is a good scheme to reduce the shifting-in power for scan test for multiple-scan-chain designs.

Table 5.8 Overall comparisons between the proposed scheme and other schemes

	<i>ATPG-Independent Methods</i>			<i>ATPG-Dependent Methods</i>	
	Fill 0/1 [53, 55, 60-61, 81, 80]	SC [82]	MDC	[37, 39, 45, 92-94]	MDCGEN
Traditional ATPG Useable?	<b>Yes</b>			No	
Low Power Test for Multiple Scan Chains?	No		<b>Yes</b>	No	<b>Yes</b>
Synchronization Problem?	Mostly Yes		<b>No</b>	<b>No</b>	
Encode/Decode Complexity?	Middle		<b>Low</b>	High	<b>Low</b>
Compression Efficiency?	Middle			<b>Highest</b>	High
Scan-In Power?	Middle	High	Middle	High	<b>Low</b>
Fault Coverage Lost?	<b>No</b>			Some	<b>No</b>
Number of Scan-In Ports?	one			two or more	one

## Chapter 6 Conclusions

We have addressed, in this thesis, two important testing issues, namely, test data volume and test power. Specifically, we have discussed the problem of the reliability reduction of the CUT and yield loss due to large test power dissipation during test. Moreover, the problem of high test cost induced from large volume of test data has been explained. In this chapter, we give concluding remarks on the study and discuss directions for further research.

### 6.1 A Scan Matrix Design for Low Power Scan-Based Test

On this topic, we pointed out that the power and energy consumed for the CUT in the test mode are much higher than in the normal operation. This may cause heating and damaging to the device. The large current surge also may cause ground bounces to disturb circuit operation. Therefore, we proposed a new Scan Matrix (SM) architecture for the scan-based design to achieve low power testing. For this scheme, scan flip-flops are connected in a matrix style with its addressing controlled by two ring generators during test for pattern scanning-in. Unlike the traditional scan, for which scan-in data need to pass through a long path and many scan flip-flops switch simultaneously, it dynamically forms low-power scan paths to reduce test energy and peak power for pattern shifting. The architecture is scalable for large designs and has minimal circuit performance penalty. Experiments on ISCAS89 benchmark circuits have shown that the scheme achieves significant power savings, for some large circuits 99% saving, both in energy consumption and peak power. Hence, this scheme is a good solution to the low power scan test.

## **6.2 Cocktail Random Access Scan for Test Data and Power Reduction**

Traditional serial scan suffers from the above mentioned testing problems due to its natural architecture. An alternative way to it is to use randomly addressable scan style, RAS. On this topic, we have proposed and demonstrated a Cocktail Scan testing scheme to save the test vector storage, the test application time, and consequently the testing power. The scheme takes two phases to generate and apply test patterns for the DUT. In the first phase, a few number of seed patterns are applied to the DUT to generate segmented random scan patterns to test the circuit. For this phase testing, the test application is simple and a large number of faults can be detected. In the second phase, deterministic patterns are generated for the remaining faults and are applied to the circuit by the RAS fashion. For this set of test patterns, a proposed process utilizing several improved strategies, namely, Test Response Abandonment, CSC and BPBTVD, is used to reduce the test vector volume and the number of bit flipping. Experimental results show that the process is very effective in reducing the number of bit flipping, which leads to an 86% reduction in test data and 10 times of speedup in test application time. Overall, experimental results also show that, for the Cocktail Scan scheme, the test data volume for this scheme increases only slowly with the size of the tested circuit, making it effective to be applied to large size circuits.

## **6.3 Adaptive Encoding Scheme Using Embedded Memory for Low-Cost and Low Power SoC Test**

On this topic, we have proposed an Adaptive Encoding scheme to encode the test data (patterns) to save the test data volume and the test application time. The scheme

handles, instead of the data themselves, the difference between two consecutive test patterns by using packets, making the test data be encoded in variable sizes to achieve better data compression. A decoder machine is proposed to make the scheme possible and to decode the encoded data. Constrained MTF, a filling strategy, is also proposed to be adopted with the scheme to simultaneously achieve test compression and test power reduction. In addition, the scheme can incorporate the techniques to generate test patterns in two phases and to reorder the test vectors to further achieve the test data reduction. Experimental results have shown that the proposed encoding scheme is very effective in reducing the volume of test data and test power, and in speeding up test application time for large-scale designs.

## 6.4 Low Power Test Compression for Multiple Scan Chain Designs



Finally, we have proposed and demonstrated a new simple yet efficient test encoding scheme: MDC, for test compression and power reduction for multiple-scan-chain designs. The scheme adopts a simple buffer which can be flexibly organized in a multilayer structure in conjunction with a simple coding strategy to fill unspecified bits of test patterns to achieve power reduction. A layer copying mechanism, which reduces transitions between neighboring slices, makes the scheme inherently power efficient. The scheme can be incorporated into an ATPG program to generate test patterns both volume and power efficient. The scheme had been applied to some benchmark and large size circuits to show that it achieved not only high compression rate for test patterns but also low test power. In addition, only one scan-in pin is required to support large number of scan chains. This facilitates the use of a low-cost ATE for this scheme. Also, the scheme is very flexible to be used: either

in an ATPG-independent way or in an ATPG-dependent way. Experiment results on benchmarks show that test sets generated by the scheme had large compression and power saving with only a small area design overhead. We also compare our scheme with many previous works and conclude that the scheme is a good scheme to reduce the shifting-in power for scan test for multiple-scan-chain designs.

## 6.5 Future Work

We have done and solved, comparatively, some of test power and test data reduction problems, however, there are still problems related with these two issues which deserved further study. In the following, we propose some of topics which could be for future research:

On low power testing:

- Reducing not only the scan-in power, but also scan-out power and capture power.
- Finding scan architecture with reasonable hardware overhead and design effort.
- Reducing the DFT impact as much as possible (such as keeping the performance of the CUT, considering the physical routing problem).
- Automating the low-power testing DFT synthesis flow.
- Combining low-power testing DFT schemes to the test compression schemes and delay test.

On test data compression:

- Increasing the test compression efficiency.
- Reducing the DFT impact and area overhead.
- Making test compression schemes scaleable to large-scale designs.
- Combining low-power testing DFT schemes to achieve low power test





compression.

- Applying the compression methods also to delay test patterns.



# Reference

- [1] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testing", in Proc. *DAC*, 462-468, 1997
- [2] H. Ando, "Testing VLSI with random access scan", *Diag. Papers Compcon 80*, IEEE pub. 80CH1491-OC, 50-52, 1980
- [3] S. Hellebrand, J. Rajski, S. Tamick, S. Venkataraman and B. Counois "Built-in test for circuits with scan based on reseeding of multiple polynomial linear feedback shift registers", *IEEE Trans. on Computers*, 1995, 44, (2). 223-233
- [4] <http://www.itrs.net/Links/2001ITRS/Home.htm>
- [5] F. Poehl et al., "Industrial Experience with Adoption of EDT for Low-Cost Test without Concessions", in Proc. *ITC*, 1211-1220, 2003
- [6] H.Vranken, T. Waayers, H. Fleury, D. Lelouvier, "Enhanced Reduced Pin-Count Test for Full Scan Design", in Proc. *ITC*, 738-747, 2001
- [7] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices", in Proc. *VTS*, pages 4-9, 1993
- [8] R. Sankaralingam, R. Oruganti, and N. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation", in Proc. *VTS*, 35-40, 2000
- [9] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, "A gated clock scheme for low power scan-based BIST", in Proc. *IOLTS*, 2001
- [10] L. Whetsel, "Adapting scan architectures for low power operation", in Proc. *ITC*, 863-872, 2000
- [11] T. Yoshida, M. Watari, "MD-SCAN method for low power scan testing", in Proc. *ATS*, 80-85, 2002
- [12] T. C. Huang, K. J. Lee, "A token scan architecture for low power testing", in Proc. *ITC*, 660-669, 2001
- [13] B. B. Bhattacharya, S. C. Seth, and S. Zhang, "Double-tree scan: a novel low-power scan-path architecture", in Proc. *ITC*, 470-479, 2003
- [14] O. Sinanoglu, I. Bayraktaroglu, A. Orailoglu, "Scan Power Reduction Through Test Data Transition Frequency Analysis", in Proc. *ITC*, 844-850, 2002
- [15] O. Sinanoglu, A. Orailoglu, "Modeling Scan Chain Modifications For Scan-in Test Power Minimization", in Proc. *ITC*, 602-611, 2003
- [16] D. Ghosh, S. Bhunia, and K. Roy, "A Low-Complexity Scan Reordering Algorithm for Low Power Test-Per-Scan BIST", *IEEE LATIN-AMERICAN TEST WORKSHOP*, 2004
- [17] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, A. Virazel, "Design of routing-constrained low power scan chains", in Proc. *DATE*, 62-67, 2004

- [18] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, "A Test Vector Inhibiting Technique for Low Energy BIST Design", in Proc. *VTS*, 407-412, 1999
- [19] S. Wang, S. K. Gupta, "LT-RTPG: A new test-per-scan BIST TPG for low heat dissipation", in Proc. *ITC*, 85-94, 1999
- [20] S. Wang, S. K. Gupta, "DS-LFSR: A New BIST TPG for Low Heat Dissipation", in Proc. *ITC*, 848-857, 1997
- [21] N. Z. Basturkmen, S.M. Reddy, I. Pomeranz, "A Low Power Pseudo-Random BIST Technique", in Proc. *IOLTS*, 140-144, 2002
- [22] S. Gerstendrfer, H. J. Wunderlich, "Minimized Power Consumption for Scan-Based BIST", in Proc. *ITC*, 77-84, 1999
- [23] S. Wang and S. Gupta, "ATPG for heat dissipation minimization during test application," *IEEE Trans. on Computer*, vol.47, no.2, 256-262, 1998
- [24] F. Corno, P. Prinetto, M. Redaengio, and M. Reorda, "A test pattern generation methodology for low power consumption," in Proc. *VTS*, 453-459, 1998
- [25] T. Yoshida and M. Watari, "A new approach for low power scan testing," in Proc. *ITC*, 480-487, 2003.
- [26] K. K. Saluja, K. Kinoshita, "On Low-Capture-Power Test Generation for Scan Testing," in Proc. *VTS*, 265-270, 2005
- [27] X. Wen et. al., "Low-Capture-Power Test Generation for Scan-Based At-Speed Testing", in Proc. *ITC*, 1019-1028, 2005
- [28] S. Remersaro et. al., "Preferred Fill: A Scalable Method to Reduce Capture Power for Scanbased Designs", in Proc. *ITC*, 2006
- [29] R. Sankaralingam, N. A. Touba, "Controlling Peak Power During Scan Testing," in Proc. *VTS*, 153-159, 2002
- [30] S. Bose, P. Agrawal, V. Agrawal, "Generation of compact delay tests by multiple path activation," in Proc. *ITC*, 714-723, Oct. 1993
- [31] J. Saxena; D.K. Pradhan, "A method to derive compact test sets for path delay faults in combinational circuits," in Proc. *ITC*, 724-733, Oct. 1993
- [32] H. K. Lee and D. S. Ha, "On the Generation of Test Patterns for Combinational Circuits", Technical Report 12-93, Department of Electrical Eng., Virginia Polytechnic Institute and State University.
- [33] Mitra, S., and K.S. Kim, "X-Compact: An Efficient Response Compaction Technique for Test Cost Reduction," In Proc. *ITC*, 311-320, 2002
- [34] N. R. Saxena and E. J. McCluskey, "Parallel signature analysis design with bounds on aliasing," *IEEE Trans. on Computer*, vol. 46, 425-438, Apr. 1997
- [35] K.J. Lee, J. J. Chen and C. H. Huang, "Using a single input to support multiple scan chains," In Proc. *ICCAD*, 74-78, 1998

- [36] I. Hamzaoglu and J. Patel, "Reducing Test Application Time for Full Scan Embedded Cores" IEEE International Symposium on Fault Tolerant Computing (FTC), 260-267, 1999.
- [37] A. Orailoglu, I. Bayraktaroglu, "Test Volume and Application Time Reduction through Scan Chain Concealment", In Proc. *DAC*, 151-155, 2001.
- [38] S. Samaranayake et. al., "A Reconfigurable Shared Scan-in Architecture," in Proc. *VTS*, 9-14, 2003.
- [39] H. Tang, S. M. Reddy and I. Pomeranz, "On Reducing Test Data Volume and Test Application Time for Multiple Scan Chain Designs", in Proc. *ITC*, 1079-1088, 2003.
- [40] G. Mrugalski, J. Rajski, and J. Tyszer, "Ring Generators—New Devices for Embedded Test Applications," *IEEE Trans. on CAD*, vol. 23, no. 9, 1306-1320. 2004
- [41] B. Koenemann, "LFSR-Coded Test Patterns for Scan Designs," in Proc. European Test Conf.(*ETC*), 237-242, 1991
- [42] C.V. Krishna, A. Jas, and N.A. Touba, "Reducing Test Data Volume Using LFSR Reseeding with Seed Compression," in Proc. *ITC*, pp 321-330, 2002
- [43] B. Koenemann et al., "A SmartBIST Variant with Guaranteed Encoding," in Proc. *ATS*, 325-330, 2001
- [44] C.V. Krishna, A. Jas, and N.A. Touba, "Test Vector Encoding Using Partial LFSR Reseeding," in Proc. *ITC*, 885-893, 2001
- [45] J. Rajski et al., "Embedded Deterministic Test," *IEEE Trans. on CAD*, vol. 23, no. 5, 776-792, 2004
- [46] N. A. Touba, "Survey of Test Vector Compression Techniques," IEEE Design & Test of Computers, 294-303, 2006
- [47] D. Baik, K. K. Saluja, S. Kajihara, "Random access scan: a solution to test power, test data volume and test time", International Conference on VLSI Design, 883-888, 2004
- [48] O. Sinanoglu, A. Orailoglu, "Fast and Energy-Frugal Deterministic Test Through Test Vector Correlation Exploitation", In Proc. *DFT*, 325-333, 2002
- [49] S. Reda and A. Orailoglu, "Reducing Test Application Time through Test Data Mutation Encoding", In Proc. *DATE*, 387-393, 2002
- [50] F. Karimi, Z. Navabi, W. Meleis, and F. Lombardi, "Using Data Compression in Automatic Test Equipment for System-on-Chip Testing", *IEEE Trans. on Instrumentation and Measurement*, vol. 53, no. 2, 308-317, 2004
- [51] A. Jas and N. A. Touba, "Using an Embedded Processor for Efficient Deterministic Testing of Systems-on-a-Chip", In Proc. *ICCD*, 418-423, 1999
- [52] K. J. Balakrishnan and N.A. Touba, "Relating Entropy Theory to Test Data

- Compression”, In Proc. *ETS*, 94-99, 2004
- [53] A. Chandra and K. Chakrabarty, “System-on-a-chip test data compression and decompression architectures based on Golomb codes”, *IEEE Transactions on CAD*, vol. 20, 355–368, March 2001.
- [54] A. Jas, J. G. Dastidar, and N. A. Touba, “Scan Vector Compression/Decompression Using Statistical Coding,” In Proc. *VTS*, 114-120, 1999.
- [55] P. T. Gonciari, B. M. Al-Hashimi, N. Nicolici, “Variable-Length Input Huffman Coding for System-on-a-chip Test”, *IEEE Trans. on CAD*, vol. 22, no 6, 783-796, 2003.
- [56] A. Chandra, and K. Chakrabarty, “Frequency-Directed Run-Length Codes with Application to System-on-a-chip Test Data Compression”, In Proc. *VTS*, 42-47, 2001
- [57] P. T. Gonciari, B. Al-Hashimi and N. Nicolici, “Reducing Synchronization Overhead in Test Data Compression Environments”, IEEE European Test Workshop (ETW), 2002.
- [58] S.M. Reddy et al, “On Test Data Volume Reduction for Multiple Scan Chain Designs,” in Proc. *VTS*, 103-108, 2002
- [59] A. Chandra and K. Chakrabarty, “Combining Low-Power Scan Testing and Test Data Compression for System-on-a-Chip”, In Proc. *DAC*, 166-169, 2001
- [60] A. Chandra and K. Chakrabarty, “A unified approach to reduce SOC test data volume, scan power, and testing time”, *IEEE Transactions on CAD*, vol. 20, 355-368, 2003
- [61] M. Nourani and M. Tehranipour, “RL-Huffman Encoding for Test Compression and Power Reduction in Scan Applications”, *IEEE Transactions on TODAES*, vol. 10, 91-115 2005
- [62] Y. Shi, N. Togawa, S. Kimura, M. Yanagisawa and T. Ohtsuki, “Low Power Test Compression Technique for Designs with Multiple Scan Chain”, in Proc. *ATS*, 386-389, 2005
- [63] J. Lee, N. A. Touba, “Low Power Test Data Compression Based on LFSR Reseeding”, in *ICCD*, 180-185, 2004
- [64] L. Xu, Y. Sun, H. Chen, “Scan array solution for testing power and testing time”, in Proc. *ITC*, 652-659, 2001
- [65] K. J. Lee, J. J. Chen, “Reducing test application time and power dissipation for scan-based testing via multiple clock disabling”, in Proc. *ATS*, 338-343, 2002
- [66] A. Orailoglu, O. Sinanoglu, “A Novel Scan Architecture for Power-Efficient, Rapid Test”, in Proc. *ICCAD*, 299-303, 2002
- [67] J. Saxena, K. Butler and L. Whetsel, “An analysis of power reduction techniques

- in scan testing”, in Proc. *ITC*, 670-677, 2001
- [68] N. Nicolici, B. M. Al-Hashimi, “Multiple scan chains for power minimization during test application in sequential circuits”, *IEEE Trans. on Computers*, 51(6), 721-734, 2002
- [69] Rosinger, P., Al-Hashimi, B. and Nicolici, N., “Scan Architecture With Mutually Exclusive Scan Segment Activation for Shift- and Capture-Power Reduction”, *IEEE Trans. on CAD.*, 23(7), 1142-1153, 2004
- [70] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan and J. Rajski, “Logic BIST for large industrial designs: real issues and case studies”, in Proc. *ITC*, 358-367, 1999
- [71] W. Rao, I. Bayraktaroglu and A. Orailoglu, “Test Application Time and Volume Compression through Seed Overlapping”, in Proc. *DAC*, 732-737, 2003
- [72] A. R. Pandey, J. H. Patel, “An incremental algorithm for test generation in Illinois scan architecture based designs,” in Proc. *DATE*, 368-375, 2002
- [73] R. Dorsch and H. Wunderlich, “Reusing Scan Chains for Test Pattern Decompression”, IEEE European Test Workshop, 124- 132, 2001
- [74] A. Jas, B. Pouya, and N.A. Touba, “Test Data Compression Technique for Embedded Cores Using Virtual Scan Chains”, *IEEE Trans. on VLSI*, vol. 12, no. 7, 775-780, 2004
- [75] A. Jas, C.V. Krishna, and N.A. Touba, “Hybrid BIST Based on Weighted Pseudo-Random Testing: A New Test Resource Partitioning Scheme”, in Proc. *VTS*, 2-8, 2001
- [76] D. Das and N.A. Touba, “Reducing Test Data Volume Using External/LBIST Hybrid Test Patterns”, in Proc. *ITC*, 115-122, 2000
- [77] G. Zeng and H. Ito, “Hybrid BIST for System-on-a-Chip Using an Embedded FPGA Core”, in Proc. *VTS*, 355-360, 2004
- [78] P. F. Flores, J. C. Costa, H. C. Neto, J. C. Monteiro and J. P. Marques-Silva, “Assignment and Reordering of Incompletely Specified Pattern Sequences Targetting Minimum Power Dissipation”, in Proc. *VLSI*, 37-41, 1999
- [79] F. Karimi, Y. B. Kim, F. Lombardi and N. Park, “Compression of Partially Specified Test Vectors in an ATE Environment”, in Proc. *IMTC*, 999-1004, 2003
- [80] A. El-Maleh and R. Al-Abaji, “Extended frequency-directed run-length codes with improved application to system-on-a-chip test data compression”, in Proc. *ICECS*, 449-452, 2002
- [81] A. Chandra and K. Chakrabarty, “Reduction of SOC test data volume, scan power and testing time using alternating run-length codes”, in Proc. *DAC*, 673-678, 2002
- [82] A. Jas, J. Ghosh-Dastidar, Ng, and N. A. Touba, “An efficient test vector

- compression scheme using selective huffman coding”, *IEEE Trans. on CAD*, no. 6, 797-806, 2003
- [83] M. Tehranipour, M. Nourani and K. Chakrabarty, “Nine-Coded Compression Technique with Application to Reduced Pin-Count Testing and Flexible On-Chip Decompression”, in Proc. *DATE*, 1284-1289, 2004
- [84] Mitra, S., and K.S. Kim, “X-Compact: An Efficient Response Compaction Technique for Test Cost Reduction,” in Proc. *ITC*, 311-320, 2002
- [85] D. H. Baik and K. K. Saluja, “State-reuse Test Generation for Progressive Random Access Scan: Solution to Test Power, Application Time and Data Size”, in Proc. *ATS*, 272-277, 2005
- [86] S. P. Lin, C. L. Lee and J. E. Chen, “A Cocktail Approach On Random Access Scan Toward Low Power And High Efficiency Test,” in Proc. *ICCAD*, 94-99, 2005
- [87] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, K. H. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, J. Qian, “Embedded Deterministic Test for Low-Cost Manufacturing Test”, in Proc. *ITC*, 301-310, 2002
- [88] C. V. Krishna, N. A. Touba, “3-Stage Variable Length Continuous-Flow Scan Vector Decompression Scheme,” in Proc. *VTS*, 79-86, 2004
- [89] C. Shi and R. Kapur, “How power-aware test improves reliability and yield,” <http://www.eetimes.com>.
- [90] H. Vranken, F. Hapke, S. Rogge, D. Chindamo, E. Volkerink, “ATPG Padding and ATE Vector Repeat Per Port For Reducing Test Data Volume”, in Proc. *ITC*, 1069-1078, 2003
- [91] Z. Wang and K. Chakrabarty, “Test data compression for IP embedded cores using selective encoding of scan slices”, in Proc. *ITC*, 581-590, 2005
- [92] I. Bayraktaroglu and A. Orailoglu “Decompression hardware determination for test volume and time reduction through unified test pattern compaction and compression” , in Proc. *VTS*, 113-120, 2003
- [93] L. Li and K. Chakrabarty, “Deterministic BIST Based on a Reconfigurable Interconnection Network”, in Proc. *ITC*, 460-469, 2003
- [94] Synopsys Inc. <http://www.synopsys.com/>.
- [95] P. Girard, “Survey of Low-Power Testing of VLSI Circuits,” *IEEE Design & Test of Computers*, vol. 19, 82-92, 2002
- [96] Y. Kim, M. Yang, Y. Lee and S. Kang, “A New Low Power Test Pattern Generator using a Transition Monitoring”, in Proc. *ATS*, 230-235, 2005
- [97] P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici, “Test Data Compression: The System Integrator's Perspective”, in Proc. *DATE*, 726-731, 2003

[98] TurboScan, <http://www.syntest.com/>

[99] Leon, <http://www.gaisler.com>





## 學經歷

姓 名：林世平

性 別：男

籍 貫：臺灣省台中市

出生日期：民國六十七年二月九日

通訊住址：台中縣大里市長榮里和平街 20 號，電話：(04)24819136

學 經 歷：民國九十二年九月至民國九十六年九月

國立交通大學電子研究所博士班

民國八十九年九月至民國九十一年六月

國立交通大學電子研究所碩士班

民國八十五年九月至民國八十九年六月

國立交通大學電子工程學系

論文題目：低耗能並考慮低成本效益之系統晶片測試策略

Low Power and Low Test Data Volume Testing for Scan Design VLSI

## 著作目錄 (新法)

### (A) International Journal:

- [1] (長) S.-P. Lin, C.-L. Lee, J.-E. Chen, J.-J. Chen, K.-L. Luo, and W.-C. Wu, “A Multilayer Data Copy Test Data Compression Scheme for Reducing Shifting-in Power for Multiple Scan Design,” *accepted by IEEE Trans. VLSI Systems*, 2007

### (B) Other Journal Papers:

- [1] (長) S.-P. Lin, C.-L. Lee and J.-E. Chen, “Cocktail Random Access Scan for Test Data and Power Reduction,” *Journal of the Chinese Institute of Electrical Engineering*, Vol. 13, No. 3, pp. 293—303, 2006

### (C) International Conference:

- [1] S.-P. Lin, C.-L. Lee and J.-E. Chen, “A Cocktail Approach on Random Access Scan toward Low Power and High Efficiency Test,” in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 94--99, 2005.
- [2] S.-P. Lin, C.-L. Lee and J.-E. Chen, “A Scan Matrix Design for Low Power Scan-Based Test,” in *Proceedings of IEEE Asian Test Symposium (ATS)*, pp. 224--229, 2005.
- [3] S.-P. Lin, C.-L. Lee and J.-E. Chen, “Adaptive Encoding Scheme for Test Volume/Time Reduction in Soc Scan Testing,” in *Proceedings of IEEE Asian Test Symposium (ATS)*, pp. 324--329, 2005.
- [4] S.-P. Lin, C.-L. Lee, J.-E. Chen, J.-J. Chen, K.-L. Luo, and W.-C. Wu, “A Multilayer Data Copy Scheme for Low Cost Test with Controlled Scan-In Power for Multiple Scan Chain Designs,” *accepted by International Test Conference (ITC)*, 2006.

### (C) Local Conference:

- [1] S.-P. Lin, C.-L. Lee and J.-E. Chen, “Scan Matrix: A Low Power Scan Architecture,” in *Proceedings of The 16th VLSI Design/CAD Symposium*, Hualien, Taiwan, Aug. 2005.
- [2] S.-P. Lin, C.-L. Lee and J.-E. Chen, “Cocktail Scan for Low Power and High Efficiency Test,” in *Proceedings of The 16th VLSI Design/CAD Symposium*, Hualien, Taiwan, Aug. 2005.
- [3] S.-P. Lin, C.-L. Lee and J.-E. Chen, “Adaptive Encoding: A Test Compression Scheme for Soc Testing,” in *Proceedings of The 16th VLSI Design/CAD Symposium*, Hualien, Taiwan, Aug. 2005.