國 立 交 通 大 學

電子工程學系電子研究所

博 士 論 文

以振盪信號測試超大型積體電路之串音障礙

VLSI Crosstalk Fault Testing with Oscillation Signals

研 究 生 ：吳 明 學

指導教授 ：李 崇 仁 教授

中華民國 九十五 年 七 月

以振盪信號測試超大型積體電路之串音障礙
VLSI Crosstalk Fault Testing with Oscillation Signals

研 究 生：吳明學　　Student：Ming-Shae Wu

指導教授：李崇仁 博士　　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

A Dissertation

Submitted to Department of Electronics Engineering & Institute of

Electronics College and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electronic Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中華民國 九十五 年 七 月

以振盪信號測試超大型積體電路之串音障礙

學生：吳明學　　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

本論文就相關於震盪訊號的創新方法針對數位電路之串音障礙測試與內建自我測試技術的幾個子題加以研究探討。首先，提出一個以基於震盪訊號測試串音障礙所引發的突波效應之測試方法，藉由採用一個震盪訊號源輸入，以此使侵略性接線震盪，然後測試受害接線在兩者之間存有串音障礙下所被引起的脈波效應。我們定義了一組符號和代數來運算訊號之傳遞及偵測。此方法簡單且消除了傳統方法在產生串音障礙測試圖樣時的複雜時間考量。在此研究中之測試圖樣產生器及障礙模擬器即是根據此測試方法所發展出來的。實驗結果顯示產生器流程可對此測試方法產生有效率的測試圖樣。

其次，我們提出一個以方波測試訊號，同樣是檢測串音障礙所引發之突波效應的自我測試方法。此方法運用於深次微米超大型積體電路之周邊掃瞄測試環境，藉此來測試其內藏式電路的串音障礙。本測試方法利用提供一個測試方波，並結合亂數產生器，以此來引發並測試待測電路中因串音障礙所引起的短暫脈波。僅需對周邊掃瞄電路單元加入簡單的測試電路，即可進行本方法測試。實驗結果顯示利用本方法來測試一些大型的

樣本電路，可以很容易達到百分之九十以上的障礙涵蓋率。

最後，我們提出一個利用路徑延遲慣性原理，來測試系統電路連線之串音障礙的新測試方法。此方法無需使用時間上的量測，藉由在侵略性連線的輸入端提供一個轉換的訊號，以及在受害連線的輸入端提供一個特定寬度的脈波—CWP，並且偵測此 CWP 脈波是否可以傳輸到受害連線的輸出端，以此來測試是否有串音障礙存在。實驗結果顯示此方法在測試訊號無時間差及沒有製程參數漂移的情況下有高的判別率。此方法僅簡單，且在偵測連線上的串音障礙上有相當大的效率。

VLSI Crosstalk Fault Testing with Oscillation Signals

Student: Ming-Shae Wu Advisor: Prof. Chung Len Lee

Department of Electronics Engineering
& Institute of Electronics
National Chiao Tung Unviersity

Abstract

This dissertation studies several topics on testing and built-in self-testing of crosstalk faults of digital circuits with innovative methods which are related with oscillation signals. First, a test scheme for the induced-glitch type of the crosstalk fault by applying an oscillation signal on an aggressor line and detects induced pulses on a victim line if a crosstalk fault exists between these two lines of the circuit under test (CUT). A set of symbols and associated algebra are defined to compute the propagation and detection of signals. It is simple and eliminates the complicated timing issue during test generation for the crosstalk fault in the conventional approaches. The test generation and fault simulation based on the scheme are described. Experimental results are also presented to show that the described test generation procedure is effective in generating test patterns for this scheme.

Next, a BIST scheme based on a squarewave test signal to test also the induced-glitch

type of the crosstalk fault of embedded circuits of SoC in the boundary scan environment for deep sub-micron VLSI is proposed. The scheme applies squarewave test signals in conjunction with pseudo random patterns to induce glitches which are caused by crosstalk faults and tests them. Modifications on boundary scan cells with simple added detection circuits to facilitate this test scheme are presented. Experimental results show that the average fault coverage obtained by applying the scheme to large size benchmark circuits can easily reach 90%.

Finally, a new test scheme to detect the crosstalk fault, based on the path delay inertia, for interconnection lines in SoC is proposed. The scheme, without using timing measurement, applies a transition on the aggressor line and a pulse of specified width, CWP, to the victim line and detects the propagation of the CWP at the output of the victim line to detect the existence of crosstalk faults. Experimental results show that the detection quality is high when no time skew at test signals and no process variation on interconnects occurs. The scheme is simple and is considered effective in detecting crosstalk faults.

誌　謝

首先，感謝我的父母及家人長期以來所給予的支持與鼓勵，使我有足夠的信心及勇氣繼續攻讀博士學位。特別地，謹獻上最誠摯的謝意與祝福，給我的指導教授李崇仁老師，於十餘年間在論文研究上提供寶貴的建議與細心的指導，以及在生活上的照顧，使我得以順利完成學業，也學習到研究的方法與態度。同時也對口試委員們，於碧利斯(BILIS)颱風襲台期間，仍不畏風雨來參與我的口試，獻上我無限的感激與敬意。

這些年來，對於實驗室的夥伴們，有著深刻的感情，大家的相處彷彿是一個相互扶持的大家庭一般。竹一、孟烈、碧茵、慧民、文慶、新聰、國展、永嘉、志文、順志、俊偉、淑敏、世平等學長(姐、弟、妹)們、同學，以及族繁不及備載的學弟們的指導與鼓勵，是所難忘。也感謝你們為我的生活增添許多的光彩與歡樂，沖淡許多研究不順遂所帶來的煩惱。

最後，感謝工研院的崑崙、繼展，以及其他工作夥伴們，讓我得以獲得在工作上的寶貴經驗。

十餘年的研究生涯雖然漫長，有苦、有樂、有喜，亦有悲，然而最重要的記憶，是這個過程豐富了我的人生與歷練。曾經聽聞一段話：Who we are never changes, but what we are never stops changing，正因如此，對我而言彌足珍貴。

吳明學

謹誌於 新竹交大

九十五年 七月

Contents

Table Captions

Figure Captions

Chapter 1

Introduction

For the advanced modern high density and high speed VLSI of the deep submicron technology, the distances between wires and devices become closer and closer. Parasitic capacitors, which are due to the narrow distances between interconnection lines and insulating layers between conducting layers, or due to unexpected manufacturing defects, become important sources of internal noises for the circuit. For the 0.18 μm technology, for example, two parallel lines of a 240 μm length with a spacing of 0.23 μm will produce a 25 fF parasitic capacitance in the normal fabrication condition [1]. If there is a manufacturing defect, the parasitic capacitances created will be even larger. The noise induced by these parasitic elements interfere the normal operation of the VLSI system in generating unexpected pulses, speeding up or slowing down the transition speed of the affected lines [10, 14, 43]. For one case, if the unexpected pulses appear and are caught by flip-flops at their sampling time, the system will fall into erroneous states. For the other case, if the slowed-down transitions exceed the clock period of flip-flops, erroneous states are also resulted.

On the other hand, the design of system on a chip (SoC) becomes a trend in the present semiconductor industry. System engineers integrate several modules, which may include the CPU, memory, ADC, DAC, or DSP cores, into a chip. Hundreds or even thousands of interconnects travel hundreds of micro meter between these cores. The parasitic capacitances,

inductances and intrinsic resistors between interconnection lines also become an important issue for the system engineers. According to the ITRS roadmap [2], the gate delay will further decrease for the future technology generation but the interconnect delay will increase dramatically. Furthermore, the scaled voltage supply will make the crosstalk effect even worse. Traditional stuck-at fault models can not precisely model the crosstalk fault effect and the conventional test pattern generation tools can not generate the pattern to test them. In addition, the highly unpredictability and signal dependence make testing the crosstalk fault even a difficult, if not unsolvable, task.

1.1 Review

Many works had been dedicated in solving the above problems from various aspects. In the following subsection, we categorize them based on their motivation and application.

1.1.1 Crosstalk Reduction and Analysis

Since the crosstalk effect becomes an importance issue in the modern IC technology, circuit designers have to face the crosstalk reduction problem during the circuit design stage. References [3-9] used the graph-based approach or integer linear programming (ILP) approach to estimate the risk of crosstalk on the channel routing, so that they could minimize the crosstalk effect between interconnects by setting up the design constraint during the circuit synthesis or channel placement and routing stage. Although most of the crosstalk can be

eliminated during the circuit design stage, defects during the manufacturing process make the test for crosstalk become an inevitable task. For more understanding the crosstalk fault effects, many researches focused on their topics on the analysis of crosstalk propagation behavior. For example, in order to obtain insight into effects, including the width, amplitude, energy, and timing characters of the pulses that cause errors, [10] analyzed the crosstalk from the frequency domain to obtain a closed form of the voltage transfer function, and then further transferred the function into expressions in the time domain to make the problem handleable. Reference [11] analyzed the characterization of the crosstalk signals, and they found the relationship between the limitation of crosstalk fault effect propagation and the width and the amplitude of crosstalk-induced pulse. The most difficult work on the crosstalk effect prediction is the uncertainty of signal arrival time between aggressor and victim lines and this makes the crosstalk induced delay time unpredictable. The bound of such the crosstalk effects were analyzed in [12-13].

1.1.2 Crosstalk Fault Models

We can categorize the crosstalk fault models into RLC models [14-20] and symbolic models [21, 32-33]. The research used the RLC models tried to construct simple networks for the existence of crosstalk. These networks are composed of passive elements such as resistors, inductances and capacitors. Their purpose is to try to model the crosstalk environment so that

consistent results between simulation and measurement can be obtained. For example, in [14], it used the simple KVL approach to analyze the crosstalk noise by a three-step approach; in [15], it used KVL and KCL matrix to compute the crosstalk response; [16-17] computed the crosstalk effect based on the coupled-transmission line theory, and especially in [17], it used the effective lumped RC configuration. Because the RLC models target the accuracy of crosstalk behavior, they spent considerable computation time.

Symbolic models [21, 32-33] were often used for the purpose of test pattern generation and fault simulation. Because they used symbols to model fault effects, such as speedup, slowdown, and crosstalk-induced pulse, etc, they sacrifice the accuracy for saving computation time. A 6-vector to test each interconnect was developed in [22]. They provided a complete test set by applying a total 6N test vectors to test N interconnects.

1.1.3 Fault Identification, Simulation and Diagnosis

Because test generation time is a time consuming process, especially for those based on RLC models, the fault identification technique provides a mechanism to identify which fault is effective. Reference [23] proposed a sieve procedure which involves extractors and filters. The extractor extracts the required parameters from the circuit under test with certain accuracy. The filters compute the bound of the crosstalk induced pulses. The sieve procedure determines which fault is discarded or retained. Reference [24] expanded the ability of [23] by

adding a pruning tool to identify which crosstalk fault can induce potentially Boolean errors. On the other way, [25-26] used the topological and timing information to reduce the target faults in synchronous sequential circuits.

Wire bridging are familiar faults in IC manufacturing process. Reference [27] proposed a Verilog transistor model by connecting an NMOS transistor's drain and source terminals to adjacent two wires, and the gate terminal is connected to ground. If a bridging fault occurs, it means that an equivalent stuck at 1 fault happened at the gate terminal. In [28-29], crosstalk fault induced pulse is their target fault. For more improvement of the fault simulation speed, [29] used a concurrent fault simulation technique and unit delay model to save computation time.

In [30], it used two simulation-based methods to identify which faults were suspects for crosstalk. The first one is to build a suspect fault list, and the second used state transition information to save computation time.

1.1.4 Test Pattern Generations for Crosstalk

Most of the test pattern generation for crosstalk fault targeted at the crosstalk induced delay fault. Non-robust and robust test generations based on the single precise crosstalk induced path delay fault model were proposed in [31-32]. Both of them followed the criterions of non-robust and robust delay fault testing. Reference [33] used 11 symbols to

model the fault effects during the test generation procedure. In [34], a timing-independent approach was proposed to generate tests for the crosstalk induced delay. Considering that aggressors have different amount of interference, some are speedup and some are slowdown, on each gate of victim paths, [35] developed an ATPG for finding the pattern to maximize the crosstalk induced delay. Some test pattern generations were developed from the delay test pattern generation [36-38]. They expanded their test pattern generation for delay testing with considering the crosstalk (or noise) effects which are induced from neighbor wires [36-38] or power supply [36], and the crosstalk induced delay were their concern.

Another test pattern generation for the fault effect, crosstalk induced pulse, was developed by [39-42]. Such the fault effect damages the dynamic logics more easily than those static logics. This is because the unexpected pulses can charge or discharge the blocks which are under the evaluation phase, and no way to discharge or charge back to their normal function. The test generation for the dynamic logic, domino, was developed in [43-45].

Some test pattern generation tried to find crosstalk behavior, they used RLC models to develop their tools [46-49]. They utilized a timing analysis technique to help generate test patterns for crosstalk faults at the desired time window, where the timing analysis takes much computation time.

1.1.5 Test Schemes

Many test schemes were proposed to detect the crosstalk faults. Reference [50] designed a test chip to measure the real noise effects of dynamic logics. Combining the statistical analysis, the noise parameters were fed back to the CAD tool to optimize the noise control procedure. Reference [51] tested the SoC interconnects based on the oscillation ring test methodology. They developed the algorithm to form the aggressor line in a loop with odd parity inversion, so that the oscillation signal on the aggressor line will induce pulses on the victim line to increase the value of counter in detectors. In [52], they used the embedded processor cores to test interconnect crosstalk. In [53], it tested the crosstalk for the weak bridge defect in the circuit and [54] tested the crosstalk while the tri-state and bi-directional nets exist on the design. In [55-56], they tested the crosstalk between data line lines and clock lines. In [55], the aggressor line is clock line and the victim line is data line. On the contrary, [56] considered the situation that the aggressor line is the data line and victim line is the clock line.

Self-Test [56-61] and on-line testing [62-66] techniques were widely used in the crosstalk testing for logics [56, 59, 61, 64-66] and interconnects [57-60, 62-63]. Reference [57-58] proposed at-speed test schemes and [62] proposed a test for testing victim lines in parallel. Both of above were used the maximum aggressor model [22]. In [60-61], a

software-based self-test methodology to detect the crosstalk faults on interconnects was also proposed.

1.2 Crosstalk Effects

Although similar studies [67-68] on the crosstalk effects caused by crosstalk faults were reported, in the following subsection, a circuit level simulation for an aggressor line and a victim line is still done to show how the crosstalk fault effects, speedup, slowdown, and induced pulse, which are induced by aggressor line, interfere the output of victim line. Figure 1-1 shows the simulated circuit where the parasitic coupling capacitance is 25 *f*F, and the W values of each driver of the lines are as shown. The L of each drive is 0.18μm.

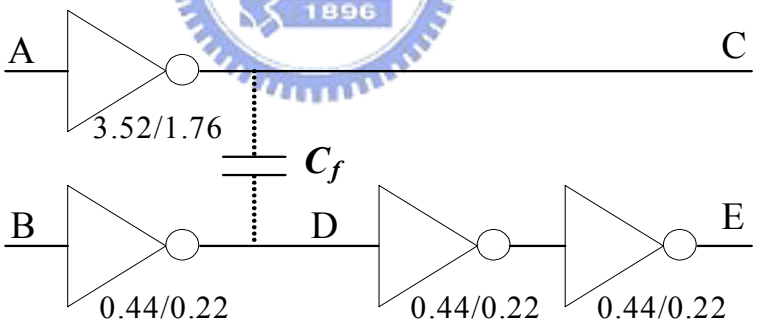
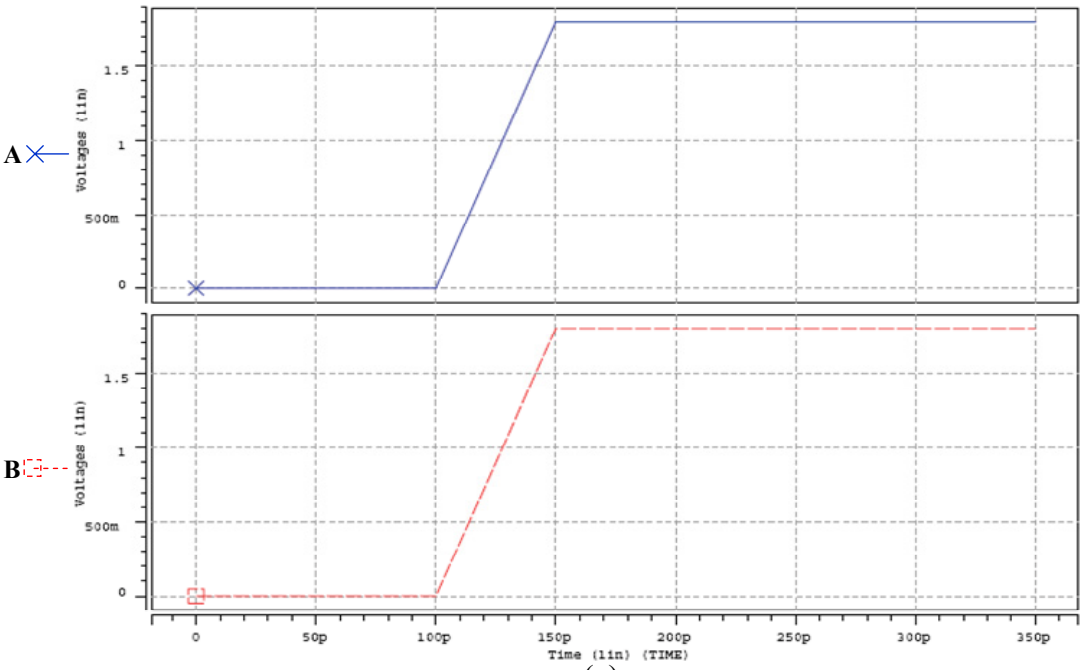


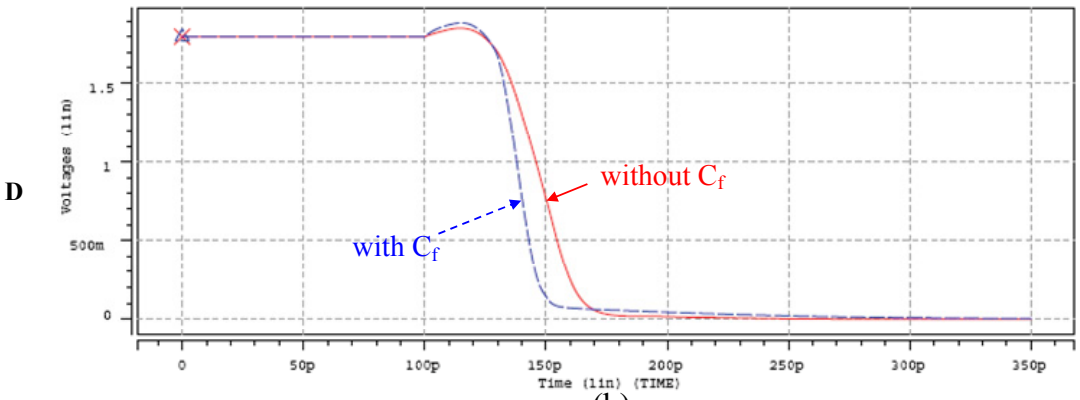
Figure 1-1. The simulated circuit where the parasitic coupling capacitance is 25 *f*F and the L of each driver is 0.18μm while W's of each driver of the aggressor line, and the victim line are as shown

1.2.1 Speedup

The speedup effect occurs at both interconnection lines when two lines, i.e., aggressor line and victim line, have the same-polarity transition in a very close time interval. Basically, the speedup effect will not cause problems on the interconnection lines. On the contrary, it



(a)



(b)

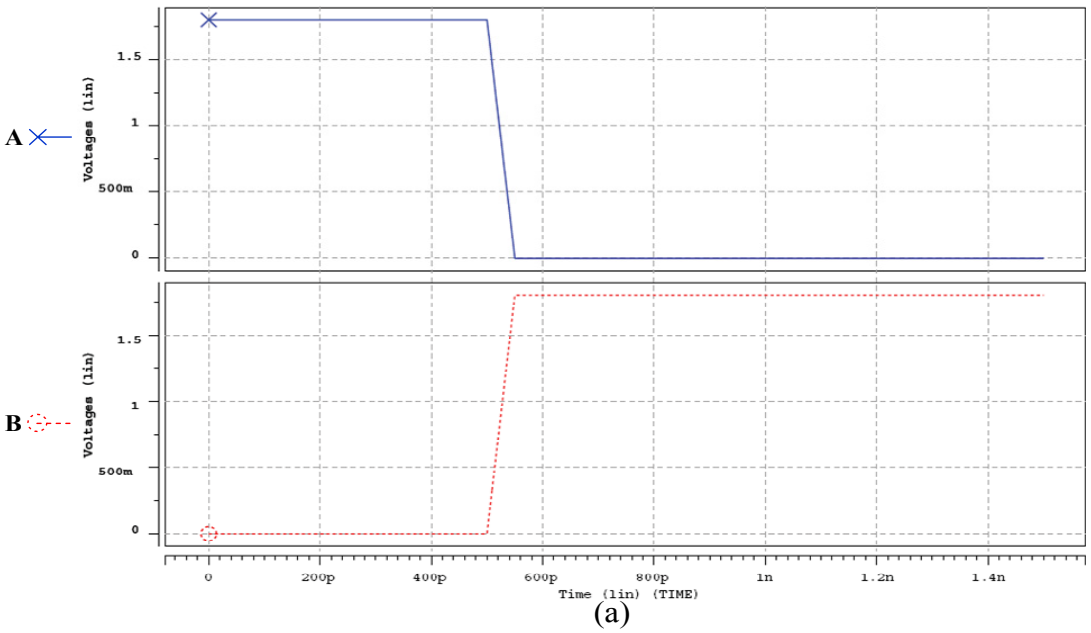
Figure 1-2. The speedup effect of crosstalk fault: (a) The input signals at node A and B. (b) The output response at the aggressor node D with(dash line) / without(solid line) coupling capacitor C_f.

"improves" the path's performance. But on the viewpoint of delay fault testing, the speedup effect may mask the delay fault effect, if it exists, on the victim path, and mistake the path under test to be path delay fault free. In figure 1-2 shows the simulation result of speedup effect caused by the crosstalk fault. In figure 1-2(a), both of the inputs at the node A and the node B are applied a rising transition at time 100 ps simultaneously. In the figure 1-2(b), the

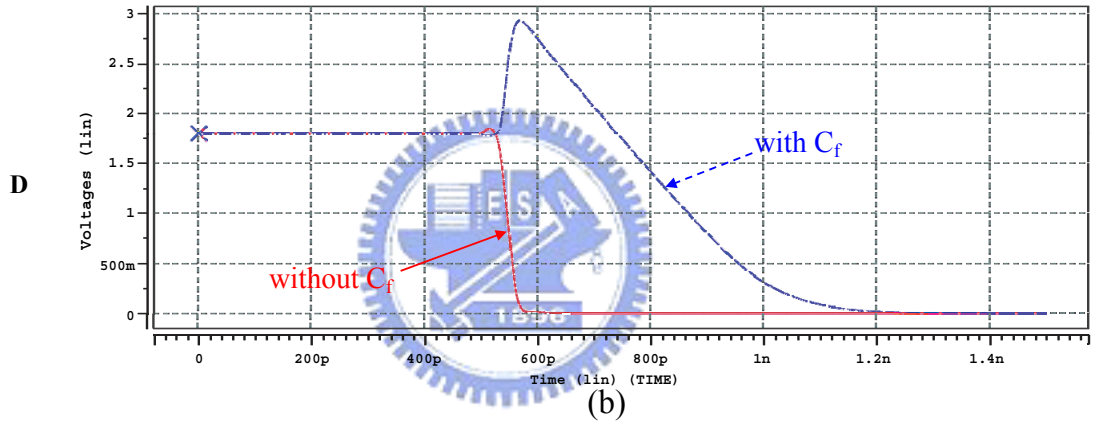
output response at node D is shown. The simulation result shows that if there exists a coupling capacitor C_f between the aggressor line and the victim line, the transition time at the victim line will go faster than that without the coupling capacitor when the aggressor and victim line have the same-polarity transitions simultaneously.

1.2.2 Slowdown

The slowdown effect occurs at both the interconnection lines when two lines have the opposite-polarity transition in a very close time interval. The slowdown effect will worsen the signal propagation time of the signal path. In the traditional delay fault testing, it is sufficient to assure the quality of circuit by testing its critical paths. However, the crosstalk effect may delay transition time of the non-critical paths. When the delayed paths can not be qualified under the speed requirement, the crosstalk induced delay faults occur. It means that more paths must be tested to insure crosstalk effect will not affect the circuit's performance. Figure 1-3 shows the simulation result of the slowdown effect caused by the crosstalk fault. In figure 1-3(a), the node A is applied a falling transition at the time 0.5ns. At the same time, the node B is applied a rising transition. In the figure 1-3(b), the output response at node D is shown. The simulation result shows that if there exists a coupling capacitor C_f between aggressor line and victim line, the transition time at the victim line will delay than that without coupling capacitor when the aggressor and victim line have the opposite-polarity transitions simultaneously.



(a)

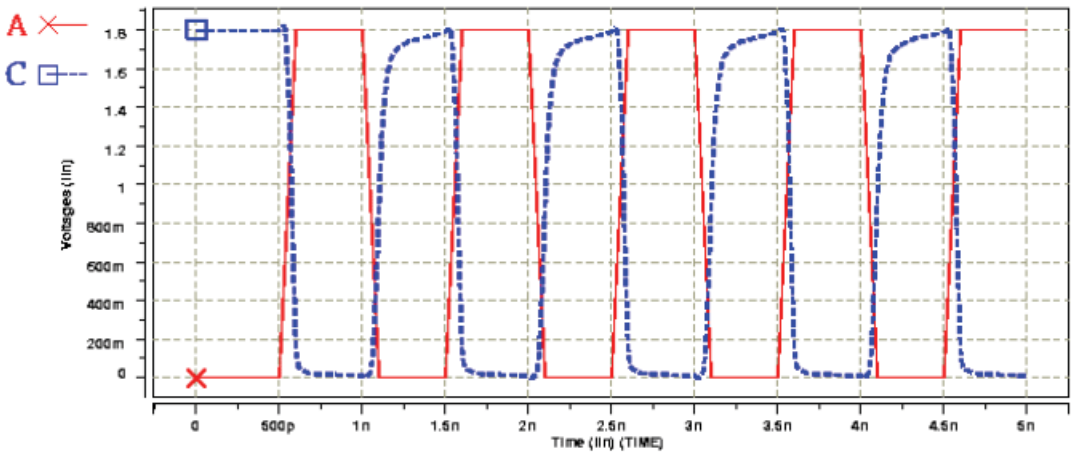


(b)

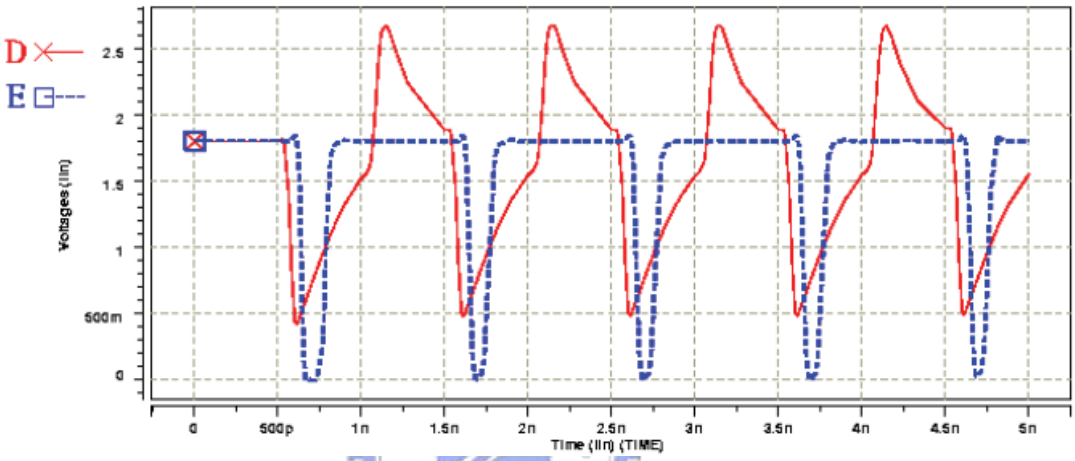
Figure 1-3. The slowdown effect of crosstalk fault: (a) The input signals at node A and B. (b) The output response at the aggressor node D with(dash line)/without(solid line) coupling capacitor C_f.

1.2.3 Induced Pulses

The crosstalk induced pulses occur when the victim line has a transition from the same voltage level to opposite voltage level with the victim line. In Figure. 1-4(a), the input of the aggressor line is applied an oscillating square wave, the victim line is at a static value. The simulated output waveforms at the output E of the victim line are shown in Figure 1-4(b) with those waveforms at A, C, D and E respectively.



(a)



(b)

Figure 1-4. The circuit level simulation for the induced pulses caused by a crosstalk fault: (a) The input waveform at node A and the simulated waveforms at nodes C. (b) The simulated waveforms at affected node of D and E.

Besides the crosstalk effects mentioned above, the crosstalk fault effects will consume more power than the normal circuit operation [2]. More current is drawn to charge or discharge the coupling capacitors. In table 1-1, the issues caused by the crosstalk fault effects are listed.

Table 1-1 Issues of crosstalk fault effects.

Effect	Causes		Phenomenon	Issues
	aggressor	victim		
Speedup	rising	rising	Faster rising on the victim line	Increase testing escape rate: *- require change test pattern* Slightly voltage overshoot Extra power consuming
	falling	falling	Faster falling on the victim line	
Slowdown	rising	falling	Slower falling on the victim line	Increase testing escape rate: *- required additional test* Heavily voltage overshoot Extra power consuming
	falling	rising	Slower rising on victim line	
Induced pulse	rising	0	Induce pulse 1 on the victim line	Unexpected pulses Error capture Extra power consuming Over testing
	falling	1	Induce pulse 0 on the victim line	

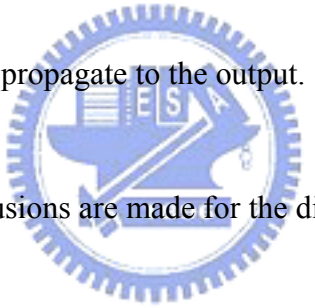
1.3 Outline of Dissertation

In Chapter 2, a random test pattern generator and deterministic test pattern generator are developed and described to detect the crosstalk effect caused by coupling capacitors. The deterministic test pattern generator is based on the oscillation ring test scheme [69-70]. Unlike [69], where one of the CUT's outputs is connected one of its output for generating oscillation signals, the oscillating signal here is directly applied at one of the CUT's inputs. If there is any aggressor line located at the oscillation path, there will induce a crosstalk fault effect. It is to activate the fault and propagate the fault effect to the output.

In chapter 3, the application of the oscillation ring test scheme is extended to the boundary scan environment, and a BIST scheme is proposed to detect crosstalk faults in cores.

With modifying the boundary scan registers and the test patterns are applied by the LFSR, the test scheme can operate self test mechanism. Some simulations for the test scheme are made, and the simulation results show that the fault coverage are up to 90% for most of benchmark circuits. The self test scheme will reduce the memory requirement on the ATE for testing the crosstalk fault.

In chapter 4, the crosstalk faults between interconnects were studied and a test scheme based on the path delay inertia principle is proposed. In the test scheme, a pulse is applied at the input of interconnection line. If there is a coupling capacitor which exceeds the limit of specification, the pulse will not propagate to the output.



Finally, in chapter 5 conclusions are made for the dissertation.

Chapter 2

A Testing Scheme for Crosstalk Faults Based on the Oscillation Test Signal

2.1 Preliminary

In 1996, an oscillation test methodology was first proposed to test the analog circuit and mixed-signal IC [71-72]. Under the test mode, the CUT is converted to be an oscillation circuit. If there is any fault (including the catastrophic or the soft fault) in the CUT, the observed oscillation frequency will deviate from the normal oscillation frequency. The test scheme is easy for the test engineer by only observing the oscillation frequency of the CUT without applying any test vector. In 1997, the oscillation test methodology was first applied on the digital circuits [69], which was called oscillation ring test. The test scheme firstly found the sensitized paths. When the circuit under the test mode, the output of one of sensitized path is connected to its input either directly or with inserting an inverter by considering the polarity of input and output are same or opposite respectively. The test scheme could detect all stuck faults on the oscillating path, and some faults on side inputs.

In this work, a test scheme which is based on an oscillation square wave signal is proposed. In the scheme, the oscillation signal is applied on the *aggressor* line and the induced pulses at the *victim* line are detected if a crosstalk fault between these two lines exists.

The scheme eliminates the need to consider the timing issue of faults, since as long as the induced pulses, which are induced by the crosstalk fault and usually induced unexpectedly, exist, they will be detected at the output of the victim line. This makes test generation for faults very simple with no need to consider timing of fault occurrence. Test generation and fault simulation for crosstalk faults based on the scheme are described and experimental results are presented.

2.2 The Testing Scheme Based on the Oscillation Test Signal

Figure 2-1 (a) is an example circuit with which the idea of the proposed testing scheme is demonstrated, where a crosstalk fault exists between aggressor line X and victim line Y. A test pattern, which includes an oscillating square signal, is applied to inputs of the circuit and the oscillation signal is propagated to the aggressor line X. Due to the crosstalk fault, a series of induced pulses will appear at the victim line. When the induced pulses are propagated to the output under the applied pattern and detected, the crosstalk fault is detected. Once the fault is detected, another pattern is applied to the circuit to test another coupling fault. The timing diagram of the pattern is shown in Figure 2-2(b), where another pattern is also included for demonstration. As it is easily seen that, in order to observe a clean pulse train at the output of the victim line, patterns applied to inputs of the circuit should be a “hazard-free” patterns. The patterns are very similar to those which are used to detect the path delay fault of the circuit

[73].

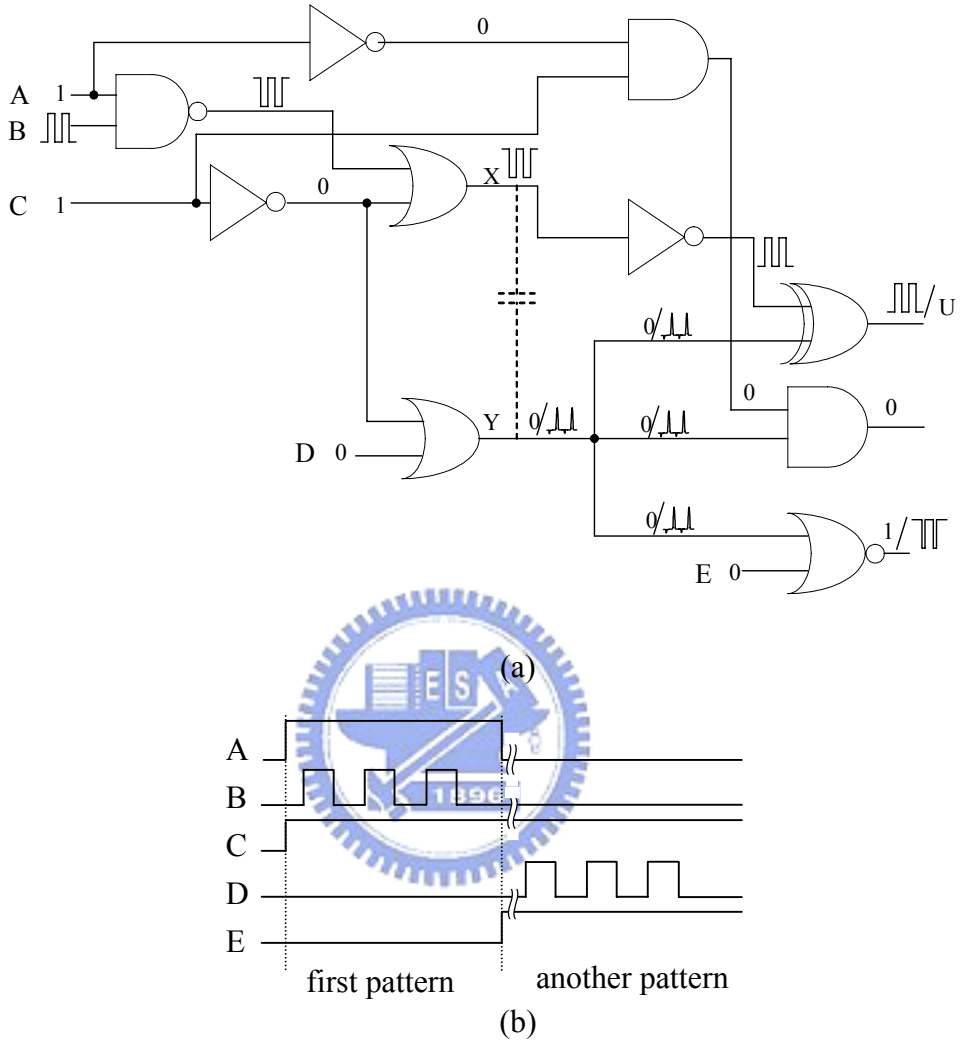


Figure 2-1. (a) A circuit example to demonstrate the test scheme for crosstalk fault detection; (b) The timing diagram of the applied patterns.

2.3 Test Generation and Fault Simulation

As it was mentioned above, the test patterns for this test scheme should be the "hazard-free" pattern. In order to generate the test patterns, a set of symbols are first introduced and a set of operation rules are defined for each type of gates to guarantee "hazard-free". The set of symbols are shown in Figure 2-2(a), where C, and C^* are fault effect

signals which are to be propagated and detected at the output, PF and PF* are potential fault effects which could be 1 or C and 0 or C* respectively, S and S* are oscillation signals which are to be applied at the fault site, and ***x*** represents for an "unknown" value. Figure 2-2(b) shows examples on how these symbols are operated through an & (AND) gate. For examples, C* & C* or C & C* results in PF* and C & C results in C. Figure 2-2(c) shows the truth tables for AND and OR operation respectively. In the table, it can be seen that these sets of operations are rather conservative. For examples, when S operates on S (or S*), an *x* is given, which indicates an "unknown", while for practical cases, most often an S (or S*) is obtained.

0 : static 0

1 : static 1

C* : positive pulses at logic 0

C : negative pulses at logic 1

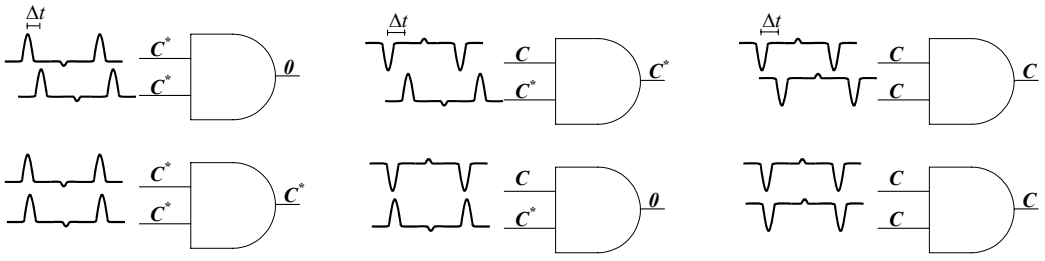
PF* : ***0*** or ***C****

PF : ***1*** or ***C***

S*, S : oscillation signals

x : Unknown value

(a)



(b)

	0	1	C*	C	PF*	PF	S*	S	x
0	0	0	0	0	0	0	0	0	0
1	0	1	C*	C	PF*	PF	S*	S	x
C*	0	C*	PF*	PF*	PF*	PF*	x	x	x
C	0	C	PF*	C	PF*	C	x	x	x
PF*	0	PF*	PF*	PF*	PF*	PF*	x	x	x
PF	0	PF	PF*	C	PF*	PF	x	x	x
S*	0	S*	x	x	x	x	x	x	x
S	0	S	x	x	x	x	x	x	x
x	0	x	x	x	x	x	x	x	x

AND-gate

	0	1	C*	C	PF*	PF	S*	S	x
0	0	1	C*	C	PF*	PF	S*	S	x
1	1	1	1	1	1	1	1	1	1
C*	C*	1	C*	PF	C*	PF	x	x	x
C	C	1	PF	PF	PF	PF	x	x	x
PF*	PF*	1	C*	PF	PF*	PF	x	x	x
PF	PF	1	PF	PF	PF	PF	x	x	x
S*	S*	1	x	x	x	x	x	x	x
S	S	1	x	x	x	x	x	x	x
x	x	1	x	x	x	x	x	x	x

OR-gate

(c)

Figure 2-2. (a) The symbol table for test generation and fault simulation for crosstalk faults; (b) examples of symbol operation through an AND gate; (c) truth table for the AND and OR operations.

With the set of variables and operation rules defined above, the conventional stuck-at fault test generation is used to generate test patterns. The patterns generated will be ones which do not create re-convergent sensitized path. However, also due to the stringent conditions imposed above, patterns are sometimes very difficult to be generated and the test generation is time consumed. And very often, undetected or aborted faults are resulted during the test generation process.

For the reason stated above, to generate patterns for this test scheme, a random test generation is first adopted and the generated patterns are crosstalk fault simulated to determine the fault coverage. After the fault coverage has reached a certain value, the remaining undetected faults are then applied a deterministic test generation procedure to

generate remaining test patterns. This reduces the test generation time and increases the test generation efficiency. In the following, the two approaches to generate test patterns to crosstalk faults are described:

2.3.1 Random Test Pattern Generation

For this approach of test generation, no specific fault is targeted and random patterns are generated and fault simulated to find detected faults (out of a fault list). However, as experimental results show that this approach is generally inefficient, i.e., the generated patterns detect only a few patterns, resulting much wasted fault simulation time, a guided random pattern generation method is used.

For the guided random pattern generation, the circuit is firstly processed with the input cone of the aggressor line of each fault identified (shown in Figure 2-3). At first, a random pattern is generated and applied to the circuit with one of inputs of the input cone applied with signal ***S***. The pattern is then fault simulated to see if ***S*** is propagated to the faulty site and a faulty effect ***C*** is induced and propagated to any of outputs. If it fails, ***S*** is applied to the next input of the input cone and another fault simulation is done. With all the inputs of the input cone scanned by ***S***, if the pattern still does not detect any fault, another random pattern is generated and the above process is repeated to see if the generated pattern is a test. Due to the fact that ***S*** is guidedly applied to the input cone of the aggressor line, the chance to achieve a

successful pattern is much enhanced.

Figure 2-4 shows an example that a pattern is simulated and a fault (f_1) is detected by the pattern and another fault (f_2) is potentially detected. For this random test generation process, as no "justification" step is involved, it is usually very fast as compared to the approach of the deterministic test generation which is to be explained in the later section.

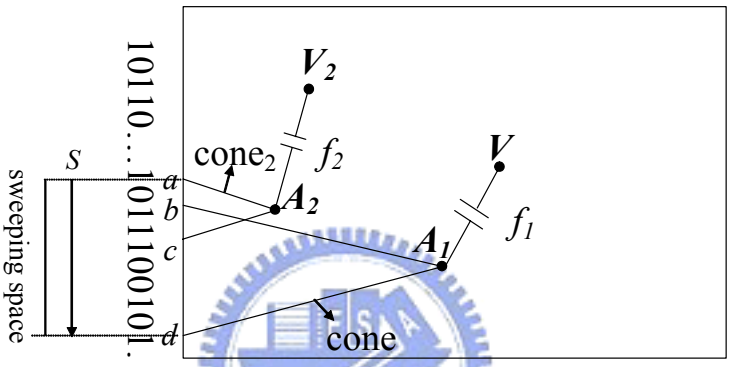


Figure 2-3. The example to show the input cones of aggressor lines A_1 and A_2, where V_1 and V_2 are victim lines of crosstalk faults f_1 and f_2 respectively.

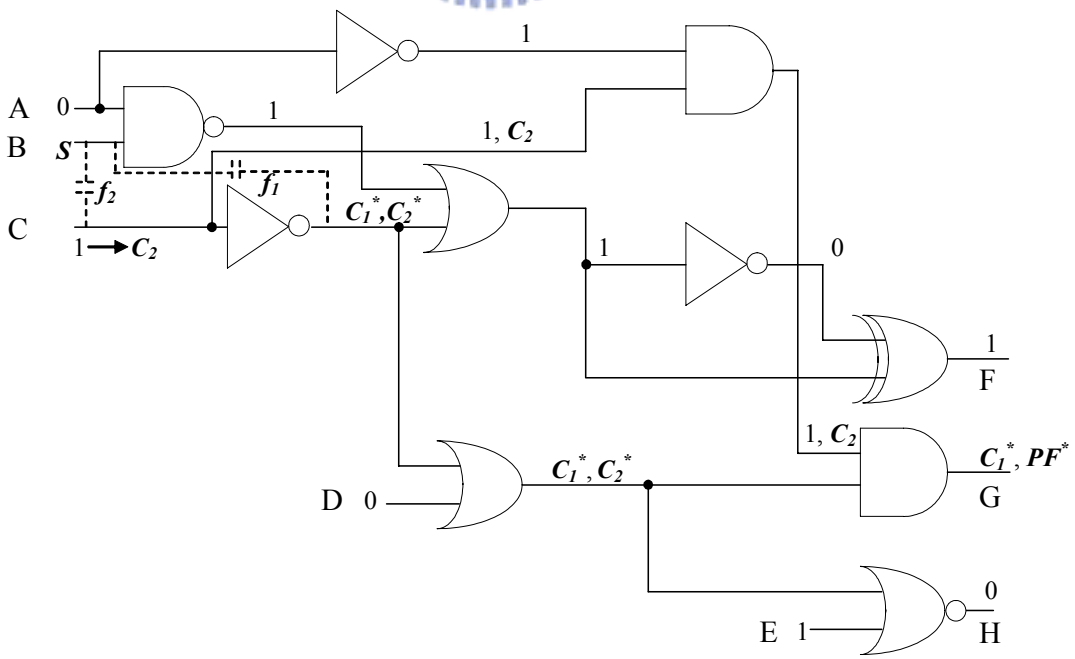


Figure 2-4. A fault simulation example where f_1 can be detected at G, and f_2 can be potentially detected at G.

2.3.2 Deterministic Test Pattern Generation

For this approach of test generation, a specific fault is targeted. The test generation process is similar to that of the conventional stuck-at fault test generation except that, as mentioned previously, the sensitized path of the fault activation does not intersect with that of the fault propagation.

Figure 2-5 shows the flow chart of the test generation procedure. In the procedure, a cost function and observability [74] for ***C*** for all faults are computed respectively to aid the later test generation. Then a target fault is selected to be generated the test pattern. For the selected targeted fault, another similar computation for the cost function and observability [74] for the activation line ***S*** is done. Then a fault analysis step is performed to avoid the intersection of the potential sensitized activation path and the potential sensitized propagation path during the later justification. An example circuit is shown in Figure 2-6 to explain the above step. In the figure, Y is the victim line which is at value ***C*** which has three paths to be propagated to the outputs of the circuit. Line X is the aggressor line which is assigned value ***S.*** Both ***C*** and ***S*** need to be justified later. Through the fault analysis, for line X, its input lines are assigned N_C and N_1, which mean that they can not be signal ***C*** and ***1*** respectively since the output of the OR gate is ***S*** signal. Similarly, the output line of gate X can not be ***C***, ***1***, or ***0***, so N_C, N_1, and N_0 are assigned. All these values are then propagated backwardly and forwardly respectively

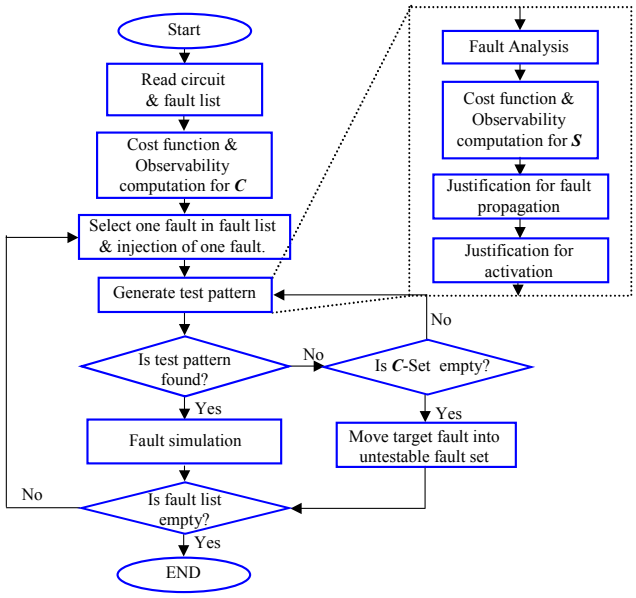


Figure 2-5. Flow chart of the deterministic test pattern generation

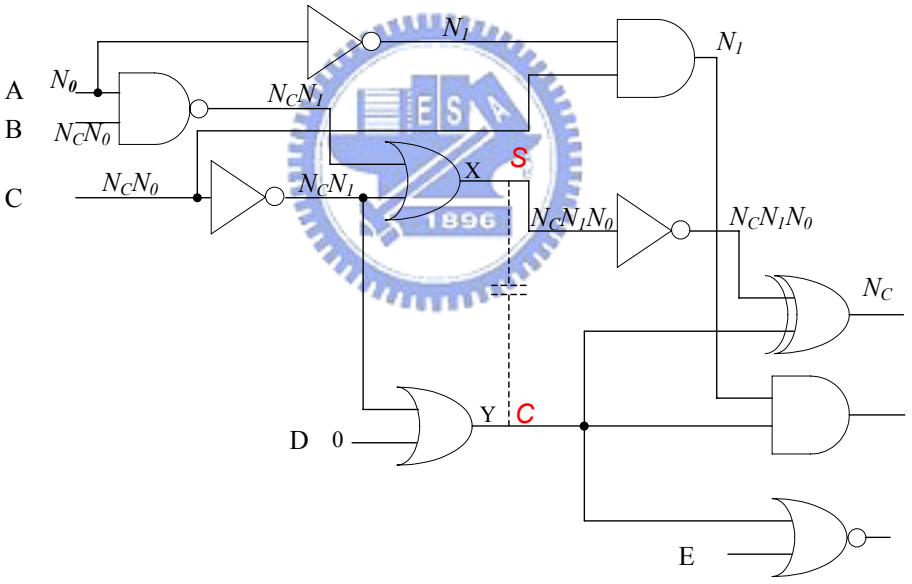


Figure 2-6. An example circuit to demonstrate the fault analysis step, where X is the aggressor line which is at ***S*** and Y is the victim line which is at ***C***.

to find the feasible sensitized paths. During propagating these values, the cost functions and observabilities computed previously for ***C*** and ***S*** will be used as reference to determine the propagation path. After this step justification of paths are done, which are in two phases. In

phase 1, the justification for fault propagation is done and in phase 2, the justification for the fault activation is done. Once the pattern is found, it is fault simulated to see if it can detect other faults.

2.3.3 Fault Simulation

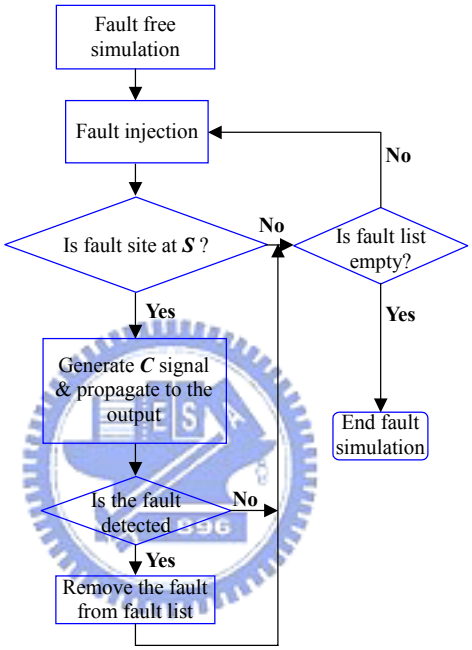


Figure 2-7. Flow chart of the fault simulation

For the fault simulation, when a test pattern is applied to the circuit, a fault free logic is firstly computed with the set of variables and operation rules defined above. Then a fault is injected to see if the aggressor line is at the ***S*** (oscillation) path. If the aggressor line is at the oscillation path, the victim line of the fault site is set to ***C***, and it is propagated toward the output of the circuit. If the propagation is successful, the crosstalk fault is detected by the pattern and another fault is injected. The above process is repeated until all faults are injected

and another pattern is simulated. The flow chart of the fault simulation is shown in Figure 2-7.

2.4 Experimental Results

The above two approaches were applied to benchmark circuits [75] to generate test patterns for crosstalk faults. Both of them were implemented in C language and run on SUN Ultra60. Table 2-1 shows results for the random test pattern generation approach where both the results of the conventional random pattern generation approach and the guided random pattern generation approach are listed. In the experiment, since crosstalk faults are dependent on the circuit layout, which was not available, 100 crosstalk faults, some of them might be undetectable, were randomly assumed and injected and 100 patterns were randomly generated. In the table, for each circuit, the number of inputs is listed, where PD is the potential detected faults, FC (fault coverage) includes the PD faults, and TG is the total test generation time which mainly is the fault simulation time. From the table, it is seen that for this random test generation approach, fairly high FC, over 85% in average, can be obtained, especially for larger size circuits such as s13207c ~ s38584c. The test generation time in general is acceptable and the guided approach has less test generation time than does the conventional approach, especially for larger circuits which have large number of inputs.

Table 2-2 shows the results of the deterministic test generation. In the table, the faults injected were the same as those of Table 1, and UD represents for identified undetectable

Table 2-1. Test generation results of both the conventional and guided random pattern generation approaches.

Circuits	No. of PI	PD	F.C. (%)	TG time(sec)	
				conven.	guided
c5315	178	0	91	0.78	0.31
c6288	32	31	47	0.71	0.70
c7552	206	2	88	1.07	0.56
s3271c	142	0	96	0.33	0.17
s3330c	172	0	59	0.45	0.26
s3384c	209	0	90	0.49	0.13
s4863c	153	0	91	0.83	0.52
s5378c	214	0	89	0.55	0.22
s6669c	312	0	98	1.44	0.19
s9234c	247	1	77	0.88	0.50
s13207c	650	2	86	2.82	0.43
s15850c	600	0	89	2.81	0.62
s35932c	1763	0	94	19.49	1.59
s38584c	1462	0	92	13.19	1.19

Table 2-2. Test generation results of deterministic test pattern generation approach

Circuits	UD	Abort	F.C. (%)	Test efficiency	Time (sec)
c5315	7	0	93	100%	0.76
c6288	26	47	27	53%	516.48
c7552	7	1	92	99%	4.46
s3271c	4	0	96	100%	0.36
s3330c	7	0	93	100%	0.19
s3384c	0	0	100	100%	0.17
s4863c	0	2	98	98%	3.84
s5378c	2	0	98	100%	0.44
s6669c	2	0	98	100%	0.74
s9234c	7	0	93	100%	0.43
s13207c	2	0	98	100%	0.75
s15850c	2	0	98	100%	0.73
s35932c	6	0	94	100%	1.30
s38584c	5	0	95	100%	1.33

faults and Abort represents for faults aborted when their test generation time exceeded for a certain limit (10 seconds for c6288 and 1 second for all other circuits). It is seen that for most of circuits, the test efficiency are 100%, i.e., UD faults can be identified and all faults can be

generated test patterns. The test generation time for each circuit is usually longer than that of the guided random test generation approach but the fault coverage obtained is generally somewhat higher than that of Table 1.

Table 2-3. Test generation results of the guided random pattern generation approach (*) combined with the deterministic test pattern generation approach (**).

Circuits	P.D.		F.C. (%)		Total time(sec)		Total Pattern
	*	**	*	**	*	**	
c5315	0	0	91	93	0.31	0.48	102
c6288	31	29	47	47	0.70	45.08	102
c7552	2	2	88	93	0.56	0.60	105
s3271c	0	0	96	96	0.17	0.25	100
s3330c	0	0	59	93	0.26	0.34	134
s3384c	0	0	90	100	0.13	0.14	110
s4863c	0	0	91	99	0.52	1.54	108
s5378c	0	0	89	98	0.22	0.48	109
s6669c	0	0	98	98	0.19	0.23	100
s9234c	1	1	77	94	0.50	0.67	117
s13207c	2	2	86	99	0.43	0.52	113
s15850c	0	0	89	98	0.62	0.72	109
s35932c	0	0	94	94	1.59	1.63	100
s38584c	0	0	92	95	1.19	1.23	103

Table 2-3 shows the results that at first the guided random test pattern generation approach was applied with 100 generated patterns and then the deterministic test pattern generation method was applied again to generate more test patterns to test the remaining faults. In the table, Total Pattern means the total number of patterns generated which include the guided random patterns approach and the deterministic patterns. It can be seen that further

improvement can be obtained by combining these two approaches to generate test patterns for crosstalk faults. The total test generation time is acceptable for all circuits except for c6288, which is a multiplier circuit with many XOR gates.

2.5 Summary

In this chapter, we propose a test scheme to test crosstalk faults. It uses an oscillation signal applied on an aggressor line and detects induced pulses on a victim line if a crosstalk fault exists between these two lines. It is simple and eliminates the complicated timing issue during test generation for the crosstalk fault. The scheme is very simple and easy to be implemented. Two test generation approaches, i.e., the guided random test generation and the deterministic test generation are described and experimental results are presented. The experimental results show that the proposed test generation approach, i.e., it first uses the guided random test generation then a deterministic approach can effectively generate crosstalk fault test patterns for circuits.

Chapter 3

A BIST Scheme for Detecting Crosstalk Faults Using Squarewave Test Signal in Boundary Scan Environment

3.1 Preliminary

The basic difficulty on testing crosstalk faults is that they are pattern-dependent and are highly unpredictable in nature. To generate test patterns to deterministically detect them, a timing analysis program of high precision needs to be employed, and this takes much computation time. In chapter 2, a test scheme which is based on an oscillation signal was proposed. In the scheme, the oscillation test signal is applied on the *aggressor* line and the induced glitches at the *victim* line are detected if a crosstalk fault between these two lines exists. The scheme eliminates the need to consider the timing issue of faults, since as long as the induced glitches, which are induced by the crosstalk fault and usually are induced unexpectedly, exist, they will be detected at the output of the victim line. A guided random test pattern generation was also proposed in subsection 2.3.1 of chapter 2, and the test scheme was shown in Figure 2-3. It is very suitable to apply the test scheme to be BIST, because the random pattern can be generated by LFSR. The simulation result for the test scheme was list in table 2-3. It showed that the test scheme had high fault coverage for most benchmark circuits except the C6288.

In this chapter, the test scheme is modified to be applied to test this type of crosstalk fault

of the unexpected glitches of embedded circuits in the boundary scan environment of a VLSI. The scheme tests crosstalk faults with the squarewave test signal under applied random patterns which are generated by the LFSR which is trans-configured by boundary scan cells of the embedded circuit. The scheme is very simple in test generation and test application with no need to consider the timing issue of fault occurrence. Experimental results show that considerable high (> 90%) fault coverage can easily be obtained for detecting the induced glitch type of crosstalk fault for large size benchmark circuits [75]. In the following sections, the test scheme is described in section 3.2, and the detail BIST circuit is shown in section 3.3. The boundary scan circuits, including boundary scan-in cells and boundary scan-out cells, should be modified for implementing the test scheme. The modified boundary scan cells and the detection circuit are described in section 3.4. In section 3.5, the experimental results which are simulated by C program are shown to manifest the practicability of the test scheme.

3.2 The Test Scheme

The basic idea, which is similar to the idea shown in chapter 2, behind the test scheme is described in subsection 3.2.1. In subsection 3.2.1, the test scheme architecture which is based on the boundary scan environment.

3.2.1 Basic Idea

Figure 2-1 is a circuit example which demonstrates the basic idea of the detection of

crosstalk faults by using the oscillation squarewave signal. In the figure, A, B, C, D and E are inputs, O_1, O_2 and O_3 are outputs, line X is the *aggressor* line, line Y is the *victim* line, and the coupling capacitor *C* represents a coupling fault between lines X and Y. A random test pattern, {***1, S, 1, 0, 0***}, where ***S*** stands for an oscillation square wave signal, is applied to the input and the output will be {O_1, O_2, O_3 = ***S***, ***0***, ***1***} if the circuit is fault free, where O_1 = ***S*** means that an oscillation squarewave signal appears at O_1. However, if the coupling fault *C* exists between the line X and the line Y, a pulse train will be induced at line Y. The induced pulse train will be propagated to output O_3 and detected under the applied pattern. The coupling fault *C* is thus detected.

3.2.2 Test Architecture in Boundary Scan Environment

Figure 3-1 shows the architecture of the test scheme in the boundary scan environment. The input cells store a random pattern, which is generated randomly by an LFSR which is transformed from boundary scan cells. When testing is started, the first bit of the input scan cells is flipped in polarity and all the outputs of the circuit are checked to see if there are any polarity changes on them. If there are any output changes, say on outputs O_k's, this indicates that there are sensitized paths existing between the outputs O_k's and the input. Then, an oscillation squarewave signal ***S*** is applied to the input. At this time if any pulse trains are detected at any outputs other than O_k's, there exist coupling faults between the sensitized

paths and *victim* lines and the coupling faults are detected by this pattern. When the above testing is finished, the above procedure is repeated again to the second bit of the inputs. Additional coupling faults will be caught if they do exist. And this procedure is repeated for all inputs until to the last bit. At this moment, another new random pattern is generated and stored in the input cells and another run of the above testing procedure is done again. When **p** random patterns are applied, **m** x **p** testings are applied, where **m** is the number of the inputs. Hence, it can be expected that the detection efficiency of patterns should be very high and this is verified by the experiment results of the later section.

In the above, the oscillation square wave signal can be provided by an oscillator, which can be either built in locally or provided by the SOC chip in which the CUT is embedded.

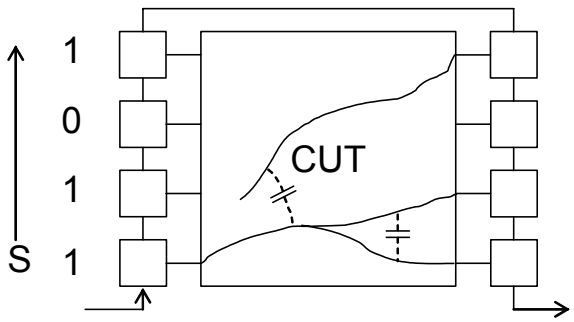


Figure 3-1. The architecture of the squarewave test scheme for crosstalk faults in the boundary scan environment. For one random test pattern applied to the inputs of the circuit, the excitation signal ***S*** is scanned from the first bit to the last bit of the inputs.

3.3 Detail BIST Circuits

The general description about the test scheme is first described in the following

subsection 3.3.1. In this subsection, the timing diagram of control signals and oscillation squarewave signal ***S*** are shown to explain how the test scheme operates.

3.3.1 General Description of the Circuits

Figure 3-2 shows the more detail circuit for the BIST test architecture of the previous section. The circuit contains a ***random pattern generator block*** and a ***detector block***, which are located at the input and output of the CUT, which has **m** inputs and **n** outputs, respectively. The random pattern generator block contains an m-bit **LFSR**, where clock $\mathbf{TCK_0}$ is the signal which controls the **LFSR** to change its state and **Sq** is the oscillation squarewave, which as mentioned previously could be built-in locally or provided externally by other circuits in the SOC. The shift register is to shift a single logic 1 value from its lowest bit to its highest bit. The outputs of shift register are connected to an m-bit 2-to-1 multiplexer, of which the outputs could be from the LFSR or from **Sq**. The detector block contains two D-type flip-flops, which are output boundary cells of the CUT and are to store the test results of the output, two Exclusive-OR gates ($\mathbf{Xor_1}$ and $\mathbf{Xor_2}$), a detection mode latch (**DM**), which is enabled by the signal **DMLE** to latch the output of **Xor1**, and an error detector (**ED**), which is to detect risky pulses of the output signal.

As described previously, for the circuit, there are three operation phases, i.e. *the random test pattern generation phase*, *the oscillation signal port selection and stable logic latch phase,*

and the oscillation test phase. In the first phase, one pulse arrives at **TCK_0**, and **LFSR** changes its state to apply a random pattern to the CUT. Shift register (**SR**) begins to load first logic 1 from its LSB, and the multiplexor selects its LSB to be **Sq**. In the second phase, it is to identify the sensitized path and **Sq** applies "0" and "1" in sequence to the CUT. The output result of the first "0" is stored in **DFF_1,** which is a positive-edge-triggered flip flop, and the output result of the second "1" stays at the output of the CUT. **Xor_1** compares these two bits and its output is latched at **DM**, which is enabled by the signal **DMLE** at the beginning of this phase. If **DM** is "1", it means the output of the CUT is a sensitized path. In the third phase, which is *the oscillation test phase*, **Sq** is applied an oscillation square wave signal at the rising edge of test clock to test the circuit. The capture clock, update clock and detection clock (**Det_{ck}**) of boundary scan cells are synchronized with the test clock (see Figure 3-3 of timing diagrams of all the signals). The circuit operates at the normal speed, and its output data are shifted into **DFF_1** and **DFF_2**. **Xor_1** compares the outputs of **DFF_1** and **DFF_2**. If **DM** = 0, which means the output is not a sensitized output, but the error detector detects a "1", an unexpected pulse induced by a crosstalk fault is detected.

In the above, it can detect the glitches which propagate to DFF_1 when DFF_1 is activated at the rising edge of Cap_{cko}. This is the case that the crosstalk-induced glitches affect the circuit operation in the normal mode. The glitches which arrive before or after the triggering edge of Cap_{cko} are not considered since they do not affect the circuit operation.

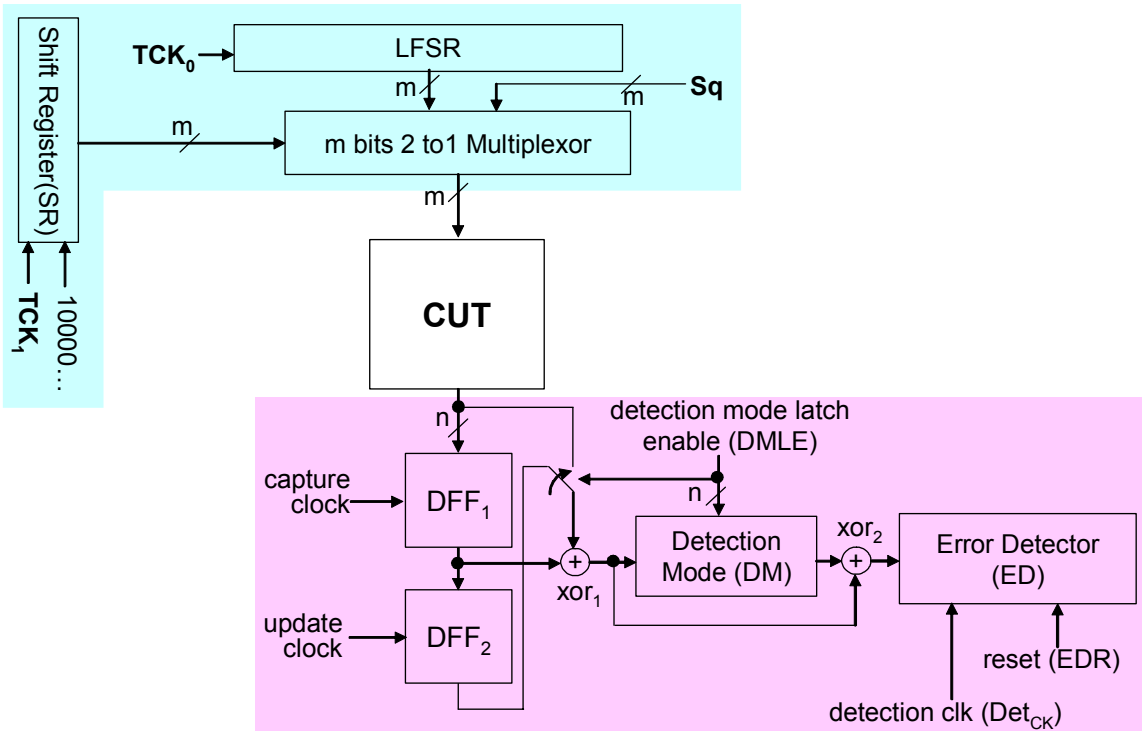


Figure 3-2. The detail circuit of the BIST architecture for the crosstalk fault detection scheme.

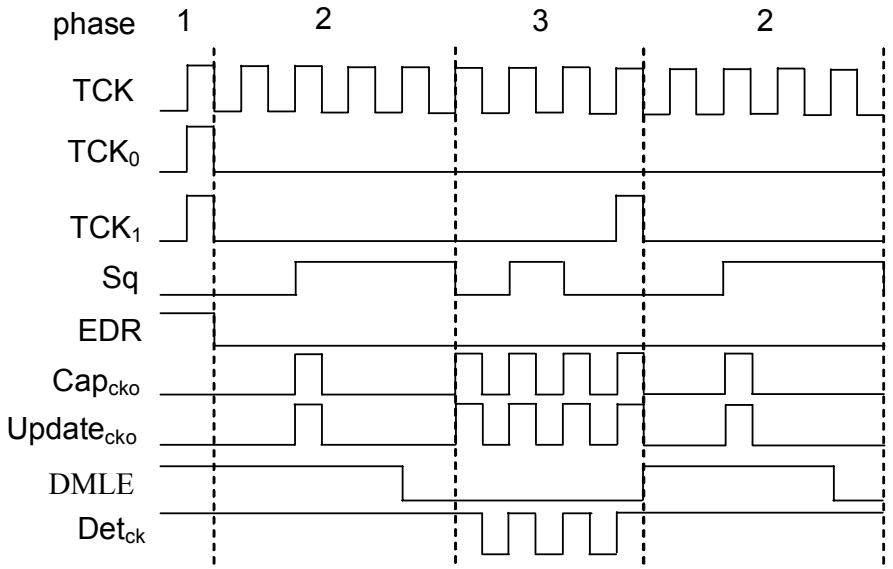


Figure 3-3. The timing diagrams of the BIST circuit. The procedure is in three phases, i.e., *the test pattern generation phase*, *the oscillation signal port selection and stable logic latch phase, and the oscillation test phase*, for one pattern.

Figure 3-3 shows timing diagrams of all the signals of the circuit for the test procedure,

where **TCK** is the test clock which may be the system clock of the CUT, and **EDR**, $\mathbf{Cap_{cko}}$, and $\mathbf{Update_{cko}}$, are the error detector reset signal, the capture register, the update register signals of the boundary scan cell.

It is to be mentioned that, the scheme is to detect crosstalk induced glitch faults of a combinational block. To apply it to a sequential circuit, similar to the case of stuck-at fault testing, the sequential circuit needs to be scan-designed and only the combinational part is to be tested.

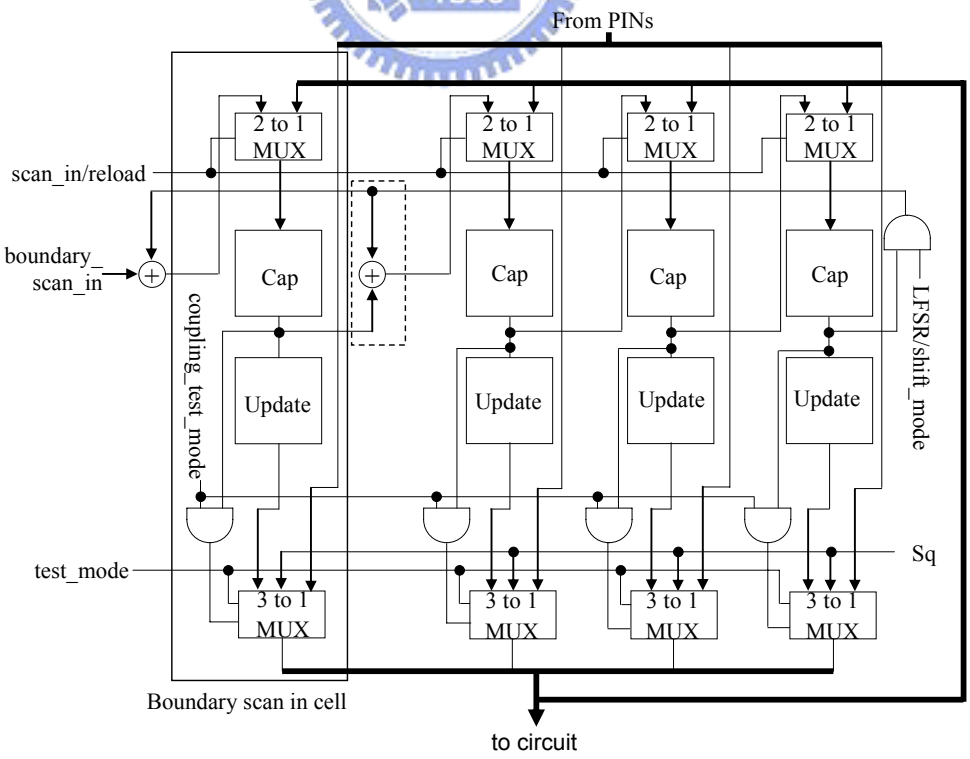
3.4 The Boundary Scan Cell Design

Since in the above circuit, the input and output sub-circuits are transformed from the boundary scan cells, the original designs of the scan cells need to be modified.

3.4.1 Boundary Scan-in Cell

The boundary scan-in cells of this scheme should be able to implement the function of LFSR in addition to their original shifting function. Figure 3-4 shows the modified 4-bit boundary scan-in cell design, where the boundary scan cells are conventional boundary scan cells with a 3-to-1 multiplexor added to each cell, and **Cap** and **Update** are capture registers and update registers respectively. When transformed into the **LFSR** mode, it has a seed of the primitive polynomial $1 + x + x^4$. As described above, in the first *phase*, $\mathbf{TCK_0}$ =1 and the cells

are in the **LFSR** state, where **LFSR/shift_mode** and **scan_in/reload** are set to "1". The capture clock $\mathbf{Cap_{cki}}$ (= $\mathbf{TCK_0}$) at this moment starts to clock capture registers until the capture register' outputs (i.e., the random pattern) are finally stored into update registers when the update clock $\mathbf{Upd_{cki}}$ arrives. In the second *phase*, the capture registers become shift registers with **LFSR/shift_mode** set to "0" and **boundary_scan_in** is set to "0". Also, **Coupling_test_mode** and **test_mode** are set to "1" in this *phase*. The 3-to-1 MUX will select **Sq** as its input when the corresponding Cap's output is "1". The truth tables for the 2-to-1 MUX and the 3-to-1 MUX are also shown in Figure 3-4.When **scan_in/reload** is set to "1" and **LFSR/shift_mode** is set to "0", the modified boundary scan-in cells still support the original boundary scan-in function.



(a)

scan_in/reload	2 to 1 MUX
0	3 to 1 MUX
1	previous Cap

test_mode, Cap coupling_test_mode	3 to 1 MUX
0xx	From PIN
10x	Update
110	Update
111	Sq

(b)

Figure 3-4. (a) The modified 4-bit boundary scan-in cells and (b) truth tables for the 2-to-1 MUX and 3-to-1 MUX.

When **scan_in/reload** is set to "1" and **LFSR/shift_mode** is set to "0", the modified boundary scan-in cells still support the original boundary scan-in function.

3.4.2 Boundary Scan-out Cell

Figure 3-5 shows the detail design of the boundary scan-out cell for the ***detector block***. In the figure, **EDR** is detector clock and error detector reset. The error detector is reset to "0" by **EDR** which was set to "1" in the first *phase* of the circuit operation. As described previously, in the second *phase*, **DMLE** was set to "1", **Sq** is applied a sequence of "1" and "0" alternatively to determine the sensitized output, and a pulse arrives at $\mathbf{Cap_{cko}}$ to capture the first "1" output to be compared with the second "0" output which appears at line A by $\mathbf{Xor_1}$. Finally, **DMLE** returns to "0". In the third *phase*, $\mathbf{Cap_{cko}}$, update clock of boundary scan out cell ($\mathbf{Upd_{cko}}$), and $\mathbf{Det_{ck}}$ are clocked synchronously with the test clock. The outputs are latched at the **Cap** and **Update** registers in the normal operation speed in this phase. Two consequent outputs are compared by $\mathbf{Xor_1}$ which gives a "1" if these two outputs differ from each other. The detection_mode latch, **DM**, which is controlled by **DMLE**, is shown in Figure

3-6. The error detector circuit is shown in Figure 3-7, which samples its input F when **Det_{ck}** becomes “0”.

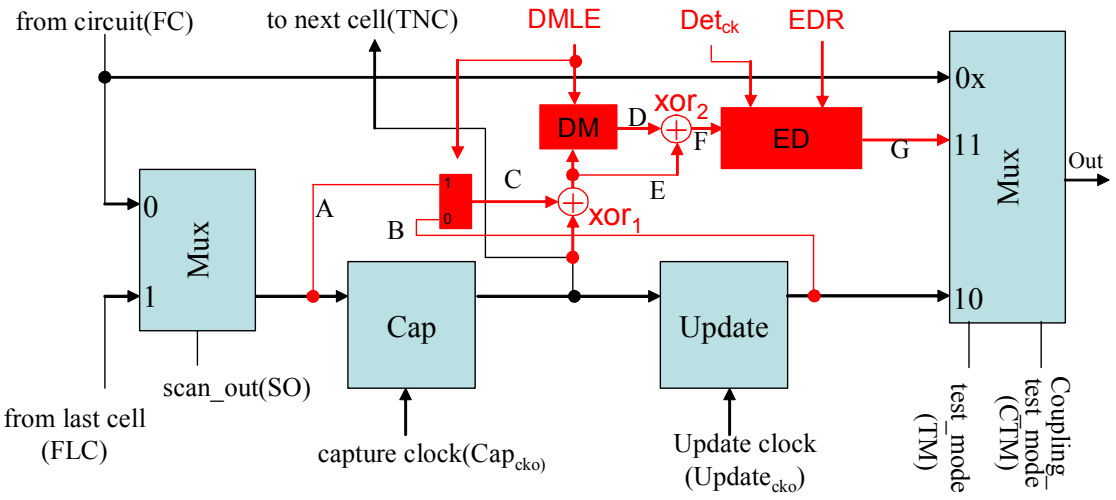


Figure 3-5. The modified boundary scan-out cell.

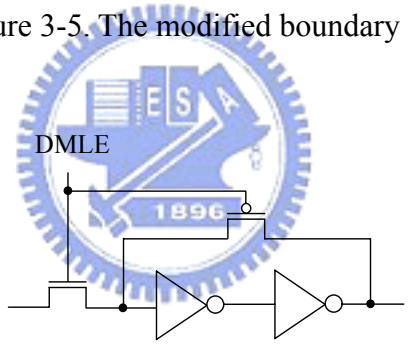


Figure 3-6. Detection_mode latch.

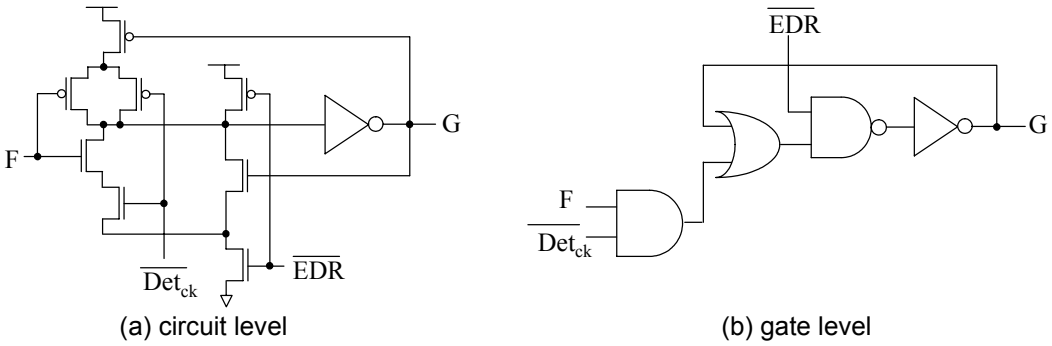


Figure 3-7. The error detector circuit. Line F is sampled when **Det_{ck}** is “0”.

3.5 Fault Coverage Estimation through Fault Simulation

For this test scheme, to evaluate the fault coverage of the randomly generated patterns, fault simulation needs to be done.

3.5.1 Symbol Definition and Operation

To perform fault simulation, a set of symbols are first introduced and defined and a set of operation rules on them are defined for each type of gates. The set of symbols are shown in Figure 2-2, where ***C***, and ***C**** are fault effect signals which are to be propagated and detected at the output, ***PF*** and ***PF**** are potential fault effects which could be ***1*** or ***C*** and ***0*** or ***C**** respectively, ***S*** and **S*** are oscillation squarewave signals which are to be applied at the fault site, and ***x*** represents for an "unknown" value. Figure 2-2(b) shows examples on how these symbols are operated through an **&** (**AND**) gate. For examples, ***C* & *C**** results in ***PF**** and ***C* & *C*** results in ***C***. Figure 2-2(c) shows the truth tables for **AND** and **OR** operation respectively. In the table, it can be seen that these sets of operations are rather conservative. For examples, when ***S*** operates on ***S*** (or ***S****), an ***x*** is given, which indicates an "unknown", while for practical cases, most often an ***S*** (or ***S****) is obtained.

3.5.2 Fault Simulation

For fault simulation, when a test pattern is applied to the circuit, a fault free logic is

firstly computed with the set of variables and operation rules defined above. Then a fault is injected to see if the aggressor line is at the ***S*** (oscillation) path. If the aggressor line is at the oscillation path, the victim line of the fault site is set to ***C***, and it is propagated toward the output of the circuit. If the propagation is successful, the crosstalk fault is detected by the pattern and another fault is injected. The above process is repeated until all faults are injected and another pattern is simulated.

3.6 Experimental Results

Firstly, the circuit level simulation for the proposed detection circuit was done. Figure 3-8 shows the simulation results of the modified boundary scan-out cell of Figure 3-5 under the patterns of Figure 3-3. In the figure, the clock signal **TCK**, the squarewave test signal **Sq**, the control signals **DMLE** and $\mathbf{Det_{ck}}$ for **DM** and **ED** respectively are displayed with the simulated signals at **FC** (the input of the cell where the induced glitches comes in), **F** (the output from XOR_2), and **Out** (the output of the detector of the boundary cell). When **DMLE** is "high", i.e., the circuit is at the *stable logic latch phase*, for which Det_{ck} is at "high", even there is an induced glitch at **FC** (at 3.3 ns) the **DM** does not detect the glitch and the output of the boundary cell, **Out** is "low". When DMLE is "low" (the *oscillation test phase*), if the glitch occurs at the time slot when $\mathbf{Det_{ck}}$ is "low" (at 7 ns), it is detected by the circuit and **Out** is "high".

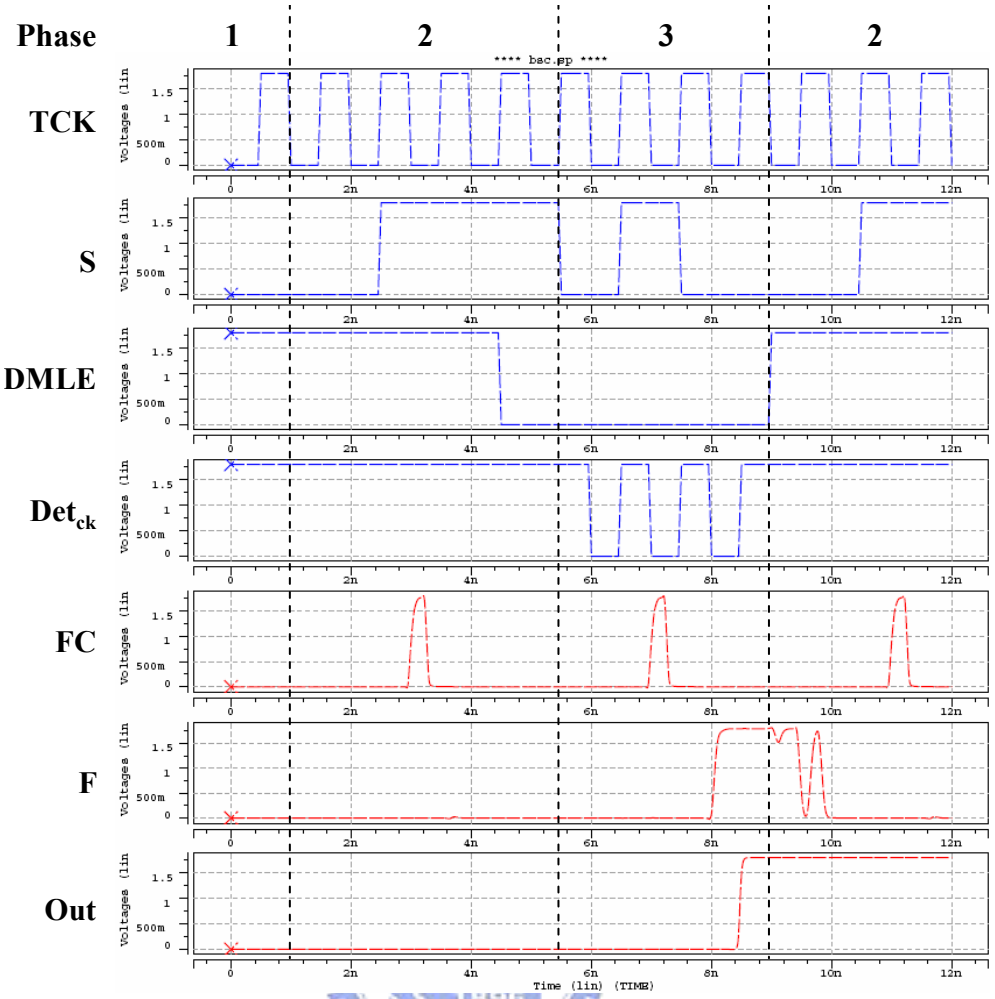


Figure 3-8. Hspice simulation waveforms of modified boundary scan-out cell in Figure 3-5.

To estimate the efficiency of random test patterns to detect crosstalk faults, we did experiments on randomly generating patterns to apply to benchmark circuits [22] for which crosstalk faults are randomly injected to be detected by the patterns. A fault simulator based on the symbols and operation rules in the previous section was implemented to evaluate the fault coverage of the patterns. In the experiment, for each circuit, two different non-overlapped separate sets of random crosstalk faults were generated, and the number of faults is twice the number of gates. For each fault set, two separate sets, each with different seed for pattern generation, of random patterns were generated. Figure 3-9 shows results for

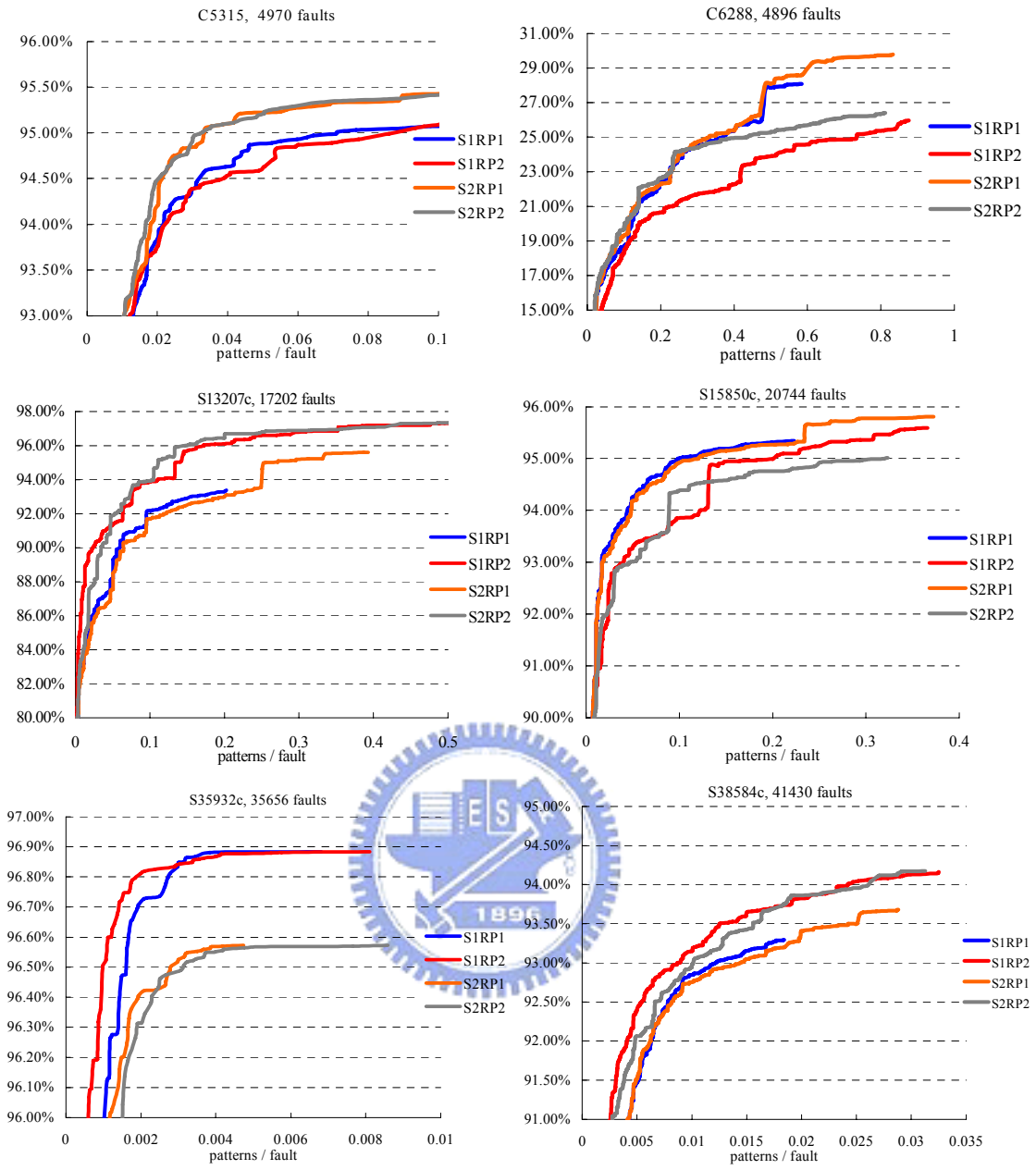


Figure 3-9. The fault coverages versus the number of patterns/number of faults for circuits C5315, C6288, S13207c, S15830c, S35932c, and S38584 respectively.

some benchmark circuits (C2670, C5315, C6288, S13207c, S15830c, and S35932c, where c represents for the combinational part of the circuit) where fault coverage curves are plotted with respect to the number of patterns/fault. In the figure, S1 and S2 represent for two different fault sets and RP1 and RP2 represent for two different sets of random patterns. It can be seen that, similar to the stuck-at fault random testing, the fault coverage increases fast

initially for some initial number of patterns and when it reach a "*saturated*" number it then increases slowly afterward. The *saturated* fault coverage depends strongly on the type of the circuit.

For examples, for circuits: C5315, S13207c, S15830c, and S35932c, which mainly are circuits of larger sizes, it can easily reach 90%, but for circuit C6288, which is a multiplier, it can only reach 30%. Also, since the faults were randomly selected, some of them may be untestable faults. The fault coverage also depends somewhat on the set of random test patterns which are generated with different seeds of LFSRs. However, the difference of the fault coverage between two sets of patterns is generally small, which is within 2%. This means that for an arbitrary set of random patterns, it can detect crosstalk fault quite efficiently. This makes this BIST scheme very attractive in the practical application.

Table 3-1 compiles more detail simulation results of some benchmark circuits, in addition to the above four circuits. In the table, similarly, for each circuit, the number of faults is twice the number of gates of the circuit and the faults were randomly selected. The column of "**Saturated Pattern**" is the number of random patterns which were generated by the software LFSR when the fault coverage reaches the "*saturated*" value, which is shown in the column " **Saturated Fault Coverage**". The "**Potential Detectable Fault Coverage**" is the fault coverage for which the faults were potentially detected when one (or more) ***PF*** or ***PF***** is detected at the circuit output(s). In the table, it can be seen that for most of circuits, the

saturated fault coverage can be reached over 90% with a number of patterns considerable less than the number of faults.

Table 3-1. Fault simulation results of the proposed BIST scheme for crosstalk faults.

Ckt Name	Total PI	Total Fault	Sturated Patterns	Saturated Fault Coverage(%)	Potential Detectable Fault Coverage (%)
c2670	157	2700	2700	69.07	7.11
c3540	50	3438	3426	83.33	8.20
c5315	178	4970	4899	94.81	2.90
c6288	32	4896	3741	28.82	13.01
c7552	206	7436	3896	90.14	3.28
s3271c	142	3428	2430	97.52	0.90
s3330c	172	3922	3879	79.40	0.33
s3384c	209	3788	2352	93.98	0.37
s4863c	153	4990	4115	94.71	0.54
s5378c	214	5986	3918	97.24	0.60
s6669c	312	6784	1643	98.28	0.25
s9234c	247	11688	1718	83.10	0.48
s13207c	650	17202	436	92.34	0.11
s15850c	600	20744	419	92.98	0.10
s35932c	1763	35656	87	96.85	0.44
s38584c	1462	41430	94	90.61	1.51

3.7 Summary

In this chapter, a BIST scheme to detect crosstalk faults in the boundary scan environment for deep sub-micron VLSI is proposed. The scheme uses an oscillating squarewave signal, which can be easily either generated locally or supplied externally, to test crosstalk faults of the circuit if they possibly exist, with no need of any external ATE. It needs only minor modification on boundary scan cells, which are transformed into a signal generator and some simple detection circuit to detect pulses which are induced by crosstalk faults of the circuit. Simulation results show that the scheme can detect most of the crosstalk

faults, up to a fault coverage over 90%, with a small number of random patterns for most of large size of circuits.

Chapter 4

Crosstalk Fault Detection for Interconnection Lines Based on Path Delay Inertia Principle

4.1 Preliminary

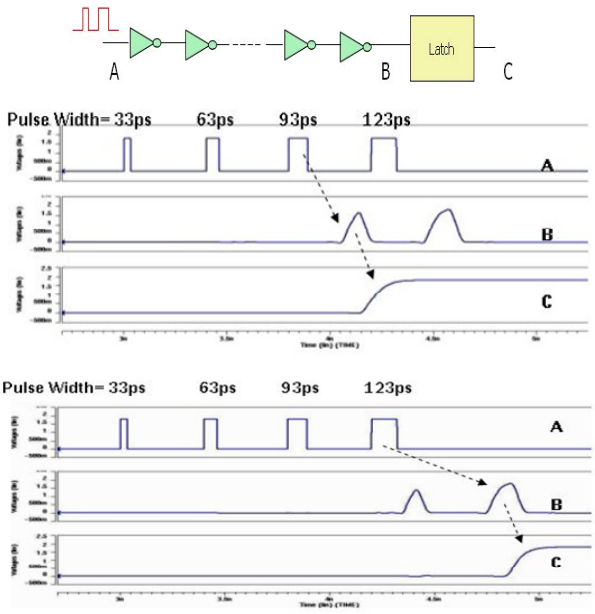
As mentioned in the Introduction chapter, as the process technology enters the deep sub-micron SoC era, asides the conventional stuck-at fault testing, delay and crosstalk of a circuit are two important issues to be tested to guarantee the performance of a semiconductor chip. For the delay testing, it is to detect timing defects to ensure that the manufactured chips meet the desired timing specifications. For the crosstalk testing, it is to test if the desired function of a circuit goes erroneously or even corrupted due to interference between internal lines or nodes within the circuit when the circuit operates at the specified frequency. Much research had been dedicated to topics of testing these two types of faults [10, 14, 22, 28, 33, 41, 43, 49, 58, 62, 76-88].

For delay fault testing, fault models were first issue considered [76]. Among many delay fault models, the path delay fault model is considered to be the most realistic model since it covers cumulative effects of defects occurring on a path, either on devices or interconnection lines, caused by wafers themselves or various process misalignments. Test generation and fault simulation targeting for this fault model had been extensively investigated [76-85]. For

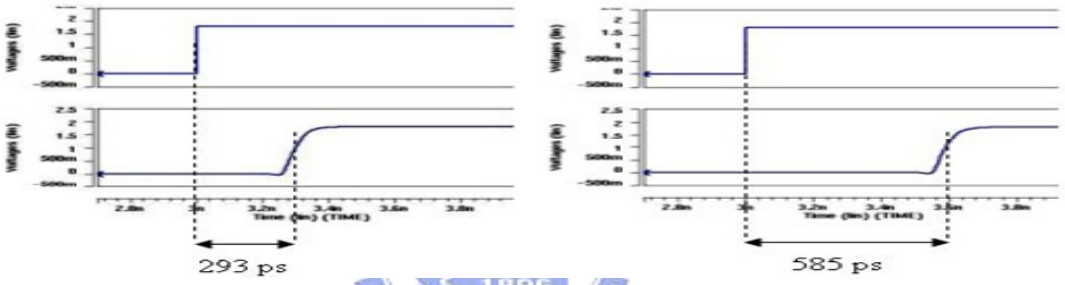
examples, Smith, in addition to first addressing the path delay fault, presented a framework for its fault simulation [76]. Test generation for robust path delay faults was addressed by Lin and Reddy [77], and also by Hsieh, et al [78]. An ATPG system for path delay fault was proposed in [79]. Pseudo-random test generation approach was also investigated [80]. Simulation issues on delay faults were discussed in [81]. An approach by formulating the delay testing problem as an energy minimization problem was also proposed in [82]. Statistical issues and approaches on detecting delay faults were also presented [83-85]. For example, in [83], a technique to statistically estimate path delay fault coverage is presented. In [840], a framework to diagnose parametric path delay faults based on statistical timing analysis is proposed, and in [85], a statistically timing analysis based on Monte Carlo simulation is presented. In all the approaches of testing the path delay fault, two patterns are needed, i.e., the first pattern is to initialize the path and the second pattern is to activate the path and propagate the fault effect to be observed at the output. Also, very often a special timing consideration should be taken in applying the above two patterns.

In [94], an innovative scheme was proposed and demonstrated to detect path delay faults. The scheme, instead of applying a two-vector pattern, propagating it along the path-under-detect (PUD) and observing the transition at the output of the PUD at the specified time, applies a pulse to the input of the PUD. The pulse has a critical width which contains just enough energy to conquer the path delay inertia to be propagated to the output of the PUD.

Since the path delay of the PUD is proportional to the path delay inertia of the PUD, any erratically longer delay on the PUD will increase the path delay inertia of the PUD. As a result, the pulse can not be propagated to and detected at the output of the PUD. Hence, by detecting the pulse of the critical width, we can tell if the PUD has a path delay exceeding the specified value. Figure 4-1 is an inverter chain example of the test scheme. The simulation waveforms at the nodes B and C, which are the nodes before and after the connected latch, of an inverter chain for which several pulses of different widths are applied. In the figures, the horizontal axis is time and the vertical axis is voltage amplitude of waveforms. The pulses, as propagating along the chain, deteriorate gradually. For the pulses of their widths are not large enough, they deteriorate to a degree that they can not trigger the latch (pulse widths: 33 and 63 ps). For the pulse of the width of 123 ps, it can propagate through the inverter chain of 20 stages. Since the delay of the chain is directly related to the length of the chain (Figure 4-1(b)), by controlling the width of the applied pulse and observing the triggering of the latch, one can tell if the path delay exceeds the specified value. The scheme is very simple in detecting the delay faults of a circuit since it eliminates the complicated timing issues in generating and applying two-vector patterns.



(a)



(b)

Figure 4-1.(a) Two inverter chains of 10 stages and 20 stages applied with input pulses of different widths and the simulated waveforms at nodes B and C; (b) The propagation delays of two inverter chains. The horizontal axis is time and the vertical axis is the amplitude voltage of the waveforms

In this chapter, we extend the above mentioned scheme to detect the crosstalk faults of interconnects of SoC. This makes the detection of crosstalk faults be simple and effective. In the text which follows, we first describe the basic idea how the scheme can be extended to detect the crosstalk fault in section 4-2; then we present the sensitivity analysis of the scheme in section 4-3; and then present the experimental results of the scheme in section 4-4.

4.2 Basic Idea of the Detection Scheme

As mentioned previously, the scheme is based on the path delay inertia principle. In the following, a transmission line model is utilized to explain the working principle of this scheme.

Figure 4-2 shows two transmission lines, one of which is the aggressor line and the other is the victim line and there is a coupling capacitance, i.e., coupling fault, in-between these two lines. In the figure, R_s and C_s are distributed intrinsic resistance and capacitance of transmission lines respectively, C_c is the intrinsic distributed coupling capacitance between two interconnection lines and C_f is the coupling capacitance which causes the crosstalk fault. For the victim line, there is a pulse detector connected at the output. As the victim line is applied a pulse with large enough width, the pulse will be propagated along the line and be detected by the pulse detector. However, if the width of the pulse is not large enough, i.e., it contains not enough energy to "conquer" the path delay inertia of the victim line, the pulse will not be detected by the detector. The critical width of the pulse is called "critical width", CW, and the pulse of the critical width is called "critical width pulse", CWP. Figure 4-3 shows the SPICE simulation of two pulses applied at the input of the victim line (Figure 4-3(a)), where P_a has a pulse width smaller than the CW and P_b has a width larger than CW. The P_a pulse, when propagating to the output of the line, has a degraded shape (Figure 4-3(b)) and

not able to trigger the detector (Figure 4-3 (c)), while P_b can still maintain its shape, i.e., contains enough energy to trigger the detector.

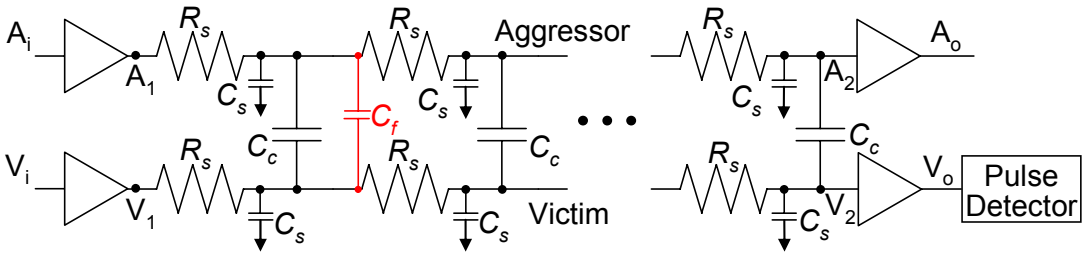


Figure 4-2. The transmission lines model used to explain the technique, where one is the aggressor line and the other is the victim line, and C_f is the crosstalk capacitance which causes the crosstalk fault.

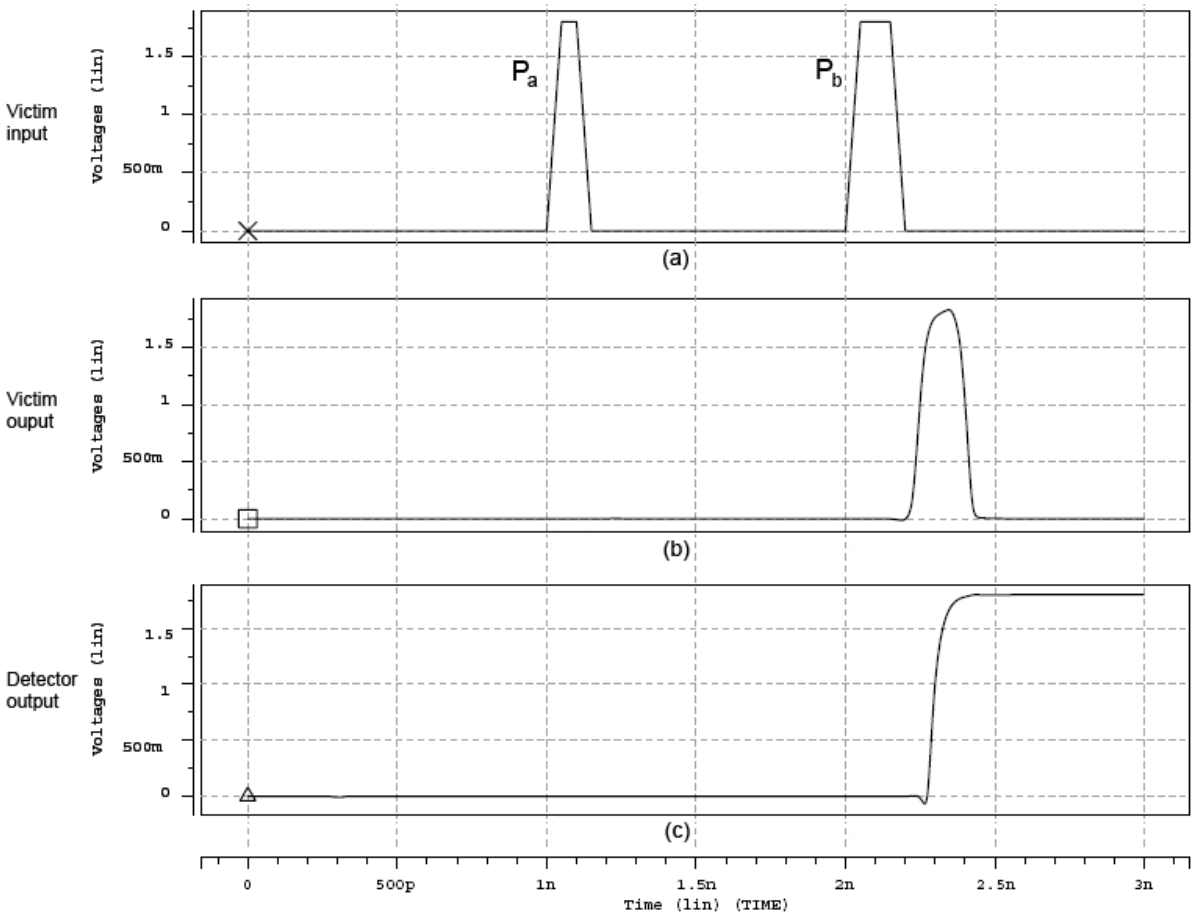


Figure 4-3. The SPICE simulation waveforms of two pulses, P_a and P_b, of various widths (a) at the input and (b) at the output of the victim line and (c) at the output of the detector.

Now, in figure 4-2, as the victim line has been applied a CWP pulse which causes triggering of the pulse detector, but a transition signal which transits in the opposite polarity of the CWP pulse is applied at the aggressor line. This transition signal will affect the CWP pulse in shape. Figure 4-4 shows the SPICE simulation waveforms of the interconnection lines of Figure 4-2 with a transition wave applied at the aggressor line of node A_1 (Figure 4-4(a)). The CWP pulses arrive at the node V_2 of victim line (Figure 4-4(b)) and the output of the detector (Figure 4-4(c)) are also shown. In Figure 4-4 (b), there are two CWP pulses applied at two different time. For the CWP pulse where there is no transition wave applied at the aggressor line, it maintains its original shape, but for the CWP where a transition wave applied at the aggressor line at the same time, its waveform is the superposition of its original wave and the induced glitch caused by the aggressor line transition wave through the crosstalk capacitance [14]. This affected CWP pulse can not trigger the detector anymore. The larger C_f, the more the CWP is affected. To make the CWP able to trigger the detector again, it needs to increase the width of the CWP. That is, the presence of C_f and the applied transition testing pattern at the aggressor line have changed the CW of the CWP of the victim line! Hence, by applying a CWP to the victim line and an opposite-polarity transition pattern to the aggressor line, we can detect the crosstalk fault by observing the switching of the pulse detector.

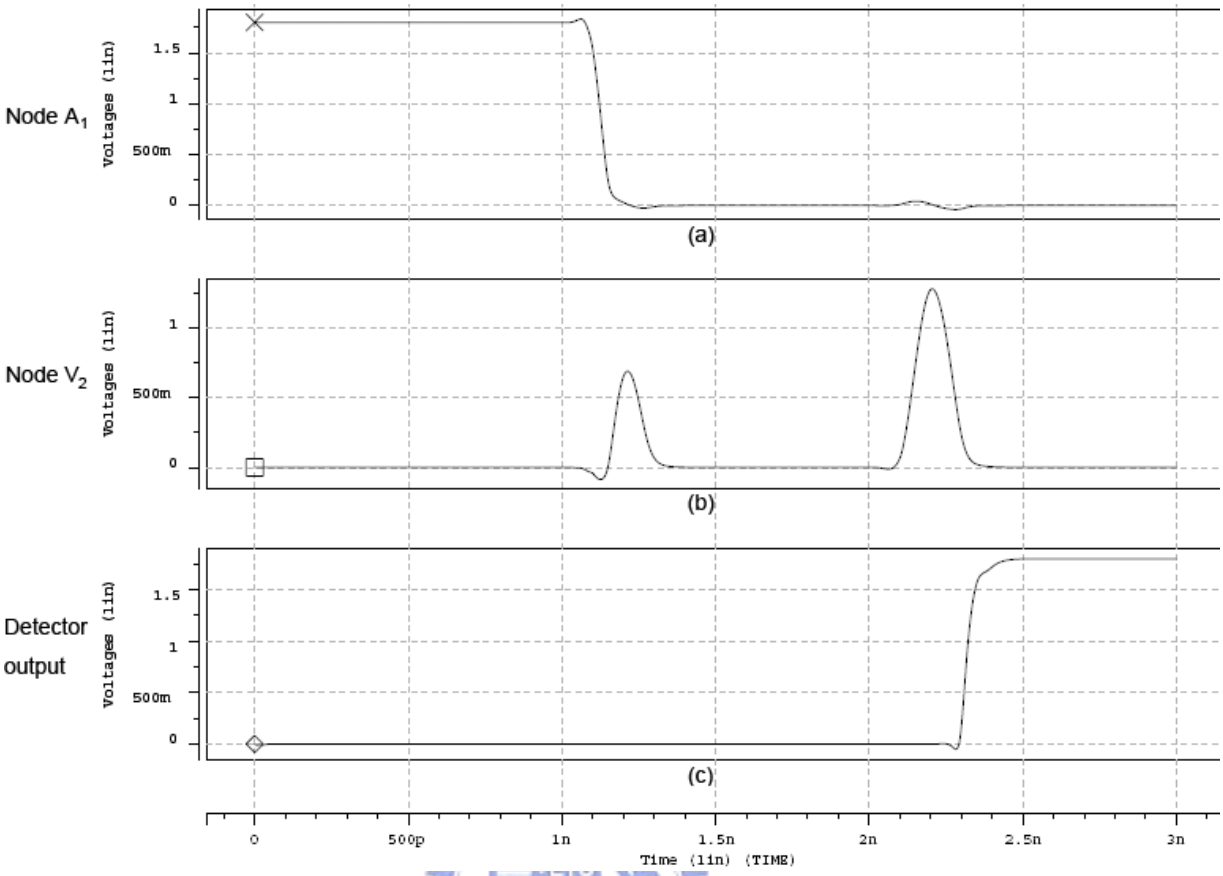


Figure 4-4. The SPICE simulation waveforms of (a) the applied transition wave at the node A_1 of the aggressor line, (b) the affected CWP pulses of the victim line at the node V_2, and (c) the output of the detector.

4.3 Analysis of the Scheme via Simulation

This section shows the analysis of the scheme via SPICE simulation for an interconnection line pair. The parameters for the transmission line model are: transmission lines are 400um long, 0.23um wide with a 0.23um line spacing, R_s = 7.8*10^{-2} Ω/□, C_s = 1.987*10^{-2} fF/um and C_c = 1.05*10^{-1} fF/um.

4.3.1 Relationship of CW of Pulse versus the Crosstalk Fault

It is first to investigate the relationship between the CW of the CWP with respect to the

magnitude of the crosstalk fault. The simulation is for both cases that the victim line is applied a “1” pulse (P1) and a “0” pulse (P0) respectively and the aggressor line is applied an opposite transition wave respectively. Figure 4-5 shows the results where the CW of the pulses is plotted in terms of the magnitude of the crosstalk fault, C_f. It is seen that the relationship is linear. The $C_f = 0$ corresponds to the case that there is no crosstalk fault coupling between the aggressor line and the victim line. For example, to detect a crosstalk fault of 45 fF, which is about the value of the intrinsic coupling capacitance between the two interconnection lines, the P1 CWP to be applied to the input needs to have a CW of 0.228 ns as compared to the original CW = 0.161 ns for which there is no crosstalk fault. The results for P1 and P0 are somewhat different. For the P0 case, it needs a larger CWP than that of the P1 case, but the slope, the P1 case (1.48ps/fF) has a higher value than that of the P0 case (0.83ps/fF). This is because, for the P1 pulse, the rising edge of the pulse needs the pMOS of the driver of the interconnection line to charge the line but the pMOS has a less driving ability than that of the corresponding nMOS of the driver. To facilitate the crosstalk fault testing, the higher sensitivity pattern, i.e., P1, is preferred since this makes differentiate crosstalk easy.

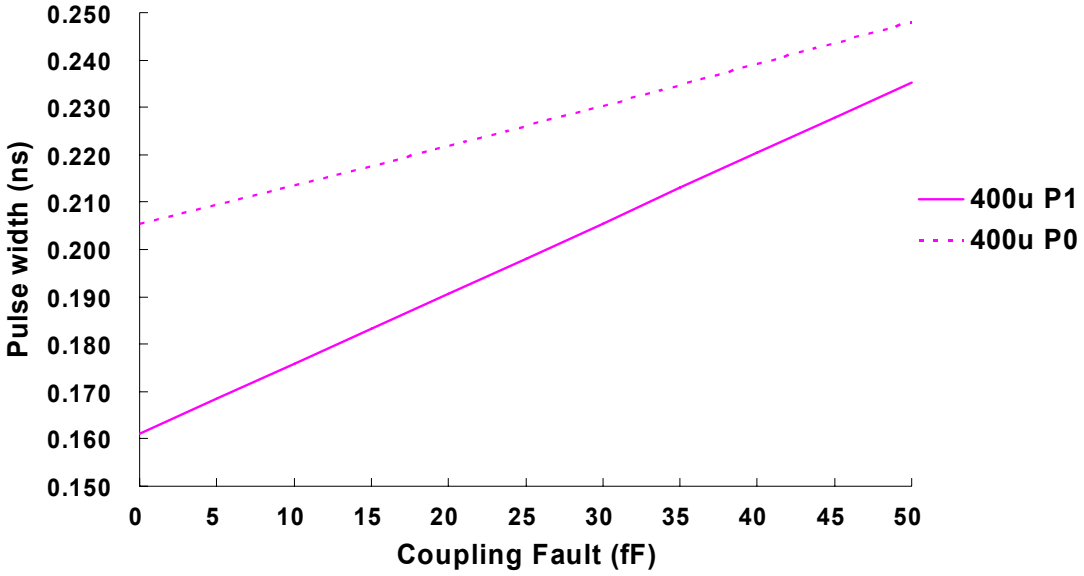


Figure 4-5. The CW plotted in terms of the crosstalk fault, C_f, for the cases of P1 and P0. The slope for the P1 case is 1.48 ps/fF and that for the P0 case is 0.83 ps/fF.

Figure 4-6 plots the same CW curves with respect to C_f for P1 and P0 cases for several different lengths of interconnection lines. It is seen that the longer interconnection lines, the larger CWP's, but the slope is the same.

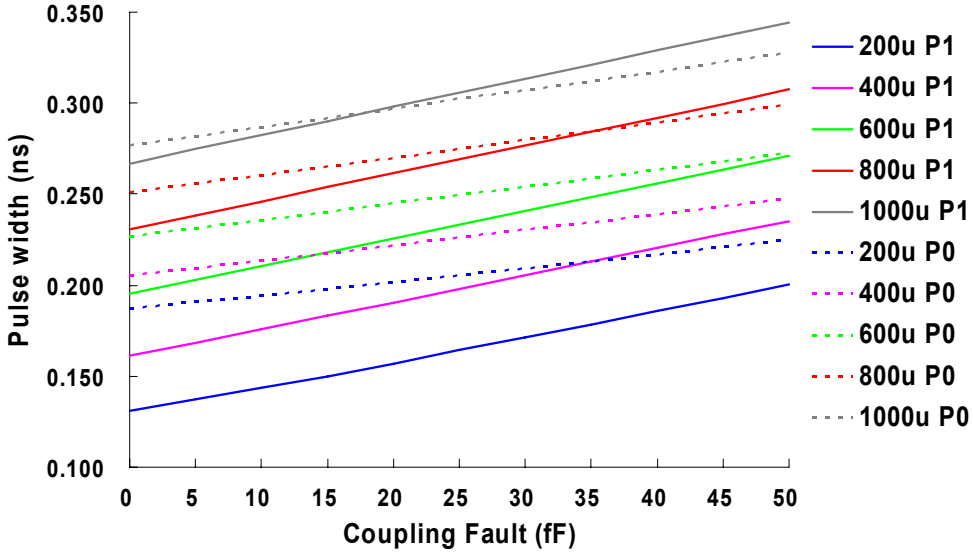


Figure 4-6. The CW plotted in terms of C_f for the cases of P1 and P0 for several different lengths of interconnection lines.

4.3.2 Relationship between Induced Victim Line Delay and Crosstalk Fault

Crosstalk faults induce delay on the victim line when the aggressor line is applied an opposite-polarity going transition wave [10, 33, 73, 79, 88]. The relationship between the induced delay and the magnitude of the crosstalk fault of the system in Figure 4-2 is also investigated. Figure 4-7 shows the results for the P1 case for several lengths of the interconnection line. The relationship is also linear as it is apparent to see since the system in Figure 4-2 is a linear system. In the figure, the points $C_f = 0$ correspond to the case that no crosstalk faults exist and the delays are the original delays of each interconnection line. For example, for the 400 um interconnection line, the delay on the victim line without the crosstalk fault is about 0.284 ns, and for a 45 fF crosstalk fault, it will increase to 0.330 ns.

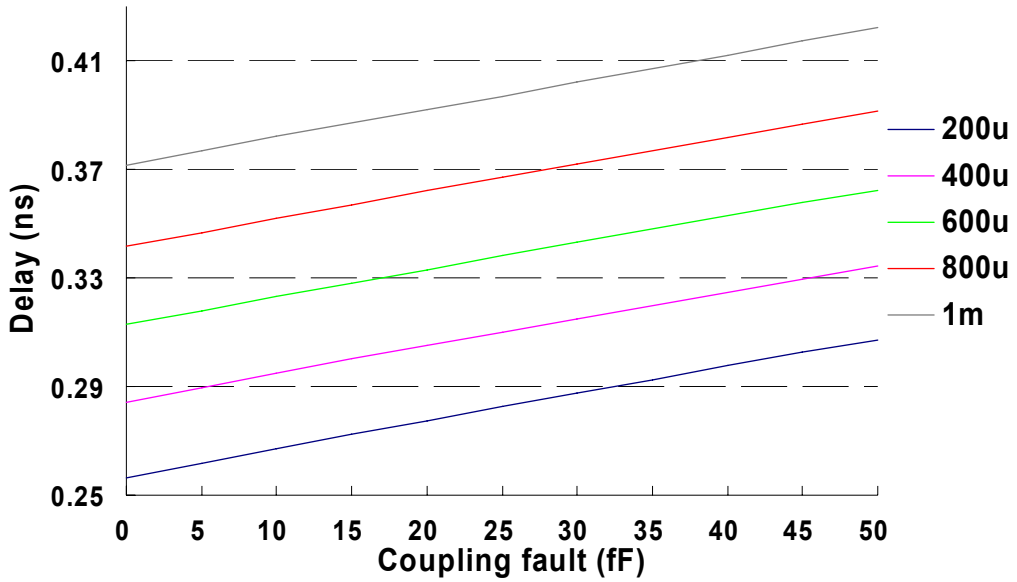


Figure 4-7. The simulated induced delay on the victim line of different lengths in terms of the magnitude of the crosstalk fault when the aggressor line is applied an opposite-polarity going transition.

The slope of each line in the figure is about 1.02ps/fF. Combing this value with that of the P1 case, which is 1.48 ps/fF, of Figure 4-5, we can derive the sensitivity of the CW of the CWP applied to the victim line for crosstalk fault testing with respect to the induced delay caused by the aggressor line to be 1.45. This indicates that, to detect 1 ps induced delay caused by the aggressor line to the victim line, it needs to extend 1.45 ps of the original CW of the applied CWP. Hence if the specified allowable crosstalk induced delay is 0.1 ns, additional 0.145 ns needs to be put to the CWP to detect the fault.

Figure 4-8 plots the above mentioned relationship of the crosstalk fault induced delay with respect to the CW for the P1 case for several different lengths of interconnection lines. The relationship basically is linear and the slope is also almost the same for all the lengths. The inverse of the slope is approximately 1.46 as obtained in the above paragraph.

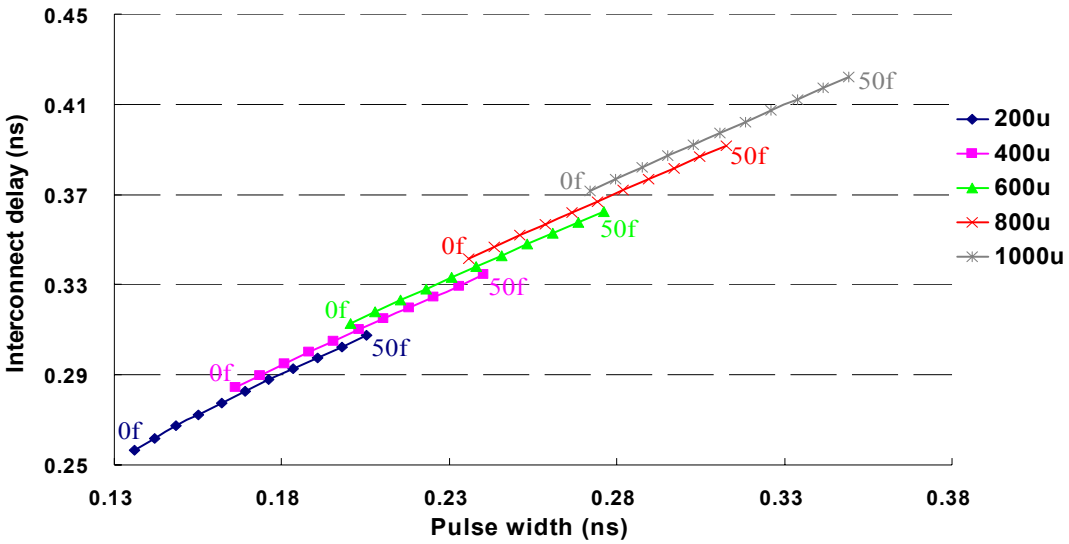


Figure 4-8. The relationship between the induced delays caused by crosstalk faults with respect to the required minimum pulse width, CW, which need to detect crosstalk faults.

4.3.3 The Effect of the Signal Skew between Aggressor and Victim Lines

To test the crosstalk fault, one difficulty is that the signals applied to the aggressor line and the victim line should be in coincidence. If two signals are separated too far away, the induced delay fault effect will disappear. The same difficulty exists for this proposed scheme. In the above, the CW to detect the crosstalk fault was derived under the condition that two waves applied to the aggressor and the victim lines are in coincidence. If the skew between two signals is considered, the CW value may have some variation. Figure 4-9 shows the simulated CW variation (in percentage with respect to the CW when no crosstalk fault exists) in terms of the skew between the aggressor's transition time and the applied P1 of the victim line. On the horizontal axis, 0 coordinate corresponds to the case that there is no skew between the two signals, the negative value represents that the aggressor's signal leads the victim's signal and the positive value represents that the aggressor's signal lags the victim's signal. The interconnection line length is 400um. In the figure, it can be seen that the maximum CW occurs at the position that the aggressor line's signal leads the victim line's signal by 0.05ns to 0.2ns. This result is in agreement with that obtained in [13, 91]. The physical meaning for this result is that for the CW derived to detect a crosstalk fault, for example, C_f = 25 fF, a CWP of 42% more of CW than the fault-free CW should be applied to the victim line, but this may not detect the fault if the aggressor line's signal is 0.05 ns lagging, or 0.1 ns leading, than that of the victim line. This causes an "Escape"

during testing. Furthermore, the CWP of this CW will detect the fault less than 25 fF, for example, 20 fF, if the signal of the aggressor line is leading the victim line in a range of 0.03 ~ 0.06ns. This causes an "Overkill" during the testing. This is a drawback of this scheme. In the next section, we present the experimental results to show how seriously this skew problem affects the testing result of this scheme.

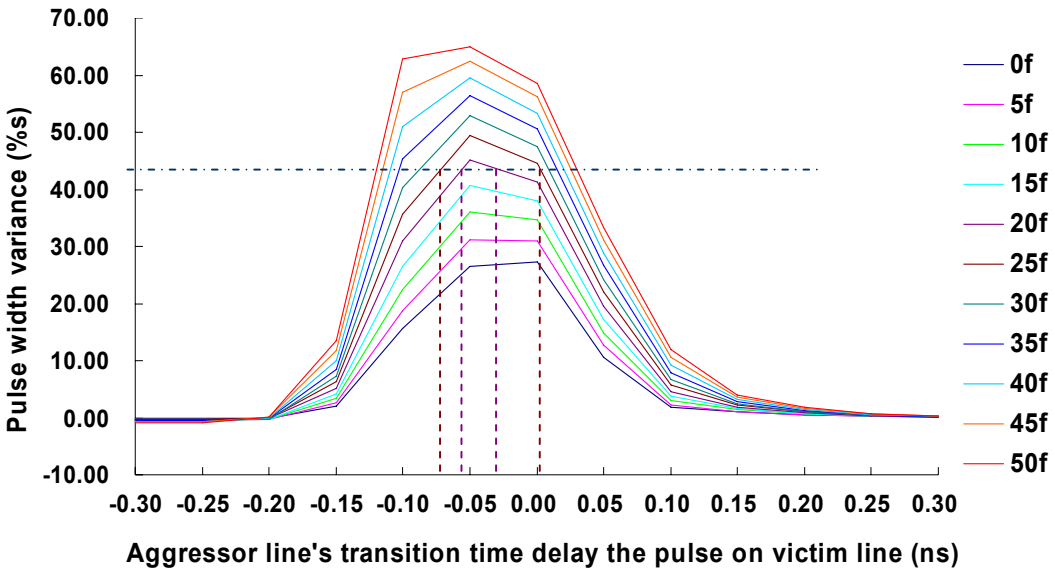


Figure 4-9. The CW plots with respect to the skew between the aggressor line's signal and the victim line's signal for different magnitudes of crosstalk faults.

4.4 Experimental Results

In this experiment, a pair of interconnection lines of Figure 4-2 was set up, where the skew between the applied signals of the aggressor line and the victim line was considered. Two-interconnection lines model was used because multiple-aggressor lines can be lumped into a single aggressor line with a stronger driving capability [92]. In addition, the process

variations on each component, i.e., widths and heights of interconnection lines and spacing between interconnection lines, are also considered. Monte Carlo simulation was done by applying signals on the aggressor line and the victim line respectively and the signal at the output of the victim line was captured by the pulse detector to see if the propagated signal triggers the detector. The above parameters were selected randomly allowing their values varying around their nominal values in 3σ =20% respectively. For each length of lines, 1000 samples were simulated. The results are shown in Table 4-1 and Table 4-2 for the cases of "without signal skew" and "with signal skew" respectively. For Table 4-2, the skew between the aggressor line and the victim line was selected randomly in a normal distribution with 3σ = 0.05ns. In the table, "Detection Quality" is the ratio of the cases that when a CWP was applied, it detected the injected crosstalk fault even when the selected parameters had 3σ =20% of variation with respect to their nominal values to the total number of simulation. "Escape" is the case that when the applied CWP was detected by the detector, i.e., the interconnection lines are considered to be free of the injected crosstalk fault, but the induced delay caused by the injected fault exceeds the specified fault free value. "Overkill" is the case that when the applied CWP was not detected by the detector, i.e., the interconnection lines are considered to be faulty under the injected crosstalk fault, but the induced delay caused by the injected fault is within the specified fault free value.

The tables are listed for five pairs of different lengths of interconnection lines and the

results are listed for 12 different magnitudes of crosstalk faults (including the fault free case, i.e., $C_f = 0$). For Table 4-1 of the "without signal skew" cases, it can be seen that the scheme exhibits a fairly good testing accuracy, i.e., for most of injected faults, "Detection Quality" are 100% except for a few cases that the magnitude of injected faults is small as compared to the total capacitances, i.e., C_s and C_c, of interconnection lines. Also, it is observed that for those cases that "Detection Quality' not equal to 100%, the "Overkill" percentage are always larger than that of the "Escape". That means that to decrease this value, it can enlarge the CW of the CWP somewhat when applying this scheme.

For Table 4-2 of the "with signal skew" cases, "Detection Quality" deteriorates significantly for those injected faults whose magnitudes are less than 20fF. This indicates that, to facilitate this test scheme, the application of two signals on the two interconnection lines should be sought to be in coincidence as possible. Also, for this table, the "Escape" data are much larger than the "Overkill" data. This is because in Figure 4-9, the regions which cause the "Escape" cases are much larger than the region of "Overkill". It is also seen that as the size of the crosstalk fault increases, "Detection Quality" increases steadily. For the injected faults larger than 45 fF, "Detection Quality" can generally reach over 99% .

Table 4-1. Experimental data for the testing scheme for cases of "without signal skew".

line length	Fault free delay (ns)	Signals	No signal skew										
		Coupling fault (fF)	0	5	10	15	20	25	30	35	40	45	50
200	0.224	Over kill (%)	1.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Detection Quality	98.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
400	0.264	Over kill (%)	2.2	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Detection Quality	97.6	99.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
600	0.304	Over kill (%)	1.2	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Detection Quality	98.5	99.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
800	0.344	Over kill (%)	0.9	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	0.3	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Detection Quality	98.8	99.3	99.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
1000	0.384	Over kill (%)	0.7	0.6	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	0.2	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Detection Quality	99.1	99.0	99.8	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 4-2. Experimental data for the testing scheme for cases of "with signal skew".

line length	Fault free delay (ns)	Signals	With signal skew										
		Coupling fault (fF)	0	5	10	15	20	25	30	35	40	45	50
200	0.224	Over kill (%)	23.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	23.9	40.6	28.0	17.9	9.6	5.1	1.9	1.0	0.6	0.4	0.2
		Detection Quality	52.4	59.4	72.0	82.1	90.4	94.9	98.1	99.0	99.4	99.6	99.8
400	0.264	Over kill (%)	21.2	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	21.8	38.1	26.6	15.8	8.9	3.9	1.9	1.0	0.7	0.3	0.2
		Detection Quality	57.0	60.7	73.4	84.2	91.1	96.1	98.1	99.0	99.3	99.7	99.8
600	0.304	Over kill (%)	19.9	4.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	19.7	33.0	27.0	17.3	9.1	4.7	2.1	1.0	0.7	0.3	0.2
		Detection Quality	60.4	62.5	72.9	82.7	90.9	95.3	97.9	99.0	99.3	99.7	99.8
800	0.344	Over kill (%)	18.6	6.6	1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	18.6	28.8	25.9	16.9	9.9	4.9	2.4	1.2	0.7	0.4	0.2
		Detection Quality	62.8	64.6	73.0	83.1	90.1	95.1	97.6	98.8	99.3	99.6	99.8
1000	0.384	Over kill (%)	17.0	9.0	2.3	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		Escape (%)	17.3	24.0	24.6	16.9	10.6	5.3	2.6	1.2	0.6	0.4	0.2
		Detection Quality	65.7	67.0	73.1	82.7	89.4	94.7	97.4	98.8	99.4	99.6	99.8

4.5 Summary

In this chapter, we have proposed a test scheme to test the crosstalk fault between interconnection lines based on the path delay inertia principle. For this scheme, it does not need to measure the interconnection line delay with sophisticated circuit set-up but just applies a CWP of a designated width, CW, which was first computed (simulated) during the design stage under the assumption of known crosstalk fault (capacitance) between two interconnection lines. By detecting if the applied CWP is propagated to the output of the

victim line, it can tell if there exists the crosstalk fault of the assumed magnitude. The only drawback of the scheme is that the signals applied to the aggressor line and the victim line should be in exact coincidence. Even there is some skew between two applied signals, experimental results show that the scheme has high detection quality for testing the crosstalk fault if the magnitude of the crosstalk fault is comparable with that of the intrinsic coupling capacitance, C_c, of interconnection lines.

Chapter 5

Conclusions

As stated in previous chapters, the crosstalk inside the VLSI chip due to signal interaction between lines or gates of circuits become a more important issue for the today's deep submicron technology, and its testing is a difficult task due to its un-predictability. In this dissertation, we have proposed and studied two innovative testing schemes for testing the crosstalk faults in three topics. For the first topic, we make use of an oscillation signal as the test signal to test the crosstalk induced glitch fault. The scheme eliminates the complex timing issues in testing crosstalk as well as delay faults. For the second topic, the scheme is then further modified and developed to be applied to test the crosstalk glitch faults of SoC cores in the boundary scan environment. For the third topic, we propose another scheme to test the crosstalk induced delay faults between interconnects of SoC based on the path delay inertia principle. This test scheme is also very simple in setup and does not require to consider the complex timing issue of applying the two test patterns as in the conventional delay testing approaches.

For the first topic, which is Chapter 2, we study the scheme which utilizes an oscillation signal as an excitation which will induce glitches if there are crosstalk faults within the circuit. A set of symbols with operation algebra have been proposed for generation of test patterns.

The test pattern generations are implemented in two manners. The first manner is the deterministic ATPG which is to find test patterns for the specific crosstalk faults. It is based on the PODEM algorithm by applying the oscillation signal, which activates the aggressor line and propagates the fault effect to primary outputs. The fault simulation for the crosstalk fault in the ATPG was also studied. It categorizes fault syndromes while propagating the fault effects, so that we can verify the target fault to be detected or potentially detected. The second manner is a guided random pattern generation. The simulation shows that it obtains high fault coverage (>90%) for most benchmark circuits. It makes the BIST to be possible to test the crosstalk faults. The experimental results show that the test generation time in general is acceptable and the guided approach has less test generation time than does the conventional approach, especially for larger circuits which have large number of inputs. By combining both test generators, the test generation time can be reduced dramatically for those circuits with many XOR gates.

For Chapter 3, in order to implement the BIST for crosstalk faults detection, the boundary scan circuits are modified in second topic. By modifying the scan-in register with adding a few gates, we can perform the LFSR function for the BIST scheme required in phase 1. For those circuits with large inputs, we also divide the LFSR to several smaller LFSRs for the reasons of circuit complexity and pattern randomness. For the boundary scan-out register, we plan the Cap and Update FFs to store the stable outputs status of PO in phase 2, so that we

can recognize if the PO is under sensitized path or not. If they are consistent, it means that the PO is not under the sensitized path, the error detector is activated to detect the crosstalk fault in phase 3. The Hspice simulation shows that the design can detect the crosstalk fault successfully. We also study the test efficiency of the test scheme by randomly injecting crosstalk faults. The applied number of crosstalk faults is twice the number of gates, and the fault simulation shows the scheme can detect most of the crosstalk faults, up to a fault coverage over 90% for most of benchmark circuits.

In the third topic, which is Chapter 4, we study the crosstalk faults detection between interconnects. The scheme is based on the path delay fault inertia principle. By applying a pulse with a specified width and observing the output response, we can detect the fault between interconnects. Timing issues are not necessary to be considered for the test scheme. The experimental results show that the detection quality is high when no time skew at test signals and no process variation on interconnects happen. Even considering the time skew and the process variation, the scheme has high detection quality for testing the crosstalk fault if the magnitude of the crosstalk fault is comparable with that of the intrinsic coupling capacitance of interconnection lines.

So, in conclusion, this thesis has studied three subjects which are novel and different from the conventional approaches in detecting crosstalk faults. The results of the theoretical

study show that the techniques proposed are very promising and simple. However, to further demonstrate the feasibility of the techniques, they should be implemented in real chips and the actual testing should be performed. This can be regarded as the topics for future research.

In the following, we have some suggestions for future research: At first, try to investigate a circuit to realize the scheme of using the path delay inertia principle to detect crosstalk faults. Second, apply the above path delay inertia scheme to detection of crosstalk faults in logic circuits. Finally, implement a chip by intentionally injecting crosstalk faults and realize real test set-ups to test the faults to physically verify all the schemes.

Bibliography

[1] TSMC 0.18um mixed signal 1P6M salicide 1.8/3.3v RF SPICE models.

[2] International Technology Roadmap for Semiconductor, 2005.

[3] T. Xue, E. S. Kuh, and D. Wang, "Post Global Routing Crosstalk Risk Estimation and Reduction", *IEEE ICCAD*, pp. 302-309, 1996.

[4] T. Xue, E, S, Kuh, and D. Wang, "Post Global Routing Crosstalk Synthesis", *IEEE ICCAD*, pp. 1418-1430, 1997.

[5] T. Gao and C. L. Liu, "Minimum Crosstalk Channel Routing", *IEEE ICCAD*, pp. 465-474, 1996.

[6] J. Lou and W. Chen, "Crosstalk-Aware Placement", *IEEE DATC*, pp. 24-32, 2004.

[7] C. C. Chang and J. Cong, "Pseudopin Assignment with Crosstalk Noise Control", *IEEE TCAD*, pp. 598-611, 2001.

[8] M. A. Margolese and F. J. Ferguson, "Using Temporal Constraints for Eliminating Crosstalk Candidates for Design and Test", *IEEE VTS*, pp. 80-85, 1999.

[9] K. C. Hsu, Y. C. Lin P. X. Chiu, and T. M. Hsieh, "Minimum Crosstalk Channel Routing with Dogleg", *IEEE ISCAS*, pp. 73-76, 2000.

[10] W. Chen, S. K. Gupta, M. A. Breuer, "Analytic Model for Crosstalk Delay and Pulse Analysis under Non-Ideal Inputs", *IEEE ITC*, pp. 809-818, 1997.

[11] F. Moll, M. Roca, D. Marche, and A. Rubio, "Detectability of Spurious Signals with Limited Propagation in Combinational Circuits", *IEEE ATS*, pp. 176-181, 1994.

[12] Y. Sasaki, M. Sato, M. Kuramoto, F. Kikuchi, and T. Kawashima, "Crosstalk Delay Analysis of a 0.13-μm Node Test Chip and Precise Gate-Level Simulation Technology", *IEEE JSSC*, pp. 702-708, 2003.

[13] K. S. M. Li, C. L. Lee, C. Su, and J. E Chen, "A Unified Approach to Detecting Crosstalk Faults of Interconnects in Deep Sub-micron VLSI", *IEEE ATS*, pp. 145-150, 2004.

[14] P. B. Sabet, F. Ilponse, "A Model for Crosstalk Noise Evaluation in Deep Submicron Processes", *IEEE ISQED*, pp. 139-144, 2001.

[15] A. K. Palit, L. Wu, K. K. Duganapalli, W. Anheier, and J. Schloeffel, "A New, Flexible and Very Accurate Crosstalk Fault Model to Analyze the Effects of Coupling Noise between the Interconnects on Signal Integrity Losses in Deep Submicron Chips", *IEEE ATS*, pp. 22-27, 2005.

[16] K. Agarwal, D. Sylvester, and D. Blaauw, "Modeling and Analysis of Crosstalk Noise in Coupled RLC Interconnects", *IEEE TCAD*, pp. 892-901, 2006.

[17] Y. Eo, W. R. Eisenstadt, J. Y. Jeong, and O. K. Kwon, "A New On-Chip Interconnect Crosstalk Model and Experimental Verification for CMOS VLSI Circuit Design", *IEEE Trans. on Electron Devices*, pp. 129-140, 2000.

[18] S. Kundu, S. T. Zachariah, Y. S. Chang, and C. Tirumurti, “On Modeling Crosstalk Faults”, *IEEE TCAD*, pp. 1909-1915, 2005.

[19] L. Ding, D. Blaauw, and P. Mazumder, “Accurate Crosstalk Noise Modeling for Early Signal Integrity Analysis”, *IEEE TCAD*, pp. 627-634, 2003.

[20] S. Chary and M. L. Bushnell, “Analog Macromodeling for Combined Resistive Vias, Resistive Bridges, and Capacitive Crosstalk Delay Faults”, *IEEE VLSID*, Jan., 2006.

[21] W. Sirisaengtaksin and S. K. Gupta, “Enhanced Crosstalk Fault Model and Methodology to Generate Tests for Arbitrary Inter-core Interconnect Topology”, *IEEE ATS*, pp. 163-169, 2002.

[22] M. Cuviello, S. Dey, X. Bai. Y. Zhao, “Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects”, *IEEE ICCAD*, pp 297-303, 1999.

[23] S. Nazarian, H. Huang, S. Natarajan, S. K. Gupta, and M. A. Breuer, “XIDEN: Crosstalk Target Identification Framework”, *IEEE ITC*, pp. 365-374, 2002.

[24] M. A. Breuer, S. K. Gupta, and S. Nazarian, “Efficient Identification of Crosstalk Induced Slowdown Targets”, *IEEE ATS*, pp. 124-131, 2004.

[25] K. J. Keller and K. K. Saluja, “On Reducing the Target Fault List of Crosstalk-Induced Delay Faults in Synchronous Sequential Circuits”, *IEEE ITC*, pp. 568-577, 2001.

[26] H. Takahashi, K. J. Leller, K. T. Le, K. K. Saluja, and Y. Takamatsu, “A Method for Reducing the Target Fault List of Crosstalk Faults in Synchronous Sequential Circuits”, *IEEE TCAD*, pp. 252-263, 2005.

[27] J. Bradford, H. Delong, I. Pilian, and B. Becker, “Simulating Realistic Bridging and Crosstalk Faults in an Industrial Setting”, *IEEE ETW*, pp. 75-80, 2002.

[28] N. Itazaki, Y. Idomoto and K. Kinoshita, “A Fault Simulation Method for Crosstalk Faults in Synchronous Sequential Circuits”, *IEEE FTCS*, pp. 25-27, 1996.

[29] M. Phadoongsidhi, K. T. Le, and K. K. Saluja, “A Concurrent Fault Simulation for Crosstalk Faults in Sequential Circuits”, *IEEE ATS*, pp. 182-187, 2002.

[30] H. Takahashi, M. Phadoongsidhi, and Y. Higami, “Simulation-Based Diagnosis for Crosstalk Faults in Sequential Circuits”, *IEEE ATS*, pp. 63-68, 2001.

[31] P. F. Shen, H. W. Li, Y. J. Xu, and X. W. Li, “Non-robust Test Generation for Crosstalk-Induced Delay Faults”, *IEEE ATS*, pp. 120-125, 2005.

[32] H. W. Li, P. F. Shen, and X. W. Li, “Robust Test Generation for Precise Crosstalk-Induced Path Delay Faults”, *IEEE VTS*, pp. 300-305, 2006.

[33] W. Y. Chen, S. K. Gupta, and M. A. Breuer, “Test Generation for Crosstalk-Induced Delay in Integrated Circuits”, *IEEE ITC*, pp. 191-200, 1999.

[34] S. Irajpour, S. K. Gupta, and M. A. Breuer, “Timing-Independent Testing of Crosstalk in the Presence of Delay Producing Defects using Surrogate Fault Models”, *IEEE ITC*, pp. 1024-1033, 2004.

[35] Aniket, R. Arunachalam, “A Novel Algorithm for Testing Crosstalk Induced Delay Faults in VLSI Circuits” *IEEE ICVD*, pp. 479-484, 2005.

[36] A. Krstic, Y. M. Jiang, and K. T. Cheng, “Pattern Generation for Delay Testing and Dynamic Timing Analysis Considering Power-Supply Noise Effects”, *IEEE TCAD*, pp. 416-425, 2001.

[37] H. W. Li, Y. Zhang, X. W. Li, “Delay Test Pattern Generation Considering Crosstalk-Induced Effects”, *IEEE ATS*, pp. 178-183, 2003.

[38] S. Chary and M. L. Bushnell, “Automatic Path-Delay Fault Test Generation for Combined Resistive Vias, Resistive Bridges, and Capacitive Crosstalk Delay Faults”, *IEEE VLSID*, 2006.

[39] A. Rubio and J. A. Sainz, “Test Pattern Generation for Logic Crosstalk Faults in VLSI Circuits”, *IEE Proc. G Circuits Devices and Systems*, pp. 179-181, 1991.

[40] A. Rubio, N. Itazaki, X. Xu, and K. Kinoshita, “An Approach to the Analysis and Test of Crosstalk Faults in Digital VLSI Circuits”, *IEEE EDAC*, pp. 72-79, 1991.

[41] A. Rubio, N. Itazaki, X. Xu, and K. Kinoshita, “An Approach to the Analysis and Detection of Crosstalk Faults in Digital VLSI Circuits”, *IEEE TCAD*, Vol. 13, Issue 3 , pp. 387-395, March 1994.

[42] R. Kundu and R. D. (Shawn) Blanton, “Timed Test Generation for Crosstalk Switch Failures in Domino CMOS”, *IEEE VTS*, pp. 379-385, 2002.

[43] K. T. Lee, C. Nordquist, J. A. Abraham, “Automatic Test Pattern Generation for Crosstalk glitches in Digital Circuits”, *IEEE VTS*, pp. 34-39, 1998.

[44] R. Kundu and R. D. Blanton, “ATPG for Noise-Induced Switch Failures in Domino Logic”, *IEEE ICCAD*, pp. 765-768, 2003.

[45] J. Liu, W. B. Jone, and S. R. Das, “Crosstalk Test Pattern Generation for Dynamic Programmable Logic Arrays”, *IEEE IMTC*, pp. 55-60, 2005.

[46] K. T. Lee, C. Nordquist, and J. A. Abraham, “Test Generation for Crosstalk Effects in VLSI Circuits”, *IEEE ISCAS*, pp. 628-631, 1996.

[47] W. Y. Chen, S. K. Gupta, and M. A. Breuer, “Test Generation in VLSI Circuits for Crosstalk Noise”, *IEEE ITC*, pp. 641-650, 1998.

[48] W. Y. Chen, S. K. Gupta, and M. A. Breuer, “Analytical Models for Crosstalk Excitation and Propagation in VLSI Circuits”, *IEEE TCAD*, pp. 1117-1131, 2002.

[49] Y. Zhao and S.Dey, “Analysis of Interconnect Crosstalk Defect Coverage of Test Sets”, *IEEE ITC*, pp. 492-501, 2000.

[50] M. Graziano, G. Masera, G. Piccinini, M. R. Roch, and M. Zamboni, “A Statistical Noise-Tolerance Analysis and Test Structure for Logic Families”, *IEEE ICMTS*, pp. 68-73, 1999.

[51] K. S. M. Li, C. L. Lee, C. C. Su, and J. E Chen, “Oscillation Ring Based Interconnect Test Scheme for SoC”, *IEEE ASPDAC*, pp. 184-187, 2005.

[52] L. Chen, X. Bai, and S. Dey, “Testing for Interconnect Crosstalk Defects Using On-Chip Embedded Processor Cores”, *IEEE DAC*, pp. 317-322, 2001.

[53] S. Irajpour, S. Nazarian, L. Wang, S. K. Gupta, and M. A. Breuer, “Analyzing Crosstalk

in the Presence of Weak Bridge Defects", *IEEE VTS*, pp. 385-392, 2003.

[54] W. Sirisaengtaksin and S. K. Gupta, "Modeling and Testing Crosstalk Faults in Inter-Core Interconnects that Include Tri-state and Bi-directional Nets", *IEEE ATS*, pp. 132-139, 2004.

[55] N. Itazaki, Y. Matsumoto, and K. Kinoshita, "An Algorithmic Test Generation Method for Crosstalk Faults in Synchronous Sequential Circuits", *IEEE ATS*, pp. 22-27, 1997.

[56] K. Shimizu, N. Itazaki, and K. Kinoshita, "Built-In Self-Test for State Faults Induced by Crosstalk in Sequential Circuits", *IEEE ATS*, pp. 469, 2001.

[57] J. Yang, C. hu, Y. Shi, Z. Zhang, and L. Shi, "A New Self-Test Structure for At-Speed Test of Crosstalk in SoC Busses", *IEEE ICASIC*, pp. 633-636, 2001.

[58] X. Bai, S. Dey, and J. Rajski, "Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects", *IEEE DAC*, pp. 614-624, Jun. 2000.

[59] K. Sekar and S. Dey, "LI-BIST: A Low-Cost Self-Test Scheme for SoC Logic Cores and Interconnects", *IEEE VTS*, pp. 417-422, 2002.

[60] T. Bengtsson, A. Jutman, R. Ubar, and S. Kumar, "A Method for Crosstalk Fault Detection in On-Chip Buses", *IEEE NORCHIP*, pp. 285-288, 2005.

[61] T. Bengtsson, A, Jutman, R. Ubar, and S. Kumar, "Software-Based Self-Test Methodology for Crosstalk Faults in Processors", *IEEE HLDVTW*, pp. 11-16, 2003.

[62] W. C. Lai, J. R. Huang, and K. T. Cheng, "Embedded-Software-Based Approach to Testing Crosstalk-Induced Faults at On-Chip Buses", *IEEE VTS*, pp. 204-209, Apr. 2001.

[63] C. Metra, M. Favalli, B. Ricco, "On-Line Testing and Diagnosis Scheme for Intermediate Voltage Values Affecting Bus Lines", *IEEE DBT*, pp. 76-81, Apr. 2000.

[64] C. Metra, M. Favalli, B.Ricco, "On-Line Detection of Logic Errors due to Crosstalk, Delay, and Transient Faults", *IEEE ITC*, pp. 524-533, 1998.

[65] M. Favalli and C. Metra, "Online Testing Approach for Very Deep-Submicron ICs", *IEEE DATC*, pp. 16-23, 2002.

[66] C. Metra, A. Pagano, B. Ricco, "On-line Testing of Transient and Crosstalk Faults Affecting Interconnections of FPGA-Implemented Systems", *IEEE ITC*, pp. 939-947, 2001.

[67] A. Deutsch, G. V. Kopcsay, "When are Transmission-Line Effects Important for On-chip Interconnections", *IEEE Trans. on Microwave Theory and Techniques*, Vol.45, No.10, Oct 1997.

[68] A. Deutsch, G. V. Kopcsay, "High-Speed Signal Propagation on Lossy Transmission Lines", *IBM J. Res. Develop*, Vol. 34, No. 4, July 1990.

[69] W. C. Wu and C. L. Lee, "Delay Testing and Fault Simulation for Digital Circuits", *Ph. D. dissertation of Dep. E.E. of National Chiao-Tung Univ.*, 1997.

[70] W. C. Wu, C. L. Lee, M. S. Wu, J. E Chen, and M.Abadir, "Oscillation Ring Delay Test for High Performance Microprocessor", *JETTA*, pp. 147-155, 2000.

[71] K. Arabi, B. Kaminska, and S. Simter, “Design for Testability of Integrated Operational Amplifiers Using Oscillation-Test Strategy”, *IEEE ICCD*, pp. 40-45, 1996.

[72] K. Arabi and B. Kaminska, “Oscillation-Test Strategy for Analog and Mixed-Signal Integrated Circuits”, *IEEE VTS*, pp. 476-482, 1996.

[73] A. M. Smith and R. Sridhar, “On-line Detection of Environmentally-Induced Delay Faults in CMOS Wave Pipelined Circuits”, *IEEE ASIC Conference and Exhibit*, pp. 57-60, 1996.

[74] M. Abramovici, M. A. Breuer, and A. D. Friedman, “Digital System Testing and Testable Design”, pp215-217, 1990.

[75] F. Brglez, D. Bryan, and K. Kozminiski, “Combinational Profiles of Sequential Benchmark Circuits”, *IEEE ISCAS*, pp. 1929-1934, 1989.

[76] G. L. Smith, “Model for Delay Faults Based upon Paths”, *Proc. IEEE ITC*., pp. 342-349, 1985.

[77] C. J. Lin and S. M. Reddy, “On delay fault testing in logic circuits”, *IEEE TCAD*, pp. 694-703, Sep. 1987.

[78] E. P. Hsieh, et al, “Delay Test Generation”, *IEEE DAC*, pp486-491, 1977.

[79] K. Fuchs, F. Fink, and M. H. Schulz, “DYNAMITE: An Efficient Automatic Test Pattern Generation System for Path Delay Faults”, *IEEE TCAD*, pp. 1323-1335, 1991.

[80] K. D. Wagner, “The Error Latency of Delay Faults in Combinational and Sequential Circuits”, *IEEE ITC*, pp334-341, 1985.

[81] S. Koeppe, “Modeling and Simulation of Delay Faults in CMOS Stuck-at Open Logic Circuits”, *IEEE ITC*, pp530-536, 1986.

[82] S. T. Chakradhar, M. A. Iyer, and V. D. Agrawal, “Energy Models for Delay Testing”, *IEEE TCAD*, vol. 14, pp. 728-739, Jun 1995.

[83] Z. Zhang, R. D. McLeod, D. M. Miller, and S. Zhang, “Statistically Estimating Path Delay Fault Coverage in Combinational Circuits”, *IEEE PACRIM*, pp. 461-464, May 1995.

[84] M. Sivaraman and A. J. Strojwas, “Diagnosis of Parametric Path Delay Faults”, *IEEE ICVD*, pp. 412-417, 1996.

[85] J. J Liou, A. Krstic, K. T. Cheng, D. A. Mukherjee, and S. Kundu, “Performance Sensitivity Analysis Using Statistical Methods and Its Applications to Delay Testing”, *IEEE ASPDAC*., pp. 587-592, 2000.

[86] F. Brglez and H. Fujiwara, “A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in FORTRAN”, *IEEE ISCAS*, 1985.

[87] M. S. Wu, C. L. Lee, C. P. Chang, and J. E. Chen, “A Testing Scheme for Crosstalk Faults Based on the Oscillation Test Signal”, *IEEE ATS*, pp. 170-175, 2002.

[88] M. S. Wu, and C. L. Lee, “Using a Periodic Square Wave Test Signal to Detect Crosstalk Faults”, *IEEE DATC*, Mar.-Apr., pp. 160-169, 2005.

[89] C. K. Tsai, M. S. Malgorzata, “Modeling Crosstalk Induced Delay”, *IEEE ISQED*, pp.

189-194, Mar. 2003.

[90] Aniket, R. Arunachalam, “A Novel Algorithm for Testing Crosstalk Induced Delay Faults in VLSI Circuits”, *IEEE ICVD*, pp. 479-484, 2005.

[91] A. E. Zain and S. Chowdhury, “An Analytical Method for Finding the Maximum Crosstalk in Lossless-coupled Transmission Lines”, *IEEE ICCAD*. pp. 443-448, 1992.

[92] D. Chai, A. Kondratyev, Y. Ran, K. H. Tseng, Y. Watanabe, and M. S. Malgorzata, “Temporofunctional Crosstalk Noise Analysis”, *IEEE DAC,* pp. 860-863, 2003.

[93] C. L. Chen, C. L. Lee, and M. S. Wu, “A New Path Delay Test Scheme Based on Path Delay Inertia”, *IEEE ATS*, pp. 140-144, 2004.

學經歷

姓　　名：吳明學

性　　別：男

籍　　貫：臺灣省臺東縣

出生日期：民國五十八年十二月二十一日

通訊住址：高雄市 807 三民區安寧街 366 之 1 號，電話：(07)3115406

學 經 歷：民國八十五年九月至民國九十五年七月

國立交通大學電子研究所博士班

民國八十三年九月至民國八十五年六月

國立交通大學電子研究所碩士班

民國七十八年九月至民國八十三年六月

國立交通大學電子工程學系

論文題目：以振盪信號測試超大型積體電路之串音障礙

VLSI Crosstalk Fault Testing with Oscillation Signals

著作目錄

吳明學

(A) International Journal:

[1] M. S. Wu, C. L. Lee, "Using a Periodic Square Wave Test Signal to Detect Crosstalk Fault," IEEE D&T of Computers, Vol. 22, Number 2, Mar.-Apr., pp. 160-169, 2005

(B) International Conference:

[1] W.C. Wu, C.L. Lee, M.S. Wu, J.E. Chen, and M.Abadir, "Oscillation Ring Delay Test for High Performance Microprocessor", JETTA, pp. 147-155, 2000.

[2] M. S. Wu, C. L. Lee, C. P. Chang, and J. E. Chen, "A Testing Scheme for Crosstalk Faults Based on the Oscillation Test Signal," IEEE ATS, pp. 170-175, 2002.

[3] C. L. Chen, C. L. Lee, and M. S. Wu, "A New Path Delay Test Scheme Based on Path Delay Inertia Principle," IEEE ATS, pp. 140-144, 2004.

[4] M. S. Wu, C. L. Lee, Y. J. Chang, and W. C. Wu, "Crosstalk Fault Detection for Interconnection Lines Based on Path Delay Inertia Principle," IEEE ATS, pp. 106-111, 2005.

(C) Local Conference:

[1] M. S. Wu, C. L. Lee, C. P. Chang, and J. E. Chen, "A Testing Scheme for Crosstalk Faults Based on the Oscillation Test Signal," Proceeding of VLSI Design/CAD Symposium, 2002.

[2] M. S. Wu, C. L. Lee, Y. J. Chang, and W. C. Wu, "Crosstalk Fault Detection for Interconnection Lines Based on Path Delay Inertia Principle," Proceeding of VLSI Design/CAD Symposium, 2005.