TEST GENERATION FOR CROSSTALK NOISE IN VLSI CIRCUITS

by

Wei-Yu Chen

Ceng 00-07

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFRONIA
In Partial Fulfillment of the
Requirements for the Degree
Doctor of Philosophy
(ELECTRICAL ENGINEERING)

May 2000

Copyright 2000 Wei-Yu Chen

Acknowledgments

I would like to begin by expressing my deep sense of gratitude to my advisors, Professor Melvin A. Breuer and Sandeep K. Gupta, for their help, support and guidance throughout this research. Their constant enthusiasm towards this research made working with them exciting and challenging. They are always accessible and have provided me knowledgeable advices. They will always have my highest admiration and deepest respect.

I also appreciate my dissertation committee members, Professor Massoud Pedram and Peter H. Baxendale for taking their time in reviewing this dissertation.

I would also like to thank my fellow students, Yishing Chang, Arani Sinha, Liang-Chi Chen, Suriyaprakash Natarajan, and Seelan Kumrasamy in our research group for making it a comfortable environment to work in and for their many helpful discussions. Through their helpful knowledge exchange and many pleasant gatherings, together we have developed friendships that I believe will last for many year to come.

This work was supported in part by Intel Corporation and by Semiconductor Research Corporation under contract number 98-TJ-646.

Finally and mostly, I would lie to thank my parents and my dear wife, Meng-Chen Chang, for their constant support, patience, encouragement and true love at every moment in my life.

Table of Contents

ACKNOWLEDGMENTS II

LIST OF FIGURES VI

LIST OF TABLES IX

ABSTRACT X

CHAPTER 1 1

INTRODUCTION 1

1.1 CROSSTALK EFFECTS 4
1.2 REVIEW OF CROSSTALK MODELS 7
1.2.1 Symmetric transmission line model 7
1.2.2 Distributed models 11
1.2.3 Simplified lumped model 13
1.3 EXISTING TEST GENERATION TECHNIQUES FOR CROSSTALK NOISE 15
1.4 MOTIVATION AND ORGANIZATION OF THE DISSERTATION 19

CHAPTER 2 22

ANALYTIC MODELS FOR CROSSTALK EXCITATION 22

2.1 CROSSTALK EFFECTS 23
2.2 TECHNOLOGY TRENDS AND PROCESS VARIATIONS 26
2.2.1 Technology Scaling 26
2.2.2 Impacts of Process Variations 28
2.3 NEW DESIGN VALIDATION AND TEST ISSUES 30
2.4 CROSSTALK MODEL AND ANALYSIS 32
2.4.1 Driver Modeling and Approximation of Distributed RC Network Using Lump Models 33
2.4.1.1 Driver Modeling 33
2.4.1.2 Approximation of Distributed Network Using Lump Models 35
2.4.2 Analytical Equations from Crosstalk Waveforms 36
2.4.2.1 Analysis of Crosstalk Pulse 37
2.4.2.2 Analysis of Crosstalk Delay 46
2.4.3 Dependence of Crosstalk Effects on Input Transition Times and Skews 51
2.5 DESIGN VALIDATION FOR CROSSTALK NOISE 56
2.6 SUMMARY 58

CHAPTER 3 61

ANALYTIC MODELS FOR NOISE PROPAGATION 61

3.1 A NEW INVERTER MODEL ... 61
3.2 A METHOD TO COLLAPSE CMOS GATES ... 65
3.2.1 Series MOS ... 66
3.2.2 Parallel MOS ... 72
3.2.3 Internal Capacitance ... 73
3.2.4 Multiple Input Transitions ... 74
3.3 A PIECE-WISE LINEAR MODEL FOR NOISE ... 75
3.4 TERMINATION CONDITIONS FOR NOISE (OUTPUT RECEIVER CHARACTERIZATION) ... 82
3.5 SUMMARY ... 86

CHAPTER 4 88

TEST GENERATION FOR CROSSTALK NOISE 88

4.1 VALUE SYSTEMS ... 89
4.2 CONDITIONS FOR MAXIMIZING CROSSTALK EFFECTS ... 93
4.3 COST FUNCTIONS FOR NOISE PROPAGATION ... 94
4.4 TIMING ANALYSIS ... 96
4.4.1 Forward "Arrival" Timing Window Calculation ... 97
4.4.2 Backward "Required" Timing Window Calculation ... 98
4.4.3 Timing-Oriented ATPG ... 99
4.4.3.1 Objectives for Crosstalk Delay ... 100
4.4.3.2 Timing-Oriented Backtrace Procedure ... 102
4.4.3.3 Incremental timing refinement ... 105
4.4.3.4 Selection of Propagation Paths ... 106
4.4.3.5 Conflicts between Objectives and Backtracking ... 107
4.4.3.6 Branch and Bound Process to Reduce the Search Space ... 108
4.5 TEST GENERATION ALGORITHM ... 110
4.5.1 Main Test Generation Algorithm ... 110
4.6 EXPERIMENTAL RESULTS ... 114
4.6.1 Crosstalk Pulse ... 114
4.6.2 Crosstalk Delay ... 124
4.7 SUMMARY ... 136

CHAPTER 5 138

FUTURE EXTENSIONS TO OUR ATPG 138

5.1 EXTENSION TO GENERAL GATES ... 138
5.1.1 General CMOS gates ... 139
5.1.2 Dynamic gates ... 140
5.1.3 Latches ... 142
5.2 MULTIPLE CROSSTALK EFFECTS ... 144
5.2.1 Multiple-way and multiple-level crosstalk ... 145
5.2.2 Static glitches ... 150
5.3 TARGET FAULT EXTRACTION ... 153
5.4 SUMMARY ... 157

CHAPTER 6 158
CONCLUSIONS 158
REFERENCES 163
APPENDIX A 169
APPENDIX B 172
APPENDIX C 175

List of Figures

Figure 1.1 Crosstalk effects: (a) basic structure of circuit; (b) crosstalk pulse; (c) crosstalk slowdown; (d) crosstalk speedup. 5

Figure 1.2 (a)-(c) Errors causes by crosstalk pulse (d) error caused by crosstalk delay. ...6

Figure 1.3 Transmission line model....................8

Figure 1.4 Simplified capacitive coupling model. 13

Figure 2.1 Simple circuit showing source of crosstalk due to capacitive coupling. 24

Figure 2.2 Crosstalk waveforms of signals in Figure 2.1: (a) crosstalk pulse; (b) crosstalk decreases/increases signal transition times (speedup/slowdown). 25

Figure 2.3 Noise trend for different technologies. 28

Figure 2.4 Capacitive coupling model. 32

Figure 2.5 (a) An input signal with transition time t_{ra} applied to a driver; (b) equivalent circuit....................34

Figure 2.6 Circuit model for crosstalk pulse analysis. (a): circuit model for a positive pulse induced on V due to a rising transition on A; (b) an equivalent circuit. 39

Figure 2.7 (a) Crosstalk pulse at V due to exponential and step inputs at A_{in}; (b) maximum amplitude vs. input transition time (time constant); (c) maximum amplitude vs. affecting/victim driver ratio; (d) maximum amplitude vs. affecting and victim lines resistance (driver resistance plus line resistance); (e) maximum amplitude vs. coupling capacitance, affecting and victim lines load capacitance (line capacitance plus load capacitance). 45

Figure 2.8 Equivalent circuit for crosstalk delay analysis....................46

Figure 2.9 Crosstalk speedup and slowdown effects assuming simultaneously switching inputs where both inputs have a transition time of 100ps. (a) effects on victim line; (b) effects on affecting line. 50

Figure 2.10 Circuit used to study influence of input signal properties and circuit parameters on crosstalk. 51

Figure 2.11 (a) The victim line slowdown-time vs. input switching rates; (b) the victim line speedup-time vs. input switching rates. 53

Figure 2.12 Voltage waveforms on affecting and victim lines for z = 25 ps. 54

Figure 2.13 Victim line speedup-time and slowdown-time vs. skew z. 54

Figure 2.14 Example circuit for test vector generation. 58

Figure 3.1 CMOS inverter and its corresponding model when N and P MOS transistors operate in different modes: (a) circuit, (b) PMOS in linear and NMOS in

saturation mode, (c) both in saturation mode, and (d) NMOS in linear and PMOS in saturation mode. 63

Figure 3.2 Comparison of analytic result of proposed model and SPICE simulations....65

Figure 3.3 Pull-Down NMOS chains; (a) single NMOS, (b) series connected NMOS, all values normalized w.r.t. V_{DD}. 68

Figure 3.4 Experimental results for selecting the empirical constant m: a) percentage error w.r.t. the output signal delay time; b) percentage error w.r.t. the output signal rise/fall times. 70

Figure 3.5 Experimental results for selecting the empirical constant α: a) percentage error w.r.t. the output signal delay time; b) percentage error w.r.t. the output signal rise/fall times. 71

Figure 3.6 (a) Circuit for collapsing NAND gate into an equivalent inverter, (b) model and SPICE simulation results. 72

Figure 3.7 (a) Pull-down subcircuit of a NAND gate, (b) corresponding RC model to obtain lumped load capacitance including internal capacitance, and (c) the circuit with all capacitance lumped into the load capacitance. 73

Figure 3.8 Crosstalk pulse passes through an inverter (a) a small input pulse, (b) a large input pulse. 77

Figure 3.9 (a) Circuit for measurement for input and output pulses amplitude H'; (b) Comparison of the model and SPICE results. 81

Figure 3.10 (a) Circuit for applying piece-wise-linear pulses; (b) Comparison of the model and SPICE results (maximum pulse amplitudes). 82

Figure 3.11 Circuit diagram for a output voltage degradation of a dynamic gate due to a input pulse. 84

Figure 3.12 Output voltage degradation of a dynamic gate due to a input pulse: (a) severity of voltage degradation w.r.t. various pulse amplitudes and widths, (b) severity of voltage degradation w.r.t. the arrival times of input pulses 85

Figure 3.13 Setup time violation of a D flip-flop causes metastability. 86

Figure 4.1 Computation of timing windows for a gate. 97

Figure 4.2 Computation of required times. 98

Figure 4.3 Timing window of transitions on the affecting (A) and victim (V) lines, where z is the skew allowed on A. 100

Figure 4.4 Amount of speedup and slowdown on the victim line V vs. skew z: a negative z implies A leads V. 102

Figure 4.5 Check for the existence of a compatible and incomplete pattern at gate inputs in processing objectives. 103

Figure 4.6 Recursive execution of the backtrace process. 104

Figure 4.7 Incremental timing refinement: a) before refinement, b) after refinement... 106

Figure 4.8 Branching and bounding process, where Δ is the delay of gate g................ 110

Figure 4.9 Flowchart of the algorithm. 113

Figure 4.10 Example circuit to illustrate the algorithm. 115

Figure 4.11 Detection rate vs. pulse threshold.. 121

Figure 4.12 Detection rate vs. coupling capacitance.. 122

Figure 4.13 Detection rate vs. ratio of affecting to victim line driver strengths. 122

Figure 4.14 Detection rate vs. signal transition times at primary inuts......................... 123

Figure 4.15 Example circuit to illustrate the algorithm. 125

Figure 4.16 Waveforms on the victim and affecting lines. 126

Figure 4.17 Detection rate vs. skew between affecting and victim lines. 131

Figure 4.18 Detection rate vs. extra delay slack. 132

Figure 5.1 Crosstalk effect on static gate and dynamic gate: (a) a basic gate G; (b) gate G implemented as a static gate; (c) corresponding input/output pulse waveform for (b); (d) gate G implemented as a dynamic gate; (e) corresponding input/output pulse waveform for (d). 141

Figure 5.2 (a) Dynamic latch, (b) static latch, (c) crosstalk pulse................................ 143

Figure 5.3 A cross-coupled inverter latch. 143

Figure 5.4 A1-A2 are affecting lines and V is the victim line 145

Figure 5.5 Victim line circuit model for multi-way coupling; C_{m1} and C_{m2} are coupling capacitance to A1 and A2; t_{r1} and t_{r2} are switch times of signals on A1 and A2, respectively... 145

Figure 5.6 (a) Different affecting signal slopes; (b) piece-wise linear approximation of pulse waveform. 147

Figure 5.7 Example of multiple-level coupling: a victim "path".................................. 148

Figure 5.8 (a) Creation of a static glitch, (b) transistor diagram of a NAND gate, (c) creation of an equivalent input waveforms 152

Figure 5.9 Crosstalk circuit with a ramp signal at the affecting line with rise time t_r. .. 155

Figure A. 1 Increased delay vs. driver ratio. 176

List of Tables

Table 2.1 Interconnect parameters for various technologies [32], [47], [48]. W is the min. width; R and C are unit length resistance and total capacitance; AR is the aspect ratio, Ca, Cf, and Cm are area, fringing and coupling capacitance, respectively 27

Table 2.2 Effect of process variations on crosstalk delay (pico seconds) 30

Table 2.3 Effect of process variations on crosstalk pulse height 30

Table 4.1 Symbols and parameters used for test generation. 90

Table 4.2 Truth table for the value system for an AND gate. 91

Table 4.3 Conditions for achieving three objectives (for a NAND gate). 94

Table 4.4 Conditions creating a fast transition at the output of a NAND gate. 105

Table 4.5 Comparison of the model and SPICE results. 116

Table 4.6 Comparison of the model and SPICE results for circuit with dynamic gate D. 116

Table 4.7 Results of experiment 1: all tests for a single fault. 119

Table 4.8 Result of experiment 2: one test for each fault; Number of faults = 100; no timing criterion set at POs. 119

Table 4.9 Result of experiment 2: one test for each fault; Number of faults = 100; the longest path delay is set as the timing criterion at POs. 120

Table 4.10 Results of Experiment 1: all tests for a single fault. 129

Table 4.11 Results of Experiment 2: one test for each fault; number of faults = 100; no timing criterion set at POs. 130

Table 4.12 Results of Experiment 2: one test for each fault; number of faults = 100; the longest path delay is set as the timing criterion at POs. 130

Table 4.13 Comparison of ATPG efficiency for crosstalk pulse with and without incremental timing refinement; no timing criterion was set at POs. 134

Table 4.14 Comparison of ATPG efficiency for crosstalk delay with and without incremental timing refinement; no timing criterion was set at POs. 134

Table 4.15 Comparison of ATPG efficiency for crosstalk pulse with and without incremental timing refinement; the longest path delay was used as the timing criterion at POs. 135

Table 4.16 Comparison of ATPG efficiency for crosstalk delay with and without incremental timing refinement; the longest path delay was used as the timing criterion at POs. 136

Abstract

This dissertation presents a general methodology for the analysis of crosstalk noise and a test generation framework for crosstalk fault. Our goal is to enable more aggressive designs, decrease redesign effort, and ensure a higher quality of chips shipped to customers.

We first focus on developing an understanding of types of crosstalk effects and their dependence on circuit parameters, signal timing and process variations. Closed form equations quantifying the dependence of crosstalk effects on circuit parameters are presented. By differentiating these equations new design corners can be identified for validation of designs that have significant crosstalk effects. We also show that crosstalk effects can be significantly aggravated by variations in the fabrication process. The results of our analysis provide conditions that must be satisfied by a sequence of vectors used for validation of designs as well as post-manufacturing testing of devices in the presence of significant crosstalk.

We present a test generation framework to efficiently and accurately generate two-vector tests for crosstalk effects, such as pulses, signal speedup and slowdown, in digital combinational circuits. Several new techniques have been developed including new models for a CMOS inverter, methods to calculate inverter output response for pulse inputs, a method for collapsing CMOS gates into equivalent inverters, and a piece-wise linear model for pulses. These techniques were integrated into a mixed-signal test

generator that incorporates classical static values as well as dynamic signals such as transitions and pulses. In addition, this ATPG algorithm includes the concept of gate delay and timing information such as signal arrival time, and rise/fall times. Conditions for the creation of the worst-case coupling and propagation of a crosstalk effect are presented. We also present a new analog cost function that is used to guide the search process. By using the path delay information obtained in circuit preprocessing and/or the analog cost function, preferred paths can be selected during the backtrace as well as propagation process. Comparison of results with SPICE simulations confirms the accuracy of this approach, and experimental results show that the method can be applied to circuits with reasonable sizes.

In the future, our test generation framework can be extended in several aspects to improve and/or optimize the test generation process. The capability of our APTG can be extended to deal with (a) more general CMOS gates, (b) different types of logic including dynamic gates and latches, (c) multiple crosstalk effects, and (d) techniques to automate the process of target fault extraction.

Chapter 1

Introduction

The dramatic increase in signal switching speed and density of integrated circuits leads to challenging design and test problems. The problem addressed by this dissertation are motivated by three area of changes, namely (1) the emergence of deep sub-micron technology, (2) high clock rates, and (3) short signal rise and fall times. These changes have made interconnection lines that were once considered to be electrically isolated now interfere with each other and have an important impact on system performance and correctness. One such interaction caused by parasitic coupling between wires is known as crosstalk, and many advanced systems do not achieve optimum performance because the impact of the crosstalk noise has been underestimated.

Crosstalk has always existed but was not of major concern for micron-level (i.e. feature size >1um) technologies. Continuous advancements in the field of VLSI have lead to a decrease in device geometry (deep sub-micron technology). This makes cross-coupling capacitance between adjacent wires increasingly significant. At higher clock rates both clock and signal transitions must be very short, thus leading to small values of rise (fall) time t_r (t_f), and hence CdV/dt values in a circuit increase. In addition, due to the consideration of power consumption, reduction of the power-supply voltage results in reduced noise margins. Hence crosstalk effects become more severe.

Crosstalk noise may cause undesirable effects including excessive overshoot, undershoot, glitches, additional signal delay (slowdown) and even a reduction in signal delay (speedup)[30]. If these anomalies are sufficiently large, they can propagate to a storage element and create a permanent error. For example, many high performance circuits make extensive use of pipelines, shallow logic blocks between storage elements, dynamic gates, latches instead of flip-flops, single phase clocking, and performance based logic design. The net result is that the timing margins between clocked elements are small. Hence delay must be well controlled and budgeted. Because crosstalk can adversely affect signal delay, coupling effect must be correctly handled to guarantee correct circuit operation. Thus the delay time for each combinational block must include the signal skews due to crosstalk. Also, the setup and hold times for latch elements must include the clock skews due to crosstalk. There is also a trend toward the increased use of asynchronous circuits. Such circuits are event-driven and should be hazard-free. Since crosstalk can induce a pulse on a circuit line, a new source of errors must be considered. If not carefully considered during design validation, crosstalk can produce logic errors in such circuits.

Current trends in integrated circuit design indicate that signal noise and skew due to crosstalk create severe design and test problems. These problems are further aggravated by variations in the fabrication process [21]. If it were not for process variations and stringent area and performance constraints, an error due to crosstalk observed during validation could be eliminated by re-routing signals or redesign [24]. However, redesign may be very expensive in terms of design effort and its impact on a

product's schedule. In addition, with process variations and aggressive design goals, it may not always be possible to eliminate all noise effects at all worst case design and fabrication corners.

An alternative is to develop techniques to generate tests for crosstalk. The resulting tests can be applied to each manufactured chip, and chips in which crosstalk does not cause any error will pass and be shipped to customers, while chips where crosstalk causes an error will be discarded. In other words, designers can either choose to eliminate potential errors caused by crosstalk via redesign, or detect crosstalk faults during post-manufacturing testing. Such a choice is often made in favor of living with the flaw when there is a time-to-market issue; a design change can be made in a future release. By providing such an alternative, test generation for crosstalk will enable more aggressive design, decrease re-design effort, and/or enable more comprehensive post-manufacturing testing.

Thus, accurate modeling and simulation of signal pulses and delay due to crosstalk is becoming increasingly important, and testing for severe process aggravated crosstalk effects is necessary to ensure the correct functionality of fabricated chips. One end product of our research will be a mixed-signal test generator that generates high quality tests for crosstalk induced errors (faults).

The remainder of this chapter is devoted to a review of crosstalk effects and a number of crosstalk models. It also provides a brief description of existing test generation techniques for crosstalk noise. Finally, the motivation and organization of this dissertation are given.

1.1 Crosstalk effects

First we will illustrate a few examples of crosstalk effects and how crosstalk can create circuit problems. There are two types of crosstalk effects, namely, crosstalk pulse and crosstalk delay. Crosstalk delay can be further divided into crosstalk speedup and crosstalk slowdown effects. A crosstalk pulse occurs due to the coupling between a circuit line having a signal transition and a line which is holding a steady value. For example, for the basic coupling circuit structure in Figure 1.1(a), a falling transition on line l_1 can cause a pulse at line l_2, which should ideally hold a steady 1 value, as shown in Figure 1.1(b). Crosstalk delay occurs when both lines have transitions in the same clock cycle. If l_1 and l_2 have transitions in the opposite directions, then each transition will occur later in time, compared to the situation where only one line is in transition, leading to crosstalk slowdown as shown in Figure 1.1(c). If l_1 and l_2 have transition in the same direction, then both transitions will occur sooner, leading to crosstalk speedup, as depicted in Figure 1.1(d).

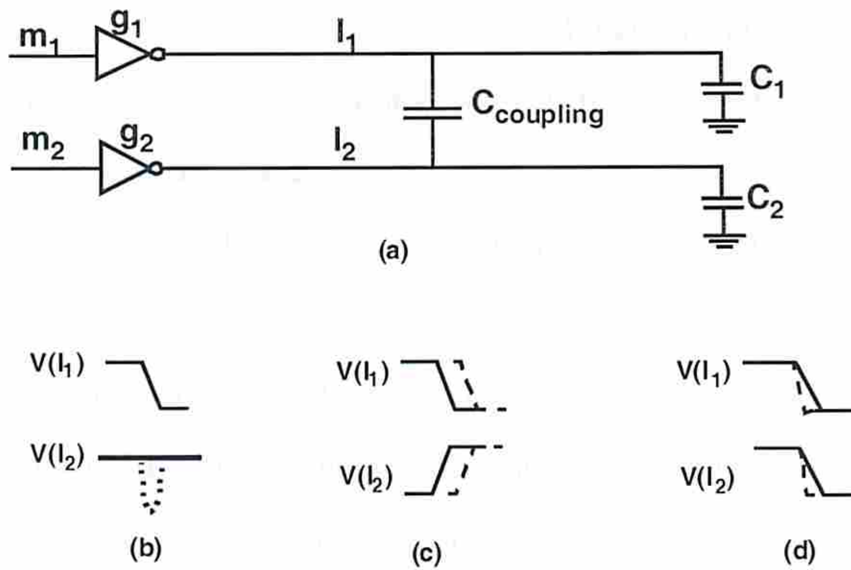


Figure 1.1 Crosstalk effects: (a) basic structure of circuit; (b) crosstalk pulse; (c) crosstalk slowdown; (d) crosstalk speedup.

Crosstalk noise can create logic errors during operation. Consider a pulse created on L2 in Figure 1.2(a). If this pulse is applied to an input of a dynamic NAND gate and all other inputs of the evaluation logic are at logic value 1, then the output may be accidentally discharged. Since the charge lost cannot be restored in the evaluation phase, this leads to a degraded voltage at the gate's output. If the degradation is substantial it may lead to a logic error. Also a degraded voltage on a line can be regarded as a weak "1" which may slowdown the operation speed of a gate in the line's fanout.

In Figure 1.2(b) a crosstalk pulse may trigger an un-wanted PRESET of a flip-flop causing data to be loss and hence an error. Similarly, if the line with a large crosstalk pulse is connected to the clock input of the flip-flop (not shown in the figure), then this pulse can be intrepreted as an additional clock pulse and cause the flip-flop to latch

erroneous data. Another example of a crosstalk pulse causing an error is shown in Figure 1.2(c). In a dense memory design, data lines usually run in parallel for long distances. If the coupling is sufficient, a signal transition on a data line can create a significant crosstalk pulse on an adjacent data line. Since the word line is enabled for the entire row, the crosstalk pulse may damage the content of a neighbor memory cell.

Finally, consider the circuit shown in Figure 1.2(d). If a signal is late to arrive due to crosstalk (both signals switch in the opposite direction) and this signal is propagated along a path that has a small delay slack, then a flip-flop setup time violation may occur and cause an erroneous logic value to be latched in the flip-flop.

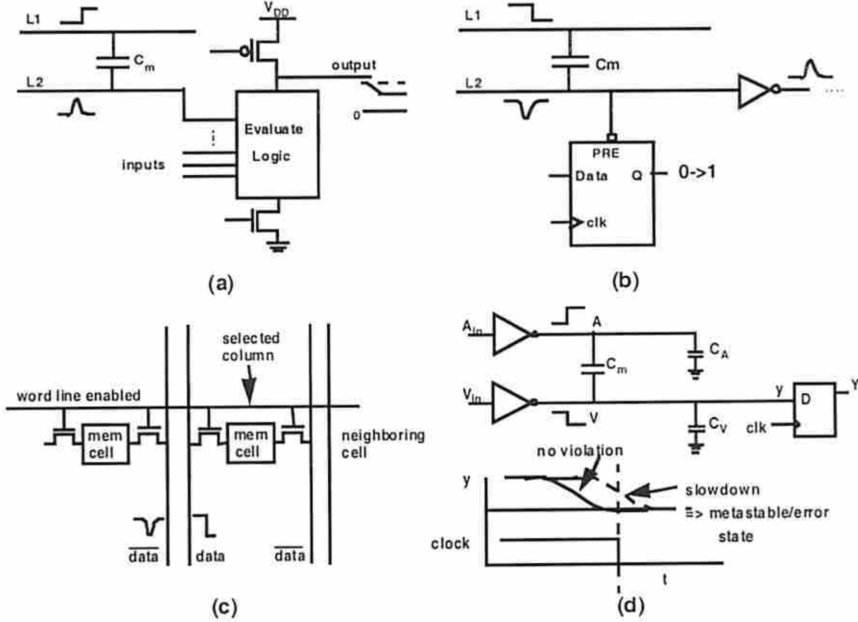


Figure 1.2 (a)-(c) Errors causes by crosstalk pulse (d) error caused by crosstalk delay.

1.2 Review of crosstalk models

1.2.1 Symmetric transmission line model

The modeling and analysis of crosstalk between interconnection lines have previously received considerable attention. Most crosstalk transient analysis techniques model interconnects as micro-strip lines and utilize the well-known multi-conductor transmission line theory [1]. The analysis of coupled lossy transmission lines has been considered by several authors [2], [3], [4], [5], [6]. Numerical methods to solve a model of lossy transmission lines in the time domain have been proposed in [7], [8]. Simulation models for interconnects and crosstalk were reported in [9], [10]. Non-linearity of the source and load networks, not addressed in these papers, were considered in [15], [16], [17], [18].

A typical transmission line model is shown in Figure 1.3. The interconnection line can be modeled as a transmission line driven by a unit step voltage source V_s having resistance R_s, loaded by the capacitive load C_L, and coupled to adjacent lines by mutual capacitance and conductance. The resistance R_s is determined by the dimensions of the driving transistor, and the load impedance consists of the gate capacitance of the transistor loading the interconnection line.

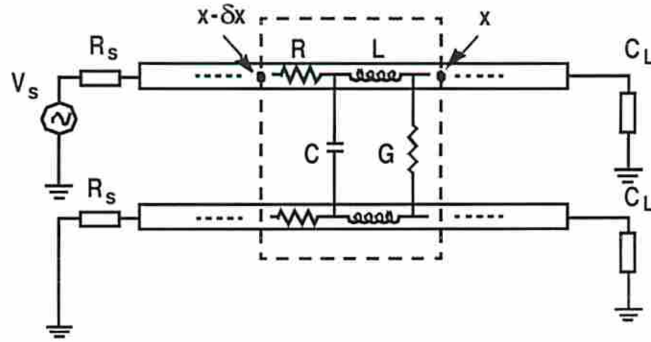


Figure 1.3 Transmission line model.

The transmission line equations are give by

$$\frac{\partial}{\partial x}V(x,t) = -\left[R + L\frac{\partial}{\partial t}\right]I(x,t) \tag{1-1}$$

$$\frac{\partial}{\partial x}I(x,t) = -\left[G + C\frac{\partial}{\partial t}\right]V(x,t) \tag{1-2}$$

where L and C are the inductance and capacitance per unit length of the interconnections, R is the resistance per unit length, G is the conductance determined by the isolation material, and x is the incremental length of the transmission line. The following analysis of the transmission line is similar to that presented in [1]. In the s domain, equation (1-1) and (1-2) can be written as

$$\frac{\partial}{\partial x}V(x,s) = -[R + sL]\,I(x,s) \tag{1-3}$$

$$\frac{\partial}{\partial x}I(x,s) = -[G + sC]V(x,s). \tag{1-4}$$

Let Z = R + sL and Y = G + sC, equations (1-3) and (1-4) can be solved in the s domain, yielding

$$V(x,s) = e^{-\sqrt{ZY}(x)}V_i(s) + e^{-\sqrt{ZY}(D-x)}V_r(s), \tag{1-5}$$

$$I(x,s) = \sqrt{\frac{Y}{Z}}[e^{-\sqrt{ZY}(x)}V_i(s) - e^{-\sqrt{ZY}(D-x)}V_r(s)], \tag{1-6}$$

where D is the total length of the transmission line, $V_i(s)$ is the voltage vector of the incident wave at x = 0, and $V_r(s)$ is the voltage vector of the incident wave at x = D.

The boundary conditions at the endpoints, i.e., x = 0 and x = D, are

$$V(0,s) = V_s(s) - R_s I(0,s),$$

and $$V(D,s) = \frac{1}{sC_L} I(D,s).$$

Solving for $V_i(s)$ and $V_r(s)$, we get

$$V_r(s) = V_s(s)\left\{-\left[1+R_s\sqrt{\frac{Y}{Z}}\right]\left[e^{\sqrt{ZY}D}\right]\left[1-\frac{1}{sC_L}\sqrt{\frac{Y}{Z}}\right]^{-1}\left[1+\frac{1}{sC_L}\sqrt{\frac{Y}{Z}}\right]+\left[1-R_s\sqrt{\frac{Y}{Z}}\right]\left[\frac{1}{e^{\sqrt{ZY}D}}\right]\right\}^{-1},$$

and

$$V_i(s) = -\left[e^{-\sqrt{ZY}D} - \frac{1}{sC_L}\sqrt{\frac{Y}{Z}}e^{-\sqrt{ZY}D}\right]^{-1}\left[1+\frac{1}{sC_L}e^{-\sqrt{ZY}D}\right]V_r(s).$$

The values for $V_i(s)$ and $V_r(s)$ can be substituted into equation (1-5) and (1-6) to obtain expressions for the current and voltage at x = 0 and x = D in the s domain. That is,

$$V(0,s) = V_i(s) + e^{-\sqrt{ZY}D}V_r(s),$$

$$V(L,s) = e^{-\sqrt{ZY}D} V_i(s) + V_r(s).$$

On the line to which the voltage source V_s is applied, V(D, s) is the voltage at the load capacitance. On the other line where V_s is not applied (i.e. the line held at constant value), V(D, s) represents the induced crosstalk voltage at its load capacitance.

In principle the time domain response can be obtained by the inverse Laplace transformation. If F(s) denotes the Laplace transform of f(t), then

$$F(s) = \int_0^\infty f(t) e^{-st} dt.$$

Let h(t) be an approximation of f(t). It has been shown [35] that h(t) is given by

$$f(t) = h(t) - E(t),$$

where

$$h(t) = \frac{1}{T}\left\{ \frac{F(a)}{2} + \sum_{k=1}^{\infty}\left(\mathrm{Re}\left[F(a + \frac{k\pi t}{T}) \right] \cos(\frac{k\pi t}{T}) - \mathrm{Im}\left[F(a + \frac{k\pi t}{T}) \right] \sin(\frac{k\pi t}{T}) \right)\right\},$$

and the error term E(t) is shown to be bounded by

$$E(t) \le M\left[\frac{e^{\beta t}}{e^{2T(a-\beta)} - 1} \right],$$

where t is in the interval (0, 2T), 1/T is the sampling frequency, M is a constant, and β is related to f(t) such that f(t) is an exponential of order a, i.e. $|f(t)| < Ce^{at}$.

Numerical computations show that if we apply the inverse Laplace transformation directly, the summation converges very slowly. Different numerical algorithm may speed

up the evaluation process, but a compromise must be made between accuracy and computation time.

Although the above techniques are very effective for some specific cases, they provide little general insights into the coupling mechanism. In addition, the circuit geometry analyzed is usually assumed to be symmetric, e.g., a BUS, with identical drivers, wires, and loads. If unbalanced circuit structures are assumed, the resulting equations will be much more complicated and the computation time will increase dramatically. Hence these techniques are often not applicable to VLSI circuits.

1.2.2 Distributed models

Reduced-order modeling techniques have become an important method for analyzing linear interconnect networks. RLC analysis has often been used to analyze clock trees, power busses, off-chip interconnects and clock skews. Similar approaches have been used to study coupling noise [39], [40], [41], [42].

In these approaches, a reduced-order modeling approach that allows for passive multi-port reduction of RC netlists as impedance macro-models while preserving the symmetric and sparsity of the state matrices has been proposed. The interconnection netlists were formulated using modified nodal analysis. The modified nodal analysis actually regards the interconnection netlists as finite distributed elements. The system of equations can be transformed into the Laplace domain and solved using Arnoldi [43] and Lanczos algorithms [44]. The macro-models are then employed to perform coupling

analysis with timing constraints to limit pessimism in the analysis. To perform the coupled noise calculation, the interconnect netlists are identified as the primary net where the noise was calculated (i.e., the victim line) and the secondary net (i.e., the affecting line) with significant coupling to the primary net. Couplings between the secondary net to nets other than the primary net are grounded and considered as load capacitance. Then the reduced-order modeling technique is applied and the modified nodal analysis used in the coupling noise calculation. Next, appropriate voltage sources are applied to the primary and secondary nets to excite the coupling noise. To calculate the worst possible noise at the primary net receiver, all pulses from different secondary nets are aligned and the superposition principle is applied to add up the peak voltages.

Although the principle advantage of the implicit techniques, such as the Arnoldi and Lanczos algorithms, is their natural extension to multiple-input, multiple-output systems where coupled networks can be analyzed, the time domain response of these algorithms still needs numerical evaluations with high computation complexity. The accuracy of the distributed model approach depends on the number of distributed elements considered. If the number of distributed elements is small, then the accuracy may not be satisfactory. But if the number of distributed elements is large, then the time complexity is high and may not be applicable to large circuits.

1.2.3 Simplified lumped model

In [11], [12], a simplified lumped RC model for crosstalk between a pair of coupled lines was proposed and the case is analyzed where the input to one line is held constant while the other has a step transition. Although the lumped model is less accurate than the transmission line model, it is feasible to obtain some insight into dependency on circuit parameters and the derived closed-form analytic equations lead to computationally tractable solutions.

Consider the simplified model of capacitive coupling shown in Figure 1.4. Here lumped capacitance are considered and other parasitic couplings are neglected. The affecting line A is assumed to have a falling transition with fall time t_f. C_{AG} and C_{VG} are the wire to ground capacitances of line A and line V, respectively. C_{AV} is the lumped coupling capacitance between line A and line V. To have a negative crosstalk pulse at line V, line V is held at logic level 1 through an active impedance R_{UV} (PMOS transistor of the line driver).

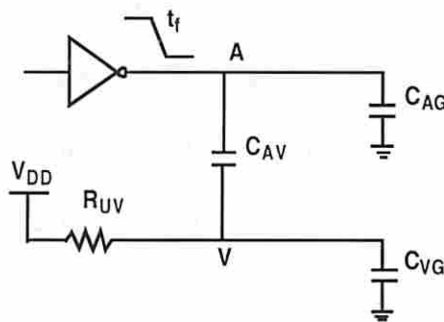


Figure 1.4 Simplified capacitive coupling model.

If the falling transition on line A is assumed to follow a linear slope going from V_{DD} to GND, the behavior of the line V voltage is given by

$$V_V(t) = \left(1 - \frac{C_{AV} R_{UV}}{t_f}\right) V_{DD} + \frac{C_{AV} R_{UV}}{t_f} V_{DD} \left(e^{-t/R_{UV} C_T}\right), \qquad for \quad 0 \le t \le t_f$$

$$V_V(t) = V_V(t_f) + (V_{DD} - V_V(t_f))\left(1 - e^{-(t-t_f)/R_{UV} C_T}\right), \qquad for \quad t > t_f$$

where $C_T = C_{VG} + C_{AV}$.

From these equations the maximum deviation ΔV_V of voltage V_V due to the falling edge of V_A, and the duration of the perturbation ($V_V < V_{DD}/2$) Δt can be derived as

$$(\Delta V_V)_{\max} = V_{DD} \frac{C_{AV}}{C_T} \frac{1}{S} (1 - e^{-S}),$$

$$(\Delta t) = t_f + R_{UV} C_T \ln\left[(1 - e^{-S})\left(\frac{2C_{AV}}{C_T} \frac{1}{S} - 1\right)\right],$$

where $S = (t_f / R_{UV} C_T)$.

In this case, the crosstalk effect manifests as a pulse on the line whose input is held constant, i.e., line V.

An analogous analysis for speedup and slowdown has not been made.

1.3 Existing test generation techniques for crosstalk noise

Logic level crosstalk fault models and PODEM based ATPG algorithms were presented in [11], [33], [36], [37]. In [11], the effect of the parasitic coupling was modeled as a logic pulse of width δ. An algorithm was presented for the detection of crosstalk induced signals considering several new logic values to represent a pulse. The set of logic values used in the algorithm are: 0 (logic 0), 1 (logic 1), X (undetermined), P0 (inverted pulse), P1 (non-inverted pulse), TU (rising transition), TD (falling transition), TUD (TU delayed signal), TDD (TD delayed signal), and a complementary set of the above associated with hazards. With these values, an algorithm based on PODEM was implemented. In that work, the limitation of propagation of the crosstalk signals was not considered, i.e., it has assumed that crosstalk signals were always strong enough to propagate to the primary outputs of the circuit. This model characterizes crosstalk effects as static hazards having a full voltage swing, and results in an overestimation of noise. Since crosstalk is a finite energy transient effect, test vectors generated using this model may not be able to actually propagate the noise to POs or flip-flops because of the inertia inherent to gates.

A more realistic model considering both width W and amplitude H of the coupling signal has been proposed [12], [36], and calculations on the number of gates that the pulse can penetrate were made. This model characterized a crosstalk signal as a square voltage pulse with an appropriate amplitude and width such that it is a more realistic model of a pulse, especially with respect to its propagation capabilities. In [12], an upper bound approach was used with the concept of covering signals. This concept

stated that a signal A covering another signal B will have a propagation capability greater than that of signal B. However, this upper bound is usually much greater than the actual propagation capability of the crosstalk signal. Therefore in [36] a modification was made by considering the width of the crosstalk signal as the time interval between the points where the signal passes the logic threshold of gates, taken to be $V_{DD}/2$. Then a penetration depth was defined in the following way. Given an ideal pulse with amplitude H and width W, the penetration depth is the maximum number of logic stages, k, such that the crosstalk signal produced at the output of the last stage has an amplitude greater than the logic threshold. The penetration depth can be used to determine whether or not a crosstalk signal is able to cause a logic effect at the output of a circuit, depending on the number of gates it has to traverse from the node where the signal was first produced.

Two algorithms for generating test vectors for crosstalk based on PODEM that take into account penetration depth have been proposed [36], [37]. In both algorithms a conventional 5-value logic (0, 1, X, D, D-Bar) was used, where D and D-bar represent the inverted and non-inverted spurious signals, respectively. It was assumed that a layout extractor existed, and was capable of identifying nodes where crosstalk can appear and calculating the corresponding penetration depth of crosstalk signals. The output of such an extractor was assumed to be a list of pairs of nodes, each of which was associated with a number representing the penetration depth of the crosstalk originated at that node. If at any time during the execution of these algorithms the number of gate levels that a crosstalk signal has propagated is greater than the penetration depth of that crosstalk signal, the propagation path is aborted and a new path is chosen.

In these algorithms the propagation of an inverted (non-inverted) crosstalk pulse was similar to the propagation of an error due to a stuck-at fault. Since a crosstalk pulse is created on the victim line by a transition on the affecting line, it is necessary to calculate two vectors so that a transition on the affecting line can be achieved. In the algorithm proposed in [36], the test generation process was carried out by a commercial ATPG tool, namely the stuck-at fault test generator called System HILO. Given a node where a crosstalk signal is suppose to be generated, a computation was made to find all the paths from this node to the outputs traversing a number of gates less than the penetration depth of that crosstalk signal. All paths traversing a number of gates greater than the penetration depth are blocked by inserting AND or NAND gates with one of their inputs connecting to GND. Therefore, a new circuit was created containing only the paths that the crosstalk signal can traverse. This modified circuit was then applied directly to the ATPG tool to obtain test vectors.

On the other hand, the algorithm presented in [37] consists of three phases. Initially, the first vector is calculated by setting the values of affecting and victim nodes. The values of these nodes are chosen following the controllability heuristic SCOAP [38]. In the second phase, the second vector is computed by setting the victim node to the same value as in the previous phase, and the affecting node to the opposite value (thus causing a transition in this node). The third phase sets unused primary inputs associated with the second vector to appropriate values in order to propagate the crosstalk signal to primary outputs. The propagation is assumed to be equivalent to the propagation of a D or D-bar value used for stuck-at faults, and the penetration depth is used in the third phase. All

three phases of the algorithm use conventional backtrace and backtrack procedures as used by stuck-at fault test generation algorithms.

In these approaches [36], [37] the dependency of detectability on the propagation ability of the crosstalk signal has been shown, but the penetration depth computation assumed that all gates (and/or all kind of gates) have the same capabitity to impede crosstalk propagation. In reality, however, some paths tend to filter out crosstalk noise. while others are very hazard-sensitive depending on the analog properties of the gates. Due to the non-linearity of CMOS gates, crosstalk noise may be attenuated or even amplified while propagating through a gate. Hence it is necessary to investigate the analog properties of CMOS gates to determine crosstalk propagation. In addition, the above models ignore timing of signals, i.e., they consider no gate delay and zero signal rise/fall times. Since the amplitude of a crosstalk pulse depends on the affecting line switching speed and the crosstalk delay has a strong relationship with a signal arrival time and rise/fall time (see Chapter 2), it is necessary to consider timing information in the test generation process.

Another approach for generating test for crosstalk was proposed in [33]. This approach uses the multiple backtrace technique and utilizes a "forward-evaluation" technique in its backtracking phase which searches for the right entry to select by propagating suggested values to minimize the number of backtracks. Therefore the efficiency of the test generation process is significantly improved. This approach also considered the signal timing information by taking into account variable gate delays so that signal arrival times could be computed. However, this approach still models crosstalk

as an ideal pulse with full voltage swing and assumes zero rise/fall times for signal transitions. Hence penetration capability of a pulse is not well characterized and again an overestimation of crosstalk noise strength may occur. The test vector generated using the model may not be able to propagate the actual crosstalk to primary outputs.

Therefore the ability to *efficiently* and *accurately create* a *large* crosstalk effect and *propagate* it with *minimal attenuation* has not been previously addressed.

1.4 Motivation and organization of the dissertation

As can be seen from the preceding sections, all the cited models and algorithms for characterizing and test generation for crosstalk noise involve trade-offs between computation speed and desired accuracy. In this work, we will focus on the development of a general methodology to analyze and obtain greater insight into the crosstalk phenomenon, and an efficient mixed-signal test generation mechanism where characteristics of crosstalk induced noise are accurately modeled.

This dissertation address the problems of validation and testing issues related to crosstalk. In Chapter 2 we will develop a general methodology to analyze and obtain greater insight into the crosstalk phenomenon. First the source of crosstalk effects will be described. Next a methodology is presented and used to characterize cases where inputs to one or both coupled lines have transitions with arbitrary transition times and directions. Our analysis starts with a model in the frequency domain (s domain) to obtain a closed form voltage transfer function. This is then transformed to obtain expressions in the time

domain. These expressions are used to characterize the amplitude, width, energy, and timing of the pulse, as well as the speedup or slowdown of transitions due to crosstalk. Experimental results show that process variations can have significant impacts on crosstalk effects. New design validation and test issues are identified, and a simple test generation scheme is presented.

Chapter 3 provides analytic models for propagating crosstalk noise through CMOS gates. Several new techniques for a 1st-order model are developed so that tests can be efficiently and accurately generated. These techniques includes new models for a CMOS inverter, methods to calculate inverter output response for pulse inputs, a method for collapsing CMOS gates into equivalent inverters, and a piece-wise linear model for pulses. These techniques are integrated into a test generation framework described in Chapter 4.

Chapter 4 presents a mixed-signal test generation process where characteristics of crosstalk induced noise are accurately modeled. This algorithm not only considers noise effects as new logic values, but also takes into consideration analog information such as finite noise energy and input arrival skews to accurately characterize noise strength. In addition, this ATPG algorithm includes the concept of gate delay, signal arrival time, signal strength and rise/fall times. Conditions for the creation of the worst-case coupling and propagation of a crosstalk effect are presented. We also present a new analog cost function that is used to guide the search process. By using the path delay information obtained in circuit preprocessing and/or the analog cost function, preferred paths can be selected during the backtrace as well as propagation process. A branch-and-bound

technique is also proposed to reduce the effort for searching through the whole PI combinations. While most ATPG algorithms attempt to only satisfy a set of logical constraints, our algorithm also maximizes an objective function. Experimental results show that our approach can generate tests for circuits of reasonable sizes (such as a functional unit) within acceptable amount of computation time.

Chapter 5 proposes possible future extensions to our work that focuses on improving the capability and efficiency of the test generator. Additional macromodels can be developed to enable the propagation of crosstalk effects via a wider range of circuit elements (complex COMS gates, dynamic gates, and latches) and under a wider range of conditions such as multiple crosstalk effects and simultaneous presence of crosstalk pulse and delays.

In Chapter 6 we present our conclusions.

Parts of the work presented in this dissertation have already been published. The analytic models for crosstalk delay and pulse under non-ideal inputs have appeared in the Proceeding of the International Test Conference, 1997 [30]. The test generation algorithm for crosstalk noise was presented in the Proceeding of the International Test Conference, 1998 [46] and 1999 [63].

Chapter 2

Analytic Models for Crosstalk Excitation

Traditionally, SPICE simulations have been used to estimate crosstalk noise in signal lines. Although accurate, these simulations are too time-consuming and inefficient for chip-level circuits. A rapid and acceptable accurate crosstalk noise estimation alternative is needed. In this chapter we develop a general methodology to analyze crosstalk to obtain insight into effects that are likely to cause errors in deep submicron high speed circuits. We focus on crosstalk due to capacitive coupling between a pair of lines. A methodology is presented and used to characterize cases where inputs to one or both coupled lines have transitions with arbitrary transition times and directions. Our analysis starts with a model in the frequency domain (s domain) to obtain a closed form voltage transfer function. This is then transformed to obtain expressions in the time domain. These expressions are used to characterize the amplitude, width, energy, and timing of the pulse, as well as the speedup or slowdown of transitions due to crosstalk. We first consider the case where crosstalk noise manifests as a pulse and characterize the maximum amplitude, width, energy and timing of this pulse. Closed form equations quantifying the dependence of these pulse attributes on the values of circuit parameters and the rise time of the input transition are derived. We also consider how crosstalk causes slowdown (speedup), i.e. increases (decreases) the rise/fall times and arrival time

of signals on coupled lines when their inputs have transitions in the opposite (same) directions. Expressions relating the slowdown (speedup) to circuit parameters, the rise/fall times of the input transitions, and the skew between the transitions are derived. We show that crosstalk effects can be significantly aggravated by variations in the fabrication process. New design corners are identified for validation of designs that have significant crosstalk effects. Finally, the results of our analysis provide conditions that must be satisfied by a sequence of vectors used for validation of designs as well as post-manufacturing testing of devices in the presence of significant crosstalk.

The chapter is organized as follows. In section 2.1 a brief review of source of crosstalk effects is described. In section 2.2 shows the impact of scaling and process variation on crosstalk effects. In section 0 new validation and test issues are discussed. In section 2.4 the proposed methodology to analyze crosstalk is presented, followed by the derivation of closed form expressions for the frequency domain transfer functions and time domain signal waveforms. In section 2.5 we discuss design and test issues for various crosstalk situations. Finally in section 2.6 we provide a summary.

2.1 Crosstalk Effects

In VLSI circuits it is very common to have wires running adjacent to one another. In submicron designs, due to the closer proximity of adjacent wires on the same layer, increase in the height of wires (relative to their widths), and increase in the switching speeds of signals, the parasitic coupling effects are significant. Coupling effects produce

interference between signals, referred to as crosstalk noise, and may increase or decrease signal delays and decrease signal integrity.

Parasitic coupling includes inductive and capacitive effects. There is a low inductance value that becomes significant at very high frequency in certain lines, such as V_{DD} and GND global buses, which are very long and wide (so R is comparable to ωL) and may conduct large switching current. For most signal interconnects it is still feasible to accurately model crosstalk without considering inductance because of the voltage-controlled nature of MOS devices [64]. Figure 2.1 shows a simple circuit with mutual capacitance C_m between two signal lines A and V. The values of the parasitic capacitance C_m can be determined as described in [57], [58], [59].

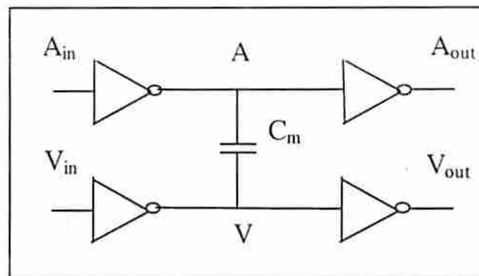


Figure 2.1 Simple circuit showing source of crosstalk due to capacitive coupling.

Crosstalk noise may cause undesirable effects including excessive overshoot, undershoot, glitches, addition signal delay and even a reduction in signal delay. These effects can lead to possible circuit malfunction (permanent errors) and increased power dissipation. Figure 2.2 shows the effects of crosstalk obtained by SPICE simulation on the signal V in Figure 2.1. In the simulation, we assume that wire resistance and coupling

capacitance can be modeled as a lumped resistance, R_{line}, and a lumped capacitance, C_m, (lumped RC model), respectively, and the size of the transistors in the inverters and lengths of the metal wires are chosen to obtain a nominal driver output response with a rise time of about 130ps, which is realistic assuming a clock period of 4 ns. For reliable operation some aspect of the worst case crosstalk pulse, such as energy or maximum amplitude, should be bounded and the input patterns that maximize these aspects of crosstalk should be used during design validation. In Figure 2.2(a) we see that a pulse is generated on line V, which should ideally have a constant zero, due to a rising transition on line A. Figure 2.2(b) shows that when A and V have transitions in the same (different) directions, the results is a decrease (increase) in the signal transition time (speedup/slowdown) of signal V.

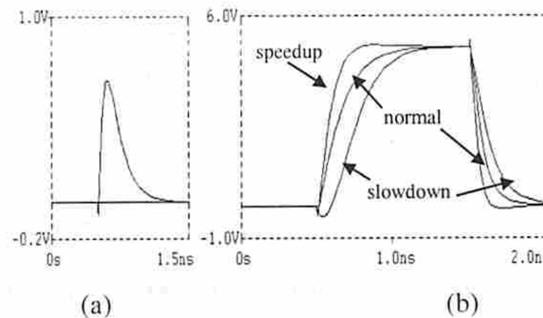


Figure 2.2 Crosstalk waveforms of signals in Figure 2.1: (a) crosstalk pulse; (b) crosstalk decreases/increases signal transition times (speedup/slowdown).

2.2 Technology Trends and Process Variations

2.2.1 Technology Scaling

In this section we will study the effect of scaling on crosstalk noise for several deep sub-micron technologies. The technology parameters are based on the SIA roadmap [32] and extracted values from [47], [48]. Table 2.1 lists the main characteristic and interconnect parameters we used for our scaling experiments.

The experimental circuit consists of two unbalanced drivers with a 4000μm long affecting line with minimum width driven by the larger driver (20 times minimum size inverter), and a 2000μm long victim line with minimum width driven by the smaller driver (5 times minimum size inverter). Both lines are metal 4 lines running in parallel with minimum spacing between them. While scaling down the device sizes for different technologies, the affecting and victim line lengths are also scaled down according to the trend of the interconnect scaling projections presented in [49].

Table 2.1 Interconnect parameters for various technologies [32], [47], [48]. W is the min. width; R and C are unit length resistance and total capacitance; AR is the aspect ratio, Ca, Cf, and Cm are area, fringing and coupling capacitance, respectively.

Tech. (um)	0.35	0.25	0.18	0.15	0.13	0.1	0.07
clock (MHz)	400	700	1100	1300	1600	2100	2500
V_{DD}(V)	2.5	2.5	1.8	1.5	1.5	1.2	1.0
Metal eff. resistivity (μΩ-cm)	3.3	3.3	2.2	2.2	2.2	2.2	1.8
M1 Interconnect							
W (um)	0.4	0.3	0.22	0.17	0.15	0.11	0.08
Pitch (um)	0.6	0.45	0.33	0.27	0.25	0.16	0.12
R (Ω/um)	0.15	0.19	0.29	0.40	0.56	0.67	0.7
C (fF/um)	0.17	0.19	0.21	0.23	0.25	0.27	0.27
AR	1.5	1.8	2.0	2.1	2.4	2.7	3
M4 Interconnect with min. width and space							
W (um)	1.0	0.76	0.56	0.47	0.38	0.28	0.20
R (Ω/um)	0.04	0.05	0.06	0.076	0.11	0.17	0.18
Ca(fF/um)	0.031	0.025	0.021	0.021	0.018	0.018	0.017
Cf(fF/um)	0.046	0.042	0.040	0.040	0.040	0.042	0.037
Cm(fF/um)	0.056	0.072	0.086	0.090	0.100	0.107	0.119

Figure 2.3 shows the effect of technology scaling on crosstalk noise. Crosstalk effect tends to increase for each successive technology generation because of two possible reasons: 1) the increase in aspect ratio and decrease in minimum spacing makes the coupling capacitance more dominant; and 2) the reduction of interconnect dimensions increases the line resistance and makes it more difficult to discharge the crosstalk voltage.

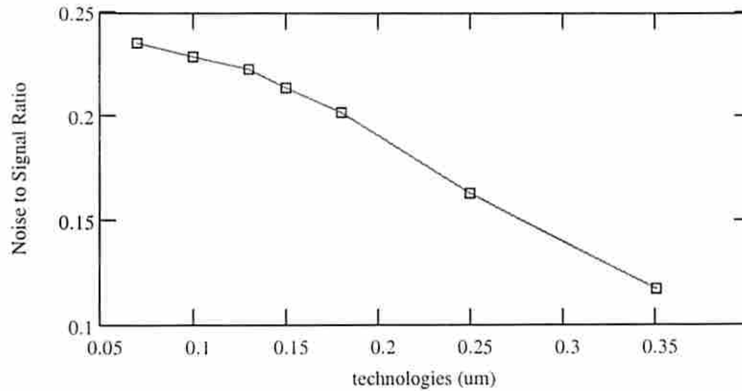


Figure 2.3 Noise trend for different technologies.

2.2.2 Impacts of Process Variations

In this section we illustrate the effects on crosstalk due to variations in the values of electrical parameters caused by manufacturing. The results are obtained by SPICE simulation of the circuit shown in Figure 2.10 . Different values for the electrical parameters are selected that are consistent with the correlations that exist in actual process data [22]. The parameter data presented is based on a 0.8 micron process with a single poly and three metal layers.

The delay of a signal V is influenced by capacitive coupling, relative drivers strength, and the transitions that occur at A and V. For simplicity, only the case where V has a rising transition is considered.

The nominal delay values are calculated for each case using nominal values for all electrical parameters. The worst case behaviors are excited by selecting the appropriate value for each parameter. From the discussion about dependence in section 2.4, the

maximum delay is obtained by selecting the maximum value of mutual capacitance between the interconnects (C_m), the minimum value of interconnect to substrate capacitance (C_a), minimum value of C_v, the maximum transistor gain values (minimum equivalent on-channel resistance) for line A driver, the minimum transistor gain for line V driver, and the minimum and maximum values of line resistances for the affecting and victim lines, respectively. The minimum delay case situation is obtained in a similar way. All parametric values selected are within the acceptable range of the process. We assume inputs switch simultaneously with a rise/fall time of 100ps.

The delay values obtained are shown in Table 2.2. Due to process variations, the worst case delay varies by about 25% around the mean value. Cross-coupling plus process variation increase the delay from normal to the worst case maximum by almost 85% (from 217.3ps to 400ps); the deviation between the minimum and the maximum delay is 234.5ps (from 165.5ps to 400ps), i.e. 108% of the nominal delay.

The height of the crosstalk pulse is determined for the case where A has a falling transition while V_{in} is stable at 0V. Table 2.3 shows the values of the pulse height. For the worst case process values, the pulse height is 43% larger than its nominal value.

These examples show that there are significant variations about the mean values for crosstalk pulse and delay due to process variations. The noise margin in a typical circuit may not be large enough to tolerate the effects of both crosstalk and process variations.

Table 2.2 Effect of process variations on crosstalk delay (pico seconds)

Simulation Cases	Mean Delays (ps)	Minimum Delay (ps)		Maximum Delay (ps)	
		Value	% deviation from mean	Value	% deviation from mean
Nominal Delay	217.3	177.4	18.4	269.5	24.0
Crosstalk Slowdown	315.0	248.0	21.3	400.0	27.0
Crosstalk Speedup	165.5	128.2	22.5	205.7	24.3

Table 2.3 Effect of process variations on crosstalk pulse height

Simulation Cases	Mean Value (V)	Minimum Height (V)		Maximum Height (V)	
		Value	% deviation from mean	Value	% deviation from mean
Crosstalk pulse height	0.65	0.43	33.8	0.93	43.0

2.3 New Design Validation and Test Issues

In high speed circuits, signal integrity and timing are important issues for correct circuit operations. From the previous section, crosstalk can have a significant impact on signal integrity and delay and even result in erroneous circuit operation.

Due to the high complexity of crosstalk analysis, the development of a methodology to identify pairs (or, in general, sets) of lines where crosstalk noise is likely to exceed the noise or timing margin is essential to any practical validation methodology. Since process variations have a significant impact on the severity of crosstalk effects, parts of a circuit where crosstalk does not cause errors for nominal values of parameters can operate erroneously for other parameter values in the design envelope. The correctness of a design at all points in the design envelope is verified by validating the circuit at various design corners, i.e., extreme combinations of parameter values where

the design is likely to fail. However, the design corners that are commonly used during validation do not represent the combination of parameter values where the severity of crosstalk is maximized [21]. In addition, the noise-to-signal ratio tends to increase as feature sizes reduce. Hence validation for crosstalk noise is essential in designing high speed circuits.

If a design is found to fail at some extreme points in a design envelope, a circuit may not necessarily be redesigned, especially if redesign would make the attainment of some design objectives impossible or have an impact on a product's schedule. Thus, each manufactured device must be tested to ensure that it works correctly. Therefore we need to develop a test generation framework for crosstalk noise. Since the amplitude of a crosstalk pulse depends on the affecting line switching speed and the crosstalk delay has a strong relationship with signal arrival times and rise/fall times (see section 2.4), it is necessary to consider analog properties and timing information of signals in the test generation process. Therefore the ability to *efficiently* and *accurately create* a *large* crosstalk effect and *propagate* it with *minimal attenuation* has not been previously addressed. Thus, it is important to develop models to analyze crosstalk effects, and integrate these models into a mixed-signal test generator for crosstalk noise.

2.4 Crosstalk Model and Analysis

To obtain insight into the nature of crosstalk and its dependence on the circuit parameters associated with the coupled lines, consider the lumped model of capacitive coupling shown in Figure 2.4.

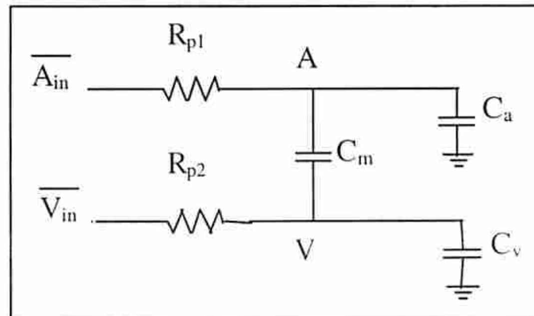


Figure 2.4 Capacitive coupling model.

In this model, each pulling resistance, R_{p1} or R_{p2}, is composed of the line resistance and the on-channel resistance associated with the line driver, where we assume the complementary device is off immediately after the inputs are applied. In [13], [14] it is shown that the impact of neglecting the short circuit current is small provided that the transition time is short. The load capacitances, C_a and C_v, consist of the line capacitance and the gate capacitance of the load driven by the line. Thus the line driver is equivalent to a pulling resistance, and the coupling network can be viewed as a network of capacitors (C_m, C_a, C_v). Compared with the simplified model in [11], which assumed a linear rise/fall time on the node A, our expanded model allows for a more general model of the signals A_{in} and V_{in}, not only in terms of their switching rates but also their relative skew.

2.4.1 Driver Modeling and Approximation of Distributed RC Network Using Lump Models

Using the lumped model in Figure 2.4 one can derive analytical expressions for crosstalk waveforms. For example, by using Laplace transformations we can obtain an expression for crosstalk in the s-domain, which we can transform back to the time domain. However, as interconnect lengths become longer, the error introduced by the lumped model increases. Hence two enhancements are made to (1) model the driver considering the rise time of the input signal, and (2) account for the distributed nature of the interconnect RC network by using a model for effective coupling and load capacitance. Once this is done, the accuracy of this model approaches the accuracy of a distributed model, but the resulting analytical equations are nearly as simple as a lumped model.

2.4.1.1 Driver Modeling

The most popular representation of a driver driving a wire consists of an input voltage source and an ON-channel resistance, as shown in Figure 2.5. However, because of the non-linearity of the driver characteristics and the finite input transition time, certain modification must be made to minimize the error in this modeling.

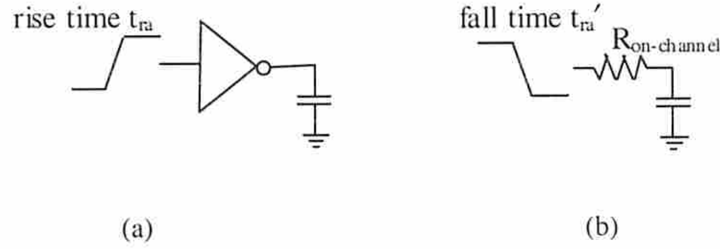


Figure 2.5 (a) An input signal with transition time t_{ra} applied to a driver; (b) equivalent circuit.

First, the ON-channel resistance of a CMOS inverter is a function of V_{ds} and I_{ds} of a MOS transistor. Instead of using only the resistance in the linear region of the device, an expression for $R_{on\text{-}channel}$, namely $0.5(V_{ds}/I_{ds})_{Vds=0.5VDD} + 0.5(V_{ds}/I_{ds})_{Vds=VDD}$, is often used [52]. Second, assume that the transition time of the input signal to the driver is t_{ra}. The output transition time of the driver, t_{ra}', is given by: $t_{ra}' = t_i + t_s + t_c$, where t_i, t_s, and t_c are the intrinsic delay dependency, input slope dependency, and the interconnect load dependency, respectively [50]. The intrinsic delay dependency t_i is empirically expressed as $t_i = k_i V_{DD} C_j / I_{dsat}$, where V_{DD} is the supply voltage, I_{dsat} is the saturation source-to-drain current, and C_j is the junction capacitance. The term k_i is a "fitting coefficient", and k_i is approximately 0.4 for many technologies [50]. The term t_i is usually small (~5ps) and is independent of the input transition time.

The input slope dependency, t_s, is a linear function of the input transition time of the signal applied to the driver, i.e., $t_a = k_a t_{ra}$, where k_s is a technology dependent fitting parameter and is typically between 0.1-0.2 for deep sub-micron technologies.

The interconnect load dependency can be expressed as

$$t_c = k_c \delta C_i \left[\left(\frac{V_t - 0.1V_{dd}}{I_{dsat}} \right) + \left(\frac{V_{dd} - V_t}{2I_{dsat}} \ln \left(\frac{19V_{dd} - 20V_t}{V_{dd}} \right) \right) \right],$$

where C_i is the interconnect capacitance, V_t is the transistor threshold voltage, δ is an empirical constant accounting for the loss due to short circuit current and is typically equal to 1.2, and k_c is an empirical expression to account for capacitance shielding caused by interconnect resistance [51]. The value of k_c is given by

$$k_c = 1 - \left(\frac{R_i}{R_i + R_d\sqrt{3}} \right)^4,$$

where R_i and R_d are the interconnect line resistance and device on-channel resistance, respectively [50].

The technology dependent fitting coefficients in the above equations can be obtained by running SPICE for several calibration cases. The model error for the output transition time prediction compared with SPICE simulation is shown to be less than 10 % for various interconnection lengths up to 10000um [50].

2.4.1.2 Approximation of Distributed Network Using Lump Models

To account for the distributed nature of the RC interconnect, the following approximation model has been proposed [50]. Based on the Elmore Delay model [53], the lumped line capacitance C_a and C_v are scaled by a factor of 0.5. The lumped coupling capacitance C_m is scaled by a semi-empirical and technology dependent factor $\varphi = (1 - \eta)e^{-t_{ra}'/\tau} + \eta$. In this expression t_{ra}' is the output transition time of the driver

described in the previous section. τ is a function of circuit parameters and is expressed as

$$\tau = \sqrt{[R_{p1}(C_a + C_m) + R_{p2}(C_v + C_m)]^2 - 4R_{p1}R_{p2}(C_aC_v + C_aC_m + C_vC_m)}\ .$$

The parameter η accounts for the presence of the victim line driver resistance, and is given by $\eta = 0.5[1+R_{vd}/(R_{vi}+R_{vd})]$, where R_{vi} and R_{vd} are the victim line interconnect resistance and driver resistance, respectively. η is close to 1 if the interconnect resistance is negligible, and monotonically decreases to 0.5 as interconnect becomes more dominant. The scaling factor φ is equal to η for a slow transition time, but monotonically approaches 1 for signals with fast transition times.

2.4.2 Analytical Equations from Crosstalk Waveforms

We de-couple the system shown in Figure 2.4 into an input waveform stage, a driver characterization stage and a coupling network stage. By doing this, we can employ more complex models to obtain more accurate results, take into account input waveforms other than ideal step functions, and thus analyze crosstalk induced speedup and slowdown (delay).

By using Laplace transformations, we can accomplish the following:

1. for the input waveform stage, obtain the Laplace transfer expressions for fairly complex inputs,
2. for the driver characterization stage, obtain the transfer function of the line driver model at a desired degree of accuracy,

3. for the cross-coupling network, obtain the transfer function from A to V.

By cascading these three stages we can obtain an expression for crosstalk in the s-domain that can be transformed back into the time domain. The analytic response derived is based on the first order model of MOS device behavior, commonly referred as the LEVEL 1 model, assuming that the channel modulation is negligible. This model was selected because more sophisticated models that take into account higher order effects are intractable for analytic manipulation. The insights gained from the results obtained using this simple model are sufficiently useful for our applications.

2.4.2.1 Analysis of Crosstalk Pulse

To illustrate the analysis procedure, consider the case of a positive crosstalk pulse induced on node V (victim line) due to a rising transition at node A (affecting line). The input A_{in} to the inverter driving the affecting line in Figure 2.1 is a falling transition, and the input V_{in} to the inverter driving the victim line is kept high so that the victim line should remain at a constant low. The values for coupling capacitance and load capacitance can be obtained by techniques described in section 2.4.1.2.

After the input to A_{in} is applied, the pulling device (PMOS) of the inverter driven by A_{in} can be modeled by its on channel resistance, R_{on}, connecting A to VDD; the corresponding NMOS device is off. The inverter driven by V_{in} can be modeled by the channel resistance of its NMOS device connecting V to GND. For computational convenience, we normalize VDD to be 1 and GND to be 0. Figure 2.6(a) shows the

circuit model for the situation just described. Figure 2.6(b) shows an equivalent circuit of Figure 2.6(a).

Some notation used throughout the rest of this chapter is described next:

A - node or signal on line A,

A(t) - voltage at A in time domain,

A(s) - voltage at A in frequency domain,

$A_{exp}(t)$ - voltage at A when $A_{in}(t)$ is an exponential signal and $V_{in}(t)$ is stable at low (or high),

$A_{step}(t)$ - voltage at A when $A_{in}(t)$ is a step function and $V_{in}(t)$ is stable at low (or high),

$A_{su}(t)$ - voltage at A when A_{in} and V_{in} are exponential inputs with identical directions of transition (speedup),

$A_{sd}(t)$ - voltage at A when A_{in} and V_{in} are exponential inputs with opposite directions of transition (slowdown).

(In the above, A and V can be interchanged to derive another set of notation.)

Let "H" indicate a transfer function and its subscript indicate a node name or the conventional output/ notation.

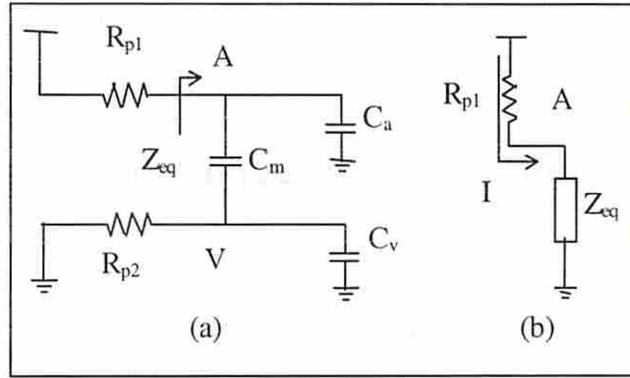


Figure 2.6 Circuit model for crosstalk pulse analysis. (a): circuit model for a positive pulse induced on V due to a rising transition on A; (b) an equivalent circuit.

From Figure 2.6, solving for the impedance at node A, we have

$$Z_{eq} = (C_m + C_a) - C_m \cdot \frac{V(s)}{A(s)} .$$

The transfer function from A to V is

$$\frac{V(s)}{A(s)} = \frac{sC_m}{\frac{1}{R_{p2}} + s(C_m + C_v)} .$$

Therefore, Z_{eq} can be expressed as

$$Z_{eq} = \frac{(C_m + C_a) + sR_{p2}C_t}{1 + sR_{p2}(C_m + C_v)} ,$$

where $C_t = C_m C_v + C_m C_a + C_a C_v$.

Now, $A(s) = H_A = H_{A/A_m} H_{A_m}$, where H_{A/A_m} is the characteristic transfer function of the low pass circuit composed of the pulling resistance R_{p1} and the equivalent reactance Z_{eq},

namely $H_{A/A_{in}} = \frac{\tau}{s+\tau}$, where $\tau = 1/R_{p1}Z_{eq}$. If the input is a unit step function then $H_{A_{in}} = 1/s$ and

$A(s) = H_A = (\frac{\tau}{s+\tau})\frac{1}{s}$.

One can think of this transfer function as the product of the driver characteristic transfer function stage and an input waveform transformation stage. Thus, the transfer function of node V to the input is

$$V(s) = H_V = H_{V/A} \cdot H_{A/A_{in}} \cdot H_{A_{in}}$$

where $H_{V/A}$ is the coupling network transfer function.

Carrying out the algebra we get

$V(s) = H_V = \frac{C_m}{R_{p1}C_t}(\frac{1}{w-u})(\frac{1}{s-w} - \frac{1}{s-u})$, and

$A(s) = H_A = \frac{1}{s} - \frac{1}{w-u}(\frac{1}{s-w} - \frac{1}{s-u})(s + \frac{C_m + C_a}{R_{p2}C_t})$,

where w, u are solutions to the quadratic equation

$$s^2 + s(\frac{R_{p1}(C_m + C_a) + R_{p2}(C_m + C_v)}{R_{p1}R_{p2}C_t}) + \frac{1}{R_{p1}R_{p2}C_t} = 0,$$

and both w and u are negative.

The time domain voltage waveform V(t) is obtained by taking the inverse Laplace transformation of its corresponding s-domain expression, resulting in

$$V(t) = (\frac{C_m}{R_{p1}C_t})(\frac{1}{w-u})(e^{wt} - e^{ut}).$$

For arbitrary input waveforms instead of step functions, we can modify the above input waveform transformation stage. Inputs such as step functions, ramps, exponentials

or combinations of the above are commonly seen in electrical models and are easy to use in transformation analysis. For example, assume the input to the driver stage is an exponential rising waveform with known transition time (time constant). The output transition time (time constant x) of the output waveform ($1 - e^{-t/x}$) for the driver is obtained using the output transition time prediction technique described in the previous section. The s-domain expression of this waveform is [(1/s)-(1/(s+1/x)]. The transfer function at A, namely H_{Aexp}, under this exponential input is

$$H_{Aexp} = H_{A/A_{in}} \cdot H_{A_{in}} = \frac{\tau}{s+\ \tau}\left(\frac{1}{s} - \frac{1}{s+1/x}\right) = \left(\frac{1}{s} - \frac{1}{s+\ \tau}\right)\frac{1}{s+1/x} = H_{Astep} \cdot \frac{1/x}{s+1/x},$$

where H_{Astep} is the transfer function H_A discussed previously (the voltage seen by the driver stage is a step function), and the subscripts are used to indicate the type of input waveform.

Hence, an exponential input results in a modulation term 1/x(s+1/x). Using this technique, the corresponding victim line time domain response is

$$V(t) = \frac{C_m}{R_{p1}C_t}\left(\frac{1/x}{(w+1/x)(w-u)}e^{wt} + \frac{1/x}{(u+1/x)(u-w)}e^{ut} + \frac{1/x}{(w+1/x)(u+1/x)}e^{-t/x}\right).$$

Finding the maximum amplitude of the pulse can be done by differentiating the above equation and setting the result to zero. However, the above equation contains 3 exponential terms and it is very difficult to find a closed-form expression for the amplitude. Hence we expand the exponential terms by using the Taylor series expansion technique. The most important process in the Taylor series expansion is finding the expansion center, t_0, where the approximation error is minimal. Since we know that the time when the maximum amplitude of the pulse at V occurs will not be earlier than when

the step input is applied, and is near the time when the affecting line finishes its transition, one can empirically derive the following expression for the expansion center:

$$t_0 = \xi \cdot t_{step}(1 - e^{-x/t_{step}}) + t_{step},$$

where ξ is an empirical fitting constant (typically it is 1.2), and $t_{step} = \ln(u/w)/(w-u)$ is the time when the maximum amplitude occurs for the case of a step input.

As the time constant x decreases to 0, t_0 monotonically decreases to t_{step}. As x increases, the expansion center t_0 increases. The useful range for this approximation for time constants is from 0 to 250 ps, which includes the range of rise/fall times in todays technologies.

By expanding the expression for the crosstalk pulse into a Taylor series, it becomes a polynomial equation and can be solved directly to find the time (t_x) when the maximum amplitude occurs. Then the maximum crosstalk amplitude is obtained by substituting t_x back into the crosstalk pulse equation. For the derivation of the crosstalk amplitude expression please see Appendix A. The approximation error in estimating the amplitude using this technique is less than 3%.

In Figure 2.7 we show the dependence of a crosstalk pulse amplitude on various circuit parameters. The default values for the following parameters are: R_{p1} = 120ohms, R_{p2} = 250ohms, C_m = 300fF, C_a = 174fF, and C_v = 87fF. Figure 2.7(a) shows the crosstalk pulse on line V due to a unit step transition and an exponential transition on line A_{in}, where for the latter case the rise time is 100ps, i.e. x = 44. It can be seen that the pulse due to a step transition at A_{in} has a larger maximum amplitude than when an exponential

transition occurs at A_{in}. Figure 2.7(b) shows that crosstalk pulse amplitude decreases as the input transition time increases. This is because there are less high frequency components contained in slower inputs (if we do an energy spectrum analysis) and hence less energy passed through the coupling capacitance to create the crosstalk pulse. Figure 2.7(c) shows the effect of affecting-to-victim line driver ratio on maximum crosstalk amplitude for a fixed coupling capacitance. It is seen that as the driver ratio increases, the maximum amplitude also increases and tends to saturate. The amplitude approaches a value determined by the coupling capacitance and load capacitance of both lines. Therefore, the total energy that can be coupled to the victim line is fixed, even if the driver ratio becomes extremely large. Figure 2.7(d) shows the amplitude as a function of the affecting (victim) line resistance Rp1 (Rp2) for a fixed value of victim (affecting) line resistance Rp2 (Rp1). It is seen that as Rp1 gets larger, i.e. the driving capability of the affecting line driver becomes weaker, the dV/dt value on the affect line decreases and hence the energy coupled through the coupling capacitance becomes less and results in a smaller pulses. On the other hand, as the victim line resistance Rp2 increases, it is more difficult for the victim line to discharge the crosstalk voltage. Thus the maximum pulse amplitude becomes larger. Figure 2.7(e) shows the impact of coupling capacitance, affecting and victim lines load capacitance (line capacitance plus load capacitance) on crosstalk amplitude. Typically, the longer the interconnect lines, the larger the coupling capacitance. Therefore, we expect the magnitude of the crosstalk pulse to be larger as the coupling line length become longer. However, a very long line has a large RC and thus the affecting line driver gets overloaded. Thus the dV/dt change on the affecting line

become smaller, and consequently the crosstalk pulse amplitude does not increase at the same rate as the coupling length increases. Figure 2.7(e) also shows the impact of the line loads on pulse amplitude, for fixed Rp1 and Rp2. It is seen that a larger load will result in a smaller pulse at the victim line. Similarly, for the affecting line the larger the line load becomes, the smaller the crosstalk. This is because a larger line load C_a implies a larger RC on the affecting line, and hence the dV/dt decreases and results in smaller crosstalk amplitudes. For the victim line load C_v, the crosstalk amplitudes decreases as C_v increases. This is because C_v can hold charge and compensate the charging process from the affecting line through the coupling capacitance. Therefore the crosstalk amplitude become smaller.

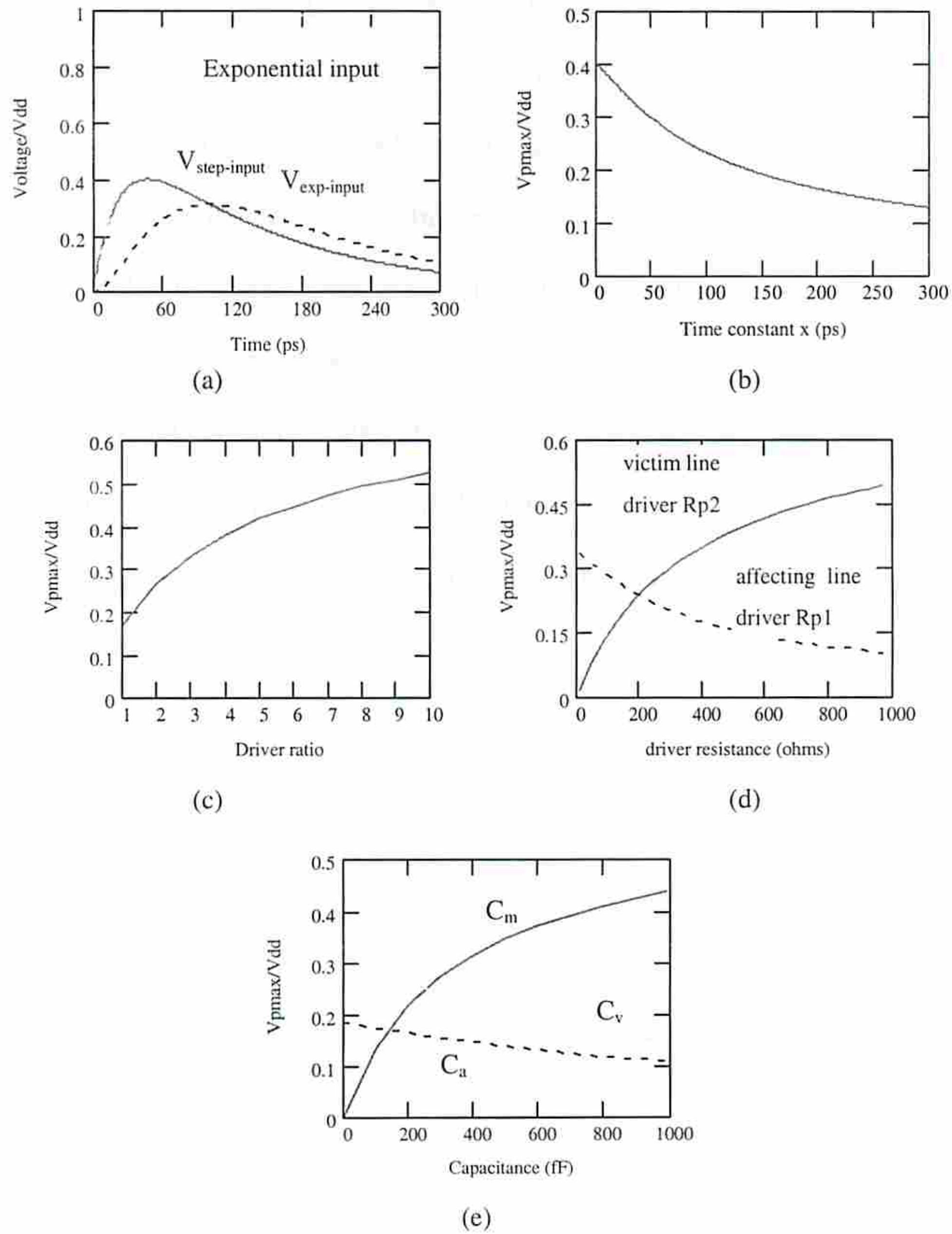


Figure 2.7 (a) Crosstalk pulse at V due to exponential and step inputs at A_{in}; (b) maximum amplitude vs. input transition time (time constant); (c) maximum amplitude vs. affecting/victim driver ratio; (d) maximum amplitude vs. affecting and victim lines resistance (driver resistance plus line resistance); (e) maximum amplitude vs. coupling capacitance, affecting and victim lines load capacitance (line capacitance plus load capacitance).

2.4.2.2 Analysis of Crosstalk Delay

By using the techniques described above we can analyze effects such as (1) when both signals A and V change simultaneously and in the same direction to cause signal speedup, (2) change in the opposite direction to cause extra delay, or (3) change with a relative timing skew.

Consider the case where the affecting line A has a falling transition and line V has a rising transition. The equivalent circuit model is shown in Figure 2.8. Here we assume that exponential waveforms of time constants x and y are applied to A_{in} and V_{in}, respectively, and the A_{in} signal has a time skew of z units with respect to signal V_{in}, where z can be positive or negative.

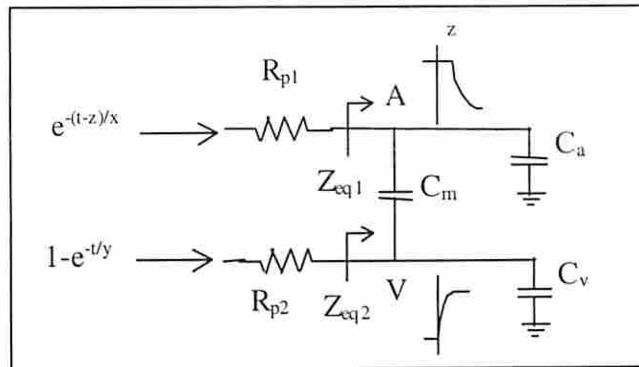


Figure 2.8 Equivalent circuit for crosstalk delay analysis.

By using the techniques described in the previous section, the Laplace transformation for shifting the time-axis, and proper initial conditions, we have the following results (for detail derivation please see Appendix B).

$$A_{sd}(s) = [(\frac{1}{s+\frac{1}{R_{p1}Z_{eq1}}})(\frac{1/x}{s+1/x}) + \frac{1}{s+1/x}]e^{-sz} - \frac{e^{-sz}}{s} + \frac{1}{s},\text{ and}$$

$$V_{sd}(s) = (\frac{1}{s})(\frac{\frac{1}{R_{p2}Z_{eq2}}}{s+\frac{1}{R_{p2}Z_{eq2}}})(\frac{1/y}{s+1/y}),\text{ where}$$

$$Z_{eq1} = [\frac{1}{sA_{sd}(s)-1}(sA_{sd}(s)(C_m + C_a) - sC_mV_{sd}(s) - C_a - C_m)],\text{ and}$$

$$Z_{eq2} = C_m + C_a - C_m\frac{A_{sd}(s)}{V_{sd}(s)} + \frac{C_m}{V_{sd}(s)}.$$

By solving the above system of equations, we obtain

$$A_{sd}(s) = [A_{step}(s)(\frac{1/x}{s+1/x}) + \frac{1}{s+1/x}]e^{-sz} + (\frac{C_m}{R_{p2}C_t}\frac{1}{(s-w)(s-u)})(\frac{1/y}{s+1/y}) + [\frac{1}{s} + \frac{e^{-sz}}{s}],\text{ and}$$

$$V_{sd}(s) = V_{step}(s)(\frac{1/y}{s+1/y}) - (\frac{C_m}{R_{p1}C_t}\frac{1}{(s-w)(s-u)})(\frac{1/x}{s+1/x})e^{-sz},\text{ where}$$

$$A_{step}(s) = \frac{1}{(s-w)(s-u)}(s + \frac{C_m + C_a}{R_{p2}C_t}),\text{ and}$$

$$V_{step}(s) = \frac{1}{s} - \frac{1}{(s-w)(s-u)}(s + \frac{C_m + C_v}{R_{p1}C_t}).$$

We can interpret these equations in the following way.

Total response on line V = (signal due to step input at V_{in})*(modulation on line V) + (coupling from line A)* (modulation on line A)*(skew on line A)

where $(\frac{1/y}{s+1/y})$ represents the modulation on line V due to the finite signal transition time, $(\frac{1/x}{s+1/x})$ represents the modulation on line A, and e^{-sz} represents the time skew of line A with respect to the signal on line V.

$A_{sd}(s)$ can be interpreted in a similar manner except that there are also terms resulting from initial conditions.

The waveforms for A and V are given by the expressions

$$A_{sd}(t) = A_{exp}(t) + \frac{1}{y}[\frac{c}{(w+1/y)(w-u)}e^{wt} + \frac{c}{(u+1/y)(u-w)}e^{ut} + \frac{c}{(w+1/y)(u+1/y)}e^{-t/y}],$$

$$V_{sd}(t) = V_{exp}(t) - [\frac{1}{x}[\frac{b}{(w+1/x)(w-u)}e^{w(t-z)} + \frac{b}{(u+1/x)(u-w)}e^{u(t-z)} + \frac{b}{(w+1/x)(u+1/x)}e^{\frac{-(t-z)}{x}}]U(t-z).$$

where

$$A_{exp}(t) = \{\frac{1}{x}[\frac{w+a}{(w+1/x)(w-u)}e^{w(t-z)} + \frac{u+a}{(u+1/x)(u-w)}e^{u(t-z)} + \frac{a-1/x}{(w+1/x)(u+1/x)}e^{\frac{-(t-z)}{x}}] + e^{\frac{-(t-z)}{x}}\}U(t-z)$$
$$+ U(t) - U(t-z),$$

$$V_{exp}(t) = 1 - e^{\frac{-t}{y}} - \frac{1}{y}[\frac{w+f}{(w+1/y)(w-u)}e^{wt} + \frac{u+f}{(u+1/y)(u-w)}e^{ut} + \frac{f-1/y}{(w+1/y)(u+1/y)}e^{\frac{-t}{y}}],$$

$$a = \frac{C_m + C_a}{R_{p2}C_t}, \quad c = \frac{C_m}{R_{p2}C_t}, \quad b = \frac{C_m}{R_{p1}C_t}, \quad f = \frac{C_m + C_v}{R_{p1}C_t}, \quad C_t = C_m C_v + C_m C_a + C_a C_v,$$

and U(t) is a unit step function.

The terms in $A_{sd}(t)$, except for $A_{exp}(t)$, contribute to the slowdown effect caused by the mutual capacitance. The terms in $V_{sd}(t)$ contribute in a similar way.

Similar equations can be derived for speedup. The waveforms for A and V are given by the expressions

$$A_{su}(t) = A_{exp}(t) - \frac{1}{y}[\frac{c}{(w+1/y)(w-u)}e^{wt} + \frac{c}{(u+1/y)(u-w)}e^{ut} + \frac{c}{(w+1/y)(u+1/y)}e^{-t/y}],$$

$$V_{ud}(t) = V_{exp}(t) + [\frac{1}{x}[\frac{b}{(w+1/x)(w-u)}e^{w(t-z)} + \frac{b}{(u+1/x)(u-w)}e^{u(t-z)} + \frac{b}{(w+1/x)(u+1/x)}e^{\frac{-(t-z)}{x}}]U(t-z).$$

By using the sensitivity analysis on the equations described previously, we can observe that the severity of crosstalk is directly proportional to the mutual capacitance and line V resistance, and inversely proportional to the line A resistance and the load capacitance on each line.

Figure 2.9 shows the degree of speedup or slowdown due to coupling effects, assuming that the input signals switch simultaneously, i.e. $z = 0$. The circuit configuration is shown in Figure 2.10. The circuit consists of two unbalanced drivers with a 4000μm long and 4μm wide metal2 affecting line A driven by the larger driver (32μ/0.35μ PMOS and 16μ/0.35μ NMOS), and a 1000μm long and 2 μm wide metal1 victim line V driven by the smaller driver (8μ/0.35μ PMOS and 4μ/0.35μ NMOS). Figure 2.10 also shows the dimensions of the components. R, C and gain values used for analysis and simulations have been extracted from a layout, where R_{m1}, R_{m2} are metal1 and metal2 line resistances, C_{m1g}, C_{m2g} are metal1 and metal2 line-to-substrate capacitances, and C_m is the mutual capacitance.

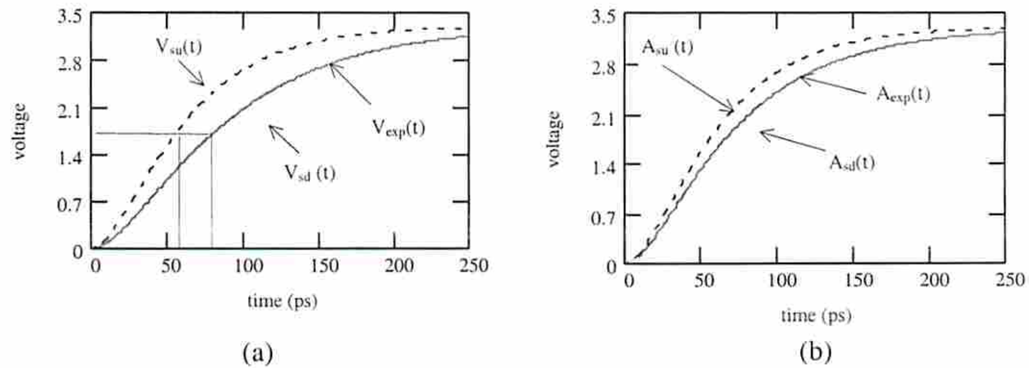


Figure 2.9 Crosstalk speedup and slowdown effects assuming simultaneously switching inputs where both inputs have a transition time of 100ps. (a) effects on victim line; (b) effects on affecting line.

Consider the case where V has a rising transition and A remains constant. Then from Figure 2.9(a) we see that V reaches $V_{DD}/2$ = 1.65 volts at about t = 77ps. Now if A simultaneously has a rising transition, then V reaches 1.65 volts at t = 54ps, i.e. 23ps earlier. This illustrates the concept of crosstalk speedup.

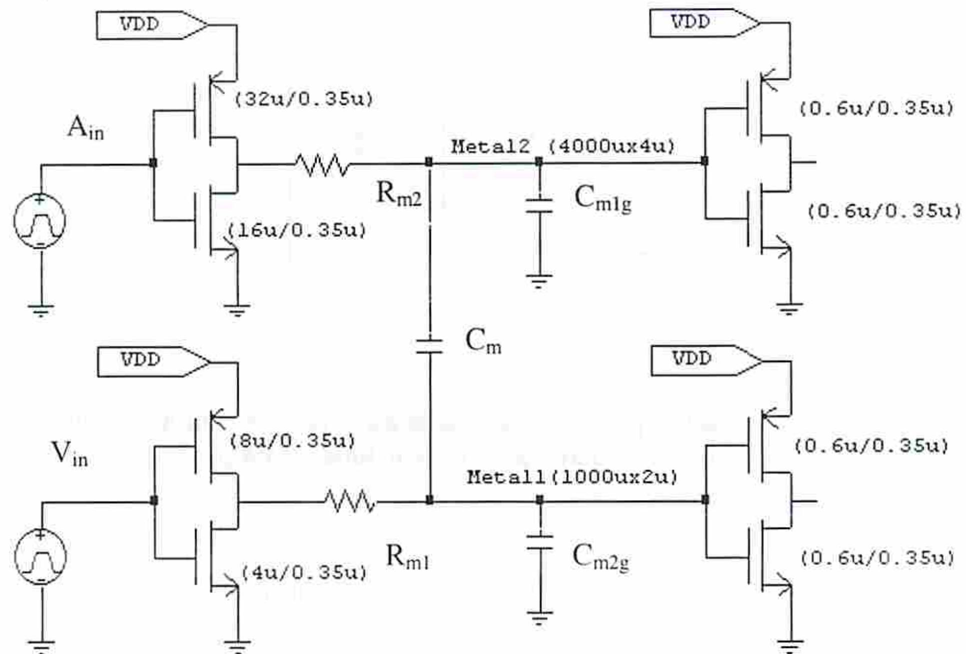


Figure 2.10 Circuit used to study influence of input signal properties and circuit parameters on crosstalk.

2.4.3 Dependence of Crosstalk Effects on Input Transition Times and Skews

In this section we investigate in more detail the dependence of crosstalk on input transition times and skew.

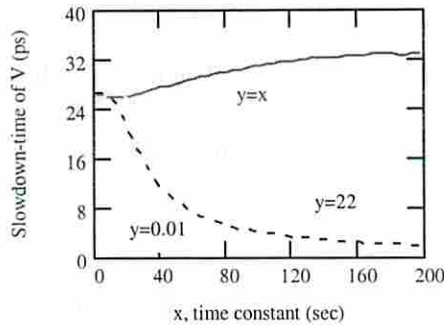
Let the delay time of a falling transition on a line be t_d when the other line is static, $t_{d\text{-}su}$ when both lines have transitions in the same direction, and $t_{d\text{-}sd}$ when both lines have transitions in opposite directions. The speedup-time due to the coupling effect is (t_d - $t_{d\text{-}su}$), and the slowdown-time is ($t_{d\text{-}sd}$ - t_d). Figure 2.11(a) shows the effects of slowdown with respect to input waveform switching rates, i.e. the time constants, x and y, of the

exponential inputs. For simplicity, we again assume both signals switch simultaneously. As x decreases, the exponential waveforms $e^{-t/x}$ and $(1-e^{-t/x})$ approach ideal step functions. In modern CMOS technologies the signal rise/fall times range from 50ps to 300ps, thus x ranges from 22 to 130. The curve with y = 22 corresponds to the case when the victim line input V_{in} has a rise time of 50ps; the one with y = .01 corresponds to a step function.

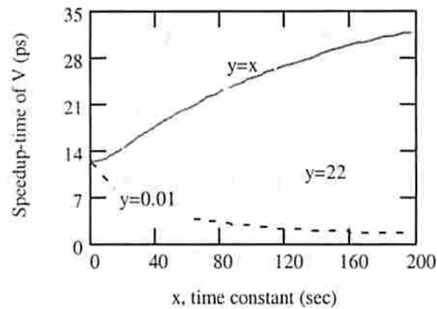
From Figure 2.11(a) we can see that when the input signal to the victim line is kept at a fixed switching rate, then the faster the affecting line changes the larger the slowdown of the victim line. Also for the case x=y, we see that the absolute amount of slowdown increases as x increases. For example, an exponential signal with a rise/fall time of 50ps (x = 21.7, t_d = 39ps) has a slowdown-time of 26ps and one with a rise/fall time of 200ps (x = 83.33, t_d = 108ps) has a slowdown-time of 30ps. However, the percentage change in delay decreases as both x and y increase. The slowdown-time in the former case (50ps) represents a 67% increase in delay, while that in the later case represents a 28% increase in delay. This implies that slow transition signals have a smaller effect than fast transition signals.

Similar results are obtained for the case of speedup and shown in Figure 2.11(b). Again for the case x=y, we see that the absolute amount of speedup increases as x increases. For example, an exponential signal with a rise/fall time of 50ps (x = 21.7, t_d = 39ps) has a speedup-time of 12ps and one with a rise/fall time of 200ps (x = 83.33, t_d = 108ps) has a speedup-time of 22ps. However, the percentage change in delay decreases as both x and y increase. The speedup-time in the former case (50ps) represents a 30%

decrease in delay, while that in the later case represents a 20% decrease in delay. This implies that slow transition signals have less effect than fast transition signals.



(a)



(b)

Figure 2.11 (a) The victim line slowdown-time vs. input switching rates; (b) the victim line speedup-time vs. input switching rates.

We now consider the amount of speedup and/or slowdown as a function of z, the time skew between the two signal transitions when A_{in} and V_{in} have rise/fall time of 100ps. Figure 2.12 shows the voltage waveforms on both lines, assuming the transition on the affecting line occurs first (z=25ps). In the time interval $0<t<z$, A either pre-charges or discharges V. Eventually, when V_{in} changes, A and V affect each other and lead to a

speedup or slowdown. From Figure 2.12 we can see that the coupling effect between these two lines are different due to the difference in line driver strengths and loads.

Figure 2.13 shows the influence of the input signal time skew z on the amount of speedup and slowdown where A_{in} and V_{in} have rise/fall times of 100ps.

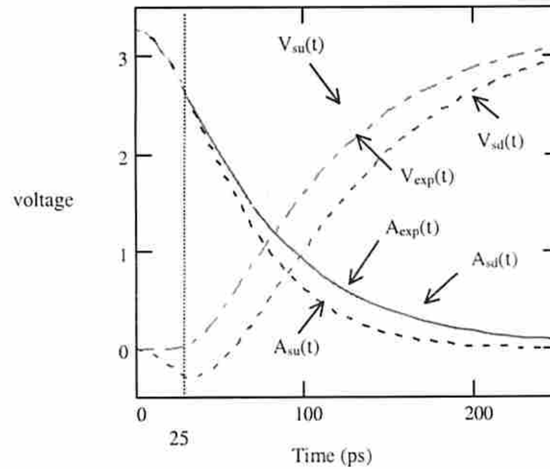


Figure 2.12 Voltage waveforms on affecting and victim lines for z = 25 ps.

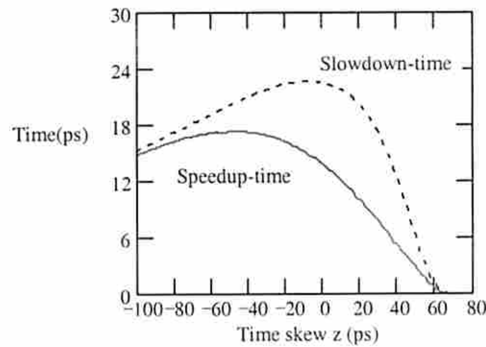


Figure 2.13 Victim line speedup-time and slowdown-time vs. skew z.

From Figure 2.13 we observe that as z increases the amount of speedup and slowdown on V both first increase and then decrease. For the speedup situation, if A

switches from low to high (high to low) earlier than V, then A helps charge (discharge) V before V changes, increasing the speedup. The amount of speedup reaches its maximum value when the coupling effect from A is maximum, i.e. where the pulse at V due to A is maximum. After that, the remaining effect of A on V starts to dissipate and hence the speedup decreases. Also we can observe that for our example the slowdown is maximum when the signals switch simultaneously. This can be explained by considering the fact that a rapid change in the voltage at A transfers charge to node V via the coupling capacitance C_m. If the transition at V occurs concurrently with the transition at A, then the entire charge transferred is discharged via the pull down of the inverter driving line V, increasing its fall time. On the other hand, if A switches earlier than V, then some of the transferred charge is discharged via the pull up of the inverter driving the line V prior to its switching. Hence only a part of the charge transferred from A is discharged when V begins to fall, decreasing the slowdown. If A switches later than V, then for a time z, V transits toward its target value before A affects it, hence the slowdown is decreased. For z greater than some fixed amount z_o, A cannot impact V since V has already reached 50% of VDD, which defines the delay time t_d. As the driver ratio increases (decreases), the skew for the maximum slowdown to occur also increases (decrease), i.e., not necessarily switching simultaneously. The skew associated with the maximum slowdown occurs at approximately $z_{\max}=(1-e^{-(r-1)})\cdot t_{peak}\cdot k_1$ (see Appendix C), where r is the ratio of the drivers strength, k_1 is a empirical constant, and t_{peak} is the time when the pulse at V due to A is maximum. z_{max} is approximately equal to $t_{peak} \cdot k_1$ for large driver ratios, and is zero if the drivers are the same size.

2.5 Design Validation for Crosstalk Noise

From previous discussion, crosstalk can have a significant impact on signal delay and even result in erroneous circuit operation. For example, consider a clocked D flip-flop. Due to crosstalk effects, a transition on D may arrive early and/or the clock edge may arrive late. These may cause hold-time violations. Also, if the transition on D arrives late and/or that on the clock arrives early, a setup-time violation may occur. Either of these scenarios can either cause meta-stability or the flip-flop to go into the wrong state.

Due to the high complexity of crosstalk analysis, the development of a methodology to identify pairs (or, in general, sets) of lines where crosstalk noise is likely to exceed the noise margin or timing is essential to any practical validation methodology. The results presented above provide a methodology to identify such pairs of lines by showing that the severity of crosstalk depends on three main factors, namely (a) the circuit parameters associated with the coupled lines, (b) the nature of inputs that can be applied to these lines, and (c) the nature of the circuit driven by them. The last factor has been discussed in some detail in [21] where it is shown how certain properties of the circuit driven by the coupled lines determines the type of crosstalk effect, e.g., pulse or delay, and for each type of effect, the characteristic of the effect that in turn determine if an error can occur. In this context, the results of our analysis can be used to identify pairs of circuit lines where crosstalk may be significant and hence should be analyzed explicitly. We have also shown that process variations have a significant impact on the severity of crosstalk. Hence, parts of circuits where crosstalk does not cause errors for nominal values of parameters can operate erroneously for other parameter values in the

design envelope. Our analysis helps identify new design corners where the candidate lines must be simulated to ensure correct operation even in the presence of crosstalk. For example, we have shown that crosstalk interference is proportional to the mutual capacitance, ratio of strengths of drivers driving the coupled lines, and inversely proportional to the load capacitance on each line. This is obviously different from the most commonly used *fast* and *slow* design corners.

Even if a design is found to fail at some extreme points in a design envelope, a circuit may not be redesigned, especially if redesign would make the attainment of design objectives impossible. In such a case, the resulting circuit will typically be guaranteed to operate at a vast majority of points within the envelope, but not all. For such a design, each manufactured device must be tested to ensure that it works correctly. The above results specify conditions that a test must satisfy to detect errors caused by crosstalk. For example, it shows that a sequence of two patterns must be applied to cause nearly simultaneous transitions in opposing directions to invoke worst case crosstalk slowdown. The resulting slowdown must then be propagated along paths with low delay slacks to circuit outputs. The application of a test sequence that satisfies these conditions will identify devices with excessive crosstalk slowdown. (Note that traditional path delay testing tests for excessive delay along logical paths in the circuit, while here excessive delays are caused by coupling between logically unrelated paths.) In a similar manner, the above results provide conditions that a sequence of patterns must satisfy to detect errors caused by other crosstalk effects.

Figure 2.14 shows an example of test pattern generation for crosstalk pulse. Assuming that we want to create a positive pulse at V, at least a or b must be set to 1 to provide a constant low at V. However, to decrease the total pulling resistance to GND, only input a is set to 1. Since a sharp transition on A is preferred, both c and d are assigned rising transitions. In addition to the pulse excitation, to propagate the resulting pulse through the next stages, proper values must be set on side fan-in's of each gate, i.e. values for e, f, g must be set accordingly. Backward implication of these conditions will give rise to a sequence of test patterns.

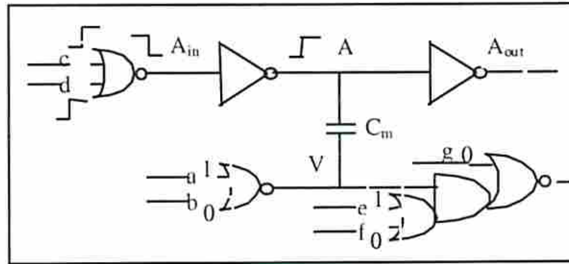


Figure 2.14 Example circuit for test vector generation.

2.6 Summary

The objective of this chapter is to develop a general methodology to analyze crosstalk in order to obtain insight into effects that are likely to cause errors in high speed VLSI circuits. We studied crosstalk due to capacitive coupling between a pair of lines. Closed form equations quantifying the dependence of the pulse attributes on the values of circuit parameters and the rise time of the input transition were derived. These expressions show that the severity of the crosstalk pulse is directly proportional to the

coupling capacitance and the ratios of the strengths of the drivers driving the two lines, and inversely proportional to the load capacitance on each line. These facts can be used to identify pairs of circuit lines where crosstalk may be significant and hence should be analyzed explicitly. Further, it is shown that while the maximum amplitude of the crosstalk pulse diminishes rapidly as the rise/fall time of the input increases, the energy of the pulse is almost independent of the input rise/fall time for a realistic range of rise/fall time values (see Appendix A). If the rise/fall time of the input to a candidate pair of lines is known to be large, then it may not be necessary to analyze the effect of crosstalk. We also studied how crosstalk causes speedup/slowdown when signals change in the same/opposite directions. Qualitatively, the dependence of slowdown and speedup on circuit parameters is similar to that observed for crosstalk pulse. Also, it was found that the faster the transition at A, the greater is the slowdown at V. Finally, it was found that the skew for the maximum crosstalk slowdown to occur is proportional to the ratio of the drivers driving the two lines. If the drivers are the same size, crosstalk slowdown is the highest when both inputs have simultaneous transitions. The magnitude of slowdown decreases as the skew between the input transitions increases.

The crosstalk effect was shown to be significantly aggravated by variations in the fabrication process. The significance of the process variations necessitates the identification of new design corners for validation, some of which have been presented here. Finally, the results of our analysis provide conditions that must be satisfied by a sequence of vectors used for validation as well post-manufacturing testing.

For 0.18μm technology, the aspect ratio of and spacing between wires are such that the capacitance between metal wires on the same layer exceeds the interlayer capacitance. Since there is a high likelihood of having long parallel wires on the same layer, we believe the effects of crosstalk will be more severe. Finally, the results of our analysis provide conditions that must be satisfied to detect errors caused by crosstalk by a sequence of vectors used for validation as well post-manufacturing testing. For example, it shows that a sequence of two patterns must be applied to cause nearly simultaneous transitions in opposing directions to invoke worst case crosstalk slowdown. The resulting slowdown must then be propagated along paths with low delay slacks to circuit outputs. The application of a test sequence that satisfies these conditions will identify devices with excessive crosstalk slowdown. (Note that traditional path delay testing tests for excessive delay along logical paths in the circuit, while here excessive delays are caused by coupling between logically unrelated paths.) In a similar manner, the above results provide conditions that a sequence of patterns must satisfy to detect errors caused by other crosstalk effects.

Chapter 3

Analytic Models for Noise Propagation

To accurately propagate noise through gates, we need to (1) characterize the noise waveform, (2) construct gate transfer functions, and (3) compute output noise waveforms. Since many CMOS gates in a random logic circuit have different electrical characteristics, our approach is to first model CMOS logic gates as equivalent inverters and then calculate the output response of noise through this gate using the transfer function of the equivalent inverter.

In Section 3.1 a new inverter model is presented that reduces the error found in other approaches caused by neglecting the short circuit current. In Section 3.2 we propose a method to determine an inverter that is equivalent, in the sense of a transfer function, to a given CMOS logic gate (NAND, NOR). This method can also be generalized to complex gates. In section 3.3 we characterize the noise waveform and calculate the propagated output noise waveform through the equivalent inverter.

3.1 A New Inverter Model

Several analytic models have been proposed for the transient response of CMOS inverters [13], [14], [25], [26]. Although these models take into account the influence of

the input waveform on the propagation delay, the short-circuit current is neglected. For current technology where the signal transition time is near 100ps and the gate load is in the range of 10-50fF, neglecting short-circuit current can result in errors in the estimation of the propagation delay and output waveform. Since crosstalk noise is a finite energy transient phenomenon, we proposed an improved model for a CMOS inverter to take into account the short-circuit current so that the error in estimating the propagated noise can be significantly reduced. The derivations assume a rising input transition. Similar results have been obtained for falling input transitions.

Consider the CMOS inverter in Figure 3.1(a). We wish to determine the falling output waveform $V_o(t)$ due to a rising input ramp $V_{in}(t)$ with rise time t_r. Assume all circuit capacitance is lumped into one grounded load capacitance C at the inverter's output and all voltages have been normalized with respect to V_{DD}. The charging of the capacitance C can be expressed by

$$I_p = -I_n - C\frac{dV_o}{dt},$$

where

$$I_n = \beta_n\left[(V_{in} - v_{tn})V_o - V_o^2/2\right], \qquad for \ (V_{in} - v_{tn}) > V_o, \text{ and}$$
$$= \frac{\beta_n}{2}(V_{in} - v_{tn})^2, \qquad for \ (V_{in} - v_{tn}) < V_o;$$

$$I_p = \beta_p\left[(V_{in} - 1 - v_{tp})(V_o - 1) - (V_o - 1)^2/2\right], \ for \ (V_{in} - v_{tp}) < V_o, \text{ and}$$
$$= \frac{\beta_p}{2}(V_{in} - 1 - v_{tp})^2, \qquad for \ (V_{in} - v_{tp}) > V_o.$$

Here β_n (β_p) is the gain factor, and v_{tn} (v_{tp}) is the transistor threshold voltage normalized with respect to V_{DD} of the NMOS(PMOS) transistors.

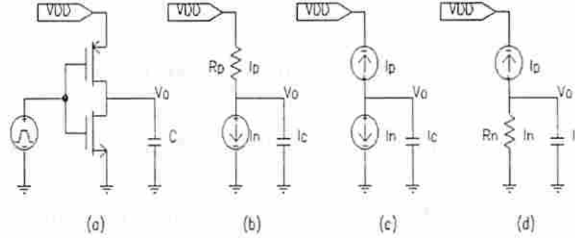


Figure 3.1 CMOS inverter and its corresponding model when N and P MOS transistors operate in different modes: (a) circuit, (b) PMOS in linear and NMOS in saturation mode, (c) both in saturation mode, and (d) NMOS in linear and PMOS in saturation mode.

When the input is first applied, the NMOS (PMOS) is in the saturation (linear) region and can be modeled as shown in Figure 3.1(b), where we replace the NMOS by a current source and the PMOS by a resistance. As long as the PMOS is in the linear region, the circuit can be characterized by the differential equation

$$\frac{1-V_o}{R_p} = C\frac{dV_o}{dt} + \frac{\beta_n V_{DD}}{2}(V_{in} - v_{tn})^2 .$$

With the initial condition $V_o = 1$ when $V_{in} = v_{tn}$, integration yields

$$V_o = P \cdot e^{\frac{-(t - t_r v_{tn})}{R_p C}} + A(V_{in} - v_{tn})^2 + B(V_{in} - v_{tn}) + D, \tag{3-1}$$

where

$K = \beta_n V_{DD}/2C$, $P = 2R_p^{\,3}C^3K/t_r^{\,2}$, $A = -R_pCK$, $B = 2R_p^{\,2}C^2K/t_r$, and

$D = 1 - 2R_p^{\,3}C^3K/t_r^{\,2}$.

However, the on-channel resistance R_p of the PMOS transistor in this model is not constant during the input transition. R_p is small (P-channel is fully ON) when the input is small and becomes very large when the PMOS transistor saturates to become a current source. Taking this non-constant property into account we modify the channel resistance as a function of input waveform, namely, we set

$$R_p = \frac{1}{\left|\beta_p V_{DD}\left(V_{in} - 1 - v_{tp}\right)\right|},$$

where $V_{in}(t) = t/t_r$.

When the input is rising and the output voltage drops to $(V_{in}-v_{tp})$, the PMOS transistor goes into saturation. The circuit can now be modeled as shown in Figure 3.1(c) and can be described by the equation

$$\frac{\beta_p V_{DD}}{2}\left(V_{in} - 1 - v_{tp}\right)^2 = \frac{\beta_n V_{DD}}{2}\left(V_{in} - v_{tn}\right)^2 + C\frac{dV_o}{dt}.$$

Integrating the above equation we obtain

$$V_o = \frac{\beta_p V_{DD} t_r}{6C}\left(V_{in} - 1 - v_{tp}\right)^3 - \frac{\beta_n V_{DD} t_r}{6C}\left(V_{in} - v_{tn}\right)^3 + M, \tag{3-2}$$

where M is a constant and can be obtained by using the boundary condition ($V_o = V_{in}-v_{tp}$) in both equations (3-1) and (3-2).

As the output voltage continues to drop, the NMOS transistor will eventually operate in the linear region. The circuit can now be modeled as shown in Figure 3.1(d). The equations characterizing this region are similar to the case in Figure 3.1(b).

Figure 3.2 shows the result of our new model. The input waveform is assumed to be a ramp having a rise time of 250ps, and the load capacitance is 15fF. We use a rise

time of 250ps because, as shown in [30], when an affecting line has a transition with a 100ps rise time, the slope of the rising edge of the crosstalk noise on the victim line is about 250ps. The results using our model match SPICE results very well except for the tail portion of the response. Note that the result based on ignoring the PMOS transistor has a significant error.

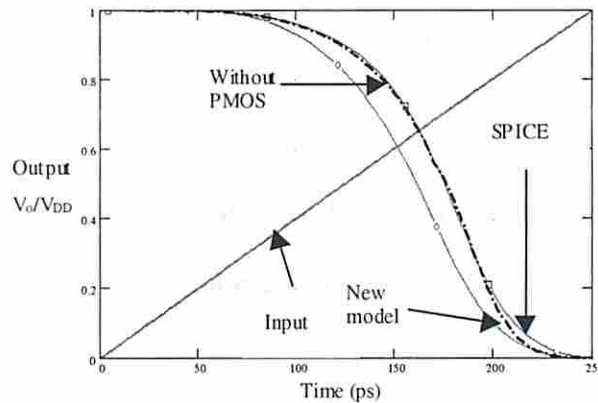


Figure 3.2 Comparison of analytic result of proposed model and SPICE simulations.

3.2 A Method to Collapse CMOS Gates

In this section we deal with the problem of propagating a pulse (noise) through a NAND or NOR gate. We set the side fan-in's to their non-controlling values. Our approach for computing the output noise for a general CMOS gate is to collapse the gate to an equivalent inverter and then apply the results in section 3.1. Collapsing techniques have previously been used for computing propagation delay [26], [27], [28], [29]. The methods presented in [26] treat series transistors as series resistors and add the widths of parallel devices. This leads to an inaccurate estimate of delay. The approaches described

in [27], [28] need either pre-characterization or DC analysis to determine some necessary parameters, which is technology dependent and is not applicable in the ATPG process. Although the approach in [29] provides a good estimation of propagation delay, the predicted output waveforms do not match well with SPICE simulations. Since the propagation of the noise depends heavily on the gate's response, we have developed a new but simple approach to collapse CMOS gates into equivalent inverters.

3.2.1 Series MOS

The effective transconductance, β_{eff}, of n series-connected transistors is traditionally approximated as β/n. This approximation is valid only when the input is a step function, all transistors operate in their linear regions, and they all have the same β value. Consider the pull-down NMOS chain of a CMOS NAND gate in Figure 3.3(b), where the V_{DS} and/or V_{GS} of each MOSFET in the series-connected chain is smaller than that of the inverter (Figure 3.3(a)). Assume that all devices have identical β values. Also assume that there are no more than 5 MOSFETs connected in series. When the input transition is applied, the switching MOS first operates in the saturation mode and then moves into the linear region. In addition, during the first part of the input transition all transistors above the switching MOSFET operate in saturation and all those below the switching MOSFET operate in the linear region. This results in the primary source of error in the use of the β/n approximation. Thus, to take this into account we need to estimate β_{eff} under various conditions of operations.

When the input transition first occurs, both the NMOS in Figure 3.3(a) and the switching NMOS in Figure 3.3(b) are in the saturation region and thus V_{GS} determines the device current. For the single NMOS in Figure 3.3(a), assume $V_{GS} = v_{inv}$ is the input voltage at which $V_{out} = 0.5$ (i.e. $V_{DD}/2$). For the switching NMOS in Figure 3.3(b) to conduct the same amount of current so that V_{out} can drop to $0.5+\sum_i V_{DS}^{\,i}$, the input voltage applied to the switching device must be $v_{inv} + \sum_i V_{DS}^{\,i}$, where i ranges over all q transistors below the switching NMOS. At the instant that the switching NMOS moves from the saturation region into the linear region, the voltages across the q MOSFETs below the switching device are as indicated in Figure 3.3(b). Hence the summation term is approximately equal to 0.14×q, and the estimated input voltage is (v_{inv} + 0.14×q).

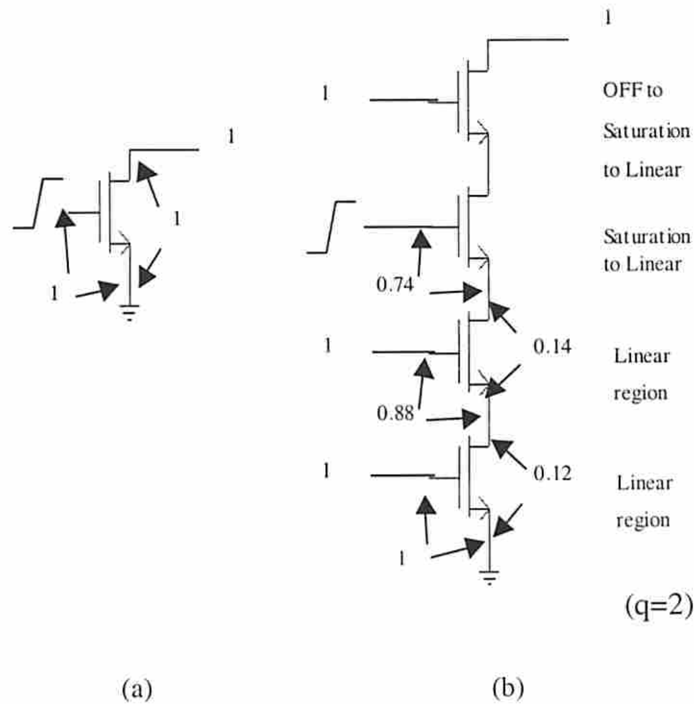


Figure 3.3 Pull-Down NMOS chains; (a) single NMOS, (b) series connected NMOS, all values normalized w.r.t. V_{DD}.

Therefore when the switching NMOS is in the saturation region and the PMOS transistor with its own effective β_p in the linear region, the effective β for the pulldown network is

$$\beta_{eff} \cong \beta\left[\frac{V_{inv}}{\left(v_{inv} + 0.14 \times q\right)}\right] = \beta_1.$$

As V_{out} continues to drop, the PMOS transistor in the pull-up network will go into its saturation region and change its effective β_p. To deal with this situation, one can either modify the effective β_p directly or continue to modify the β_{eff} of the pull-down network to compensate for the change in β_p. We chose the later approach because we can use interpolation to easily approximate the modification for β_{eff}.

Before developing the interpolation approach, consider the next region where the switching NMOS goes into the linear region. Here the NMOS can be modeled as an on-channel resistor except that its V_{DS} is not the whole output voltage drop and the devices above the switching NMOS will move into the linear region one by one. Hence instead of using the traditional β_{eff} value of β/n, a correction term is needed. The effective transconductance when a switching NMOS is in the linear region is approximated by $\beta_{eff} = \beta_2 = m\frac{\beta}{n}$ where m is a constant determined empirically. We have found that $m = 0.75$ (shown in Figure 3.4) works well when the number of devices below the switching NMOS range from 0 to 5, which is usually the case for a NAND gate.

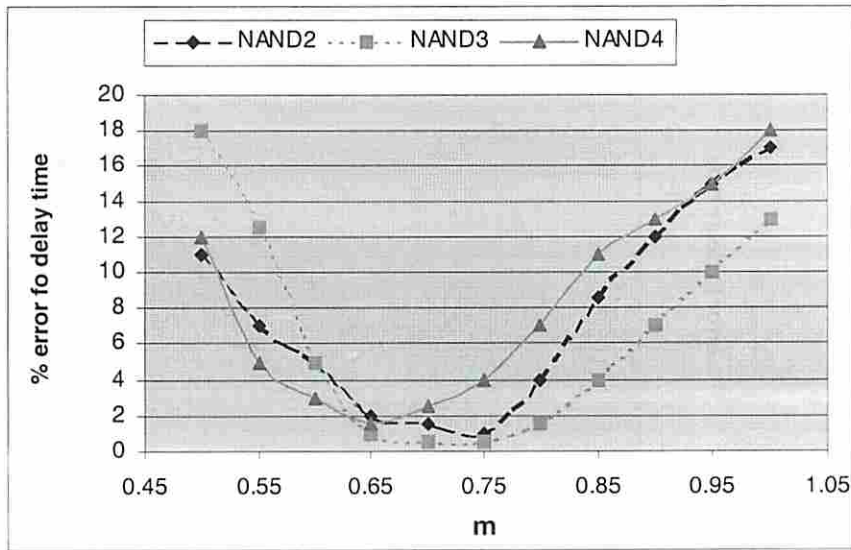
Returning to the region where both the complementary PMOS and the switching NMOS are in the saturation region, by interpolation from the other two cases presented, we get

$$\beta_{eff} = \beta[(\frac{V_{DD} - V_{in}}{V_{DD} - v_{inv}})(\frac{\beta_1 - \beta_2}{\beta}) + \frac{m}{n}].$$

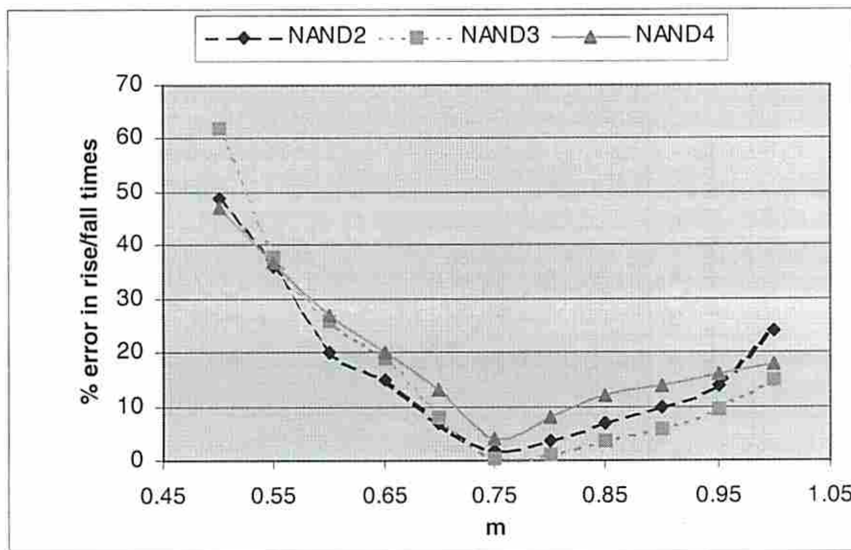
Because the above approximation involves the input V_{in} which makes it difficult to find a closed-form solution, β_{eff} can be further approximated by

$$\beta_{eff} = \alpha\beta_1 + (1 - \alpha)\beta_2,$$

where α is an experimental constant. We have found that $\alpha = 1/3$ (shown in Figure 3.5) works well when the number of devices below the switching NMOS range from 0 to 5.

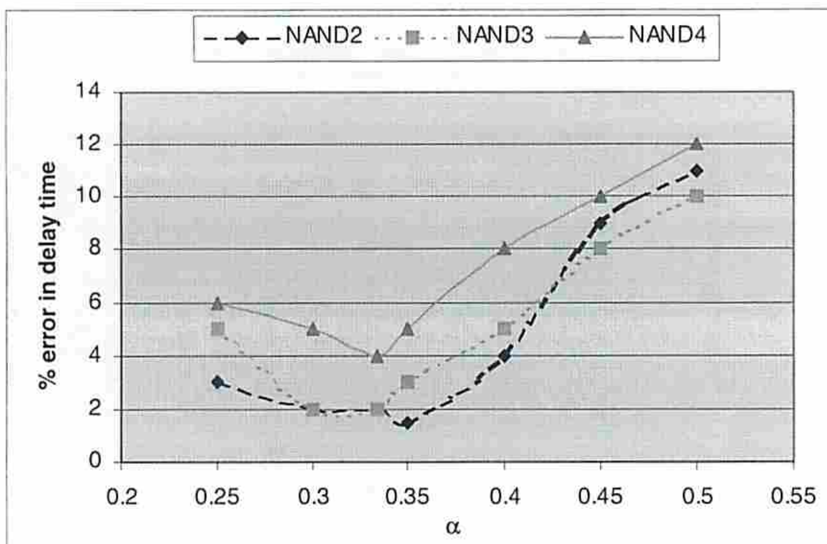


(a)

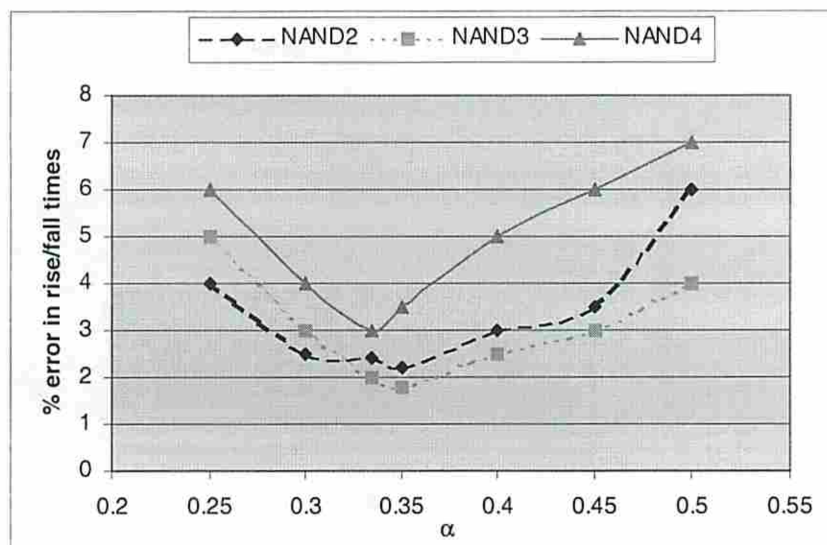


(b)

Figure 3.4 Experimental results for selecting the empirical constant m: a) percentage error w.r.t. the output signal delay time; b) percentage error w.r.t. the output signal rise/fall times.



(a)

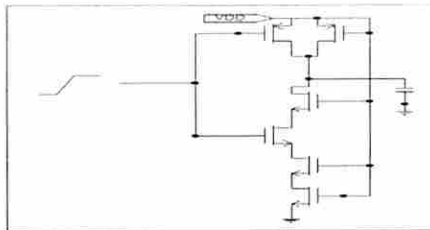


(b)

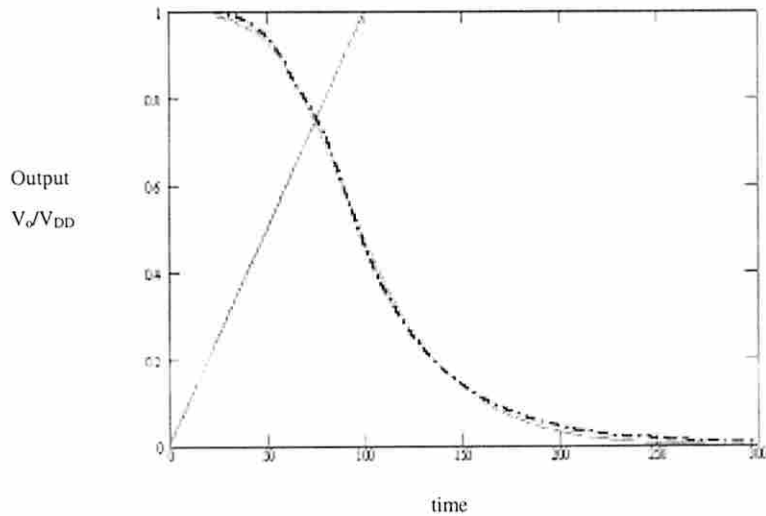
Figure 3.5 Experimental results for selecting the empirical constant α: a) percentage error w.r.t. the output signal delay time; b) percentage error w.r.t. the output signal rise/fall times.

3.2.2 Parallel MOS

When propagating noise through a CMOS gate, since all side fan-in's have to be set to their non-controlling values, the parallel network is reduced to a single transistor whose gate is connected to the switching input. Figure 3.6 illustrates the collapsing technique. The input is a ramp having a rise time of 100ps and the load is 20fF. All device sizes are (4u/0.8u) and we assume all capacitances are lumped into the output load. The dash curve is the output waveform of the equivalent inverter obtained using the collapsing technique, and the solid curve was obtained by SPICE simulation.



(a)



(b)

Figure 3.6 (a) Circuit for collapsing NAND gate into an equivalent inverter, (b) model and SPICE simulation results.

3.2.3 Internal Capacitance

Internal capacitances are usually ignored when they are small compared to the load capacitance, but often this is not the case when a large number of transistors are connected in series. The easiest way to take into account the effects of internal capacitance is to add it to the load capacitance. But this results in an overestimation of the propagation delay and output transition time. Hence our approach is to model MOS devices as ON-channel resistances and use the Elmore delay model to obtain the equivalent load capacitance at the gate output.

Consider the pull-down NMOS chain shown in Figure 3.7(a). Since the transistor M2 is ON before the input is applied to the switching device (M1), the internal capacitance C_{p2} is completely discharged. Also M0 is ON so C_{p1} is charged. When the input is applied, all transistors are turned ON and hence can be modeled by their linear resistance as shown in Figure 3.7(b).

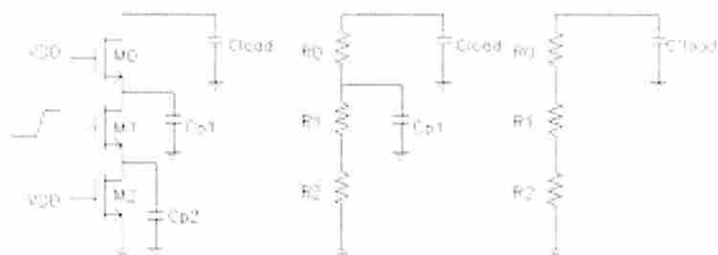


Figure 3.7 (a) Pull-down subcircuit of a NAND gate, (b) corresponding RC model to obtain lumped load capacitance including internal capacitance, and (c) the circuit with all capacitance lumped into the load capacitance.

Using the Elmore delay model, the time constant for the circuit in Figure 3.7(b) is $C_{p1}(R_1+R_2)+C_{load}(R_0+R_1+R_2)$, or equivalently $C'_{load}(R_0+R_1+R_2)$ where

$$C'_{load} = C_{p1}\frac{R_1 + R_2}{R_0 + R_1 + R_2} + C_{load}.$$

The above method is a first order approximation to lump the internal capacitance to the output load and can be extended to multiple transistors above the switching MOS. Inaccuracy can be minimized by a suitable choice of transistor resistance values.

3.2.4 Multiple Input Transitions

Computing t_r (or t_f) for transition signals is complicated when more than one input of a gate switches. Consider a NAND gate with multiple (q) switching inputs. First we apply the method in section 3.2.3 to lump all internal capacitances to the output load. That is, all C_p's below the lowest switching MOSFET are discharged to "0" and, depending on the current state of the circuit, either all other internal C_p's or only those above the highest switching MODFET are added to the output load. Then, we re-order the series connected MOSFETs so that the number of "ON" transistors below the lowest switching device remains the same as before, all q switching devices are then put in series, and finally the remaining "ON" MOSFETs are put on top. The next step is to merge all q switching MOSFETs into one equivalent switching device. This is accomplished by setting the effective β of these switching MOS to β/q. Let the switching inputs be V_{in}^1, V_{in}^2,...,V_{in}^q. The effective input is selected as the input V_{in}^i to the MOS

device such that $t_{ai} + t_{ri} \geq t_{aj} + t_{rj}$ for all j, where t_{ai} is the arrival time of transition i, and t_{ri} is the rise time of transition i.

Then the series connected MOS chain is reduced to the circuit model of Figure 3.3(b).

3.3 A Piece-Wise Linear Model for Noise

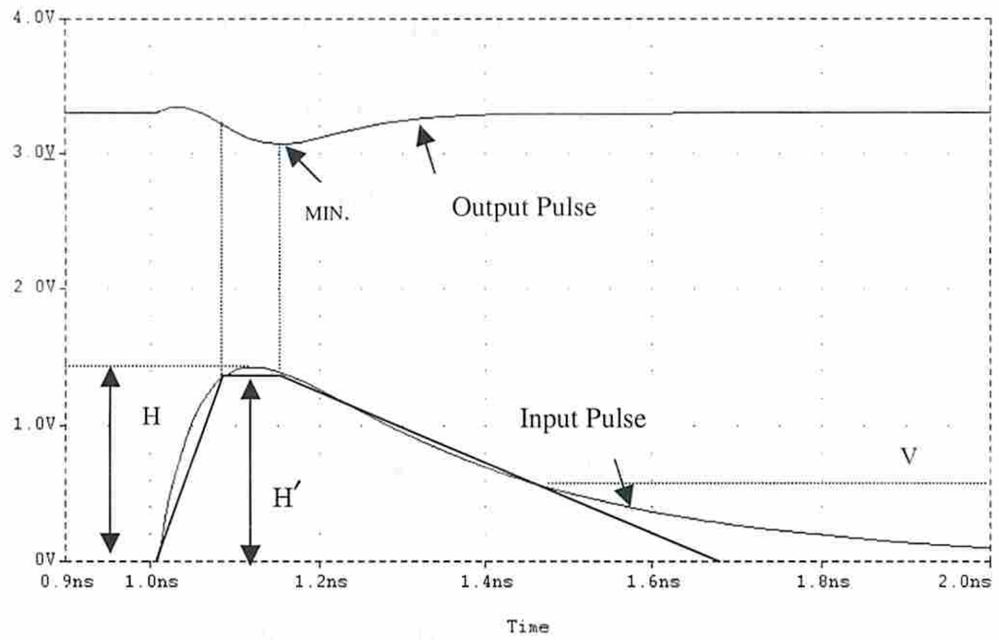
When a crosstalk noise (a pulse) passes through a gate, it can be either attenuated or amplified depending on its amplitude H and width W. Figure 3.8 shows a simulation result of crosstalk noise propagate through an inverter. In Figure 3.8(a) the output noise is small. On the other hand, the input noise in Figure 3.8(b) is sufficient to produce a large output pulse. Note that the output reaches its minimum after the amplitude of the input noise starts to decrease. In addition, the output pulse is almost symmetric with respect to t_q, the time it reaches its minimum value.

There are two obvious ways to obtain the output waveform as a function of the input waveform. The first is to use the crosstalk waveform equations developed in Section 2.4 convolved with the equations described in Section 3.1. The second is to use a piece-wise linear model of the input noise and approximate the output response using the transformation developed in the previous sub-sections. The latter technique is preferred because it is both accurate and computationally efficient. Let the value of the input voltage be H' when the output reaches its minimum. There are two instants of time where

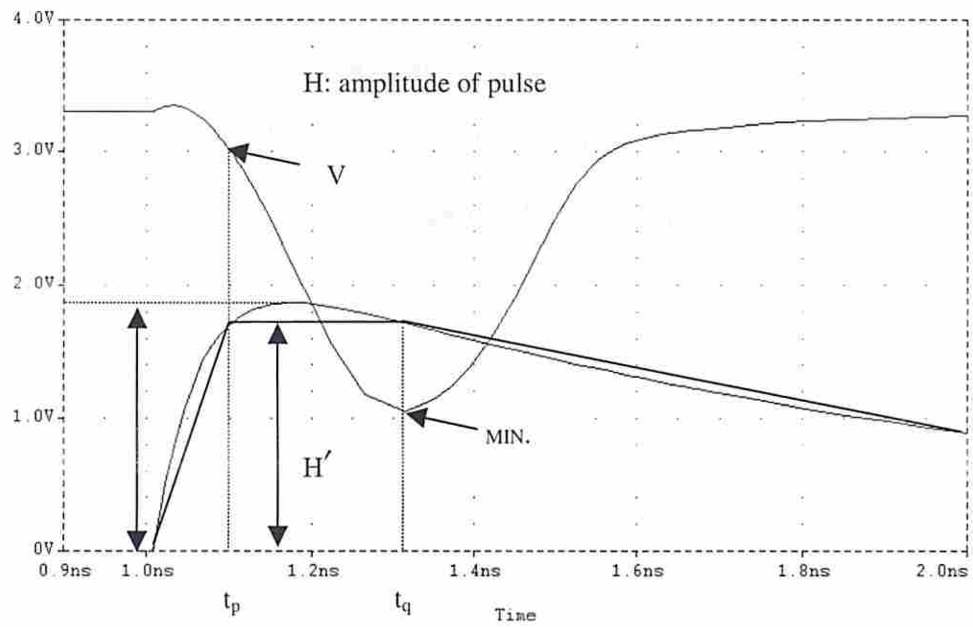
the input has the value H', labeled t_p and t_q in Figure 3.8(b). We approximate the input pulse waveform by three linear segments, as shown in Figure 3.8, namely

1. a rising ramp from the start of the noise until the input reaches the value H' at time t_p,
2. a constant value of H', and
3. a falling ramp from H' at time t_q and going through the point where the input voltage drops to v_{th}.

Assume H' is a linear function of H, i.e., $H' = \rho H$ for $0 \le \rho \le 1$. Experimental results show that when H is in the range of 1-3.3V, ρ is in the range from 0.85-0.87. By using the crosstalk pulse equations in section 2.4, the slope and time period of each segment can be easily determined. We can apply this piece-wise linear approximation of the noise waveform to the inverter model described in section 3.1 to obtain the output response.



(a)



(b)

Figure 3.8 Crosstalk pulse passes through an inverter (a) a small input pulse, (b) a large input pulse.

To complete our model we need to set a critical voltage v_x such that a pulse with amplitude less than v_x will be attenuated and one larger than v_x will be amplified. This critical voltage v_x can be defined as the input voltage such that $dV_{out}/dV_{in} = -1$. Since this point resides in the region that is modeled by the circuit in Figure 3.1(b), the following results are obtained from equation (3-1).

$$\frac{dV_{out}}{dV_{in}} = P \cdot e^{\frac{-t_r(V_{in}-v_{tn})}{R_p C}} \cdot \frac{-t_r}{R_p C} + A(V_{in} - v_{tn}) + B = -1 \cdot \tag{3-3}$$

Solving for V_{in} we obtain the critical voltage v_x. An approximate value of v_x can be found by using a Taylor series expansion for the exponential term.

If H is smaller than v_x, the circuit model in Figure 3.1(b) is used to determine the output response. First we apply the first segment of the noise waveform, i.e. the rising ramp, to the model in section 3.1 and obtain the output voltage drop to V_s at time t_p as shown in Figure 3.8(b). Then the second segment, a level voltage of value H' is applied to continuing discharge the output. Similar to the process in section 4.1, except the input is now held constant at H', we obtain the output response as

$$V_{out} = Pe^{-1/R_p C} + MR_p C, \quad t_p < t \leq t_q, \tag{3-4}$$

where

$$R_p = \frac{1}{V_{DD}\beta_p(H'-1-v_{tp})Coef1},$$

$$P = e^{t_p/R_p C}(V_s - MR_p C),$$

$$M = \frac{1}{R_p C} - \frac{V_{DD}\beta_n}{2C}(H' - v_{tn})^2 ,$$

and Coef1 is an experimental fitting coefficient and is a function of H.

The minimum output voltage obtained is $V_{out}(t_q)$.

For values of $V_{in} > v_x$, a change, dV_{in}, in the input voltage will cause a change, dV_o, in the output voltage such that dV_o will be greater than dV_{in}, i.e. the circuit is in the amplification mode. The circuit model in Figure 3.1(c) will be used to determine the output response. Similar to the above process for the case of a small pulse, we obtain the output response as $V_{out} = Z \cdot t + (V_s - Z \cdot t_p)$, $t_p < t \le t_q$ where

$$Z = \left[\frac{\beta_p V_{DD}}{2C}(H' - 1 - v_{tp})^2 - \frac{\beta_n V_{DD}}{2C}(H' - v_{tn})^2 \right].$$

Again the minimum output voltage is $V_{out}(t_q)$.

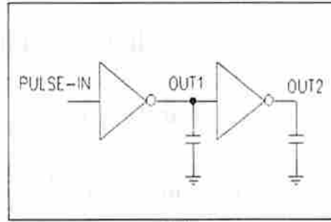
If the output voltage continues to drop, the NMOS transistor will pull out of saturation and move into the linear region, and the inverter will no longer operate in the amplification mode. The circuit model in Figure 3.1(d) is then used to calculate the output response. The resulting equations for the output response are similar to equation (3-4), except the roles of the NMOS and the PMOS transistors are interchanged and the coefficients are different. Again the corresponding minimum output voltage is $V_{out}(t_q)$.

After the output reaches its minimum voltage, the third segment of the model, the falling ramp, is applied to the inverter model to obtain the recovery portion of the output waveform. This is the reverse of the previous processes in obtaining the discharging waveform. However, since we already observed that the output waveform is almost

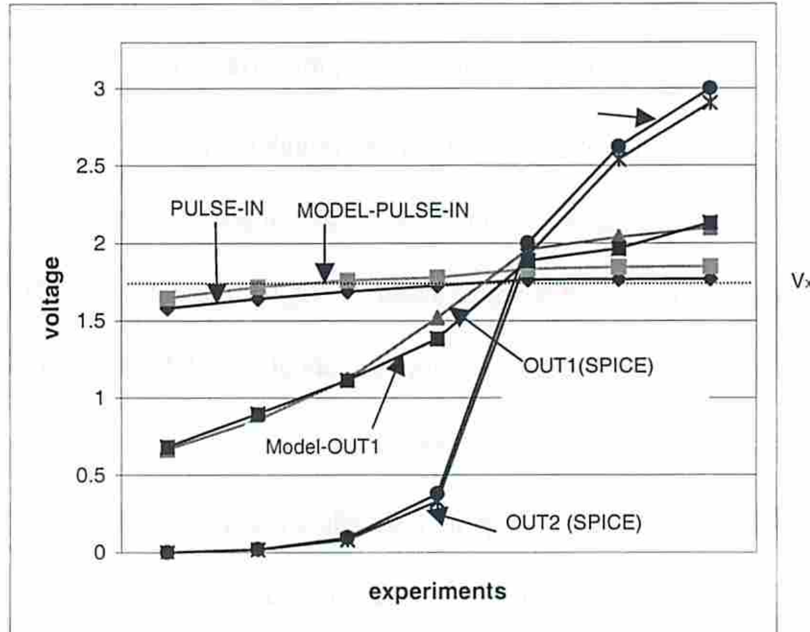
symmetric around t_q, another approach to obtain the recovery portion of the output waveform is just to reflect the discharging part of the waveform with respect to the axis t_q. The error caused by this "reflection" method is mainly in the tail portion of the output waveform. Since the variance in the tail portion is less than the device threshold voltage (v_{tn} or v_{tp}), this approximation has a negligible effect on the results.

Propagation of this output pulse through the next level of gates is done in a similar way. Figure 3.9 shows a comparison of this approach with SPICE results. Here we see that for an input height equal to about $v_x - 0.2V$, the pulse at OUT1 is about 0.7V and is essentially zero at OUT2. For an input of about $v_x + 0.2V = V^*$, the pulse at OUT1 is more than V^*, and that at OUT2 is almost 3V.

For crosstalk speedup and slowdown, a piece-wise linear model can be easily constructed by using the arrival and transition times of a signal.



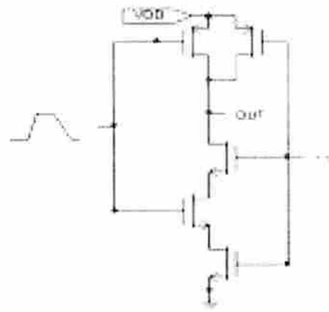
(a)



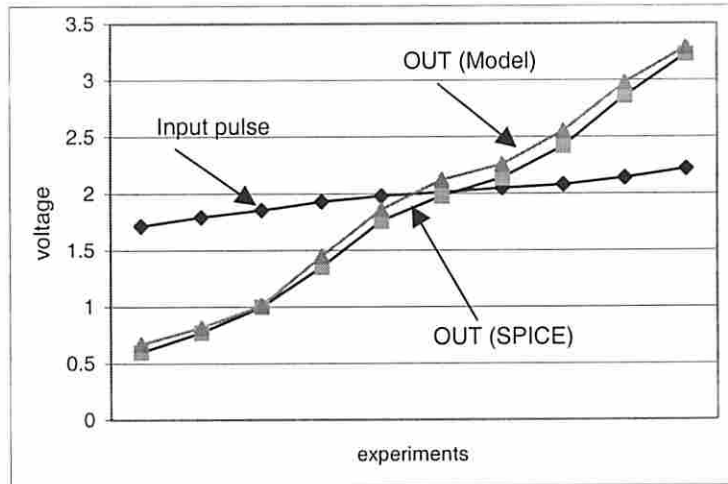
(b)

Figure 3.9 (a) Circuit for measurement for input and output pulses amplitude H'; (b) Comparison of the model and SPICE results.

Combining all the techniques described in section 3.1–3.3, i.e., the inverter model, the method to collapse CMOS gates and the piece-wise-linear model, Figure 3.10 shows the results of applying input pulses to the middle input of a 3 input NAND gate. The other 2 inputs are of course held at "1".



(a)



(b)

Figure 3.10 (a) Circuit for applying piece-wise-linear pulses; (b) Comparison of the model and SPICE results (maximum pulse amplitudes).

3.4 Termination Conditions for Noise (Output Receiver Characterization)

When a crosstalk noise effect reaches a primary output, it is important to determine whether an error has been created or not depending on the severity of the noise. We focus on combinational logic circuits whose outputs are either primary outputs or pseudo primary outputs, that are data or clock inputs to storage devices. Pseudo primary

outputs can be those devices that are very sensitive to noise such as sense amplifiers and dynamic logics. For example, Figure 3.11 shows that the output of a dynamic gate may be degraded if a pulse is applied to one of the inputs of the evaluation network. Figure 3.12(a) shows the severity of output degradation of a dynamic gate due to input pulses with various amplitudes and pulse widths. We can see that the output voltage degradation is proportional to the pulse's amplitude and width. In Figure 3.12 (b), we vary the arrival time of the input pulse with respect to the arrival time of the clock edge. The input pulse is assumed to be a fixed size with $0.5V_{DD}$ amplitude and 20ps pulse width.

A clock ϕ is assumed to arrive at time 1 ns. As shown in Figure 3.12 (b), the input pulse is completely filtered away when it arrived before 975ps, which is approximately the arrival time of the clock minus the pulse's width. If the pulse arrives after the clock edge, then all the energy contained in the pulse will contribute to the discharge of the output voltage and cause a large voltage degradation. If the pulse arrives between 975ps and 1000ps, i.e. the clock edge arrives some time during the pulse's period, then only a portion of the pulse energy will be available to discharge the output, and the severity of the output degradation will depend on the skew between the clock and input pulse.

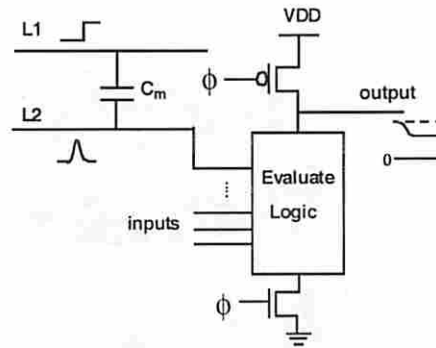
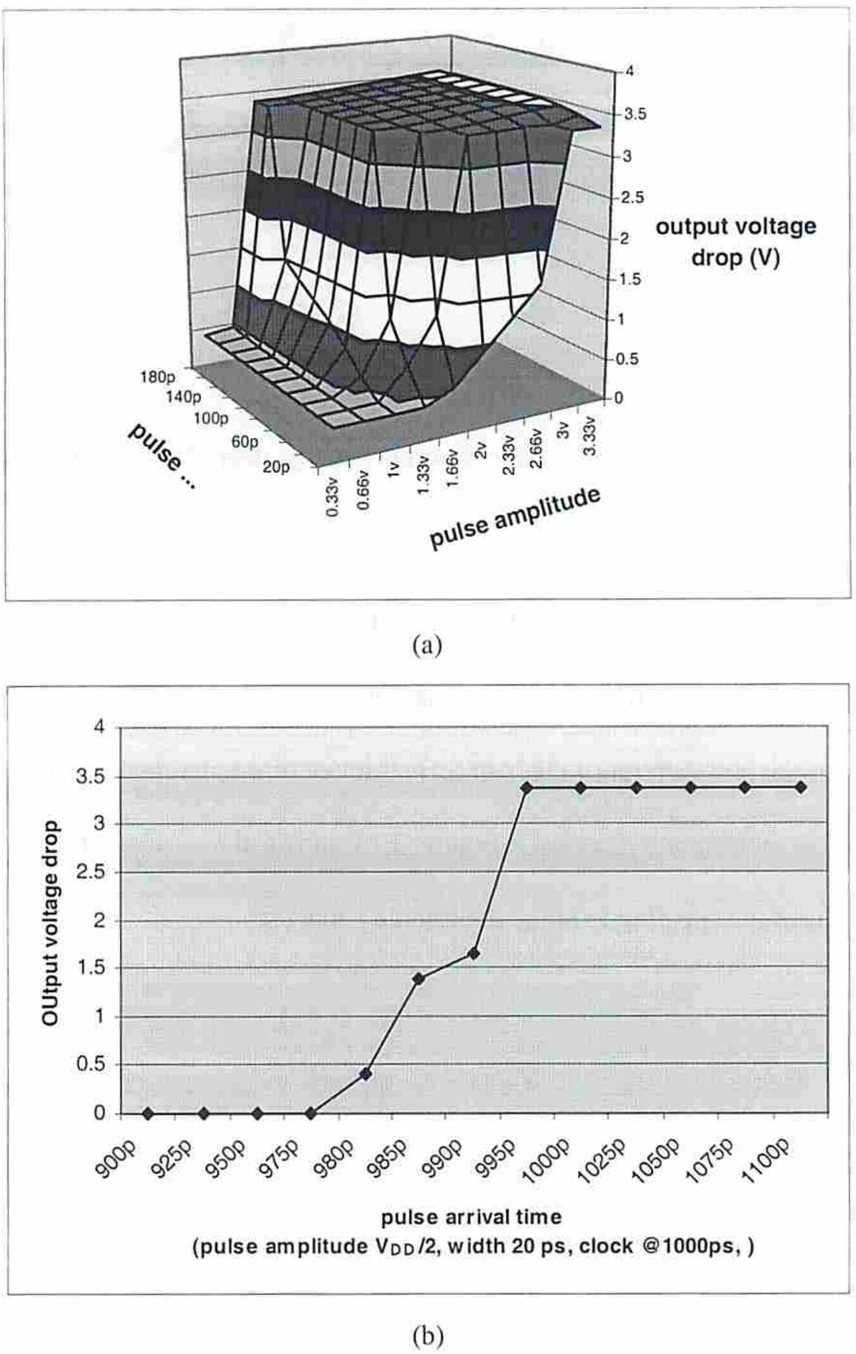


Figure 3.11 Circuit diagram for a output voltage degradation of a dynamic gate due to a input pulse.

To illustrate one problem due to crosstalk slowdown, consider a setup time violation of a flip-flop. Such a violation can result in metastability or a wrong output value. Several different ranges of arrival times of input signal D were simulated to cover a wide metastability region. Figure 3.13 shows the results of these simulations. The output of the flip-flop exhibits metastable behavior.



(a)

(b)

Figure 3.12 Output voltage degradation of a dynamic gate due to a input pulse: (a) severity of voltage degradation w.r.t. various pulse amplitudes and widths, (b) severity of voltage degradation w.r.t. the arrival times of input pulses.

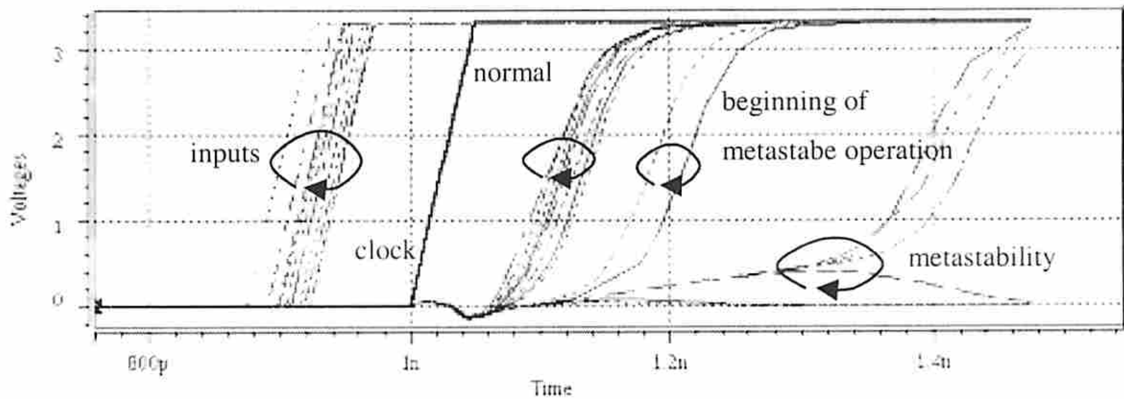


Figure 3.13 Setup time violation of a D flip-flop causes metastability.

Similar experiments can be performed to characterize the noise sensitivity of various kinds of pseudo outputs such as latches, pass gate, and flip-flops. Once the characterization process is done, when a noise reaches a pseudo PO one can determine whether an error is created or not by setting a desired criterion and performing a simple table look-up.

3.5 Summary

In this chapter several new results have been developed that can be used in our ATPG system to efficiently and accurately generate tests for what is essentially an analog effect, namely crosstalk noise. These results include new models for a CMOS inverter, methods to calculate inverter output response for pulse inputs, a method for collapsing CMOS gates into equivalent inverters, and a piece-wise linear model for pulses. These techniques were integrated into a test generation framework (see Chapter 5) that takes into account several attributes such as noise strengths and signal arrival times and

identifies test patterns that maximize crosstalk noise at POs while satisfying a given set of Boolean and analog constraints.

Chapter 4

Test Generation for Crosstalk Noise

Due to technology scaling and increasing clock frequency, problems due to noise effects lead to an increase in design/debugging efforts and a decrease in circuit performance. This chapter addresses the problem of efficiently and accurately generating two-vector tests for crosstalk induced effects, such as pulses, signal speedup and slowdown, in digital combinational circuits. These effects are becoming more prevalent due to short signal switching times and deep submicron circuitry. These noise effects can propagate through a circuit and create a logic error in a latch or at a primary output. We have developed a mixed-signal test generator that incorporates classical static values as well as dynamic signals such as transitions and pulses, and timing information such as signal arrival times, rise/fall times, and gate delay. Conditions for the creation of the worst-case coupling and propagation of a delayed signal are presented. We also present a new analog cost function that is used to guide the search process. Comparison of results with SPICE simulations confirms the accuracy of this approach, and experimental results show that the method can be applied to circuits of reasonable sizes.

This chapter is organized as follows. In section 4.1, the value systems for the proposed test generator is presented. In section 4.2 conditions for excitation and propagation maximum crosstalk effects are provided. In section 4.2 a cost function for

selecting noise sensitive path is derived. In section 4.4 we present our timing-oriented ATPG. Section 4.5 illustrate our ATPG algorithm. In section 4.6 we discuss experimental results. Finally, in section 4.7 we present a summary for this chapter.

4.1 Value Systems

In this section we present an ATPG algorithm to generate tests for crosstalk noise. We focus on combinational logic circuits whose outputs are either primary or pseudo primary outputs. This algorithm incorporates crosstalk by employing new logic values and corresponding analog information, such as signal arrival times, rise/fall times, and input arrival skews, and searches the space of all possible pairs of input patterns using a significant modified version of a backtrace procedure [62]. A signal value in our test generation system contains not only a symbol for its logic value, but also a set of parameters for its corresponding analog properties. For a specific target crosstalk coupling in a circuit and which we refer to as a c-fault, the objective of this test generator is to generate, under given timing assumptions and requirements, a pair of vectors (a test) that create a crosstalk effect at the target and either a logic error or the maximum noise effect at an output. For example, in the case of crosstalk slowdown, given the timing of a clock-edge, the test generator may generate tests that cause a victim line signal to slowdown, and propagate the delayed signal in such a way so as to violate the given timing requirement at a D input to a flip-flop. The symbols and value system shown in

Table 4.1 are used. The analytic models for the computation of the associated parameters are discussed in Chapter 3.

Table 4.1 Symbols and parameters used for test generation.

Symbols	Associated parameters	Description
1	-	constant 1
0	-	constant 0
P_p	t_a, H', t_p, t_q, t_e	positive pulse
P_n	t_a, H', t_p, t_q, t_e	negative pulse
T_u	t_a, t_r	rising transition
T_d	t_a, t_f	falling transition
S_uT_u	t_a, t_r	speedup rising transition
S_uT_d	t_a, t_f	speedup falling transition
S_dT_u	t_a, t_r	slowdown rising transition
S_dT_d	t_a, t_f	slowdown falling transition
X	-	unknown

Description of parameters.

t_a - arrival time, H', t_p, t_q, t_e as in section 3.3, t_r - rise time, t_f - fall time

Table 4.2 shows the truth table of out value system for an AND gate. Similar truth tables have been derived for other gate types. In Table 4.2 each symbol is associated with a set of parameters as shown in Table 4.1. The values of parameters of the output signal are computed using models described in Chapter 3. For example, when both a positive pulse (P_u) and a rising transition (T_u) appear at the inputs of a 2-input AND gate, the response at the output may be a positive pulse if the input rising transition arrives earlier than the input pulse. Since the transition arrives earlier, it is regarded as a "1". The computation of the output pulse is performed as described in Chapter 3. Similarly, if both

inputs have rising transitions, the output signal depends on the dominant input transition (in this case the latest one) and the output response is again computed using analytic models described in Chapter 3.

Table 4.2 Truth table for the value system for an AND gate.

IP1 / IP2	0	1	T_u	T_d	P_u	P_d	S_uT_u	S_uT_d	S_dT_u	S_dT_d	X
0	0	0	0	0	0	0	0	0	0	0	0
1		1	T_u	T_d	P_u	P_d	S_uT_u	S_uT_d	S_dT_u	S_dT_d	X
T_u			T_u	0^1	P_u, $t_{a1}>t_{a2}$ 0, $t_{a1}<t_{a2}$	T_u^3	S_uT_u, $t_{a1}>t_{a2}$ T_u, $t_{a1}<t_{a2}$	0^1	S_dT_u, $t_{a1}>t_{a2}$ T_u, $t_{a1}<t_{a2}$	0^1	X
T_d				T_d	P_u, $t_{a1}<t_{a2}$ 0, $t_{a1}>t_{a2}$	T_d^3	0^1	S_uT_d, $t_{a1}<t_{a2}$ T_d, $t_{a1}>t_{a2}$	0^1	S_dT_d, $t_{a1}<t_{a2}$ T_d, $t_{a1}>t_{a2}$	X
P_u					0^7, or P_u^2	P_u^8, or 0^4	$-^6$	$-^6$	$-^6$	$-^6$	X
P_d						P_d^5	$-^6$	$-^6$	$-^6$	$-^6$	X
S_uT_u							S_uT_u	0^1	$-^6$	$-^6$	X
S_uT_d								S_uT_d	$-^6$	$-^6$	X
S_dT_u									S_dT_u	0^1	X
S_dT_d										S_dT_d	X
X											X

1: static hazard may occur but ignored

2: multiple pulses at inputs, if pulses arrive in such a way that $t_{a1} \geq t_{a2}$ and $t_{q2} \leq t_{q1}$, select the smaller one according to simulation results; in this version only single pulse considered

3: a possible dynamic hazard, but treated as a simple transition, i.e., pulse is ignored

4: overlapped P_u and P_d may result in a small output pulse but is ignored

5: multiple pulses at inputs, select the larger one to create more significant pulse at the gate's output; in this version only single pulse considered

6: only one type of crosstalk noise is considered at a time, multiple effects or mixed-mode test generation not supported in the current framework

7: multiple pulses at inputs, but non-overlapping or partially overlapped

8: completely non-overlapping input pulses, P_d is considered as a static 1

Note that several cases have been simplified to obtain transitive closure of the truth table. For example, when a rising transition (T_u) and a falling transition (T_d) are applied to the inputs of an AND gate, there can be either a value implied by controlling

value or a static hazard at the output of the gate, depending on the arrival time of the transitions. Similarly, when both a negative pulse (P_d) and a rising transition appear at the inputs of a 2-input AND gate, the response at the output may be a rising transition or a rising transition followed by a negative pulse (i.e., a dynamic hazard). In the current version, we focus mainly on the impact due to coupling effects on circuit performance. Hence the static hazards as well as dynamic hazards are ignored in the current implementation.

Our framework is limited in its capability to process multiple pulses at gate's inputs when timing is considered. For example, when multiple negative pulses appear at the inputs of an AND gate, we propagate the largest pulse through the gate to create a larger pulse at the output. If in fact the input pulses arrive in such a way that one pulse P_{di} contains other pulses P_{dj} in time, i.e., $t_{ai} \leq t_{aj}$ and $t_{qi} \geq t_{qj}$, our approach produces a reasonable result. However, if input pulses arrive at different times, then selecting the largest input pulse to propagate may not be the optimal/correct choice. Propagating the largest pulse may result in a significant noise at primary outputs, yet the noise may arrive at such a time that it does not create a timing violation. In such a case, propagating a smaller pulse with a different arrival time may indeed result in a smaller noise at primary outputs but one that cause a timing violation. This deficiency in our current framework needs further studies and should be considered in the future version.

Since our current framework is for a single coupling effect, only one type of crosstalk noise, namely, pulse, speedup, or slowdown, is considered at a time. There is no interaction between crosstalk pulse and crosstalk delay effects in the test generation

process. If multiple coupling effects need to be considered, the truth table and/or the value system must be expanded for the interaction between different types of noise.

4.2 Conditions for Maximizing Crosstalk Effects

Conditions that a two-pattern test must satisfy to generate a crosstalk effect of maximal severity were derived using the expressions developed in section 2.4. There are three objectives in creating a crosstalk effect of large severity: a weak driver on the victim line (objective 1), a fast signal transition on the affecting line (objective 2), and a propagation path that maintains/amplifies the noise effect until it reaches an output (objective 3). These objectives are used to determine conditions to be satisfied for maximizing the observed crosstalk noise. In Table 4.3 we list the conditions for each objective for a NAND gate. Similar conditions are established for other gate types. The objective line (affecting line, victim line, ...) is assume to be fed by a NAND gate. Conditions in Table 4.3 are used by the backtrace process to select PI assignments that maximize crosstalk noise. Note that for the propagation of a pulse (objective 3), only constant values are allowed at side fan-in's. This is because a transition aligned with a noise pulse will significantly decrease the amplitude and width of the pulse. Since each signal has an arrival time t_a and transition time $t_r(t_f)$ associated with it, the algorithm can determine whether a signal transition occurs before, after, or at the same time as a pulse. That is, a transition occurring long before (after) a pulse can be modeled as the final (initial) value of that transition with respect to a pulse.

Table 4.3 Conditions for achieving three objectives (for a NAND gate).

Objective	Target value	Necessary condition	Preferred condition on side fan-in	Sufficient condition on side fan-in
1	0	All inputs are 1	-	All 1
1	1	0 at one input	All 1	1 or T_u or T_d or 0
1	T_u	T_d at one input	All 1	1 or T_d
1	T_d	T_u at one input	All T_u	T_u or 1
2	T_u	T_d at one input	All T_d	T_d or1
2	T_d	T_u at one input	All 1	1 or T_u
3	$P_p(P_n)$	$P_n(P_p)$ at one input	All 1	1 when $P_n(P_p)$ arrives
3	S_uT_u	S_uT_u at one input	1 or T_u	All 1
3	S_uT_d	S_uT_d at one input	T_d or1	All T_d
3	S_dT_u	S_dT_u at one input	T_u or 1	All T_u
3	S_dT_d	S_dT_d at one input	1 or T_d	All 1

4.3 Cost Functions for Noise Propagation

Since the objective of this TG is to create the maximum noise at a primary output, in addition to the conditions in Table 4.3 we need a cost function that can guide the search for PI assignments as well a path from the source of the noise to an output.

The cost function contains a digital and an analog part. The digital part deals with controllability and observability measures [31], and is used to break ties. The analog part of the cost function is a measurement of the gate's capability to propagate noise and is dependent on the gate's strength, i.e. effective β, load capacitance, and gate type such as static, dynamic, domino or latch. Consider a simple static gate such as the inverter in Figure 3.1(a). When a positive pulse is applied to this inverter, the circuit model in Figure 3.1(b) and/or (c) are used to obtain the output response. Since the PMOS current reaches its maximum when the transistor enters the saturation region, the influence of the PMOS

in Figure 3.1(c) is greater than that in Figure 3.1(b). Re-arranging the differential equation for the circuit in Figure 3.1(c) gives

$$C\frac{dV_o}{dt} = \frac{V_{DD}}{2}(V_{in} - v_{tn})^2\left[\beta_n\left(1 - \frac{\beta_p}{\beta_n}\left(\frac{V_{in} - 1 - v_{tp}}{V_{in} - v_{tn}}\right)^2\right)\right]$$
$$= \frac{V_{DD}}{2}(V_{in} - v_{tn})^2\beta_{neff}\ .$$

Thus the effect β_n as a function of input V_{in} is

$$\beta_{eff} = \beta_n\left(1 - \frac{\beta_p}{\beta_n}\left(\frac{V_{in} - 1 - v_{tp}}{V_{in} - v_{tn}}\right)^2\right).$$

The input V_{in} can be any value between V_{-1}, defined as the point in the DC characteristic where $dV_{out}/dV_{in} = -1$, and v_{inv} so that β_{eff} becomes a constant value and can be used as an index to define the analog cost function. Since the capability for a noise to propagate through a gate is proportional to the gates strength and inversely proportional to the gate's load, the analog cost function can be defined as:

$$Cost = \frac{C_{load}}{\beta_n[1 - \frac{\beta_p}{\beta_n}\left(\frac{V_{-1} - 1 - v_{tp}}{V_{-1} - v_{tn}}\right)^2]},$$

where C_{load} is proportional to the number of inputs and fanouts of the gate.

We have defined an analog cost for different types of gates in the same manner.

This cost function quantifies the "difficulty" which a pulse encounters in propagating through a gate. For instance, the load capacitance serves as a charge pool to mitigate the noise, therefore the larger the output capacitance the smaller the output pulse.

On the other hand, the larger the β_{eff} the stronger the pull-down strength. Hence a small pulse can easily discharge the output.

After the analog cost of each gate is obtained, the cost of a path can be obtained by combining these cost values in a manner similar to calculation of observability costs [31]. The computation of the analog cost of a path starts from the primary outputs (i.e. the last level of the gates) and then the circuit is traversed backward to accumulate the cost of each gate. Thus, to propagate a noise effect we can select a path whose cost is the lowest, i.e. propagates the noise with maximum severity. If two paths have the same analog costs, then the digital observability costs are used to break ties.

4.4 Timing Analysis

To excite a target effect at a specific time (or within a timing window), we first need to obtain some delay information about the gates and paths in the circuit. We associate with each signal line that has a transition a timing window, such as those shown in Figure 4.1. The window is defined (bounded) by the minimal arrival ($\tau1$) and maximal arrival ($\tau2$) times of the transition. Within this window the transitions with the minimum ($\tau3$) and maximum ($\tau4$) transition times can occur in either order, i.e., $\tau3$ before $\tau4$ or $\tau4$ before $\tau3$. The window consists of these four transitions. Assuming that the device sizes and output loads of all gates are given, the delay of a gate (assuming one input is switching and other inputs have non-controlling values) can be estimated using standard

delay format (SDF) [60]. For a gate g, assuming a timing window is given at an input of the gate, by using SDF and additional computations one can derive the corresponding timing window (four transitions) at the output of the gate. Figure 4.1 shows an example where input x has a falling transition and output z has a rising transition. Timing windows for various combinations of transitions (rising or falling) at the input and output of the gate can be derived in the same manner.

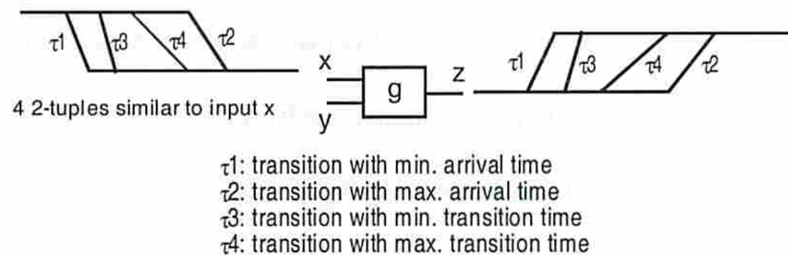


Figure 4.1 Computation of timing windows for a gate.

4.4.1 Forward "Arrival" Timing Window Calculation

The forward delay calculation performs a static timing analysis of the circuit. Given arrival and transition times for signals at primary inputs, one can traverse the circuit starting from PIs in a breadth-first manner to compute timing windows for each line. During the computations for timing windows, in addition to the timing information associated with each line, we also obtained min. and max. input-to-output delays for each input of each gate. For example, for the gate g in Figure 4.1 with rising output transitions and falling input transitions at input x, in addition to those four transitions comprising the timing window for each line (input x or output z), there are also two delay values, min.

and max. delays (indicating the minimum and maximum delays for propagating a transition from input x to output z), associated with the gate. Similar information is also obtained for input y.

If we would like a signal transition at a specific circuit node to occur near a specific time, these values provide directions and/or choices for a backtrace procedure.

4.4.2 Backward "Required" Timing Window Calculation

The backward delay calculation computes signal required times. Required times are timing windows in which signals are required to appear. Given required times of signals at primary outputs, the calculation process starts at the outputs and traverses backward through the circuit to calculate required times by subtracting proper gate delays, which are obtained in the forward delay calculation process. Figure 4.2 shows the computation of required times.

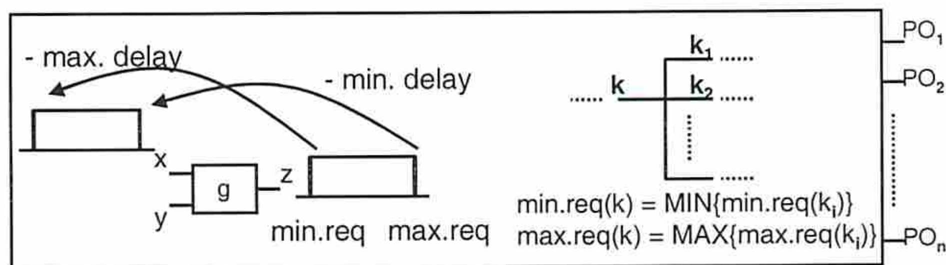


Figure 4.2 Computation of required times.

From these required times, for a given node k, we can find the shortest and longest paths in terms of time to the outputs by using require times of the outputs in the fanout cone of node k. In addition, if a signal is to arrive at a specific time at a PO, we can use these parameters to direct the search process in an attempt to satisfy this constraint.

4.4.3 Timing-Oriented ATPG

Once the static timing analysis has been performed, the timing window of transitions on the affecting line (A) and victim line (V) can be obtained, as shown in Figure 4.3, where r and s (p and q) represents the shortest and longest timing path from the PIs to the affecting (victim) line, and y and x are the shortest and longest timing paths from the victim line to an output.

Therefore a transition on A can occur no earlier than time r, and no later than time s. Similarly, the timing window for line V is within [p, q]. Also for any signal transition on the victim line to occur in the time interval [T-x, T-y], there is a chance that this signal (on V) will reach a PO at or after the time T, which will result in a possible timing violation, say, a setup time violation. However, since transitions at A and V can have a certain amount of skew and still slowdown each other, the amount of skew, z, should be included in the computation of the overlapping window. Hence the targeted timing window is the intersected time interval [T-x-z, q+z] in which both transitions on A and V can occur to have an effect that may create a timing violation at an output. Note that if

this interval [T-x-z, q+z] is null, no crosstalk effect for this target can cause a problem at an output.

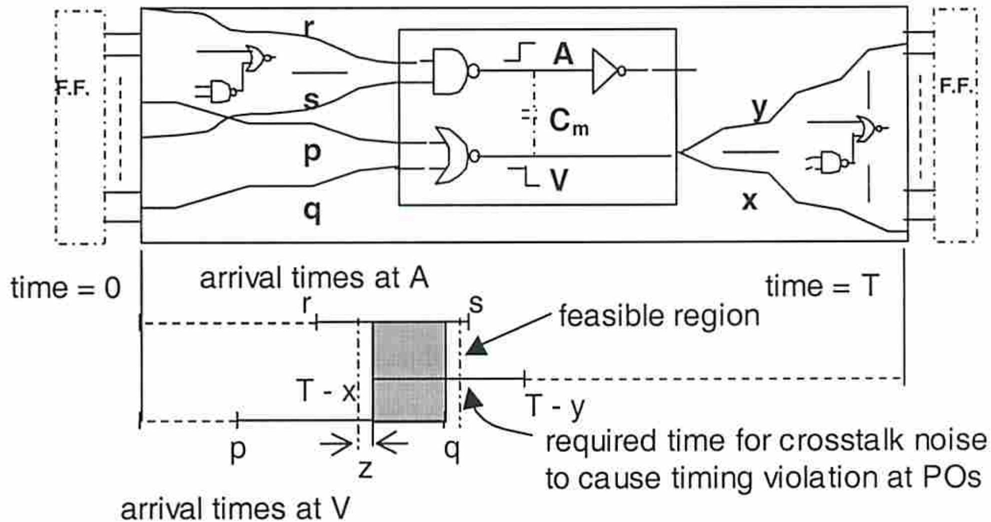


Figure 4.3 Timing window of transitions on the affecting (A) and victim (V) lines, where z is the skew allowed on A.

4.4.3.1 Objectives for Crosstalk Delay

In this section we will consider only the case of crosstalk slowdown. There are five important objectives in creating a slowdown effect of large severity at a specific time.

Objective 1: a late transition on the affecting line

Objective 2: a strong driver on the affecting line,

Objective 3: a transition (opposite direction to that of the affecting line) on the victim line, with a skew bounded by $\pm \alpha$ time units

Objective 4: a weak driver on the victim line,

Objective 5: a propagation path that delays the target slowdown-signal as much as possible until it reaches an output.

These objectives are used to determine conditions to be satisfied for maximizing the observed crosstalk noise.

Objective 2 and 4 determine conditions that a two-pattern test must satisfy to generate a crosstalk effect of maximal severity, which were presented in section 4.2 using the expressions developed in section 2.4. Objective 3 provides the maximum acceptable skew between affecting and victim line such that the crosstalk effect is significant. As stated in Section 2.4, for our example the crosstalk slowdown effect is maximum if both the affecting and victim lines switch at the same time, as shown in Figure 4.4. We can also see that signals on the affecting and victim lines can be skewed and still result in a crosstalk speedup/slowdown effect. The circuit that is used to derive the expressions for Figure 4.4 has a typical gate delay of 100ps. The maximum delay due to crosstalk is 46 ps or about 46% of a gate delay. If the skew between the affecting and victim line signals is one gate delay and the affecting line signal arrives earlier, then we still have 23/46 = 50% of the maximum crosstalk slowdown effect.

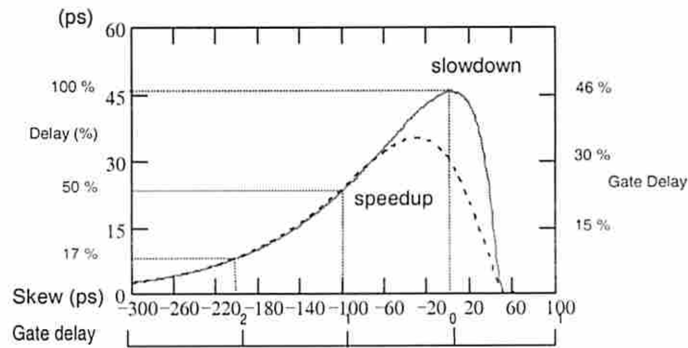


Figure 4.4 Amount of speedup and slowdown on the victim line V vs. skew z: a negative z implies A leads V.

Since some amount of skew is acceptable to have a significant slowdown effect, a skew, say, up to one gate delay, is allowed for the victim line signal, and we target the arrival time of the victim line signal to be within the window which is defined as the arrival time of the affecting line signal plus the allowed skew. Then this timing requirement is translated into the timing-oriented backtrace procedure, described in the next section, to find a test to achieve the desired transition on the victim line.

In summary, together with objective 1 and 5, we try to create on the victim line a late transition which is slowed down as much as possible due to crosstalk, and propagate this effect through a long path to an output to cause a timing violation.

4.4.3.2 Timing-Oriented Backtrace Procedure

In our test generation process, each objective is a 3-tuples of the form of *obj(value, timing, condition)*, where *value* is the desired signal value, namely a transition

or static value; *timing* is the target timing requirement that can be specified as earliest, latest, a window or a specific value; and *condition* is the constraint that specifies whether we want a fast or slow transition, or strong or weak static value.

When an objective is processed, first we check for the existence of a compatible and incomplete pattern at gate inputs. For example, consider an objective to have a falling transition at the output of gate g, as shown in Figure 4.5. We check if the desired target timing window $[z_1, z_2]$ overlaps with the timing windows $[A_f^{min}, A_f^{max}]$ that we obtained from the timing analysis described in section 4.4, where A_f^{min}/A_f^{max} denote the minimum/maximum arrival times for falling transitions. If not, then the desired objective cannot be achieved and a new objective must be selected. Next we check whether existing input signals at gate inputs violate the desired output value. For example, as shown in Figure 4.5, for a desired falling transition at the output, only rising transitions or static 0's are allowed at inputs. If during previous implication process a falling transition has been assigned to one of the inputs, then again this objective cannot be achieved and a new objective must be selected.

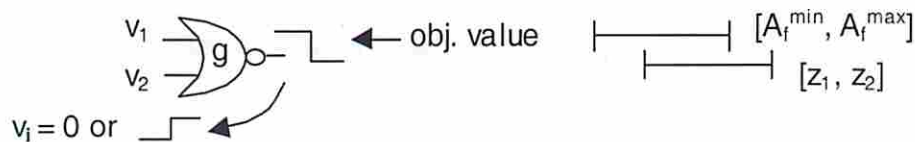


Figure 4.5 Check for the existence of a compatible and incomplete pattern at gate inputs in processing objectives.

If the objective seems to be achievable, then for inputs having unknown value "X", we backtrace and search for a pattern to achieve the objective. For the input on which we select to backtrace, we compute the new target timing window $[z_1-d_{1max}, z_2-d_{1min}]$, where d_{1max}/d_{1min} are the max/min gate delays from that input to the gate's output under the desired transition (falling transition in this case), as shown in Figure 4.6. The d_{max}/d_{min} delays are obtained from the static timing analysis described in Section 4.4. Then the new target timing window is inserted into the new objective for the input we selected to backtrace, and we continue the backtrace process recursively.

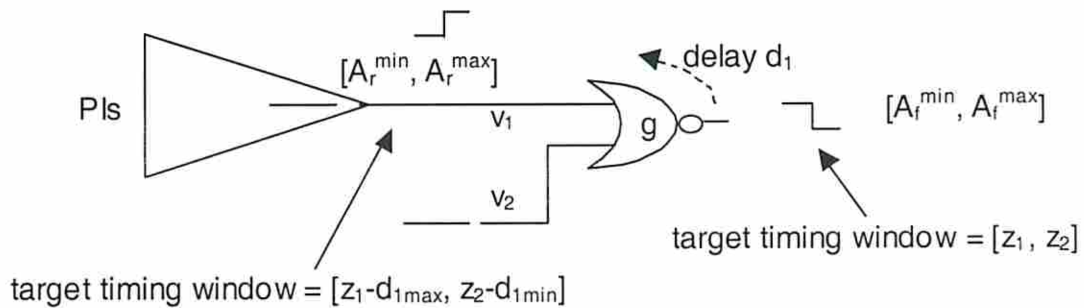


Figure 4.6 Recursive execution of the backtrace process.

The third parameter in our objective is the condition that is used for side-fanin assignments. There are many patterns that can achieve a desired transition on a line with different transition times. For example, to create a falling transition at the output of a two-input NOR gate, both inputs having a rising transition will lead to a shorter gate delay than when one input has a rising transition and the other is held at constant 0. These conditions for side-fan-in assignments that help to create a faster transition were identified in section 4.2. Table 4.4 shows conditions for creating a fast transition at the

output of a NAND gate. The procedure for side-fan-in assignments is similar to the one described in the above subsection.

Table 4.4 Conditions creating a fast transition at the output of a NAND gate.

Target value at the gate's output	Necessary Condition	Preferred condition on side fan-in	Sufficient condition on side fan-in
T_u	T_d at one input	All T_d	T_d or1
T_d	T_u at one input	All 1	1 or T_u

However, not only a transition is required, but also the arrival time of the transition is important. It is necessary to check the timing of the side-fanin assignments so that they won't invalidate the transitions already established. For example, assume that we would like to create a rising transition at the output of a 2-input NAND gate. According to the conditions in section 4.2 we would prefer having both inputs as falling transitions. But if two falling transitions are applied to a 2-input NAND gate, then the earlier transition is the one that controls the timing of the output transition. Hence if we already had a falling transition with the required timing on one input to this NAND gate and the new input has a transition with an earlier transition time, then we may want to discard this transition and try to set this input at a constant "1". Thus, conditions for setting arrival times have higher priority than conditions for switching times.

4.4.3.3 Incremental timing refinement

Static timing analysis provides a min-max range for possible transitions on each line. The min-max range is due to unspecified input values. At each ATPG step, as more

primary inputs get assigned values, more internal lines have known values and min-max timing ranges shrink due to recalculation of arrival, transition and required times. Hence as we dynamically update the timing information of signals, min/max timing ranges are refined to provide better timing information. Figure 4.7 illustrates the idea of the output incremental timing refinement.

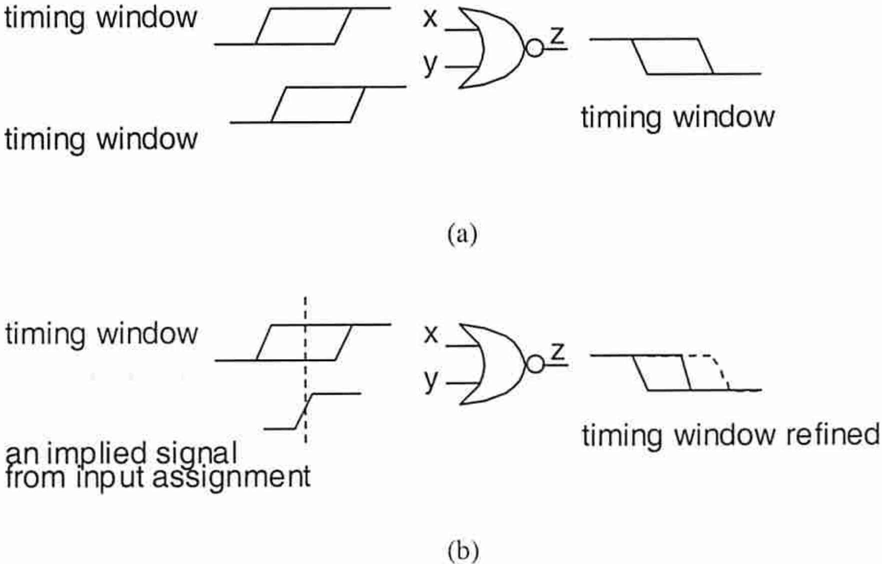


Figure 4.7 Incremental timing refinement: a) before refinement, b) after refinement.

4.4.3.4 Selection of Propagation Paths

Once the transition signal is created on the victim line and is slowed down by the coupling effect from the affecting line, we want to find a path to further delay the signal as much as possible until it reaches an output. There are two situations for slowing down a signal: by side-fan-in assignments and by fan-out branch selections. For example, consider a late (slowdown) rising transition at one input of a 2-input NAND gate. To

sensitize the gate to this slowdown signal, the other input of the NAND gate can be either a "1" or a rising transition no later than the slowdown signal. Note that if the side fan-in transition occurs much earlier than the slowdown signal then the side fan-in transition can be regarded as a static "1". However, if the side fan-in transition switches at the same time as the slowdown signal, then the output transition time will be further delayed [46]. The other alternative for a slowdown signal to reach an output late is to select a longer propagation path. Since we have already performed timing analysis of the circuit as described in Section 4.4, the longest delay path from a node to an output can be easily found.

4.4.3.5 Conflicts between Objectives and Backtracking

In an attempt to achieve the above objectives, there may be some occasions where some decisions made for earlier objectives block the chance of satisfying new objectives. Whenever these situations occur, backtracks are preformed until all objectives are achieved or it is determined that no test exists. For example, we may be able to achieve a fast affecting line transition by having all side fan-ins properly assigned, but this may make the desired transition on the victim line impossible to achieve. Hence an immediate backtracking is required to make an adjustment to the PI assignments for creating the affecting line transition so that the victim line transition can be created. Backtracks may affect the quality of the resulting test, for example, they may lead to a stronger victim line driver. The algorithm employed will explore all possible PI combinations so that the best

test, if one exists, will be eventually found. So, unlike PODEM which employs a constraint satisfaction search process, our algorithm attempts to maximize an objective, such as maximizing the delay (slowdown) of a signal transition. In the next section we describe a branch and bound technique that evaluates the quality of a partial vector obtained during the TG process, and tries to limit our exploration in the search space.

4.4.3.6 Branch and Bound Process to Reduce the Search Space

The order and procedures for processing the objectives provides a way to create a significant crosstalk effect at an output. They help to find a good solution in an efficient way but they do not guarantee finding an optimum solution. Since there are usually data dependencies between the objectives, it is hard to find an optimum solution without considering more of the circuit's electrical and topological properties. Our algorithm searches all possible PI assignments to achieve the objectives. To reduce the time complexity of the algorithm we propose to use a branch and bound process.

First, we associate with each gate G a variable $\Omega(G)$. This variable is used to record information about that crosstalk effect that has passed through gate G and has produced the largest amount of slowdown. The initial value of this variable is zero for all gates. Assume the test generation process begins and all the objectives are achieved one by one. When a test vector is found, that is, a crosstalk slowdown signal is propagated to an output at time T (hence a timing violation occurs), we record information about the

crosstalk slowdown signal on all gates along the propagation path starting from the victim line to the output.

Via backtracking, the test generation continues in order to find a "better" test. If another crosstalk effect reaches gate G, where $\Omega(G) \neq 0$, then we check whether it is possible that this new crosstalk signal will reach a PO later than the recorded Ω value. For a node k in the circuit, the longest path delay from k to a PO reachable from k is $(R_{max} - B_{min})$, where B_{min} is the minimum required time of line k and R_{max} is the maximun required time of all POs reachable from k, as shown in Figure 4.8. Required times are timing windows in which signals are required to appear. Hence we can easily predict the worst case arrival time at a PO as $\omega = \{$the arrival time of the crosstalk effect at G + delay of gate G + $(R_{max} - B_{min})\}$. If ω is greater than Ω, then we continue because the test being constructed may potentially be better than the one we found previously. If not, then we drop this gate from the noise frontier and try another gate in the noise frontier. Thus the Ω variables serve as a bound to limit our search process. If we actually find a new test, then we process the gates along the propagation path and update their Ω values accordingly. We can prune the space even more by only considering paths from G to POs that are potentially sensitizable, i.e., have not been blocked by previously assigned PI values.

As the test generation process continues and tests are generated, more gates will be assigned non zero values for the Ω variable and the search space will be further

limited. The efficiency of the ATPG process improves by about 20 % due to this branch and bound process.

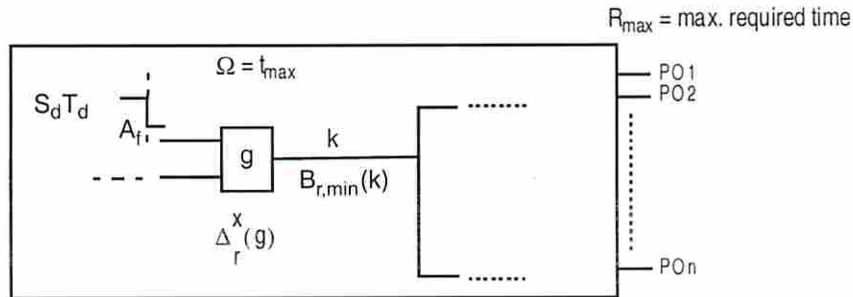


Figure 4.8 Branching and bounding process, where Δ is the delay of gate g.

We also implemented the x-path check technique to reduce the search efforts. In the x-path check process we not only check if a path has been blocked by previous assignments, but also we check if a noise can reach primary outputs and cause a timing violation. If a noise may reach a primary output, bit it arrives at such a time that it does not create a timing violation, this noise is also removed from the noise frontier.

4.5 Test Generation Algorithm

4.5.1 Main Test Generation Algorithm

The algorithm consists of five major steps to achieve the objectives. When a test is found, then it is recorded and relevant signal information along the propagation path is stored in the Ω variable to be used for branch-and-bound. Backtrack is performed to

explore the search space until all PI combinations have been implicitly tried. The flowchart of the algorithm is shown in Figure 4.9.

The outline of the algorithm is as follows.

1. Perform timing analysis of the circuit.
2. Upon START, check the timing windows for when transitions on the affecting and victim lines should occur, and determine if it is possible to create a timing violation at a PO.
3. The initial objectives are to set desired values on the affecting and victim lines. Conditions and recursive procedures discussed in Section 4.2 are used to guide the timing-oriented backtracing direction so that a weak victim line driver and a fast transition on the affecting line can be satisfied under the desired timing requirements.
4. Once these assignments are made, forward imply and evaluate the actual transition rate on every line. Then create the crosstalk signal on the victim line. The analytic models used for creation of crosstalk effect are presented in section 2.4.
5. We utilize the path delay information from timing analysis to select a path to propagate the crosstalk signal to a PO. The path delay is obtained by utilizing required times obtained in the backward delay calculation. We select a path from the current site with the longest path delay for the slowdown case (shortest for the speedup case). When propagating a slowdown signal through a gate, side fan-ins are assigned to see if they can further delay the crosstalk signal.

6. If the noise effect (crosstalk slowdown) has not reached a PO, then one of the internal signal lines that has a "noise signal" value is used as an objective to propagate a noise to outputs. If the noise effect reaches a PO, then the PI assignments are recorded as the test. The values of the Ω variables of all gates along the propagation path are updated.

7. Because we desire a test that creates the maximum slowdown at an output, we continue to backtrack so that all possible PI assignments are explored. By the branch and bound process we expect to generate better tests as we continue processing and the search space continues to be reduced in size.

8. Only the signal value 0, 1, T_u, or T_d can be assigned to primary inputs. Whenever a PI is set to a value the implication procedure is performed and the analog timing information, i.e., rise/fall times and/or arrival times, of some signals are re-computed.

The test generation process including objectives, conditions and procedures for crosstalk speedup is similar to that of the crosstalk slowdown process discussed above, except that we want to excite a speedup signal as early as possible and propagate it to a PO through a path with the shortest path delay.

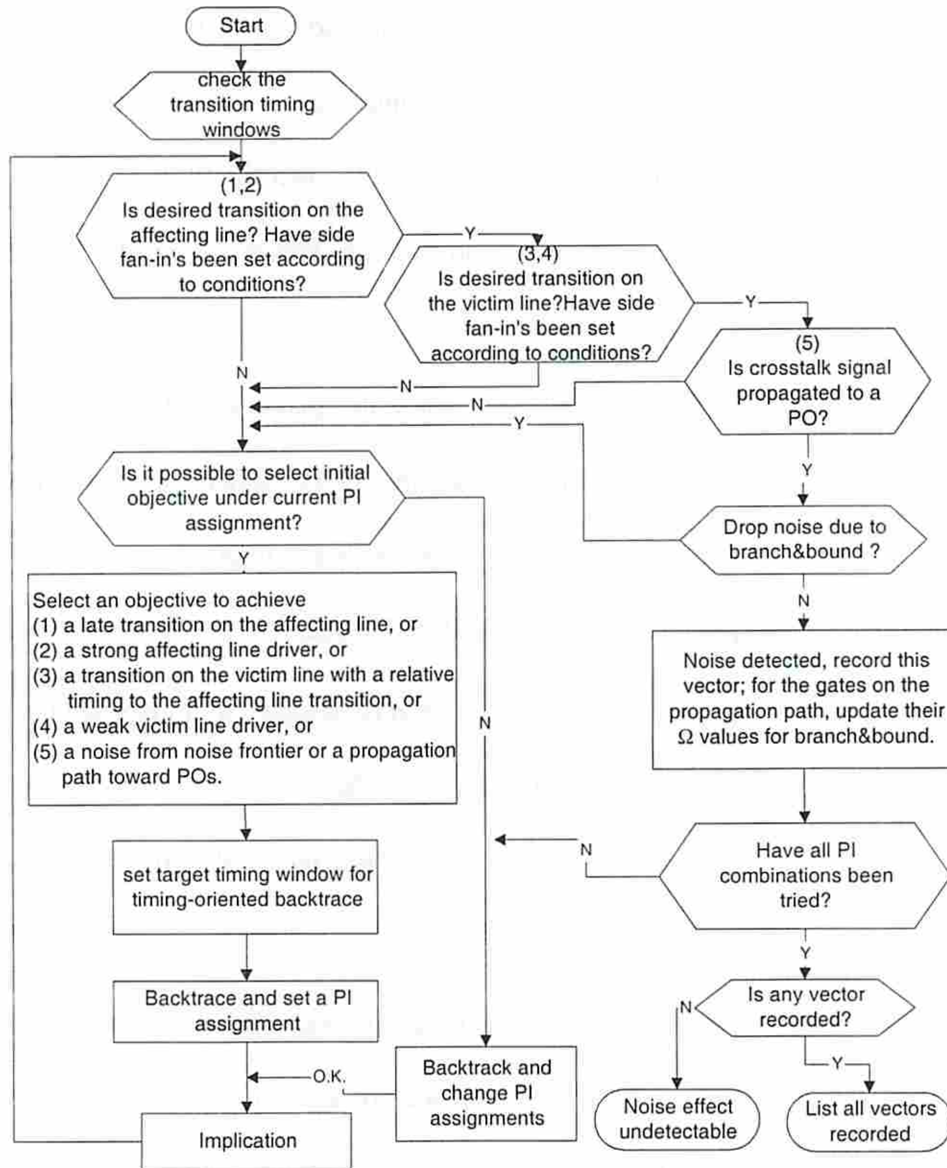


Figure 4.9 Flowchart of the algorithm.

4.6 Experimental Results

4.6.1 Crosstalk Pulse

In this section we illustrate the proposed algorithm. Consider the example in Figure 4.10. The channel lengths of all devices are 0.35um and under each gate we indicate the ratio of widths of the PMOS to NMOS FETs. The gain factor ratio of the NMOS to PMOS in the technology file used is 3.8. All wire and gate capacitances correspond to a realistic layout. Gate sizes are computed to achieve signal transition times of 100ps. Primary inputs are assumed to have signal transition times of 100ps.

Assume that the sub-circuit on the left side of the dash line is 1000um apart from the sub-circuit on the right side of the dash line. Hence lines 14, 9 and 13 are assumed to be about 1000um long and 4 um wide. In addition, assume that line 13 is the affecting line in metal1, line 9 is the affected line in metal2, and they are overlapped so that there is a significant coupling between them. The gate driving line 13 is assumed to be a buffer which has a strong driving strength.

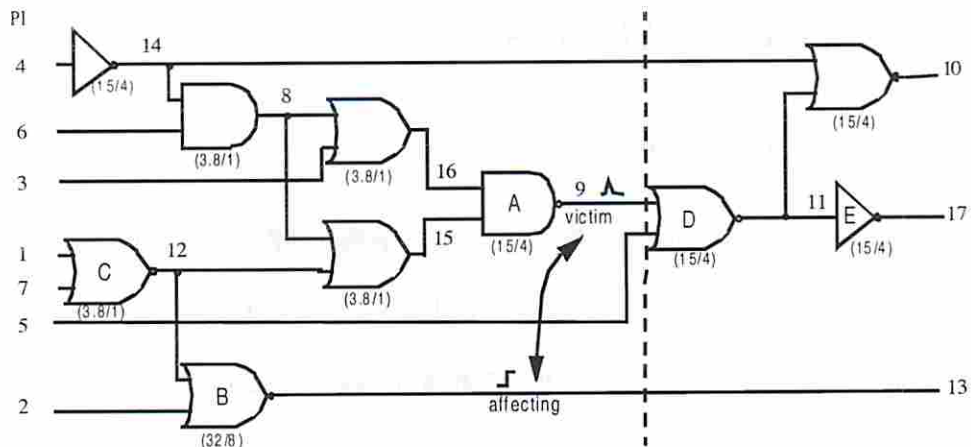


Figure 4.10 Example circuit to illustrate the algorithm.

Assume that we would like to create a positive crosstalk pulse on the affected line 9. First we examine whether a constant 0 has been set on the affected line or not. Since the gate A is a NAND gate, all its inputs have to be set to "1". By backtracing a possible PI assignment setting PI3 to 1, PI6 to 1, and PI4 to 0 is found. Next we attempt to create a rising transition on the affecting line. By using the analog cost function, gate B has a low cost compared to gate C and thus PI2 is selected first and set to a falling transition. According to the conditions for having a fast transition on the affecting line, the preferred side fan-in's of gate B are "0". Hence, either PI1 or PI7 is set to 1.

By implication the affecting line starts to transit at time 45ps with a rise time 62ps. We calculate the crosstalk noise waveform (strength) according to equations in section 2.4. The crosstalk noise has an amplitude of 1.62V with the peak time at 95ps. To propagate this noise, the propagation path has to be sensitized and hence PI5 is set to "0". Since the path through the gate driving line 10 is blocked by the assignment of PI4 to 0.

the only observation point is line 17. The noise at line 11 has an amplitude of 1.19V and the inverter attenuates this pulse so that no significant noise is obtained at the PO line 17. The comparison of the model result and SPICE is shown in Table 4.5. This result is not surprising because of the nature of the 0.35um technology and the static gates used in the example. In SCMOS 0.35um technology the dielectric material between metal 1 and metal 2 is still thick enough so that the coupling capacitance is not sufficiently large to create a severe crosstalk problem. In addition, static gates are usually well balanced for pull-up and pull-down capability which in turn weakens the noise, unless the noise is very large.

Another experiment was performed to see whether the noise is worse for dynamic logic. Gate D was replaced by a dynamic gate with a minimum-size weak keeper. This dynamic gate is very sensitive to noise. The experiment results are shown in Table 4.6. From Table 4.6 we see that significant noise is created at the primary output.

Table 4.5 Comparison of the model and SPICE results.

	Noise site (line 9)			Line 11			Primary output (line 17)		
Parameter	Noise amplitude	Peak time	Start time	Noise amplitude	Peak time	Start time	Noise amplitude	Peak time	Start time
Model	1.62	95	45	1.19	143	71	0	-	-
SPICE	1.64	100	42	1.17	150	78	0	-	-

Table 4.6 Comparison of the model and SPICE results for circuit with dynamic gate D.

	Noise site (line 9)			Line 11			Primary output (line 17)		
Parameter	Noise amplitude	Peak time	Start time	Noise amplitude	Peak time	Start time	Noise amplitude	Peak time	Start time
Model	1.62	95	45	2.40	162	70	3.24	241	120
SPICE	1.64	100	42	2.42	170	76	3.20	250	125

We next analyzed the static circuit using data from the SIA97 roadmap [32]. Assuming a clock rate of 1G Hz, we obtained a maximum pulse height of $0.67 \text{x} V_{DD}$ on line 9 and $0.76\text{x}\ V_{DD}$ on line 11. Hence a significant error exists.

The test generation algorithm described was implemented in the C programming language and applied to several ISCAS '85 benchmark circuits. The program, called XGEN, was run on a Pentium II 400 MHz desktop. Since no circuit information, such as crosstalk fault locations, polarity of transitions causing crosstalk fault, coupling capacitance, and layout information, is currently available to us for these circuits, the affecting and victim lines' driver strength and coupling capacitance value are assumed to be sufficient to excite a significant crosstalk noise at a fault site. We assume all transistors are 0.35um in length, the affecting line is driven by a large driver (28um PMOS/8um NMOS), the victim line is driven by a small driver (7um PMOS/2um NMOS), and they run parallel to each other for a distance of 1000um. All other gates and wires are assumed to have default device sizes and load capacitances.

Two sets of experiments are performed. In the first experiment a single crosstalk fault is targeted and the proposed algorithm is used to generate all possible tests for the target fault. Test vectors associated with corresponding pulses at POs are recorded so that the test creating the worst case pulse at a PO can be identified. The experimental results are shown in Table 4.7. In Table 4.7 PO denotes primary output (number is the node number), first_p_amp is the height of the pulse at the fault site, and amp at the end of each line is the amplitude of the pulse at the corresponding output. Pulse amplitudes are normalized with respect to V_{DD}. The output statistics lumps the output pulses into voltage

ranges. The results correlate well with SPICE simulations. Similar experiments can be performed on other ISCAS circuits with large number of nodes.

In the second experiment, for each circuit, 100 pairs of affecting and victim lines are selected at random without considering the circuit structure. A preprocessing step is performed so that the victim lines selected are located on critical paths. The proposed algorithm is applied to generate one test for each fault. Since a thorough search for test patterns for these many faults may require many backtracks, the maximum number of backtracks per fault is limited to 1000. The pulse size threshold is set to 0.2 V_{DD}, and any pulse smaller than the threshold will be filtered out. Results of the experiments are shown in Table 4.8 and Table 4.9. In Table 4.8 there is no timing criterion set at primary outputs and in Table 4.9 the longest path delay is set as the timing criterion at POs. For the latter case a large pulse must occur at or after the specified time value for it to be considered a problem.

In Table 4.8 and Table 4.9, Column 2 shows the percentage of faults for which tests can be successfully generated. Column 3 shows the percentage of faults for which an appropriate test does not exist to propagate a crosstalk fault to a PO with significant amplitude (i.e. >0.2Vdd), and Column 4 shows the percentage faults for which the number of backtracks exceeds the maximum setting and the TG process was aborted. Column 5 indicates the TG efficiency (Column 2 plus Column 3 divided by 100), and Column 6 is the CPU time to generate test patterns, expressed in seconds.

Table 4.7 Results of experiment 1: all tests for a single fault.

Circuit c17.i

Affecting node 16 with rising transition,

Victim node 10 with value 0

Total 3 set of vectors: 12 out of 1024 combinations

$1T_d1T_uX$ first_p_amp=0.692 PO=22 type=P_n amp=0.897

$1\ T_d$ 10X first_p_amp=0.642 PO=22 type=P_n amp=0.775

$111T_uX$ first_p_amp=0.659 PO=22 type=P_n amp=0.822

Output statistics

0.2-0.4Vdd	0.4-0.6Vdd	0.6-0.8Vdd	>0.8Vdd
0	0	1	2

Total CPU run_time = 1 seconds

Table 4.8 Result of experiment 2: one test for each fault; Number of faults = 100; no timing criterion set at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetectable			
C432	33	56	11	89	1164
C880	41	46	13	87	1324
C1355	33	48	19	81	3866
C1908	50	34	16	84	2698
C2670	33	55	12	88	4542
C3540	29	49	22	78	4133
C5315	43	48	9	91	7090
C7552	31	58	11	89	7882
Average	36.625	49.25	14.125	85.875	4087.375

Table 4.9 Result of experiment 2: one test for each fault; Number of faults = 100; the longest path delay is set as the timing criterion at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetectable			
C432	5	74	21	79	1246
C880	7	75	18	82	1648
C1355	5	70	25	75	3968
C1908	16	65	19	81	2508
C2670	7	72	21	79	4867
C3540	4	70	26	74	4614
C5315	11	74	15	85	7745
C7552	9	77	14	86	8651
Average	8.0	72.125	19.875	80.125	4405.875

As we can see from Table 4.9, if there is a timing criterion set at primary outputs, then some large crosstalk effects that reach primary outputs may not violate the timing requirement and the program continues. The process terminates when either 1) a crosstalk effect reaches a PO and violates the timing constraint, 2) the search space is exhausted hence no test exists, or 3) the backtrack limit is reached and the process aborted. Therefore the percentage numbers in Column 3 and 4 increase, but the ATPG efficiency decreases. Since it takes time to search the PI space, the CPU time increases.

Another experiment was performed to investigate the relationship between the detection rate and the threshold used to filter small pulses. Figure 4.11 shows that if we increase the threshold, some pulses that propagate to outputs are filtered away, and the percentage detection rate decreases. An obvious example is that if we set the threshold to be 1, then the detection rate becomes zero.

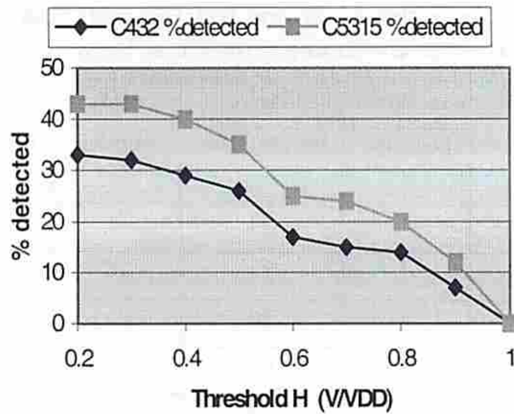


Figure 4.11 Detection rate vs. pulse threshold.

We also perform following experiments to investigate the dependence of detection rate on coupling capacitance, the ratio of affecting line to victim line driver strengths, and the signal transition times on primary inputs. Figure 4.12 shows that as the coupling capacitance increases, the crosstalk effect becomes more severe and hence the detection rate increases. Figure 4.13 shows that the detection rate increases as the affecting line to victim line driver ratio increases, because a stronger affecting line results in a larger coupling effect on the victim line. Figure 4.14 shows that the detection rate increase as the signal transition times at primary inputs decrease. This is because that if we make signal transitions faster at primary inputs, the transition that will occur on the affecting will also transit faster, which results in a larger crosstaslk effect and leads to a higher detection rate. These experimental outcomes confirm with the results we obtained from analytical expressions in Chapter 2.

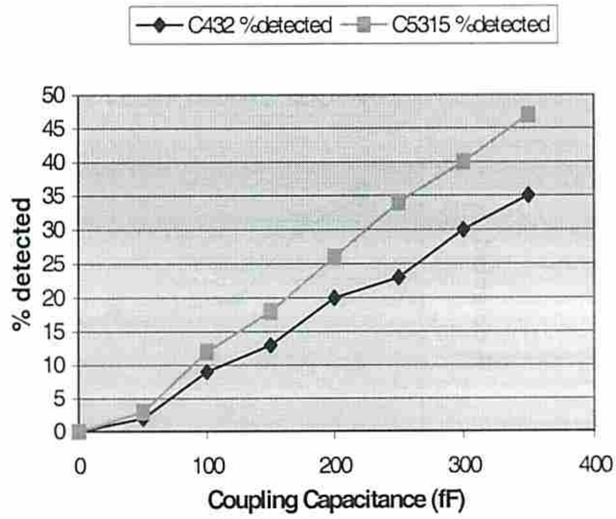


Figure 4.12 Detection rate vs. coupling capacitance.

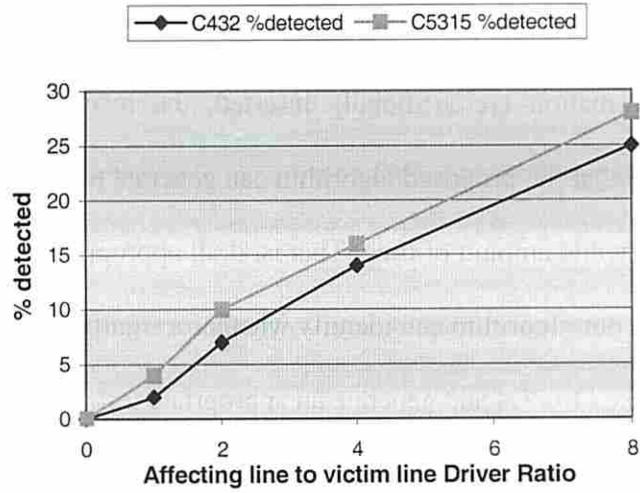


Figure 4.13 Detection rate vs. ratio of affecting to victim line driver strengths.

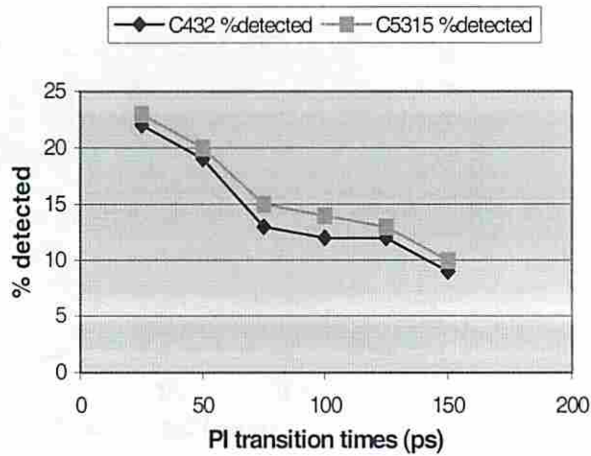


Figure 4.14 Detection rate vs. signal transition times at primary inuts.

Although in the preceding experiments the device sizes, coupling capacitance, and related information are artificially inserted, the results in Table 4.8 and Table 4.9 demonstrate that the proposed algorithm can generate tests for circuits of reasonable sizes within acceptable amount of time. That is, if all appropriate circuit and layout information is available, our algorithm can identify whether a significant crosstalk fault can be created and propagated to POs and generate an appropriate test.

Since the execution of the proposed algorithm requires a non-trivial amount of calculation time, test pattern generation for all signal pairs of a complex circuit is not practical. Therefore, only critical pairs of lines should be targeted. The selection of these critical lines should be based on the circuit configuration, manufacturing process information, layout, designer's knowledge and other relevant information. This

information is typically known in advance to the TG process, and should enable exclusion of many targets that cannot possibly cause errors at outputs. Thus, unlike stuck-at faults, we believe most circuits would have very few actual target crosstalk faults. In a separate piece of work, we are working on a preprocessor that prunes the space of targets to select a small set of potential targets that require process via XGEN.

4.6.2 Crosstalk Delay

In this section we illustrate the proposed algorithm for crosstalk delay. Consider the circuit shown in Figure 4.15. The channel length of each device is 0.35um and under each gate we indicate the ratio of widths of the PMOS and NMOS FETs. The gain factor ratio of the NMOS to PMOS FETs in the technology file (MOSIS 0.35um) used is 3.5. Assume that the sub-circuit on the left side of the dash line is separated by 1000um from the sub-circuit on the right side of the dash line. Hence lines 14, 24 and 13 are assumed to be about 1000um long and 4 um wide. In addition, assume that line 13 is the affecting line in metal2, line 24 is the victim line in metal3, and they are overlapped so that there is a significant coupling between them (C_m = 280 fF). The gate driving the affecting line 13 is assumed to be a buffer that has a strong driving strength. Gate sizes are computed to achieve signal transition times of 100ps. Primary inputs are assumed to have signal transition times of 100ps. All wire and gate capacitance correspond to a realistic layout, and gate delay is estimated as 110 ps for each gate.

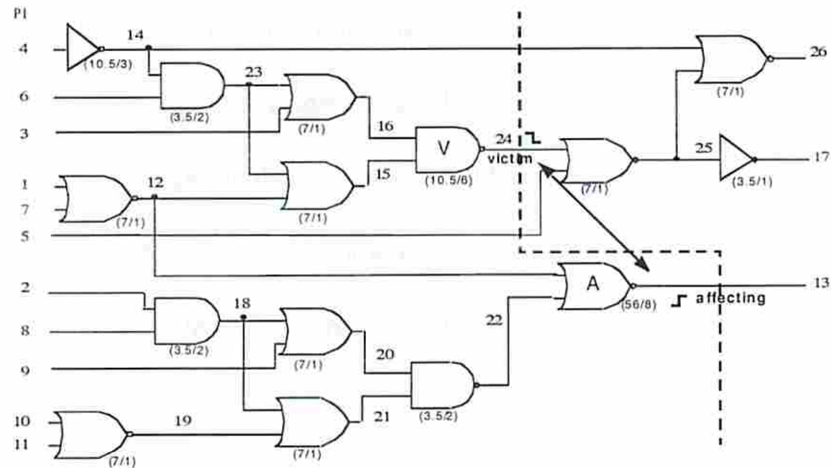


Figure 4.15 Example circuit to illustrate the algorithm.

There are six levels of gates from PI to PO. Assume that a 25 ps margin is allowed for each gate and in an aggressive design the clock period is set as (110 + 25) x 6 = 810 ps. Assume that we would like to create a crosstalk slowdown (falling transition) on the victim line 24. First we examine the timing windows for the affecting and victim line transitions. Both the affecting and victim line transitions can occur in time interval [220, 440]. By backtracing a possible PI assignment can be found where lines 1, 2, 6, and 9 are set to 1; lines 3, 10 set to 0; line 8 set to a rising transition; and lines 4 and 11 set to a falling transition. All conditions for side fan-in assignments are met to have the affecting line transition fast and the victim line transition as slow as possible.

By implication the affecting line starts to transit at time 437 ps with a rise time of 136 ps, and the victim line starts to transit at time 454 ps. We then calculate the crosstalk noise waveform according to the equations in section 2.4. The crosstalk slowdown signal on the victim line has an overshoot and a fall time of 370 ps. The waveforms for the

affecting and victim line signals are shown in Figure 4.16, with the time when the affecting line starts to transit set to zero. Because of the crosstalk effect, there is an increase of 142 ps on the signal delay (50% input to 50% output). This increase is due to both the change in signal slope and the overshoot.

To propagate this slowdown signal, the propagation path has to be sensitized and hence line 5 is set to 0. The slowdown signal propagates to line 17 at a very late time of 811 ps, which violates the timing requirements. Continuing the test generation process by backtracking in order to explore all PI combinations will find the best test that creates the worst case delay. However, since the test we found already creates the fastest affecting line transition and slowest victim line transition under the timing requirements, the branch-and-bound process keeps pruning the search space and the test we found is eventually declared as the worst case test.

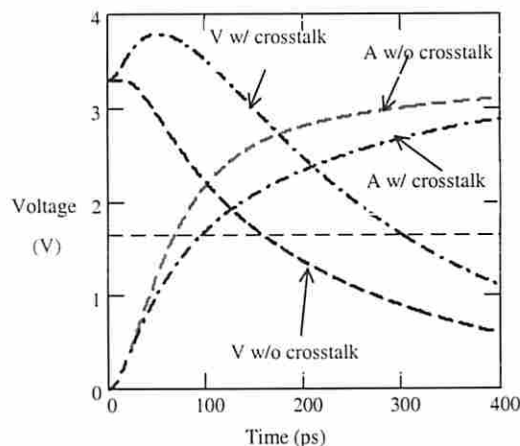


Figure 4.16 Waveforms on the victim and affecting lines.

Two sets of experiments were performed. In the first experiment a single crosstalk delay fault is targeted and the proposed algorithm used to generate all possible tests for the target fault. Tests associated with corresponding crosstalk delay that cause timing violations at POs are recorded so that the test creating the worst case timing violation at a PO can be identified. The results are shown in Table 4.10. All units are in pico second. The results correlate well with SPICE simulations. The timing criterion in Table 4.10 is the longest path delay of the circuit plus an extra delay slack of one gate delay. In Table 4.10 we can see that if there is no crosstalk effect ($C_m = 0$), then there is no timing violation at any primary output.

As we increase the coupling capacitance, the victim line signal become more delayed and its transition time increases. Also more test vectors can propagate the crosstalk delay signal to POs. This is because the delay slack is equivalently reduced and some vectors that could not cause timing violation before can do so now.

In the second experiment, for each circuit 100 pairs of affecting and victim lines are selected. If the selection of targets is completely random, then approximately 20 – 25 % of the targets have affecting and victim timing windows that do not overlap. Hence we preprocess the selection of targets so that the affecting and victim lines have overlapping timing windows. In addition, the victim lines are also located on critical paths so that a crosstalk effect propagating through these paths can have a chance to cause a timing violation. The proposed algorithm is applied to generate one test for each fault. The maximum number of backtracks per fault is limited to 1000. Results of the experiments

are shown in Table 4.11 (no timing criterion) and Table 4.12 (the longest path delay is set as the timing criterion).

From Table 4.12 we can again see that if there is a timing criterion set at the primary outputs then some crosstalk effects that reach primary outputs may not violate the timing requirement and hence become either undetectable crosstalk effects, or the TG aborts. The results in Table 4.11 and Table 4.12 again demonstrate that the proposed algorithm can generate tests for circuits of reasonable sizes, within an acceptable amount of time.

Table 4.10 Results of Experiment 1: all tests for a single fault.

Circuit C17: affecting node 10 with a rising transition, victim node 16 with a falling transition.

Tests	Arrival time of the crosstalk signal at the fault site (x100ps)	Transition time of the crosstalk signal at the fault site (x100ps)	Arrival time of the crosstalk signal at a PO (x100ps)
Cm = 0 fF 0 test found Timing criterion = 335 No slowdown timing violation at POs			
Cm = 200 fF 8 tests found Timing criterion = 335			
T_d1T_d10	213	283	335
$T_d1T_d1T_d$	213	283	335
$T_dT_u1T_d0$	214	287	339
$T_dT_u1T_dT_d$	214	287	339
T_d11T_d0	214	287	339
$T_d11T_dT_d$	214	287	339
$11T_d10$	214	287	339
$11T_d1T_d$	214	287	339
Cm = 300 fF 16 tests found Timing criterion = 335			
$T_dT_uT_dT_d0$	251	318	386
: (14 tests deleted) :			
$11T_d1T_d$	255	324	394

Table 4.11 Results of Experiment 2: one test for each fault; number of faults = 100; no timing criterion set at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetectable			
C432	35	55	10	90	1019
C880	28	63	9	91	1553
C1355	16	67	17	83	3173
C1908	33	54	13	87	2562
C2670	17	74	9	91	4914
C3540	10	72	18	82	4565
C5315	31	59	10	90	7030
C7552	14	73	13	87	8424
Average	23.0	64.625	12.375	87.625	4155

Table 4.12 Results of Experiment 2: one test for each fault; number of faults = 100; the longest path delay is set as the timing criterion at POs.

Circuit name	Successful TG (%)		TG Aborted (%)	ATPG Efficiency (%)	TG time (s)
	Detected	Undetectable			
C432	15	68	17	83	1167
C880	13	72	15	85	1664
C1355	6	71	23	77	3403
C1908	15	70	15	85	2555
C2670	9	76	15	85	4870
C3540	4	72	24	76	4661
C5315	12	74	14	86	7323
C7552	7	75	18	82	8481
Average	10.125	72.25	17.625	82.375	4265.5

Another experiment was performed to see the impact of skew on the detection rate of crosstalk delay. The result is shown in Figure 4.17. A skew of 1 implies that the transitions on the affecting and victim lines can be skewed for up to one gate delay, and a skew of zero means that both transitions have to switch simultaneously. Figure 4.17 shows that as the skew increases, the detection rate increases because it increases the search space for test vectors. However, if transitions are far apart from each other, then there will be no crosstalk delay effect and hence the detection rate saturates.

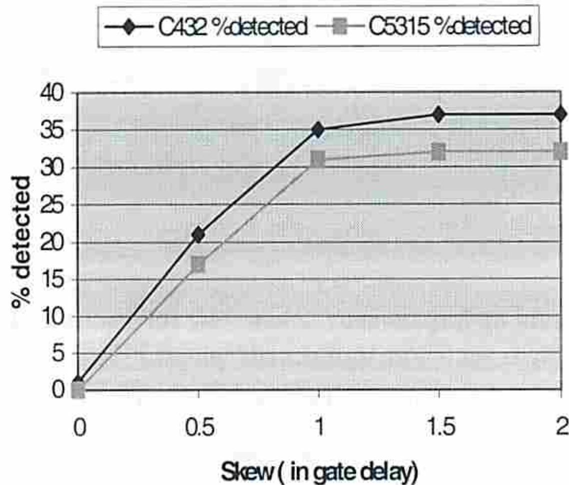


Figure 4.17 Detection rate vs. skew between affecting and victim lines.

A crosstalk delay signal can create a timing violation if there is not sufficient slack at the outputs. The following experiment was performed to study the amount of extra delay slack need to tolerate crosstalk delay. The result is shown in Figure 4.18. The amount of increased delay at a fault site is from 30-120%, and the transition time

increases from 10–110%. Because signal delay is accumulated along propagation paths, sufficient delay slack should be allocated at the outputs to avoid crosstalk slowdown causing a timing violation. Figure 4.18 shows that for these example circuits with crosstalk effects at least two and half extra gate delays should be used to ensure correct circuit operations.

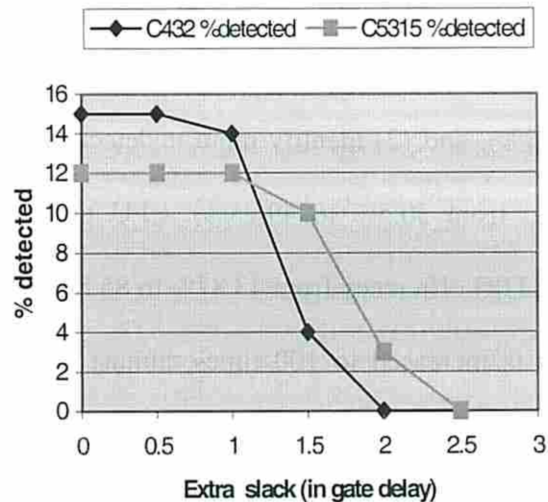


Figure 4.18 Detection rate vs. extra delay slack.

The dependence of crosstalk delay on the coupling capacitance, the ratio of affecting line to victim line driver strengths, and the signal transition times on primary inputs are similar to those for crosstalk pulse.

In [65] it was shown that the timing ranges of large circuits may shrink as fast as those for small circuits, even when only a small fraction of inputs are specified. Performance of XGEN with and without incremental timing refinement is shown in Table 4.13 for corsstalk pulse and Table 4.14 for crosstalk delay. No timing criterion is set at

primary outputs. Test targets are aborted if the number of backtracks exceeds 1000. We can see that with incremental timing refinement the ATPG efficiency increases significantly, because many objectives that are not achievable in timing-wise sense are identified early in the test generation process and lead to a backtrack. Hence the search space is reduced. For example, from Table 4.13 we found that incremental timing refinement help our timing oriented test generation algorithm by reducing the search space and so (1) find more detectable targets (the average detection rate is increased from 13.38% to 36.63%), and (2) identify more undetectable targets (the average undetected rate is increased from 30.5% to 49.25%). CPU time is increased about 10 times in improving the ATPG efficiency from 43.87% to 85.87%. Without ITR used, even the test generation time is increased to 100 times, almost no improvement on efficiency was observed.

Table 4.13 Comparison of ATPG efficiency for crosstalk pulse with and without incremental timing refinement; no timing criterion was set at POs.

Circuit Name	Successful TG (%)				TG Aborted (%)		ATPG Efficiency (%)		TG Time (s)	
	Detected		Undetectable							
	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR
C432	11	33	31	56	58	11	42	89	116	1164
C880	16	41	38	46	46	13	54	87	134	1324
C1355	9	33	22	48	69	19	31	81	414	3866
C1908	14	50	33	34	53	16	47	84	346	2698
C2670	15	33	34	55	51	12	49	88	270	4542
C3540	7	29	24	49	69	22	31	78	492	4133
C5315	19	43	38	48	43	9	57	91	177	7090
C7552	16	31	24	58	60	11	40	89	502	7882
AVE.	13.38	36.63	30.5	49.25	56.13	14.13	43.87	85.87	307	4087

Table 4.14 Comparison of ATPG efficiency for crosstalk delay with and without incremental timing refinement; no timing criterion was set at POs.

Circuit Name	Successful TG (%)				TG Aborted (%)		ATPG Efficiency (%)		TG Time (s)	
	Detected		Undetectable							
	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR
C432	12	35	31	55	57	10	43	90	101	1019
C880	18	28	35	63	47	9	53	91	86	1553
C1355	12	16	20	67	68	17	32	83	375	3173
C1908	17	33	33	54	50	13	50	87	294	2562
C2670	19	17	34	74	47	9	53	91	219	4914
C3540	8	10	28	72	64	18	36	82	457	4565
C5315	24	31	37	59	39	10	61	90	154	7030
C7552	20	14	25	73	55	13	45	87	478	8424
AVE.	16.25	23	30.37	64.63	53.37	13.37	46.63	87.63	270	4155

Similar comparisons are performed for the case when the longest path delay is used as the timing criterion at primary outputs. The results are shown in Table 4.15 for crosstalk pulse and Table 4.16 for crosstalk delay. Again we see that the ATPG efficiency increases significantly with incremental timing refinement implemented.

Table 4.15 Comparison of ATPG efficiency for crosstalk pulse with and without incremental timing refinement; the longest path delay was used as the timing criterion at POs.

Circuit Name	Successful TG (%)				TG Aborted (%)		ATPG Efficiency (%)		TG Time (s)	
	Detected		Undetectable							
	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR
C432	4	5	33	74	63	21	37	79	144	1246
C880	7	7	38	75	55	18	45	82	174	1648
C1355	5	5	23	70	72	25	28	75	483	3968
C1908	7	16	34	65	59	19	41	81	381	2508
C2670	7	7	35	72	58	21	42	79	310	4867
C3540	3	4	26	70	71	26	29	74	682	4614
C5315	10	11	39	74	51	15	49	85	441	7745
C7552	9	9	26	77	65	14	35	86	627	8651
AVE.	6.5	8	31.75	72.13	61.75	19.87	38.25	80.13	406	4406

Table 4.16 Comparison of ATPG efficiency for crosstalk delay with and without incremental timing refinement; the longest path delay was used as the timing criterion at POs.

Circuit Name	Successful TG (%)				TG Aborted (%)		ATPG Efficiency (%)		TG Time (s)	
	Detected		Undetectable							
	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR	w/o ITR	ITR
C432	5	15	32	68	63	17	37	83	138	1167
C880	9	13	40	72	51	15	49	85	112	1664
C1355	6	6	22	71	72	23	28	77	423	3403
C1908	9	15	34	70	57	15	43	85	354	2555
C2670	9	9	34	76	57	15	43	85	277	4870
C3540	3	4	30	72	67	24	33	76	539	4661
C5315	10	12	38	74	52	14	48	86	368	7323
C7552	7	7	26	75	67	18	33	82	512	8481
AVE.	7.25	10.13	32.13	72.25	60.63	17.63	39.25	82.37	340	4265

4.7 Summary

Crosstalk effects can have a significant impact on signal integrity and delay. To ensure correct circuit operation, coupling effects, such as crosstalk-induced pulses, slowdown and speedup, should be taken into consideration in validating circuit designs and estimating the timing of critical paths.

In this chapter an algorithm to generate tests for crosstalk effects is proposed. This algorithm not only considers noise effects such as speedup, slowdown and pulses as new logic values, but also takes into consideration information such as finite noise energy and input arrival skews to accurately characterize noise strength. The ATPG algorithm utilizes conditions that help excite the maximum crosstalk effect and propagate the

crosstalk signal to POs under desired timing requirements. In addition, this ATPG algorithm includes the concept of gate delay, signal arrival time, signal strength and rise/fall times. By using the path delay information obtained by circuit preprocessing and/or the analog cost function, preferred paths can be selected during the backtrace and propagation processes. Because the proposed algorithm implicitly explores all PI combinations, it is beneficial to limit the search space to improve efficiency. A branch-and-bound technique is proposed to reduce the search effort. Finally, while most ATPG algorithms attempt to only satisfy a set of logical constraints, this algorithm also maximizes an objective function. Experimental results show that the method can be applied to selected crosstalk faults in circuits of reasonable sizes.

The proposed algorithm can also generate all tests for a crosstalk effect so that a matching with functional tests can be performed to determine whether the functional tests cover the tests for the crosstalk effects.

For crosstalk-induced delay, in our TG process a transition on the victim line is created and propagated along a possible long path, and receives the largest amount of crosstalk effect under certain timing requirements. From the point of view of both crosstalk effect and signal propagation, the amount of delay imposed on the victim line signal is maximized with respect to the given constraints. Hence the test vectors generated can be considered as a complementary set of tests for the purpose of delay testing.

The algorithm has been implemented, resulting in the program XGEN. XGEN has been run on numerous examples and found to be accurate, effective and efficient.

Chapter 5

Future Extensions to our ATPG

We have developed a mixed-signal test generation algorithm that will generate test patterns for a targeted crosstalk effect. These test patterns, if applied to a circuit, would generate the largest possible crosstalk effect (pulse or speed-up, slow-down) at a memory device or a primary output, i.e., maximize the probability of creating an error. The current version of the algorithm employs macromodels for primitive CMOS gates for the case when only a single crosstalk effect can be excited. In the future, it is possible to extend our test generation framework in several ways to improve and/or optimize the test generation process.

The ideal of improvement is to extend the capability of our current test generator framework to deal with more general CMOS gates, different types of logic elements including dynamic gates and latches, and multiple crosstalk effects. In addition, we include the discussion of techniques to automate the process of target fault extraction.

5.1 Extension to general gates

Our current generation TG can deal with any circuit contains primitives such as AND, OR, NAND, NOR and NOT gates. However, both system blocks and custom-

designs may contains other elements such as complex gates, dynamic gates and latches. Additional macromodels can be developed to enable propagation of crosstalk effect via a wider range of circuit elements, such as complex COMS gates, dynamic gates, and latches, and under a wider range of conditions, such as crosstalk pulses at multiple-inputs of a circuit element, and simultaneous presence of crosstalk pulse and delays.

5.1.1 General CMOS gates

For general CMOS gates that have series and/or parallel connected networks, by using an approach similar to the one for improving MOS macromodel accuracy [28], extension of our work can be achieved by considering transistors as conducting resistors and combining them (using effective conductance β_{eff} or a scaling factor) into one or more equivalent devices connected in series. That is, given an input pattern, the complex gate is first mapped into an equivalent NAND gate, and then the NAND gate can be processed using the approach described in Chapter 3 to further reduce the NAND gate into an equivalent inverter. The key issues here are the number of switching inputs and their switching times. Several waveform representation techniques for overlapping inputs are presented in [26], [29], [45]. By applying these techniques, multiple switching inputs can be reduced to one equivalent input with an effective switching time. Once reduced to the single switching input case, nMOS transistors with a "1" at their inputs can be treated as resistors and collapsing them is straightforward.

In addition to complex gates, macro-models of commonly used custom-design elements such as full adders, MUXes, and bi-directional buffers may also be developed. These elements either contain circuit structures where the above-mentioned collapsing techniques cannot be applied, or they are used in a repetition way so that modeling them as an entity can improve efficiency. One approach to solve this problem is to develop a strategy for a simple N-port general network and apply it to these elements. The N-port general network provides a transfer function template, and for each element different requirements and conditions may be imposed for its individual functionality. Hence for each element we expect to obtain the corresponding impulse transfer function that can be used in our analytic approach. Details of the N-port general network and properties of different circuit elements needs further research and investigation.

5.1.2 Dynamic gates

Circuits often contain dynamic gates. The use of dynamic gates introduces new issues to be considered. Consider the circuit shown in Figure 5.1(a) where G is a gate, x is the input and y is the output. If the gate G is complementary (Figure 5.1(b)), then the output crosstalk manifests as a pulse as shown in Figure 5.1(c). On the other hand, if a dynamic gate is used such as the one shown in Figure 5.1(d), then the charge lost will not be restored and the same input crosstalk effect manifests as a degraded voltage (Figure 5.1(e)). While a crosstalk pulse must be propagated to a memory element before it can be treated as a permanent Boolean error, the degraded voltage can be a Boolean error by

itself if the voltage drop is sufficiently large. This difference will result in the consideration of additional factors for the selection of propagation path in the test generation process. In addition, the output of a dynamic gate can change only in the evaluation phase and it always is a falling transition. These also introduce timing and signal transition direction constraints into the propagation of crosstalk effect in the analytic models.

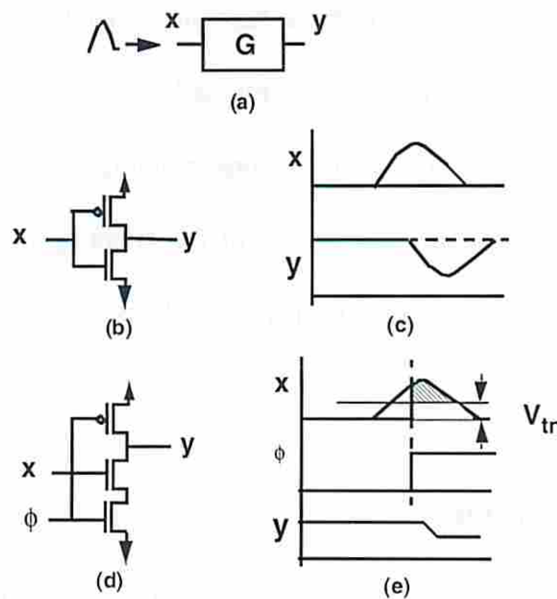


Figure 5.1 Crosstalk effect on static gate and dynamic gate: (a) a basic gate G; (b) gate G implemented as a static gate; (c) corresponding input/output pulse waveform for (b); (d) gate G implemented as a dynamic gate; (e) corresponding input/output pulse waveform for (d).

Our current analytic models can deal with static CMOS gates whose outputs are allowed to have both rising and falling transitions. To take into consideration dynamic gates, one approach is to add additional timing mechanism to consider clock phases and modifying the analytic noise evaluation procedures used in our models. The clock-phase

problem can be approached by using an additional reference time index to indicate how a crosstalk pulse aligns with a clock edge, i.e., how much of the waveform falls within the evaluation phase. For a degraded output waveform, the computation of the output waveform can be easily achieved by removing the restoring constant (or pull-up resistor) used in the analytic procedures.

5.1.3 Latches

For circuits that contain latches there are several new issues to be considered. For example, consider the circuits shown in Figure 5.2. Figure 5.2(c) shows a crosstalk pulse on the line D and its relative position with respect to the clock transition. If the latch is dynamic, as shown in Figure 5.2(a), then the severity of the effect is proportional to the area of the pulse that is shown shaded in Figure 5.2(c). Note that the effect of the pulse can be severe, even if the pulse amplitude is not very high, i.e., provided that the area of the shaded region is large. On the other hand, if the latch is static, as shown in Figure 5.2(b), then to create a Boolean error the amplitude of the pulse must be large enough to overcome the feedback provided by the weak inverter in the latch. Hence for different latch designs, there are different criteria for noise to lead to a Boolean error.

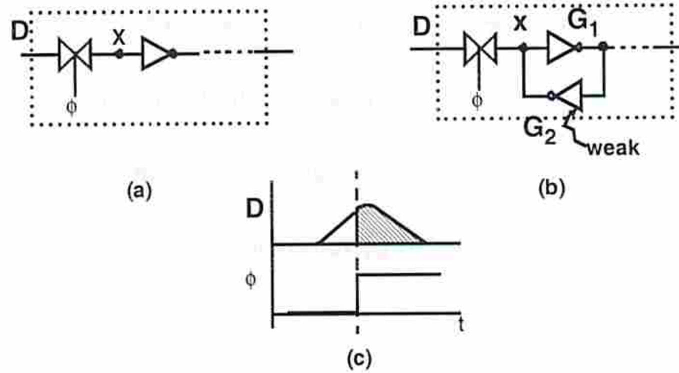


Figure 5.2 (a) Dynamic latch, (b) static latch, (c) crosstalk pulse.

Since a static latch has an internal feedback path, it may be necessary to consider a latch as a single entity instead of regarding it as a chain of gates. For example, consider the cross-coupled inverter latch shown in Figure 5.3 where a pulse is injected into the series-voltage source. In the latch's initial state, node A is low and node B is high. Simulation results show that the latch is stable until the noise amplitude reaches 1.83V (V_{DD} = 3.3V), and after that the value of A' switches. But the noise stability model for gate1 has a stability threshold of 1.46V, i.e. for a single gate1, a noise amplitude of 1.46V can make the output of gate1 switch. The difference in noise tolerance implies the conservative nature of the current approach that deals with one individual gate at a time, instead of considering the relationship between gates, such as a feedback loop forming a latch. This pessimism may introduce excessive false alarms in design validation.

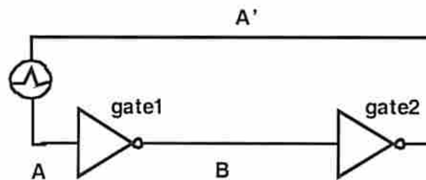


Figure 5.3 A cross-coupled inverter latch.

We propose two ways to address this pessimistic approach. The first way is a simple 1st-order approximation of the difference in noise tolerance. This can be achieved by using an empirical scaling constant (for example, 1.83/1.46 = 1.253) to adjust the noise tolerance threshold. Then the latch (two gates in the above example) can be considered as a single entity where pulses having an amplitude greater than 1.83V will flip the output value, while pulses smaller than 1.83V can be ignored. The second approach is to consider a latch as a loop of gates as in Figure 5.3. Since gate2 actually works against the input pulse, the strength of the corresponding transistor that fights against the pulse in gate2 is computed and added into gate1' pull-up strength when applying analytic models to solve the pulse response at gate1's output. Similarly, when computing pulse propagation from point B to A', the same procedure can be again performed. Details of the transistor strength conversion and feasibility of this approach need further investigation.

5.2 Multiple crosstalk effects

In our work the cross-coupling is assumed to occur between one affecting line and one victim line. In reality the coupling effect can have more complicated scenarios such as a) multi-way crosstalk, where n multiple lines are coupled to a single victim line, and b) multi-level crosstalk, where several victim lines along a circuit path are coupled with one or more lines in their vicinity and crosstalk effects accumulate. Additional macromodels need to be developed to capture these crosstalk scenarios.

5.2.1 Multiple-way and multiple-level crosstalk

In the multi-way crosstalk scenario, shown in Figure 5.4, several couplings may results in multiple crosstalk pulses on the victim line. The generalization to more affecting lines is obvious. One approach to this problem is to characterize and estimate an "effective" pulse that is equivalent to these multiple pulses. The victim line in Figure 5.4 can be modeled as shown in Figure 5.5.

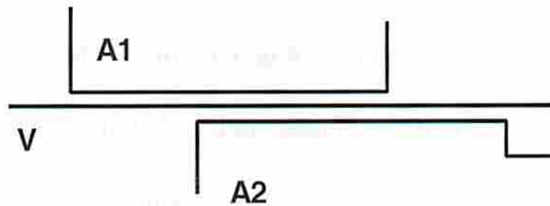


Figure 5.4 A1-A2 are affecting lines and V is the victim line

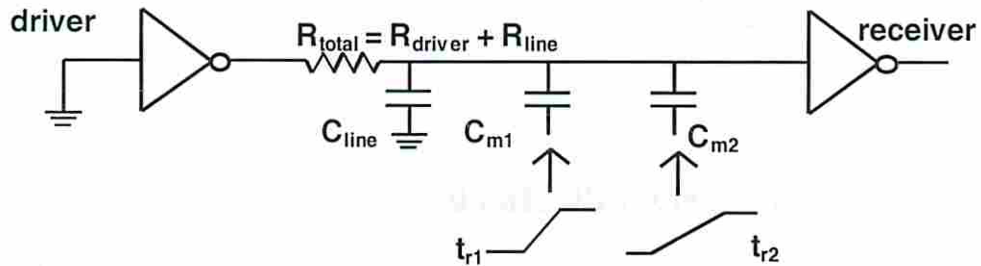


Figure 5.5 Victim line circuit model for multi-way coupling; C_{m1} and C_{m2} are coupling capacitance to A1 and A2; t_{r1} and t_{r2} are switch times of signals on A1 and A2, respectively.

As shown in Chapter 2, the slope of the affecting signal has a significant effect on the noise. Traditionally this multi-way effect has been dealt with by selecting the fastest transition at the affecting signal and lumping all the coupling capacitance together. This

results in an overestimation of noise strength, especially when affecting line signals are skewed. In real circuits, several affecting lines can switch at different times and rates.

The basic idea is to utilize the principle of superposition. With sufficiently wide overlapping windows, different affecting lines can switch as shown in Figure 5.6. We can divide the switching time into several sections. For example, from Figure 5.6(a) in the first time slot I the affecting line A1 is switching while A2 is static. In time slot II both signals are switching. In slot III A1 has finished switching while A2 is still in transition. In slot IV both have completed their transitions. The generalization to more signals is obvious, but in practical we may need to lump signals into slope categories (slow, medium, fast, etc.) to get around a potentially large number of time sections. With these time partitions, a first-order approximation of crosstalk waveform can be obtained as:

$$V(t) = R_{total} \sum_{j=1}^{i} \frac{C_{mj}}{t_{rj}} \left(1 - e^{-\frac{(t - t_{rj})}{R_{total} C_{total}}} \right), \qquad \text{for time section I} \sim \text{III} \ (i = 1 \sim 3)$$

$$V(t) = R_{total} \sum_{j=1}^{i} \frac{C_{mj}}{t_{rj}} \left(1 - e^{-\frac{(t_{s4} - t_{rj})}{R_{total} C_{total}}} \right) e^{-\frac{(t - t_{s4})}{R_{total} C_{total}}}, \qquad \text{for time section IV}$$

where R_{total}, C_{mj}, t_{rj} are shown in Figure 5.5, $C_{total} = C_{line} + \sum C_{mj}$, and t_{s4} is the start time of section IV.

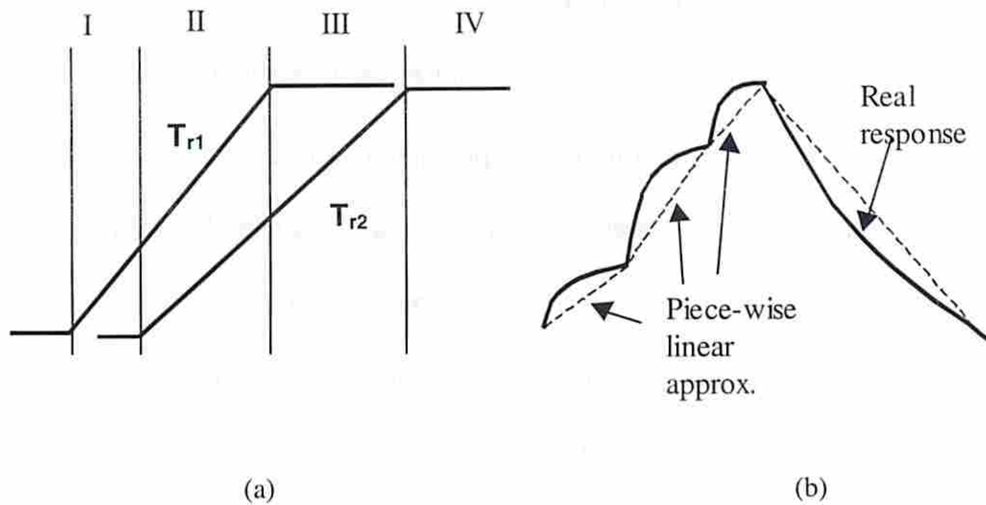


Figure 5.6 (a) Different affecting signal slopes; (b) piece-wise linear approximation of pulse waveform.

Applying this multi-slot exponential waveform to the transfer function of the receiver does not lead to a simple analytic result. Therefore we propose a piecewise linear approximation by making a line go through two endpoints of each waveform section as shown in Figure 5.6(b). This approach may give a sufficiently accurate result. A further simplification to the approximation piece-wise linear waveform is to take the average of the approximate slot slopes. This can significantly reduce the complexity by represent the pulse waveform as was done in Chapter 3.

If the pulses on the victim line do not overlap, then the above approximation is not applicable. An alternative for this situation is to deal with each individual pulse separately and then estimate the total effect at the receiver output. The previous approach for overlapped pulses is to "lump" the input pulses first and compute the output response, and the later approach for non-overlapped pulses is to "lump" the several output

responses due to multiple input pulses into one output signal. Since in the second approach the output responses are now "overlapped", then the first approach can be applied to combine the output responses into one approximated pulse. For the second approach a "signal queue" for each line has to be maintained. Although the complexity increases, with this signal queue we can not only consider multiple pulses but also take into consideration the dynamic glitch case, i.e., a pulse followed by a transition or vice versa, which is simplified as a transition only in the current version of test generator. Details of the second approach need further analysis.

For the multi-level crosstalk scenario, multiple coupling can occur across several logic levels and noise effects are accumulated along a victim "path" as shown in Figure 5.7. Depending on the type and timing of a noise effect, the noise can be increased or decreased by the various affecting signals. Since pulses at each logic level can be computed and propagated separately, by applying superposition principle the accumulation of noise effects can be dealt with in a way similar to that of multiple-way coupling.

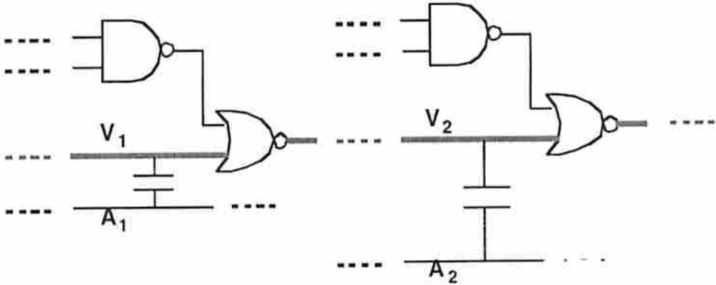


Figure 5.7 Example of multiple-level coupling: a victim "path".

Our current test generation framework processes only one affecting and one victim line. To deal with multiple affecting or victim lines, additional techniques needs to be developed. Since it is almost impossible to satisfy all requirements for all affecting and victim lines, the issue for test generation is how to select affecting/victim lines such that the coupling effects can be maximized and propagate the crosstalk effect to a primary output maximally.

For the case of multiple-way coupling (multiple affecting lines to one victim line), there are three heuristic approaches: 1) select the affecting lines in the order of their coupling capacitance; 2) select the affecting lines in the order of the sizes of their input cones, meaning, the number of primary inputs supporting them; 3) a weighted approach combining both 1) and 2). The idea behind the first approach is straight-forward, namely, we try to activate as many large coupling effects as possible. However, there is a chance that selecting the affecting line with the largest coupling capacitance may block the activation of other affecting lines and results in a smaller overall coupling effect. Therefore an alternative approach, the second approach, provides the idea of selecting the affecting lines according to the sizes of their input cones. Ideally speaking, selecting the affecting lines in this manner may increase the possibility of satisfying more affecting lines, namely, more coupling effects can be accumulated. However, it is also possible that the affecting lines with large coupling capacitance are not selected (because of existing assignments for satisfying previous affecting lines), and results in a smaller overall coupling effect. Thus a combination of approaches may be useful, i.e., a weighted technique for achieving maximum coupling effects. In this way, each affecting line is

associated with two sets of parameters: the coupling capacitance and the set of primary inputs supporting its input cone. The user can assign different weights to each set of parameters.

For the case of multiple-level coupling, the problem becomes more complicated since each level of coupling can already be a multiple-way coupling. Therefore, while trying to obtain a maximal coupling effect for each level, it is important to take all levels of couplings into consideration to achieve an overall maximum. One heuristic approach is first to find the local maximum coupling effect for each level using the approach for multiple-way coupling. Then each level of coupling can be associated with two sets of parameters: the amount of its local maximal coupling effect and the set of primary inputs needed to achieve this effect. Similarly, this can be mapped into the technique for solving multiple-way coupling (or the channel routing problem). Since this approach is greedy in nature, further study is necessary to obtain results that are close to optimal, i.e., maximum overall coupling effects.

5.2.2 Static glitches

A circuit is said to contain a hazard if there exist some possible combinations of values of delays and input transitions that will produce a glitch. Our current test generator can accept hazard-free circuits as inputs and generate tests for them to detect crosstalk faults. Since glitches created by signal skews are also pulses, we would like to extend our work to be able to deal with these cases.

Because of the existing capability of dealing with pulse inputs in our analytical models, one possible approach in considering static glitches as a source of pulses is to transform the skewed input signals into a virtual, or effective, input pulse. Then we can apply the same techniques described in Chapter 3. The idea is illustrated in Figure 5.8. Figure 5.8(a) shows an ideal situation to create a static glitch at the output of a NAND gate with two skewed input a and b, and Figure 5.8(b) is the transistor diagram for the NAND gate.

In reality signals are not step functions. Hence a more realistic case with two skewed and overlapping signals a and b is shown in the leftmost figure in Figure 5.8(c). As we can see the discharge current is always limited by the smaller input voltage applied to the transistors. That is, in section I the discharge current is limited by the amount of current allowed by transistor M1 because of the smaller voltage value at the input b. In section II the current is limited by transistor M2 for the same reason. Assume that M1 and M2 are the same size, as is usually the case. Since in section I transistor M2 is more "ON" than M1, the bottleneck of the current flow is M1. Since input b is a rising transition, the equivalent effect is that the NAND gate is turning ON, i.e., discharge current is increasing. On the other hand, in section II the transistor M1 (input b) is more "ON" than M2 (input a), and hence the bottleneck of the current flow now becomes M2. Since input a is a falling transition, the equivalent effect is that the NAND gate is turning OFF, i.e., discharge current is decreasing.

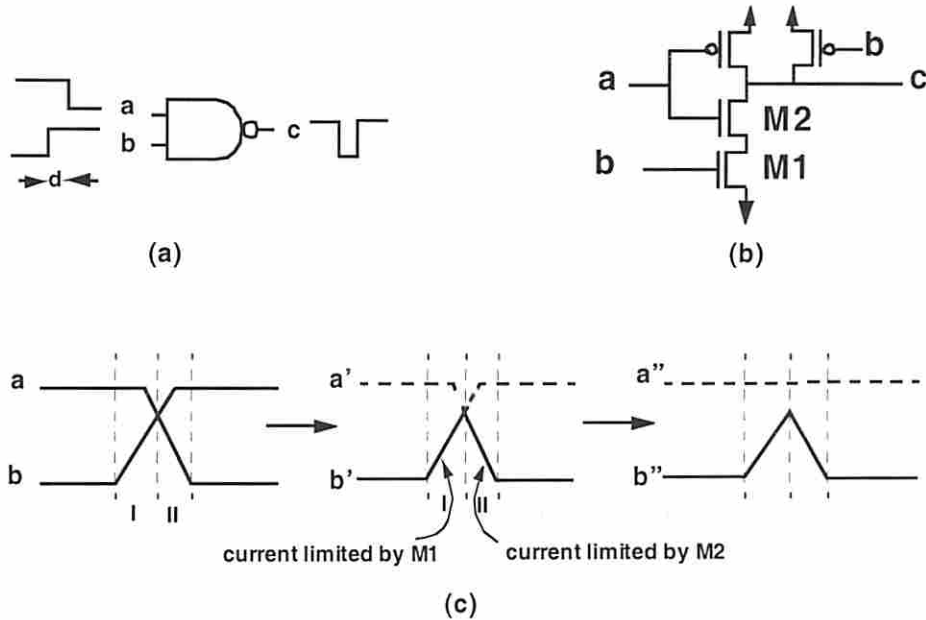


Figure 5.8 (a) Creation of a static glitch, (b) transistor diagram of a NAND gate, (c) creation of an equivalent input waveforms

Therefore, the "virtual" effective inputs applied to the NAND gate can be regarded as the case shown in the middle of Figure 5.8(c), where the solid line curve serves as the pulse input with the equivalent effect of skewed signals. Since the dashed line (input a') is not the controlling voltage of the discharge current, conceptually it can be further simplified as a stable "1". Thus, the equivalent input signals to the NAND gate are transformed into the case shown in the rightmost picture in Figure 5.8(c). This approach transforms the skewed input signals that create a static glitch into the case that one input is a pulse and the other is a stable 1. Hence we can apply the same method used in Chapter 3 to compute the output response.

Because the output static glitch is now modeled as a pulse, the interaction between static glitches and crosstalk pulses can be dealt with using the same techniques described

for multiple-crosstalk effects. In addition, the model described in this section can also be used to deal with the case when multiple pulses appear at the inputs of a gate. In such a case, one can apply the technique shown in Figure 5.8 to obtain an "effective" input pulse and then compute the corresponding output response. Details of this approach need further analysis.

5.3 Target fault extraction

A target fault site is a circuit location where the crosstalk effect is significant. The objective of this task is to identify a list of crosstalk sites to be targeted and to develop low-complexity procedures to identify parts of circuits where detailed extraction and test generation must be performed.

Extractors that analyze the 3-dimensional structure of a VLSI layout are used by designers to obtain a detailed and accurate circuit model (including parasitics) for simulation. These models are then used by simulators to identify problems associated with various types of crosstalk. To deal with various effects caused by small feature sizes and their proximity, accurate circuit extraction tools usually employ high complexity techniques that typically solve field equations with appropriate boundary conditions. Consequently detailed extraction cannot be applied to an entire chip, and circuit designers often have to manually identify parts of circuits for accurate extraction. That is, areas of a circuit targeted for crosstalk analysis are selected manually. Since crosstalk is a global

phenomenon, i.e. it deals with large circuit structures, it may be necessary to develop a new approach to identify areas of a circuit that may have potential crosstalk faults.

The proposed research will automatically identify an initial list of crosstalk sites to be targeted for test generation. The approach will begin by using information about timing (delays, rise/fall times, slack), types of logic (static or dynamic), device sizes, noise threshold of latches, and first-order layout parasitics to identify an initial list of crosstalk sites. Those targets for which the probability of a logical error is determined to be low will be eliminated. For the remaining targets, adaptive techniques can be developed to perform increasingly more accurate extraction, test generation, and simulation-based validation on increasingly larger regions of a circuit surrounding the site of a target to be validated. As the test generation process proceeds for a given target, new line values are specified and gates and wires are added to the active part of the circuit under consideration. This information can be fed back to drive another more detailed extraction process. The new extracted values are then fed to the test generation system and used to either prune certain branches or continue the search in a new direction. Thus the level of extraction depends on the need and is dynamic. This will help decrease the time required for extraction, validation, and test generation for crosstalk. It will also enhance the quality of validation and tests generated since it will use accurate extraction in all parts of the circuit where it is required, instead of in only the parts identified by the circuit designers.

The key issues that will be considered during this development are the identification of the computational complexity of extracting certain circuit parameters, the

space complexity of the resulting circuit models, and their impact on the accuracy and complexity of validation and test generation for crosstalk.

Once proper circuit extractions are available, a layout scanning and filtering mechanism to automatically identify potential crosstalk fault sites can be developed. For instance, in the following example we will illustrate a simple mechanism to locate possible crosstalk fault sites. Consider the circuit shown in Figure 5.9. The input to the affecting line is a ramp signal with rise time t_r.

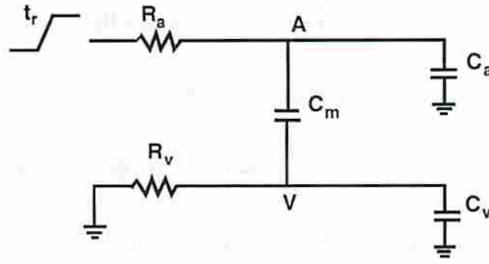


Figure 5.9 Crosstalk circuit with a ramp signal at the affecting line with rise time t_r.

By applying the procedure described in Chapter 2, we can obtain the waveform of the crosstalk pulse at the victim line as:

$$V_v(t) = \frac{R_v C_m V_{DD}}{\tau_0 t_r}(\tau_0 + \tau_1 e^{-t/\tau_1} - \tau_2 e^{-t/\tau_2}), \qquad \text{for } t < t_r, \quad (5\text{-}1)$$

$$V_v(t) = \frac{R_v C_m V_{DD}}{\tau_0 t_r}\{\tau_1 e^{-t/\tau_1}[1 - e^{t_r/\tau_1}] - \tau_2 e^{-t/\tau_2}[1 - e^{t_r/\tau_2}]\}, \qquad \text{for } t > t_r. \quad (5\text{-}2)$$

where

$$\tau_0 = \sqrt{[R_a(C_a + C_m) + R_v(C_v + C_m)]^2 - 4R_aR_v(C_aC_m + C_vC_m + C_aC_v)},$$

$$\tau_1 = \frac{2R_aR_v(C_aC_m + C_vC_m + C_aC_v)}{[R_a(C_a + C_m) + R_v(C_v + C_m)] + \tau_0},$$

$$\tau_2 = \frac{2R_aR_v(C_aC_m + C_vC_m + C_aC_v)}{[R_a(C_a + C_m) + R_v(C_v + C_m)] - \tau_0}.$$

The peak voltage of the pulse can be obtained by differentiating equation (5-2) and solving it for the peak time, and then substitute the peak time back into equation (5-1). Hence we obtain the maximum pulse amplitude as

$$V_{vmax} = \frac{R_vC_mV_{DD}}{\tau_0 t_r}\{\tau_1[1 - e^{t_r/\tau_1}]\left[\frac{1 - e^{t_r/\tau_1}}{1 - e^{t_r/\tau_2}}\right]^{\tau_2/(\tau_1-\tau_2)} - \tau_2[1 - e^{t_r/\tau_2}]\left[\frac{1 - e^{t_r/\tau_1}}{1 - e^{t_r/\tau_2}}\right]^{\tau_1/(\tau_1-\tau_2)}\}. \tag{5-3}$$

The circuit can have a potential error if V_{vmax} is greater than some value. For example, if V_{vmax} is larger than the inversion voltage of an inverter (i.e. approximately $0.5V_{DD}$), than the output of the inverter can have erroneous switching. On the other hand, if the pulse is applied to a dynamic gate, then the evaluation network may be accidentally turned ON and start to discharge the output if V_{vmax} is greater than the threshold of a transistor and has sufficient duration. Therefore the threshold for creating an error can vary for different style of gates, and hence can be set as a variable, say E_{th}. Hence the criteria function for a potential error is $V_{vmax} \geq E_{th}$.

R_a and R_v are resistors that are used to approximate the conducting channel of transistors. R_a (or R_v) can be expressed as:

$$R_a = \frac{1}{\alpha\mu C_{ox}(W/L)V_{DD}},$$

where α is a constant, μ is the electron mobility, C_{ox} is the gate capacitance per square, W is the channel width, and L is the channel length.

In equation (5-3), all parameters (R_a, R_v, C_a, C_v, and C_m) except t_r can be extracted from the layout. Setting t_r to a constant, by scanning a layout and using the criteria function we can identify possible locations that have potential crosstalk. If timing information, such as t_r is available, then this can be used to locate crosstalk sites with more accuracy.

A more complex version of the crosstalk-site extractor can be developed by adding more constraints and information such as signal timing (arrival time, rise/fall time), type of logic used in the circuit design, device sizes (driver strength ratio), ,parasitics (coupling capacitance, load capacitance) and the propagation probability through paths towards primary outputs.

5.4 Summary

For the analysis of more general gates, we could develop macromodels to include commonly used latched and primitive dynamic gates. Also new noise approximation techniques need be developed for multiple crosstalk effects. In addition, to efficiently and accurately identify target crosstalk sites, we need to develop techniques that will eliminate sites that certainly can not lead to a crosstalk error.

Chapter 6

Conclusions

Crosstalk effects can have a significant impact on signal integrity and delays. To ensure correct circuit operation, coupling effects, such as crosstalk-induced pulses, slowdown and speedup, should be taken into consideration when validating circuit designs and estimating the timing of critical paths.

The objective of this work is to develop a general methodology to analyze crosstalk and generate tests for crosstalk effects that are likely to cause errors in high speed VLSI circuits. We studied crosstalk due to capacitive coupling between a pair of lines. Closed form equations were derived quantifying the dependence of pulse attribute on the values of circuit parameters and the rise time of the input transition. These expressions show that the severity of the crosstalk pulse is directly proportional to the coupling capacitance and the ratios of the strengths of the drivers driving the two lines, and inversely proportional to the load capacitance on each line. These facts can be used to identify pairs of circuit lines where crosstalk may be significant and hence should be analyzed explicitly. Further, it is shown that while the maximum amplitude of the crosstalk pulse diminishes rapidly as the rise/fall time of the input increases, the energy of the pulse is almost independent of the input rise/fall time for a realistic range of rise/fall time values. If the rise/fall time of the input to a candidate pair of lines is known to be large, then it may not be necessary to analyze the effect of crosstalk. We also studied how

crosstalk causes speedup/slowdown when signals change in the same/opposite directions. Qualitatively, the dependence of slowdown and speedup on circuit parameters is similar to that observed for crosstalk pulses. Also, it was found that the faster the transition on the affecting line, the greater is the slowdown on the victim line. Finally, it was found that the skew required for the maximum crosstalk slowdown to occur is proportional to the ratio of the drivers driving the two lines. If the drivers are the same size, crosstalk slowdown is the highest when both inputs have simultaneous transitions. The magnitude of slowdown decreases as the skew between the input transitions increases.

The crosstalk effect was shown to be significantly aggravated by variations in the fabrication process. The significance of the process variations necessitates the identification of new design corners for validation. From the technology scaling trends, for future technologies the aspect ratio of and spacing between wires are increasing such that the capacitance between metal wires on the same layer exceeds the interlayer capacitance. Since there is a high likelihood of having long parallel wires on the same layer, we believe the effects of crosstalk will be more severe. Finally, the results of our analysis provide conditions that must be satisfied to detect errors caused by crosstalk by a sequence of vectors used for validation as well post-manufacturing testing. For example, it shows that a sequence of two patterns must be applied to cause nearly simultaneous transitions in opposing directions to invoke worst case crosstalk slowdown. The resulting slowdown must then be propagated along paths to circuit outputs that have low values of slack. The application of a test sequence that satisfies these conditions will identify devices with excessive crosstalk slowdown. Note that traditional path delay testing tests

for excessive delay along logical paths in the circuit, while here excessive delays are caused by coupling between logically unrelated paths. In a similar manner, the above results provide conditions that a sequence of patterns must satisfy to detect errors caused by crosstalk effects.

Several new techniques have been developed so that our ATPG algorithm can efficiently and accurately generate tests for what is essentially an analog effect, namely crosstalk noise. These techniques include new models for a CMOS inverter, methods to calculate inverter output response for pulse inputs, a method for collapsing CMOS gates into equivalent inverters, and a piece-wise linear model for pulses. These techniques were integrated into a test generation framework that takes into account several new attributes, such as noise strengths and signal arrival times, and identifies test patterns that maximize crosstalk noise at POs while satisfying a given set of Boolean constraints. This algorithm not only considers noise effects such as speedup, slowdown and pulses as new logic values, but also takes into consideration information such as finite noise energy and input arrival skews to accurately characterize noise strength. The ATPG algorithm utilizes conditions that help excite the maximum crosstalk effect and propagate the crosstalk signal to POs under desired timing requirements. In addition, this ATPG algorithm includes the concept of gate delay, signal arrival time, signal strength and rise/fall times. By using the path delay information obtained by circuit preprocessing and/or the analog cost function, preferred paths can be selected during the backtrace as well as propagation processes. Because the proposed algorithm implicitly explores all PI combinations, it is necessary to limit the search space to improve efficiency. A branch-and-bound technique

is proposed to reduce the search effort. Finally, while most ATPG algorithms attempt to only satisfy a set of logical constraints, this algorithm also maximizes an objective function. In short, our ATPG employs mixed-signals with the capability to deal with signal timing and is optimization-oriented. Experimental results show that the method can be efficiently applied to selected crosstalk faults in circuits of reasonable sizes.

The proposed algorithm can also generate all tests for a crosstalk effect so that a matching with functional tests can be performed to determine whether functional tests cover the tests for the crosstalk effects. Thus we only need to employ tests for those crosstalk effects that are not detectable by conventional functional tests. In addition, the test vectors generated can be considered as a complementary set of tests for the purpose of delay testing. In our TG process the transition on the victim line is created and propagated along a possible long path, and receives the largest amount of crosstalk effect under certain timing requirements. From the point of view of both crosstalk effect and signal propagation, the amount of delay imposed on the victim line signal is maximized with respect to the given constraints. Thus the path delays excited by our tests for crosstalk-induced delay are different from those of robust delay tests and can be used to validate signal delays for timing purpose.

The tools and techniques developed can be integrated into the design phase to validate and test for crosstalk. For example, our analytic results can be used to identify pairs of circuit lines where crosstalk may be significant, and thus be dealt with by a circuit designer. One option is to redesign parts of the circuit to drastically decrease the probability of the effect causing an error. Redesign may be very expensive because of its

impact on the product design schedule and/or the inability to meet aggressive design objectives. In such a case, one alternative approach is to ignore the crosstalk effect but generate tests to detect it if it occurs. The resulting tests can be applied to each manufactured chip, and chips in which the effect does not cause any error will pass and will be shipped to customers; chips where it causes an error will be discarded. By providing such an alternative, our results and ATPG system will allow a circuit designer to decide to eliminate crosstalk effects that are likely to cause logic errors via redesign or ignore them until post-manufacturing testing. In this manner, a designer can make decisions on the basis of the economics of each redesign on the one hand and the cost of testing and loss of yield on the other. Such a choice is often made in favor of living with the flaw when there is a time to market issue, and the design change can be handled in a future release. In summary, the main benefit of this work is a greater understanding of the impact of crosstalk effect on high speed circuits, and by being able to detect crosstalk faults, fewer faulty chips will produce a false positive test response and hence the defect level of chips will be enhanced.

References

[1] A. K. Goel, High-speed VLSI interconnections: modeling, analysis, and simulation, John Wiley & Sons Inc., 1994.

[2] A. E. Zain and S. Chowdhury, "An analytical method for finding the maximum crosstalk in lossless-coupled transmission lines", Int'l Conf. on Computed Aided Design, pp. 443-448, 1992.

[3] D. S. Gao, A. T. Yang and S. M. Kang, "Modeling and simulation of interconnection delays and crosstalk in high-speed integrated circuits", IEEE Trans. on Circuits and Systems, Vol. 37, pp. 1-9, January 1990.

[4] S. L. Manney, M. S. Nakhla and Q. Zhang, "Analysis of non-uniform, frequency dependent high-speed interconnects using numerical inversion of Laplace transform", IEEE Trans. on Computer Aided Design of Integrated Circuit and Systems, Vol. 13, pp. 1513-1525, December 1994.

[5] C. Gordon, K. M. Roselle, "Estimating crosstalk in multiconductor transmission lines", IEEE Trans. on Components Packaging and Manufacturing Technology, Vol. 19, pp. 273-277, May 1996.

[6] R. Kaupp, "Waveform degradation in VLSI interconnections", IEEE Journal of Solid-State Circuits, Vol. 24, pp. 1150-1153, August 1989.

[7] H. You and M. Soma, "Crosstalk analysis of interconnect lines and packages in high-speed integrated circuits", IEEE Trans. on Circuits and Systems, Vol. 37, pp. 1019-1026, August 1990.

[8] H. You and M. Soma, "Crosstalk and transient analysis of high-speed interconnects and packages", IEEE Trans. on Solid State Circuits, Vol. 26, pp. 319-330, March 1991.

[9] K. J. Chang, N. H. Chang, S. Y. Oh and K. Lee, "Parameterized SPICE subcircuits for multilevel interconnect modeling and simulation", IEEE Trans. on Circuits and Systems, Vol. 39, pp. 779-789, November 1992.

[10] M. Roca, F. Moll and A. Rubio, "Crosstalk effects between metal and polysilicon lines in CMOS integrated circuits", IEEE Trans. on Electromagnetic Compatibility, Vol. 36, pp. 250-253, August 1994.

[11] A. Rubio, N. Itazaki, X. Xu and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits", IEEE Trans. on Computer

Aided Design of Integrated Circuits and Systems, Vol.13, pp. 387-394, March 1994.

[12] F. Moll and A. Rubio, "Spurious signals in digital CMOS VLSI circuits: a propagation analysis", IEEE Trans. on Circuits and Systems, Vol. 39, pp. 749-752, October 1992.

[13] N. Hedebstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol. 6, pp. 270-281, March 1987.

[14] A. I. Kayssi, K. A. Sakallah and T. M. Burks, "Analytical transient response of CMOS inverters", Trans. Briefs, IEEE Trans. on Circuit and Systems, Vol. 39, pp. 43-45, January 1992.

[15] K. O. Jeppson, "Modeling the influence of the transistor gain ratio and the input-to output coupling capacitance on the CMOS inverter delay", IEEE Journal of Solid State Circuits, Vol. 29, pp. 646-654, June 1994.

[16] N. D. Arora, K. V. Raol, R. Schumann and L.M. Ricardson, "Modeling and extraction of interconnect capacitance for multilayer VLSI circuits", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol. 15, pp. 58-66, January 1996.

[17] J. Qian, S. Pullela and L. Pillage, "Modeling the effective capacitance for the RC interconnect of CMOS gates", IEEE Trans. on Computer Aided Design of Integrated Circuit and Systems, Vol. 13, pp. 1526-1535, December 1994.

[18] R. S. Astava and K. Fitzpatrick, "A simple model for the overlap capacitance of a VLSI MOS device", IEEE Trans. on Electron Devices, Vol. 29, pp. 1870-1880, December 1982.

[19] M. Favalli and C. Metra, "Sensing circuit for on-line detection of delay faults", IEEE Trans. on VLSI Systems, Vol. 4, pp. 130-133, March 1996.

[20] J. J. Tang, K. J. Lee and B. D. Liu, " Built-in intermediate voltage testing for CMOS circuits", Int'l Conf. on Computed Aided Design, pp. 372-376, 1995.

[21] M. A. Breuer and S. K. Gupta, " Process aggravated noise (PAN) : new validation and test problems", Proc. Int'l Test Conf., pp. 914-923, 1996.

[22] S. Natarajan, M. A. Breuer and S. K. Gupta, " Process variations and their impact on circuit operation", IEEE Int's symposium on Defects and Fault Tolerance in VLSI Systems, pp. 73-81, November, 1998.

[23] W. Y. Chen, M. A. Breuer and S. K. Gupta, "Analytic models for crosstalk delay and pulse analysis for non-ideal inputs", Computer Engineering technical report No. 97-12, Electrical Engineering - Systems Department, University of Southern California, July 1997.

[24] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 16, pp. 290-298, March 1997.

[25] T. Sakurai and A. R. Newton, "Alpha-power law MODFET model and its applications to CMOS inverter delay and other formulas", IEEE Journal of Solid-State Circuits, Vol. 25, pp. 584-593, April 1990.

[26] Y. H. Jun, K. Jun and S. B. Park, "An accurate and efficient delay time modeling for MOS logic circuits using polynomial approximation", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 8, pp. 1027-1032, September 1989.

[27] T. Sakurai and A. R. Newton, "Delay analysis of series-connected MOSFET circuits", IEEE Journal of Solid-State Circuits, Vol. 26, pp. 122-130, February 1991.

[28] J. T. Kong and D. Overhauser, "Methods to improve digital MOS macromodel accuracy", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, pp. 868-881, July 1995.

[29] A. Nabavi-Lishi and N. C. Rumin, "Inverter models of CMOS gates for supply current and delay evaluation", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 13, pp. 1271-1279, October 1994.

[30] W. Y. Chen, S. K. Gupta and M. A. Breuer, "Analytic models for crosstalk delay and pulse analysis for non-ideal inputs", Proc. Int'l Test Conf., pp. 809-818, 1997.

[31] M. Abramovici, M. A. Breuer and A. D. Friedman, Digital Systems Testing and Testable Designs, IEEE Press, 1990.

[32] The national technology roadmap for semiconductors, 1997. (See the web page http:// www.sematech.org).

[33] K. T. Lee, C. Nordquist and J. A. Abraham "Automatic test pattern generation for crosstalk glitches in digital circuits", Proc. VLSI Test Symposium, pp. 34-39, 1998.

[34] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms", IEEE Trans. on Computers, Vol. C-32, pp. 1137-1144, December 1983.

[35] K. S. Crump, "Numerical inversion of Laplace transforms using a Fourier series approximation", Journal ACM, Vol. 23, pp. 89-96, January 1976.

[36] F. Moll and A. Rubio, "Methodology of detection of spurious signals in VLSI circuits", Proc. Europe Test Conference, pp. 491-496, 1993.

[37] F. Moll and A. Rubio, "Detectability of spurious signals with limited propagation in combinational circuits", IEEE 3rd Asian Test Symposium, pp. 176-181, November 1994.

[38] L. Goldstein, "Controlability/observability analysis of digital circuits", IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 685-693, September 1979.

[39] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital systems", Proc. Int'l Conf. Computer-Aided Design, pp. 524-531, 1996.

[40] K. L. Shepard, V. Narayanan, P.C. Elmendorf and G. Zheng, "Global harmony: coupled noise analysis for full chip RC interconnect networks", Proc. IEEE Int'l Conf. Computer-Aided Design, pp.139-146, 1997.

[41] K. L. Shepard, "Practical issues of interconnect analysis in deep submicron integrated circuits", Proc. IEEE Int'l Conf. Computer Design, pp. 532-541, 1997.

[42] K. L. Shepard and V. Narayanan, "Conquering noise in deep submicron digital ICs", IEEE Design & Test of Computers, Vol. 15, issue 1, pp. 51-62, Jan-Mar, 1998.

[43] L. M. Silveria, M. Kamon, I. and J. White, "Algorithms for coupled transient simulation of circuits and complicated 3-D packaging", IEEE Trans. on Components Packaging and Manufacturing Technology, part B-Advanced Packaging, vol. 18, issue 1, pp. 92-98, February 1995.

[44] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, pp. 639-649, May 1995.

[45] V. Chandframouli and K. A. Sakallah, " Modeling the effects of temporal proximity of input transitions on gate propagation delay and transition time", Proc. Design Automation Conf. pp. 617-622, June 1996.

[46] W. Y. Chen, S. K. Gupta and M. A. Breuer, "Test generation in VLSI circuits for crosstalk noise", Proc. Int'l Test Conf., pp. 641-650, 1998.

[47] J. Cong, Z. Pan, L. He, C. K. Koh and K. Y. Khoo, "Interconnect design for deep submicron ICs", Int'l Conf. on Compuer-Aided Design, pp. 478-485, 1997.

[48] D. Sylvester, C. M. Hu, O. S. Nakagawa and S.Y. Oh, "Interconnect scaling: signal integrity and performance in future high-speed CMOS designs", Proc. Symposium on VLSI Technology, pp. 42-43, 1998.

[49] R. Ho, K. Mai, H. Kapadia and M. Horowitz, "Interconnect scaling implications for CAD", Proc. Int'l. Conf. on Computer-Aided Design, pp. 425-429, 1999.

[50] S. O. Nakagawa, D. M. Sylvester, J. G. McBride and S. Y. Oh, "On-chip cross talk noise model for deep submicrometer ULSI interconnect", The Hewlett-Packard Journal, pp. 39-45, August 1998.

[51] N. Weste, and K. Eshraghian, Principle of CMOS VLSI Design, Addison-Wesley, 1993.

[52] J. Rabaey, Digital Integrated Circuits, A Design Perspective, Prentice-Hall, 1996.

[53] M. Shoji, CMOS Digital Circuit Technology, Prentice-Hall, 1988.

[54] A. K. Goal and Y. R. Huang, "Modeling of crosstalk among the GaAs VLSI connections", IEE Proc. Part G, Vol. 136, pp. 361-368, 1989.

[55] A. Rubio and R. Anglada, "An approach to crosstalk effect analysis and avoidance techniques in digital CMOS VLSI circuits", Int'l. Journal of Electronics, Vol. 65, No. 1, pp. 3-17, 1988.

[56] S. Voranantakul and J. L. Prince, "Crosstalk analysis for high-speed pulse propagation in lossy electrical interconnections", IEEE Trans. on Components, Hybrids, and Manufacturing Technology, Vol. 16, No. 1, pp. 127-136, February 1993.

[57] Z. Q. Ning, "Capacitance coefficients for VLSI multiple metallization lines", IEEE Trans. on Electron Devices, Vol. ED-34, No. 3, pp. 644-649, March 1987.

[58] N. D. Arora, K.V. Raol and R. Schumann "Modeling and extraction of interconnect capacitance for multilayer VLSI circuits", IEEE Trans. on Computer-Aided Design and Integrated Circuits and Systems, Vol. 15, No. 1, pp. 58-67, January 1996.

[59] N. Delorme, M. Bellevile and J. Chilo, " Inductance and capacitance formulas for VLSI interconnects", Electronic Letters, Vol. 32, No. 11, pp. 996-997, May 1996.

[60] IEEE DASC standard delay format (SDF). (See the web page http://vhdl.org/vi/sdf/)

[61] W. Y. Chen, M. A. Breuer and S. K. Gupta, "Timing analysis for test generation for crosstalk-induced delay in integrated circuits", Computer Engineering technical

report No. 99-04, Electrical Engineering - Systems Department, University of Southern California, April 1999.

[62] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits", IEEE Trans. on Computer, C-30, No. 3, pp. 215-222, 1981.

[63] W. Y. Chen, S. K. Gupta and M. A. Breuer, "Test generation for crosstalk-induced delay in integrated circuits", Proc. Int'l Test Conf., pp. 191-200, 1999.

[64] A. Sinha, S. K. Gupta and M. A. Breuer, "Validation and test generation for oscillatory noise in VLSI interconnects", Proc. Int'l. Conf. on Computer-Aided Design, pp. 289-296, 1999.

[65] L. C. Chen, S. K. Gupta and M. A. Breuer, "A new framework for static timing analysis, incremental timing refinement, and timing simulation", submitted to Int'l. Conf. on Computer-Aided Design, 2000.

Appendix A

Approximation for Crosstalk Pulse Amplitude

In Chapter 2 we derived a expression for a crosstalk pulse, namely

$$V(t) = \frac{C_m}{R_{p1}C_t}\left[\frac{1/x}{(w+1/x)(w-u)}e^{wt} + \frac{1/x}{(u+1/x)(u-w)}e^{ut} + \frac{1/x}{(w+1/x)(u+1/x)}e^{-t/x}\right].$$

To find the maximum amplitude one can differentiate the above equation and set the result to zero. Differentiate the about equation we get

$$V'(t) = \frac{C_m}{R_{p1}C_t}\left[\frac{1/x}{(w+1/x)(w-u)}we^{wt} + \frac{1/x}{(u+1/x)(u-w)}ue^{ut} - \frac{1/x^2}{(w+1/x)(u+1/x)}e^{-t/x}\right].$$

Since this expression contains three exponential terms, it is very difficult to find a closed-form expression for the amplitude. Hence we expand the exponential terms by using the Taylor series expansion. The most important process in the Taylor series expansion is finding the expansion center, t_0, where the approximation error is minimal. Since we know that the time when the maximum amplitude of the pulse at V occurs will not be earlier than when the step input is applied, and is near the time when the affecting line finishes its transition, one can empirically derive the following expression for the expansion center,

$$t_0 = \xi \cdot t_{step}(1 - e^{-x/t_{step}}) + t_{step},$$

where ξ is an empirical fitting constant (typically it is 1.2), and $t_{step} = \ln(u/w)/(w-u)$ is the time when the maximum amplitude occurs for the case of a step input.

As the time constant x decreases to 0, t_0 monotonically decreases to t_{step}. As x increases, the expansion center t_0 increases. The useful range for this approximation for time constants is from 0 to 250 ps, which includes the range of rise/fall times in todays technologies.

By expanding the expression for the crosstalk pulse into a Taylor series, we get

$$\begin{aligned} V'_{apx}(t) = \frac{C_m}{R_{p1}C_t}\Big[& \frac{1/x}{(w+1/x)(w-u)} w\left(e^{w\cdot t_0} + we^{w\cdot t_0}(t-t_0) + \frac{w^2 e^{w\cdot t_0}}{2}(t-t_0)^2\right) \\ & + \frac{1/x}{(u+1/x)(u-w)} u\left(e^{u\cdot t_0} + ue^{u\cdot t_0}(t-t_0) + \frac{u^2 e^{u\cdot t_0}}{2}(t-t_0)^2\right) \\ & - \frac{1/x^2}{(w+1/x)(u+1/x)}\left(e^{-t_0/x} + \frac{-1}{x}e^{-t_0/x}(t-t_0) + \frac{\frac{1}{x^2}e^{-t_0/x}}{2}(t-t_0)^2\right)\Big]. \end{aligned}$$

This expression becomes a polynomial equation and can be solved directly to find the time (t_x) when the maximum amplitude occurs, namely

$$t_x = \frac{-Awe^{w\cdot t_0} + Aw^2e^{w\cdot t_0}t_0 - Bue^{u\cdot t_0} + Bu^2e^{u\cdot t_0}t_0 - \frac{C}{x}e^{-t_0/x} - \frac{C}{x^2}e^{-t_0/x}t_0}{\left(Aw^2e^{w\cdot t_0} + Bu^2e^{u\cdot t_0} - \frac{C}{x^2}e^{-t_0/x}\right)} - \frac{\sqrt{F-G-H}}{x},$$

where

$$A = \frac{1/x}{(w+1/x)(w-u)}w, \quad B = \frac{1/x}{(u+1/x)(u-w)}u, \quad C = \frac{1/x^2}{(w+1/x)(u+1/x)},$$

$$F = -C^2 e^{t^2_0/x^2} - A^2w^2e^{w^2\cdot t^2_0}x^2 + 2Awe^{w\cdot t_0}x^2Bue^{u\cdot t_0} + 2Awe^{w\cdot t_0}xCe^{-t_0/x},$$

$$G = B^2u^2e^{u^2 \cdot t^2_0}x^2 + 2Bue^{u \cdot t_0}xCe^{-t_0/x} - 2Aw^2e^{w \cdot t_0}x^2Be^{u \cdot t_0} + 2Aw^2e^{w \cdot t_0}x^2Ce^{-t_0/x},$$

and

$$H = 2Bu^2e^{u \cdot t_0}x^2Ae^{w \cdot t_0} + 2Bu^2e^{u \cdot t_0}x^2Ce^{-t_0/x} + 2Ce^{-t_0/x}Ae^{w \cdot t_0} + 2Ce^{-t_0/x}Be^{u \cdot t_0}.$$

Then the maximum crosstalk amplitude is obtained by substituting t_x back into the crosstalk pulse equation

$$V_{pmax} = \frac{C_m}{R_{p1}C_t}\left[\frac{1/x}{(w+1/x)(w-u)}e^{wt_x} + \frac{1/x}{(u+1/x)(u-w)}e^{ut_x} + \frac{1/x}{(w+1/x)(u+1/x)}e^{-t_x/x}\right].$$

To find the energy enclosed in the pulse, one can integrate the expression for the pulse waveform from time 0 to infinite and obtain the following expression

$$\begin{aligned}
\text{Energy} &= \frac{C_m}{R_{p1}C_t}\int_0^\infty\left[\frac{1/x}{(w+1/x)(w-u)}e^{wt} + \frac{1/x}{(u+1/x)(u-w)}e^{ut} + \frac{1/x}{(w+1/x)(u+1/x)}e^{-t/x}\right]dt \\
&= \frac{-C_m}{R_{p1}C_t}\left[\frac{1/x}{w(w+1/x)(w-u)} + \frac{1/x}{u(u+1/x)(u-w)} - \frac{1}{(w+1/x)(u+1/x)}\right] \\
&= \frac{C_m}{wuR_{p1}C_t}.
\end{aligned}$$

From this expression we can see that based on the approximations made, the energy contained in the pulse is independent of x.

Appendix B

Derivation for Crosstalk Delay Expression

From Figure 2.8, to solve for the voltage on node A, we first consider the low pass circuit composed of the pulling resistance R_{p1} and the equivalent impedance Z_{eq1}. We get

$$A(s) = \frac{\frac{1}{R_{p1}Z_{eq1}}}{s + \frac{1}{R_{p1}Z_{eq1}}}, \text{ where}$$

$$Z_{eq1} = \frac{1}{sA(s)-1}[sA(s)(C_m + C_a) - sC_m V(s) - C_v - C_a].$$

Since the input signals applied to the circuit are exponential waveforms, from the discussion in section 2.4.2.1, we have the transfer function

$$H_{Aexp} = H_{A/A_{in}} \cdot H_{A_{in}} = \frac{\frac{1}{R_{p1}Z_{eq1}}}{s + \frac{1}{R_{p1}Z_{eq1}}}(-\frac{1}{s} + \frac{1}{s+1/x}) = [(\frac{1}{s + \frac{1}{R_{p1}Z_{eq1}}})(\frac{1/x}{s+1/x}) + \frac{1}{s+1/x}] - \frac{1}{s},$$

where the term $-1/s$ indicates that before time 0 the voltage on node A is stable '1'.

However, since the input at A is skewed in time by z units with respect to V, the corresponding Laplace transformation for A becomes

$$A_{sd}(s)=[(\frac{1}{s+\frac{1}{R_{p1}Z_{eq1}}})(\frac{1/x}{s+1/x})+\frac{1}{s+1/x}]e^{-sz}-\frac{e^{-sz}}{s}+\frac{1}{s},$$

where the terms ($-e^{-sz}/s+1/s$) accounts for the boundary condition that the initial values on node A is '1' between time 0 and z.

Similarly, to obtain the expression for node V we solve the low pass network consisting of R_{p2} and Z_{eq2} and get

$$V(s)=\frac{\frac{1}{R_{p2}Z_{eq2}}}{s+\frac{1}{R_{p2}Z_{eq2}}}$$, where

$$Z_{eq2}=C_m+C_v-C_m\frac{A(s)}{V(s)}+\frac{C_m}{sV(s)}.$$

Again applying the input transformation as in section 2.4.2.1, and we get

$$V_{sd}(s)=\frac{\frac{1}{R_{p2}Z_{eq2}}}{s+\frac{1}{R_{p2}Z_{eq2}}}(\frac{1}{s}-\frac{1}{s+1/y})=(\frac{1}{s})(\frac{\frac{1}{R_{p2}Z_{eq2}}}{s+\frac{1}{R_{p2}Z_{eq2}}})(\frac{1/y}{s+1/y}).$$

By solving the above system of equations, we obtain

$$A_{sd}(s)=[A_{step}(s)(\frac{1/x}{s+1/x})+\frac{1}{s+1/x}]e^{-sz}+(\frac{C_m}{R_{p2}C_t}\frac{1}{(s-w)(s-u)})(\frac{1/y}{s+1/y})+[\frac{1}{s}+\frac{e^{-sz}}{s}],$$ and

$$V_{sd}(s)=V_{step}(s)(\frac{1/y}{s+1/y})-(\frac{C_m}{R_{p1}C_t}\frac{1}{(s-w)(s-u)})(\frac{1/x}{s+1/x})e^{-sz},$$ where

$$A_{step}(s)=\frac{1}{(s-w)(s-u)}(s+\frac{C_m+C_a}{R_{p2}C_t}),$$ and

$$V_{step}(s) = \frac{1}{s} - \frac{1}{(s-w)(s-u)}(s + \frac{C_m + C_v}{R_{p1}C_t}).$$

Appendix C

Approximation of Skew for Maximum Crosstalk Delay

Crosstalk can significantly affect the delay of a logic gate. If both affecting and victim lines have signal transitions, an increase or decrease of the gate delay may be observed. By using the analytical expression derived in section 2.4.2.2, we found that if the drivers of the affecting and victim lines have the same size, the maximum delay is obtained if simultaneously transitions occur. But if the derivers are unbalanced, then there exists a non-zero skew between the transitions on the affecting and victim lines for which the maximum delay occurs, as shown in Figure A. 1. Here the affecting line driver increases by 2, 4, and 10 times of its original size. The original balanced driver size is 16u/0.35u PMOS and 8u/0.35u NMOS, C_m =120fF, C_v = 150fF, and C_a is not constant value because the capacitance increases with the driver size of the affecting line.

To determine the value of skew that maximizes crosstalk delay, one can solve the analytical equations derived in section 2.4.2.2. However, since the expressions contain several exponential terms, and some of them also have a shift in time, it is very difficult to find a close form expression. Therefore we try to approximate the skew by curve fitting. From Figure A. 1 we can observe that the value of the maximum delay tends to saturate as the driver ratio (affecting to victim line driver size) increases. Hence the skew associated with the maximum slowdown is approximated by an exponential function

$z_{max}=(1-e^{-(r-1)})\cdot t_{peak}\cdot k_1$, where r is the ratio of the drivers, k_1 is a empirical constant, and t_{peak} is the time when the pulse at V due to A is maximum for the balanced driver case. z_{max} is approximately equal to $t_{peak} \cdot k_1$ for large driver ratios, and is zero if the drivers are the same size. In this example, t_{peak} is approximately 70ps and $k_1 = 0.606$.

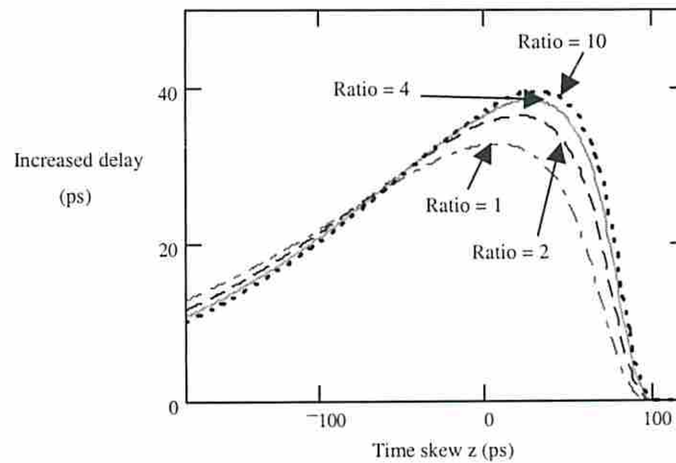


Figure A. 1 Increased delay vs. driver ratio.