

COPING WITH DELAYS AND HAZARDS IN BUSES AND RANDOM LOGIC  
IN DEEP SUB-MICRON

by

Michael N Skoufis

Master of Science in Electrical Engineering, Southern Illinois University, 2002

A Dissertation  
Submitted in Partial Fulfillment of the Requirements for the  
Doctor of Philosophy Degree

Department of Electrical and Computer Engineering  
in the Graduate School  
Southern Illinois University Carbondale  
August, 2009

## **DISSERTATION APPROVAL**

COPING WITH DELAYS AND HAZARDS IN BUSES AND RANDOM LOGIC  
IN DEEP SUB-MICRON

By

Michael N Skoufis

A Dissertation Submitted in Partial  
Fulfillment of the Requirements  
for the Degree of  
Doctor of Philosophy  
in the field of Electrical and Computer Engineering

Approved by:

Dr. Spyros Tragoudas, Chair

Dr. Themistoklis Haniotakis

Dr. Mohammad Sayeh, Dr. Haibo Wang

Dr. Wei Zhang, Dr. Jianhong Xu

Graduate School  
Southern Illinois University Carbondale  
June 18th 2009

## AN ABSTRACT OF THE DISSERTATION OF

Michael N Skoufis, for the Doctor of Philosophy degree in Electrical and Computer Engineering, presented on June 18th, at Southern Illinois University Carbondale.

TITLE: COPING WITH DELAYS AND HAZARDS IN BUSES AND RANDOM LOGIC IN DEEP SUB-MICRON

MAJOR PROFESSOR: Dr. S. Tragoudas

A new data capturing technique for a potentially coupled bus of lines is proposed that always accommodates fast operation. The proposed method utilizes multiple reference voltages available within a line's receiving logic and the initial conditions of the involved wires in order to determine early and accurately the transmitted data in the current cycle. The presented data reading technique rarely requires repeater insertion and it can considerably accelerate signal propagation. The introduced logic at the receiver-end of a victim wire entails an affordable area overhead. Experimental results are given in the  $65nm$  CMOS process for interconnects of various lengths.

An architecture is proposed that allows for data reading with fault detection capability on lines which are likely to operate under a potentially wide range of capacitive coupling. In order to develop such a methodology, multiple reference or threshold voltages in the receiving logic of the lines are considered instead of typically one. The proposed technique utilizes the additional reference voltages to evaluate whether an intermittent fault has occurred during the capture of the transmitted data. Some combinational logic is introduced on the receiver side to accomplish this task. The mechanism is initially illustrated on a line with one adjacent aggressor. Subsequently, the case of a line with two adjacent aggressors is

discussed and it is shown how to generalize the technique for wide buses. In this work the efficiency of the detection mechanism is evaluated for both single and multiple faulty occurrences.

A novel circuit to treat crosstalk induced glitches on local interconnects is presented. Design irregularities and manufacturing defects on wires may result in spurious electrical events that impact the reliability of the interconnect infrastructure. The proposed methods act by dynamically adjusting the threshold voltage of the receiving gate on the victim line. The proposed technique can be used in combination with encoding algorithms on data buses. A comparative study in the  $180nm$  CMOS process is presented that supports the applicability of the approach.

Transient faults due to radiation have become increasingly observable in combinational logic. This is due to the weakening of inherent protective mechanisms that logic traditionally held against such flawed spurious events. Further boosting of such effects is increasingly probable due to the interaction of transients appearing at the inputs of logic gates. Such multiple instances of transients can arise either because of re-convergent circuit paths or because of significant reduction in the critical charge of modern technologies. The latter, in particular, makes more than one circuit nodes susceptible to the same high energy ions. A static transient propagation is employed to address possible transient interaction and to compute its worst-case effects in logic. The quantified effects of interest are the maximum duration and slope of the resulting hazards at the circuit outputs. A hardening methodology is also proposed to protect combinational logic from such events. For this purpose, filtering circuits are inserted in logic and several placement algorithms are developed and evaluated.

## **DEDICATION**

This work is dedicated to my parents Nikolaos and Georgia Skoufis, and to my wife Carol Elizabeth for their constant and unconditional love, support and devotion.

## ACKNOWLEDGMENTS

Throughout the duration of my studies, I had substantial support from several individuals that helped me overcome various obstacles and difficulties, both academic and personal. First of all, I would like to extend my appreciation and gratitude to my adviser Dr. Spyros Tragoudas for his continuous guidance and mentoring throughout the last four years. I know that not only have I developed research and managerial skills with the help of Dr. Tragoudas, but I can confidently say that I have also gained a great friend and collaborator. In all fairness, I would also like to bestow the same acknowledgment and recognition to two of my instructors and collaborators: Dr. Themistoklis Haniotakis (my Master's Thesis adviser) and Dr. Haibo Wang.

Furthermore, I am grateful to Dr. Wei Zhang, Dr. Mohammad Sayeh and Dr. Jianhong Xu for serving in my PhD Dissertation Committee and for their constructive feedback and comments. I also feel the obligation to express my sincere thankfulness to Howard Wilson from the INTEL Microprocessor Labs in Portland, OR for providing the industry's perspective on my research and for helping me stay on a productive path of academic curiosity.

Last but not least, I would like to acknowledge the unconditional love and support of a person who has turned my life around, my lovely wife Carol Elizabeth. I am always grateful for her devotion and hard work, especially in those cases that I could not find the time to contribute as much as I wanted. Carol Elizabeth is my rock and I am hers.

## TABLE OF CONTENTS

Abstract . . . . .	i
Dedication . . . . .	iii
Acknowledgments . . . . .	iv
List of Tables . . . . .	vii
List of Figures . . . . .	ix
1 Introduction . . . . .	1
2 Fast Data Capture on Data Lines Using Multi-Threshold Receiving Logic	5
2.1 Background research on coupled data buses . . . . .	7
2.2 Description of the methodology . . . . .	9
2.2.1 Fundamentals of the method . . . . .	9
2.2.2 Coupled interconnect with one adjacent aggressor . . . . .	10
2.2.3 Coupled interconnect with two adjacent aggressors . . . . .	16
2.2.4 Circuit implementation . . . . .	18
2.2.5 Resilience and variability of the method . . . . .	22
2.3 Experimental evaluation on generalized data buses . . . . .	26
2.4 Conclusions . . . . .	32
3 Error Detection on Buses with Multi-Threshold Receiving Logic . . . . .	42
3.1 Overview of a multi-threshold receiver method . . . . .	44
3.1.1 The fundamentals of the technique . . . . .	44
3.2 On-line detection of errors for lines with one adjacent aggressor . . .	53
3.2.1 Overview . . . . .	53
3.2.2 Error-free characterization of lines . . . . .	55
3.2.3 Correlation of coupled lines . . . . .	57
3.2.4 On-line detection of single error occurrences . . . . .	58
3.2.5 On-line detection of multiple error occurrences . . . . .	60

3.3	On-line error detection for a line with two adjacent aggressors . . . . .	61
3.3.1	Preliminaries . . . . .	61
3.3.2	On-line detection of single error occurrences . . . . .	62
3.3.3	On-line detection of multiple error occurrences . . . . .	63
3.4	Experimental results and conclusions . . . . .	64
4	A Dynamically Adaptive Circuit for Hazard Tolerance on Data Buses . .	75
4.1	The proposed adaptive circuit . . . . .	76
4.2	Optimization of repeater-based configurations . . . . .	78
4.3	Experimental evaluation . . . . .	79
4.4	Conclusions . . . . .	83
5	Single Transient Effects in Combinational Logic and a Hardening Methodology . . . . .	90
5.1	Background research in single event transient effects . . . . .	93
5.2	Transient propagation in combinational logic . . . . .	99
5.2.1	Static transient propagation . . . . .	99
5.2.2	Electrical characterization for static analysis . . . . .	102
5.2.3	Logical masking . . . . .	106
5.2.4	Experimental evaluation . . . . .	106
5.3	A logic hardening technique using C-elements . . . . .	108
5.3.1	C-element overview . . . . .	108
5.3.2	C-element insertion heuristics . . . . .	111
5.3.3	Experimental evaluation . . . . .	111
5.4	Conclusions . . . . .	117
	References . . . . .	122
	Appendix . . . . .	135
	Vita . . . . .	142

## LIST OF TABLES

2.1 Typical single-threshold mapping table . . . . .	12
2.2 Two-threshold mapping table for a victim line with one adjacent aggressor (IBM 65nm) . . . . .	14
2.3 Three-threshold mapping table for a victim line with one adjacent aggressor (IBM 65nm) . . . . .	15
2.4 Two-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm) . . . . .	19
2.5 Three-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm) . . . . .	22
2.6 Four-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm) . . . . .	23
2.7 Five-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm) . . . . .	24
2.8 Clock period estimation (three-threshold logic) for a line with one adjacent aggressor (IBM 65 nm) . . . . .	28
2.9 Capacitance extraction for a two-bit neighborhood . . . . .	36
2.10 Capacitance extraction for a two-bit neighborhood . . . . .	37
3.1 Single-threshold characterization table for the victim with respect to the activity on the aggressor . . . . .	45
3.2 Two-threshold characterization table for the victim with respect to the activity on the aggressor . . . . .	48
3.3 Three-threshold characterization table for the victim with respect to the activity on the aggressor . . . . .	49
3.4 Four-threshold characterization table for a victim (with one adjacent aggressor) with respect to the activity on its only aggressor . . . . .	50

3.5	Four-threshold characterization table for the victim with respect to the activity on the two aggressors . . . . .	51
3.6	Five-threshold characterization table for the victim with respect to the activity on the two aggressors . . . . .	52
3.7	Error-free voltage ranges for a line with one adjacent aggressor using three-threshold receiving logic . . . . .	65
3.8	Error detection when clock delay and additional threshold voltages are introduced in a topology of lines with only one adjacent aggressor . . .	70
3.9	Error detection when clock delay is introduced in a topology of a line with two adjacent aggressors (using four-thresholds) . . . . .	70
3.10	Error detection when clock delay is introduced in a topology of a line with two adjacent aggressors (using five-thresholds) . . . . .	72
5.1	Electrical parameters of the injected transients used in the experimental setup ( $V_{dd} = 1\text{ V}$ ) (IBM 65nm) . . . . .	107
5.2	Electrical parameters of the injected transients used in the experimental setup ( $V_{dd} = 1.8\text{ V}$ ) (TSMC 180nm) . . . . .	107
5.3	Worst-case transient duration at primary outputs measured by static, dynamic sensitized and dynamic un-sensitized analyses with the custom generated standard cells (IBM 65nm) . . . . .	108
5.4	Worst-case transient duration at primary outputs measured by static, dynamic sensitized and dynamic un-sensitized analyses with the custom generated standard cells (TSMC 180nm) . . . . .	109
5.5	Delay overhead for a minimum-size C-element circuit . . . . .	110
5.6	Filter insertion with the several investigated heuristics (IBM 65nm) . .	118
5.7	Filter insertion with the several investigated heuristics (TSMC 180nm)	121
5.8	Five-threshold characterization table for a victim (with one adjacent aggressor) with respect to the activity on its only aggressor . . . . .	136

## LIST OF FIGURES

1.1	Parasitic coupling between lines . . . . .	1
1.2	Example of a data bus between two cores . . . . .	2
1.3	Example of a hazard or transient . . . . .	3
1.4	Propagation of a hazard in combinational logic . . . . .	4
2.1	Victim line with one adjacent aggressor . . . . .	10
2.2	Glitches for stable/quiet transitions on a victim line with one adjacent aggressor (IBM 65nm) . . . . .	11
2.3	Falling transitions on a victim line with one adjacent aggressor (IBM 65nm) . . . . .	12
2.4	Rising transitions on a victim line with one adjacent aggressor (IBM 65nm) . . . . .	13
2.5	All transitions on a victim line with one adjacent aggressor (IBM 65nm)	14
2.6	Study on delay optimal number of receiver thresholds for a two-bit 800 $\mu m$ long bus(IBM 65nm) . . . . .	16
2.7	Victim line with one adjacent and one distant aggressor . . . . .	16
2.8	All transitions for a victim line with one adjacent and one distant aggressor (IBM 65nm) . . . . .	17
2.9	Victim line with two adjacent aggressors . . . . .	18
2.10	Falling transitions for a victim line with two adjacent aggressors (IBM 65nm) . . . . .	18
2.11	Rising transitions for a victim line with two adjacent aggressors (IBM 65nm) . . . . .	19
2.12	Hazards (only) transitions for a victim with two adjacent aggressors (IBM 65nm) . . . . .	20
2.13	All transitions for a victim line with two adjacent aggressors (IBM 65nm)	21

2.14 Study on delay optimal number of receiver thresholds for a three-bit 800 $\mu m$ long bus (IBM 65nm) . . . . .	25
2.15 Victim line with two adjacent and two distant aggressors . . . . .	25
2.16 All transitions for a victim line with two adjacent and two distant aggressors (IBM 65nm) . . . . .	26
2.17 Proposed pipeline for data interconnects . . . . .	26
2.18 Data propagation, read and reset (in next clock cycle) tasks . . . . .	27
2.19 Clock periods for proposed (three-threshold logic) and conventional methods in a two-bit bus with Coupling = 10 Csub (IBM 65nm) . . . . .	29
2.20 Clock periods for proposed (three-threshold logic) and conventional methods in a two-bit bus with Coupling = 10 Csub (IBM 65nm) . . . . .	30
2.21 Clock periods for proposed (five-threshold logic) and conventional methods in a three-bit bus with Coupling = 10 Csub (IBM 65nm) . . . . .	31
2.22 Clock periods for proposed (five-threshold logic) and conventional methods in a five-bit bus with Coupling = 10 Csub (IBM 65nm) . . . . .	32
2.23 Increased noise margin for a victim wire in a two-bit bus (IBM 65nm) .	33
2.24 Wire delay change with increasing noise margin for two-bit buses using three-threshold logic (IBM 65nm) . . . . .	33
2.25 Wire delay variance for slow and fast transitions (IBM 65nm) . . . . .	34
2.26 Standard deviation of the wire delay for typical and proposed two-bit buses using three-threshold logic with Coupling = 10 Csub (IBM 65nm)	34
2.27 Wire delay change with noise margin for three-bit buses with five-threshold logic (IBM 65nm) . . . . .	35
2.28 Wire delay change with noise margin for five-bit buses with five-threshold logic (IBM 65nm) . . . . .	35
2.29 A two-bit bus partitioning (top) of a generic bus (bottom) . . . . .	36

2.30 Clock period for a 32-bit bus of (a) $250\mu m$ and (b) $500\mu m$ length with the proposed two-bit isolated neighborhoods (three-threshold logic) using $d = a$ (IBM 65nm) . . . . .	37
2.31 Clock period for 32-bit bus of (a) $250\mu m$ , (b) $500\mu m$ and (c) $750\mu m$ length with the proposed two-bit isolated neighborhoods (three-threshold logic) using $d = 3a$ (IBM 65nm) . . . . .	38
2.32 Delay change with Vdd scaling for a 32-bit $750\mu m$ long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using $D = 3\lambda$ and $d = 3\lambda = 3a$ (IBM 65nm) . . . . .	39
2.33 Delay variance for a 32-bit $250\mu m$ long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using $D = 3\lambda$ and $d = 3\lambda = 3a$ (IBM 65nm) . . . . .	39
2.34 Delay variance for a 32-bit $500\mu m$ long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using $D = 3\lambda$ and $d = 3\lambda = 3a$ (IBM 65nm) . . . . .	40
2.35 Delay variance for a 32-bit $750\mu m$ long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using $D = 3\lambda$ and $d = 3\lambda = 3a$ (IBM 65nm) . . . . .	40
2.36 Effects of coupling intensity and wire resistance on delay deviation for a 32-bit $500\mu m$ long bus ( $d = 3a$ ) with the proposed two-bit isolated neighborhoods (three-threshold logic) without any repeaters for the typical method (IBM 65nm) . . . . .	41
3.1 Falling (only) transitions for a victim with one adjacent aggressor (IBM 65nm) . . . . .	45
3.2 Rising (only) transitions for a victim with one adjacent aggressor (IBM 65nm) . . . . .	46

3.3	Voltage glitches (hazards) only for quiet transitions ( $0 \rightarrow 0$ or $1 \rightarrow 1$ ) for a line with one adjacent aggressor (IBM 65nm) . . . . .	46
3.4	All (superimposed) transitions for a victim wire with one adjacent aggressor (IBM 65nm) . . . . .	47
3.5	Falling (only) transitions for a victim with two adjacent aggressors (IBM 65nm) . . . . .	53
3.6	Rising (only) transitions for a victim with two adjacent aggressors (IBM 65nm) . . . . .	53
3.7	Voltage glitches (hazards) only for a victim with two adjacent aggressors (IBM 65nm) . . . . .	54
3.8	All (superimposed) transitions for a victim with two adjacent aggressors (IBM 65nm) . . . . .	54
3.9	Example of bus line with one adjacent aggressor . . . . .	55
3.10	Error-free characterization of ranges for all possible initial conditions of a line with one adjacent aggressor . . . . .	66
3.11	Correlated sub-ranges within the error-free ranges for all possible initial conditions on a line with one adjacent aggressor . . . . .	67
3.12	Likely errors detected in the topology of a line with one adjacent aggressor	68
3.13	Likely errors detected in the topology of a line with one adjacent aggressor	68
3.14	Likely errors detected in the topology of a line with one adjacent aggressor	68
3.15	Likely errors detected in the topology of a line with one adjacent aggressor	68
3.16	Correlated sub-ranges for a delayed three-threshold logic in a topology of lines with one adjacent aggressor . . . . .	69
3.17	Correlated sub-ranges for a four-threshold logic in a topology of lines with one adjacent aggressor . . . . .	69
3.18	Example of bus line with two adjacent aggressors . . . . .	69

3.19	Error-free characterization of ranges for a line with two adjacent aggressors using four-thresholds . . . . .	71
3.20	Detection of adjacent errors by introducing clock delay for a line with two adjacent aggressors (using four and five thresholds) . . . . .	73
3.21	Delay tradeoff for increased error detection capabilities on a three-bit bus with a five threshold receiver logic to detect any ( <i>i.e.</i> , three) simultaneous adjacent-range errors . . . . .	74
4.1	The proposed adaptive circuit . . . . .	77
4.2	Generation of the control signals for the adaptive circuit . . . . .	78
4.3	Illustrating the synchronization of the control signals in the adaptive circuit . . . . .	79
4.4	DC Analysis for the proposed hazard removal circuit . . . . .	80
4.5	Experimental optimization of typical repeater-based configurations . . . . .	81
4.6	Transmitted signal arrangement for evaluating hazard removal efficiency and potential delay degradation for legitimate signals . . . . .	82
4.7	Hazard removal and delay characteristics for the proposed circuit . . . . .	83
4.8	Transmitted signal for Configuration (I) . . . . .	84
4.9	Delay and dissipated power for Configuration (I) . . . . .	84
4.10	Transmitted signal for Configuration (II) . . . . .	85
4.11	Delay and dissipated power for Configuration (II) . . . . .	85
4.12	Transmitted signal for Configuration (III) . . . . .	86
4.13	Delay and dissipated power for Configuration (III) . . . . .	86
4.14	Transmitted signal for Configuration (IV) . . . . .	87
4.15	Delay and dissipated power for Configuration (IV) . . . . .	87
4.16	Transmitted signal for Configuration (V) . . . . .	88
4.17	Delay and dissipated power for Configuration (V) . . . . .	88
4.18	Power sweep for different levels of crosstalk reduction . . . . .	89

4.19	Delay sweep for different levels of crosstalk reduction . . . . .	89
5.1	Resulting output transient for a gate when non-controlling value transients appear at its inputs . . . . .	100
5.2	Resulting output transient for a gate when controlling value transients appear at its inputs . . . . .	101
5.3	Static propagation of transients . . . . .	102
5.4	Approximation of a five inverter chain with standard cells . . . . .	104
5.5	Input transient mapping to a slightly worse case of electrical parameters in the cell library . . . . .	106
5.6	Min-max output transient characterization for different arrival times of the input transients . . . . .	119
5.7	The basic C-element circuit . . . . .	120
5.8	Illustration of C-element placement for protection . . . . .	120
5.9	Error-free characterization of ranges in selected cases for a line with two adjacent aggressors . . . . .	137
5.10	Error-free characterization of ranges in selected cases for a line with two adjacent aggressors . . . . .	138
5.11	Error-free characterization of ranges in selected cases for a line with two adjacent aggressors . . . . .	139
5.12	Output transient plot given by both HSPICE and custom standard cells for a primary output line in the c17 benchmark circuit. Transient stimulus is applied at a primary input. . . . .	140
5.13	Output transient plot for a dynamic simulation given by both HSPICE and the custom cells for an internal line in a benchmark circuit ( <i>s9234</i> ). Random transients throughout the circuit are injected. . . . .	140



# CHAPTER 1

## INTRODUCTION

The continuous scaling of semiconductor processes into the sub-micron area has created the conditions for a whole new set of challenges in integrated circuit design. On one hand, Moore's law (outlining the anticipated technology shrinkage) has allowed the integration of an enormous amount of new functionalities on chips, making modern designs a lot more complex, dense and demanding than in previous generations. On the other hand, the electrical properties of the different components associated and directly impacted by their ever decreasing geometric dimensions have enhanced the uncertainty and unpredictability of contemporary circuits, making them thus more difficult to control and protect than in the past.

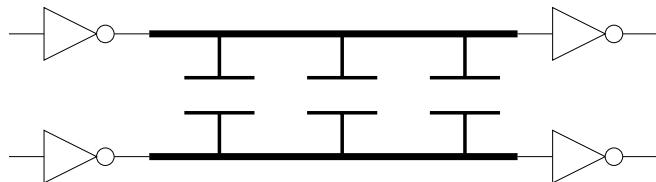


Figure 1.1. Parasitic coupling between lines

The increased component proximity contributes to the existing interference noise that is responsible for slow downs and circuit failures. Specifically, wire thickness in modern technologies is often larger than its corresponding width, whereas the wire spacing may be smaller than the separation distance between metal layers or between a layer and the substrate [4]. As a consequence, this worsens the coupling between neighboring data wires for the same metal layer (Figure 1.1). This is further exacerbated by the increasing wire resistance resulting from narrow line widths.

Furthermore, due to the complexity of designs, data lines do not scale down

as semiconductor devices do. In fact, global wire lengths increase or at least stay constant, dominating overall data transmission delay [1, 5]. High switching frequencies bring also wire inductance into the picture along with the corresponding transmission line effects for longer lines [1, 3]. Temperature increases (which lead to lower electron mobility) and lower voltage power supplies further undermine rapid and reliable signal propagation. All of the above are significant challenges considering that today's high microprocessor speeds and fast processing cores (Figure 1.2) require efficient data transmission more than ever before.

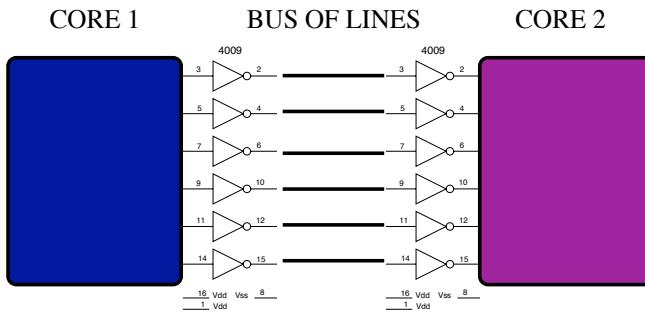


Figure 1.2. Example of a data bus between two cores

Existing research for alleviating such as the previous phenomena involves data encoding arrangements [19, 20, 21, 63, 64], repeater or buffer insertion along lines [10, 25, 26, 27, 28, 33, 50], interconnect tuning strategies [11] and simultaneous redundant switching [5], among others. The most popular approach by far is the placement of repeater inverters due to its capability to relax the wire delay from a quadratic to a quasi-linear function of its respective length. However, this approach is certainly not panacea. Increasing leakage currents, high dynamic power dissipation, required placement optimizations, frequent over the cell routing of data lines and other concerns make this approach more of a bottleneck in deep sub-micron circuits. Alternative arrangements should therefore be investigated which would not necessarily utilize repeater gates along data wires for an efficient

data transmission.

Besides data buses, combinational logic appears to be increasingly more vulnerable in new, reduced size technologies. Even though, the smaller device feature sizes reduce the overall parasitics and speed up logic operation considerably, transistors become more susceptible to radiation (cosmic particles) that produce single event transient effects in circuits [73] (Figures 1.3 and 1.4). Such events lead to functional errors and must be avoided in mission critical applications. The trend is expected to continue and get worse in the next process generations [110, 82]. Significant work in the field has been done for both estimating and alleviating the effects of these hazardous transients [107, 102, 103, 101]. However, the potential interaction of these events is not taken into consideration in existing literature and is rather underestimated. Such interaction can lead to significant boosting of hazards and it should be properly addressed and studied.

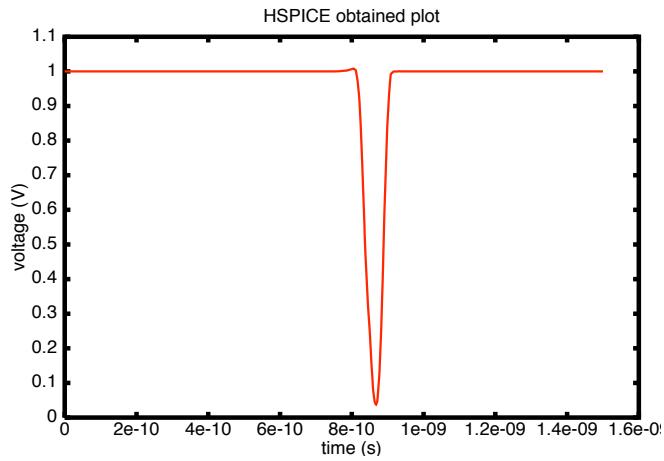


Figure 1.3. Example of a hazard or transient

In this proposed research, methods are investigated to cope with the aforementioned challenges. First in Chapter 2, a novel bus architecture is proposed

c17 benchmark circuit

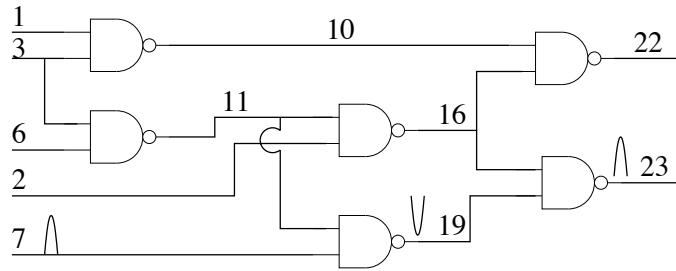


Figure 1.4. Propagation of a hazard in combinational logic

that allows for accelerating data transmission along bus wires without the need to resort to repeater inverters for a respectable length of lines. Furthermore in Chapter 3, an evaluation of the method is performed for providing data buses with error detection capabilities. Subsequently, a dynamically adaptive circuit for noise tolerance on data buses is presented and evaluated in Chapter 4. Detailed experimental results are provided that advocate and promote its applicability. Lastly in Chapter 5, a framework is proposed for computing worst-case transient effects in combinational logic incorporating transient interaction in the analysis. This framework is used to facilitate a logic hardening methodology, as different heuristics for the placement of hazard-filtering circuits within combinational logic are investigated and rated.

## CHAPTER 2

### FAST DATA CAPTURE ON DATA LINES USING MULTI-THRESHOLD RECEIVING LOGIC

Scaling down of CMOS technologies has unquestionably provided obvious performance benefits in terms of operational speed of devices. On the other hand, process shrinkage along with increasing design complexity have initiated new challenges and concerns that undermine an exploitation of modern technology to its full potential. Major burdens in contemporary IC design are the capacitive coupling exhibited between closely located circuit interconnects and their increasing wire resistance.

On-chip interconnects pose tremendous difficulties and raise serious obstacles in the gigascale era [1]. Not only do they have an undesirable impact on signal propagation delay but also on the consumed power and the actual signal integrity. Physical phenomena associated with smaller technologies like surface scattering of conducting electrons, required liner thickness, temperature increases and the typical high-frequency skin effects further exacerbate chip performance [1]. The increasing resistivity of the wires emphasizes coupling effects, worsens time delay and magnifies slew rate thus limiting the attainable bandwidth [2]. Moreover, the emergence of wire inductance likewise pronounces coupling effects, mainly in the case of global interconnects [3].

In this chapter, we explore and evaluate a new way to ensure fast data sampling in a bus of lines. The bus may or may not necessarily operate in the presence of capacitive coupling. The method is based on the premise that instead of a single receiving gate per wire (usually inverter), multiple gates operating at distinctive threshold voltage levels can be utilized. This leads to the capability of reading out the transmitted data earlier than conventional approaches suggest and

as a result the bus performance can be effectively boosted. Distinct threshold voltage levels of the receivers are appropriately selected based on an all-inclusive electrical characterization of a bus line with respect to its surroundings. Such a characterization aims to extract the patterns of the exhibited electrical behavior of a wire during data transmission.

The multiple comparison voltages derived from wire characterization allow for defining a mapping which provides a one-to-one association between previous and next states in lines. This mapping makes possible to rapidly capture data at the receiver-end of the wire and results in shorter wire delays compared to standard repeater-based buses. As a result and depending on the initial conditions of a wire, its anticipated behavior can be quickly resolved with the proposed arrangement without requiring a full voltage swing on the line. The technique is very resilient. The robustness of the method can be enhanced as much as needed for an uncertain or variational ambient.

It is worth noting that for the lines we experimented with, no repeaters are required. This could be exploited for over the cell routing purposes when embedded IP cores are used in SoC applications. Besides the improved data sampling performance, the technique also indicates better handling of interconnect delay variability that is typically due to manufacturing and environmental variations. Some preliminary exploration of this methodology was performed in [43]. The latter had significant limitations with respect to noise margin, variation tolerance, coupling intensity, clock skew, data-sampling window duration and wire inductance modeling. In this work, none of the aforementioned limitations apply.

A signal propagation speedup of 36.4% was computed for some buses in our simulations with HSPICE. Power consumption is in the same order with conventional methods. The additional energy required by some combinational logic introduced at the receiver-end of the line is negligible compared to the

amount of dynamic, switching and leakage power saved by omitting repeaters along wires. Lastly, the area overhead of the added logic is measured to be only 4%. The continuous shrinkage of the size of transistor-based logic in comparison to the area occupied by bus interconnects hints that the imposed area overhead of the method will diminish even further for smaller CMOS processes.

## 2.1 BACKGROUND RESEARCH ON COUPLED DATA BUSES

In addition to the interconnect-centric perspective and in order to develop effective remedies, it is also essential to study the uncertainty in the magnitude of coupling effects resulting from supply voltage, temperature and inherent manufacturing variations. This becomes more imperative for fabrication processes beyond the  $90nm$ . In contemporary chips, where the spread in frequency and leakage distributions due to device geometry variations can reach 30% and  $20\times$  respectively [53], ensuring noise tolerance of a design is vital. In [52], variation aware analysis for crosstalk robustness relies on noise rejection curves and magnitude-duration profiles. Thus, the vulnerability of gates to noise is determined and can be used to further harden circuit logic.

Interconnect delay variance prompted by process variations [59, 60] and by crosstalk aggressor alignment [58] is an increasing concern in the chip industry. In some cases, delay-test strategies need to be adjusted to account for newly emerged variation-related challenges that interfere with testing tasks [61]. Also, statistical modeling of noise under process variations like in [51] is useful for estimating limitations in large scale circuits, where SPICE-like tools and analysis entail prohibitive cost. Existing works aiming to alleviate variance utilize circuit techniques [56, 57], gate sizing combined with statistical approaches [55], substrate body biasing and multiple power supplies [54].

To address these increasing design challenges, several approaches have been

developed by both industry and academia. Among others, existing work suggests driver scaling, wire width and spacing optimization [31, 32, 11, 6]. Popular techniques are based on repeater insertion [25, 27, 29, 33, 10, 50]. Although effective, tedious silicon area allocation and placement of buffers on a line complicate the latter approach. There are also legitimate concerns about this technique, since the number of inserted repeaters is projected to increase and will dominate circuit logic in future technologies [34]. Furthermore, other crosstalk alleviation methods may be developed and implemented at the circuit level [12, 13], gate level [14, 15], routing level [16, 17, 18] or at any of the previous levels accompanied by some type of redundancy (space, time or voltage) [19, 49].

Encoding schemes for crosstalk, such as Hamming and Dual-Rail codes [20, 21], self-shielding code [22, 40], transition code [23], odd/even bus invert code [42] and predefined codeword generation [24] have been studied. Additional ones exist for minimizing coupled switchings on lines [45, 46, 41], reducing transitions on lines [35], minimizing dynamic [47], static [36], peak [44] and leakage power [37]. Existing codes for crosstalk impact minimization like the Dual Rail (DR) and the Modified Dual Rail (MDR) codes in [20, 21] require more than 100% wire overhead. Under minimal separation rules for any two adjacent lines, this would imply at least a doubling of the allocated design area. The bus invert [41] and the odd/even bus invert encodings [42] have small wire and area overhead and are effective at reducing the overall switching and coupling activities. However, neither one resolves crosstalk-related performance individually for each bus line. As a result, even though the overall number of potential bad-crosstalk cases is dropped considerably, transient errors due to coupling can still occur in every clock cycle. Self-shielding codes like in [22] and others [40] usually have bit redundancy in the order of 40%, while design complexity issues in encoders/decoders will impose additional delay overhead or even render them impractical. Using the predefined

codeword generation in [24] eliminates certain transition patterns and proposes a bus-partitioning scheme that entails wire increase in the order of 50%.

In other published work, the order of the interconnects is changed so that any potential crosstalk-induced delay originating from opposite-phase transitions in neighboring lines is avoided [38]. In [48], statistical and probabilistic arguments are employed to distribute the noise sources and thus render unlikely the possibility of worst-case alignment between aggressor and victim. Attempts to ease down the newly introduced adversities lead to the emergence of the on-chip interconnection networks or NoC in place of global on-chip wires [39]. However, since this new scheme also utilizes physical channels, performance challenges due to wire coupling and resistance are still of concern.

## 2.2 DESCRIPTION OF THE METHODOLOGY

### 2.2.1 Fundamentals of the method

The method assumes coupling between lines but it is certainly valid when wire interference is less than expected or even completely nonexistent. In the following, we will define a *low*→*high* (*high*→*low*) transition on a bus of lines as a *rising* (*falling*) transition. Similarly in the case of a glitch, we define a *low*→*high*→*low* (*high*→*low*→*high*) hazard on a bus of lines as a *rising* (*falling*) hazard. In essence, the presented work determines -as a function of time- *lower bounds* for the voltage value of rising transitions or falling glitches and *upper bounds* for the voltage value of falling transitions or rising glitches in bus channels. These time-defined bounds represent cutoff points or thresholds for the worst-case crosstalk effects that may be generated on a line during data transmission and they depend on the initial conditions (previous logic states) of the victim and of the aggressor lines. For instance, a rising transition on a victim line will have different lower bounds for different types of transitions on the aggressor (i.e.,

rising, falling or quiet). In this way, for each combination of input transitions in a bus we can accordingly define in time either a lower or an upper bound for the electrical voltage of victim lines.

These bounds define reference voltages used in the receiving logic of the proposed methodology. To limit the involved hardware to the minimum, the upper and lower bounds are selected in a way that they coincide to a minimum number of reference voltages. By keeping track of the initial conditions of lines and by performing voltage measurements at the receiver-end, it is possible to look up the anticipated next state of a line and thus quickly identify the transmitted data. Some combinational logic that implements this look up procedure is needed. Once the propagated data has been sampled, the transition is abruptly interrupted and terminated. Upon resetting the line, the next data can be immediately sent across. The approach is robust, *i.e.*, the selected threshold voltages extracted for worst-case crosstalk effects apply also in the absence of coupling. This is because the boundaries established for worst-case crosstalk interference are certainly not violated when coupling is minimal and hence the combinational look up logic is still functional.

### 2.2.2 Coupled interconnect with one adjacent aggressor

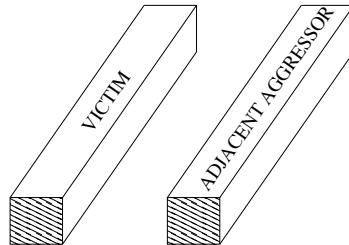


Figure 2.1. Victim line with one adjacent aggressor

We first illustrate the proposed method in the case of a single aggressor adjacent to the victim line without any distant aggressors present (Figure 2.1).

This is the typical case of a two-bit bus. To carry out a bus-interconnect characterization, all possible electrical transitions of a line are created for every type of switching that the neighboring wire might be experiencing. Figures 2.2, 2.3, 2.4 illustrate the electrical transitions on the victim wire for different activity occurring on its neighbor. It is evident that the electrical behavior of a victim wire for the same type of transition (rising, falling or stable) will differ significantly in terms of slew rate and/or attainable amplitude with the type of switching on its aggressor line. This diverse behavior can be exploited to define in each case distinctive cut-off points - implying receiver-threshold voltages - which will be used to retrieve the next state of the line. These distinct receiver-threshold voltages can be realized either by multiple receiver gates, each one of a fixed threshold, or by a single receiver of an adaptive threshold. In this research, we assume the former approach without the loss of generality.

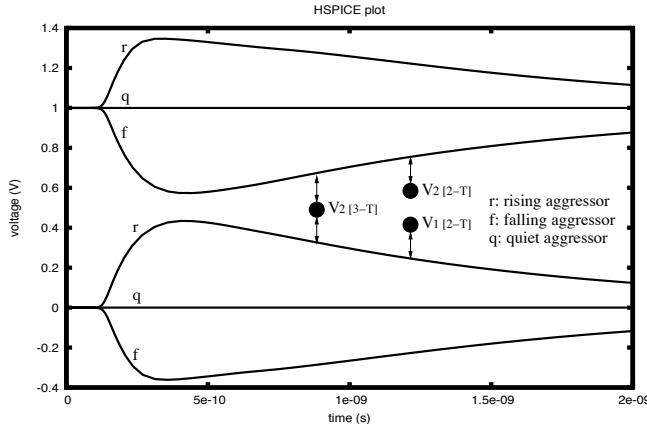


Figure 2.2. Glitches for stable/quiet transitions on a victim line with one adjacent aggressor (IBM 65nm)

Typically in a conventional bus, a single receiver inverter of a fixed threshold voltage is used at the end of a line to read the incoming data. This entails that slow transitions that are caused by line crosstalk interference will require additional time until they can sufficiently settle on wires, well above or below the

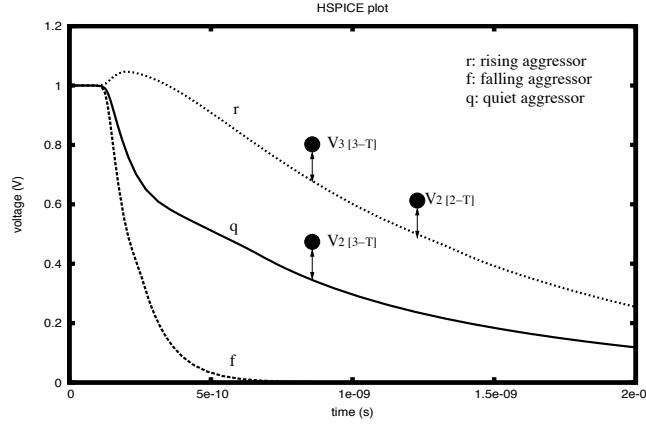


Figure 2.3. Falling transitions on a victim line with one adjacent aggressor (IBM 65nm)

inverter threshold (with typical value  $\frac{V_{dd}}{2}$ ). Therefore, the clock edge is inevitably delayed before data sampling is possible. In this case, the only way to speed up operation is to insert repeater inverters on bus lines complicating further the design phase of SoC topologies. The operation of this basic receiver-end logic follows Table 2.1. In the latter, we denote a *don't care* logic condition or value with x. It is observed that the initial conditions or previous states of the lines are labeled as don't care conditions and hence do not matter for such a scheme. When the sensed line voltage lies above the receiver gate threshold, then this indicates that a *high* logic value is being propagated. Otherwise, it is concluded that a *low* logic signal is communicated to the receiving logic block.

Table 2.1. Typical single-threshold mapping table

Victim Prev.	Aggressor Prev.	Victim Volt. Range	Victim Next
x	x	$V > V_t$	1
x	x	$V < V_t$	0

However, it is evident that resolving the slowest transitions earlier could

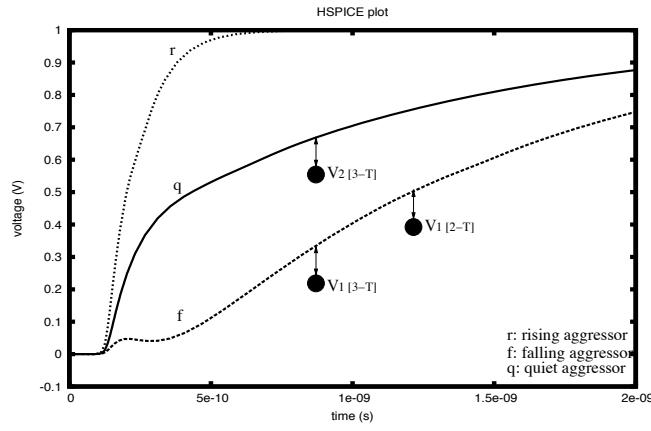


Figure 2.4. Rising transitions on a victim line with one adjacent aggressor (IBM 65nm)

significantly reduce the total propagation delay and hence dramatically improve the throughput of a bus. To achieve this, more than one reference voltages can be used in the receiving logic. This allows for adopting a higher threshold voltage to speed up slow falling transitions, and a lower threshold voltage to accelerate slow rising transitions. The potentially slow transitions are identified using the initial conditions of lines. As a result, multiple receiving gates can be employed that will operate simultaneously at different threshold voltages. In this way, it is no longer needed to wait for the signal to fully or sufficiently settle on the line. The eligibility of a potentially slow transition is depending upon the initial conditions of the bus. These threshold voltages are the extracted bounds resulting from the characterization of a wire operating under intense coupling (and are illustrated in Figures 2.2, 2.3, 2.4). In a receiving logic of that type, the initial conditions of the lines need to be taken into consideration. In the following, two-threshold and three-threshold receiving logic modules for a two-bit bus are discussed.

It will be shown that two reference voltages on the receiver-side accelerate the data sampling on the bus. As a graphical illustration, the two-threshold arrangement is defined in Figures 2.2, 2.3 and 2.4. Selecting the appropriate time

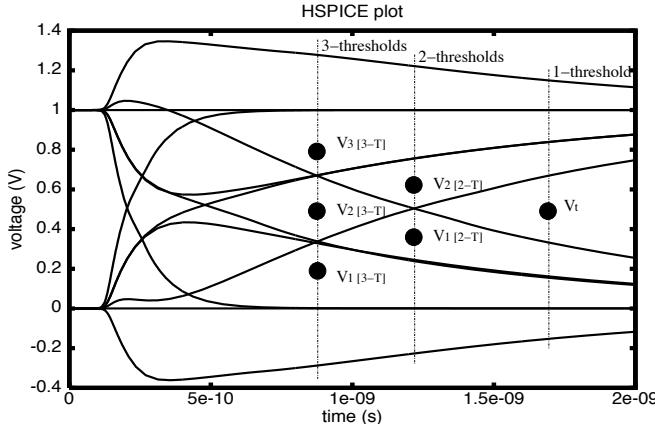


Figure 2.5. All transitions on a victim line with one adjacent aggressor (IBM 65nm)

instant permits the aforementioned voltages to remain valid from that particular instant until the end of the clock cycle. These are represented by voltages  $V_1 [2-T]$  and  $V_2 [2-T]$  in Figure 2.5 on the two-threshold axis. We will refer to the first time instant that this convergence is true as the *earliest sampling instant*. The time delay gained by not having to wait for the slowest transitions to sufficiently settle shortens the sampling clock period and this can ultimately enhance throughput. The slow transitions are identified based on the logic of Table 2.2.

Table 2.2. Two-threshold mapping table for a victim line with one adjacent aggressor (IBM 65nm)

Victim Prev.	Aggressor Prev.	Victim Volt. Range	Victim Next
x	x	$V > V_2 [2-T]$	1
1	0	$V < V_2 [2-T]$	0
0	1	$V > V_1 [2-T]$	1
x	x	$V < V_1 [2-T]$	0

Consider next the case of three total reference voltages. The method defines

different thresholds depending on how slow a transition may potentially be. Again, slow transition eligibility depends on the initial conditions of the wires. Likely slow rising transitions get assigned lower thresholds. On the other hand, likely slow falling transitions utilize higher thresholds for faster data identification. For the three-threshold case, the corresponding reference voltages are displayed in Figures 2.2, 2.3 and 2.4. The three distinct thresholds are valid bounds until the end of each clock cycle. The voltages are shown in Figure 2.5 as  $V_1$  [3-T],  $V_2$  [3-T] and  $V_3$  [3-T] along the three-threshold axis. All transitions in this case are handled by the logic implementing Table 2.3. It is also stressed that for the case of a line with one adjacent aggressor only (*i.e.*, case of a two-bit bus), a three-threshold receiver logic is delay optimal (Figure 2.6).

Table 2.3. Three-threshold mapping table for a victim line with one adjacent aggressor (IBM 65nm)

<b>Victim Prev.</b>	<b>Aggressor Prev.</b>	<b>Victim Volt. Range</b>	<b>Victim Next</b>
x	x	$V > V_3$ [3-T]	1
1	0	$V < V_3$ [3-T]	0
0	x	$V > V_2$ [3-T]	1
1	1	$V > V_2$ [3-T]	1
0	0	$V < V_2$ [3-T]	0
1	x	$V < V_2$ [3-T]	0
0	1	$V > V_1$ [3-T]	1
x	x	$V < V_1$ [3-T]	0

In the case of a distant aggressor (Figure 2.7), a different electrical behavior is exhibited by the wire (Figure 2.8). Because of the aliasing introduced by the distant neighbor, a three-threshold receiving logic doesn't seem feasible any longer.

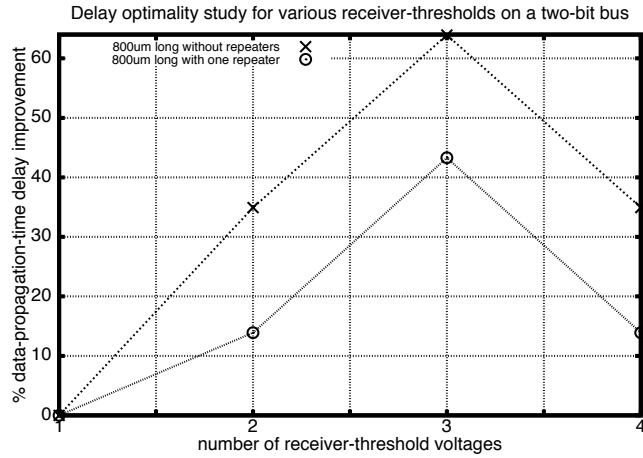


Figure 2.6. Study on delay optimal number of receiver thresholds for a two-bit  $800 \mu m$  long bus(IBM 65nm)

Instead, a two-threshold logic is recommended and applies, as described in Table 2.2.

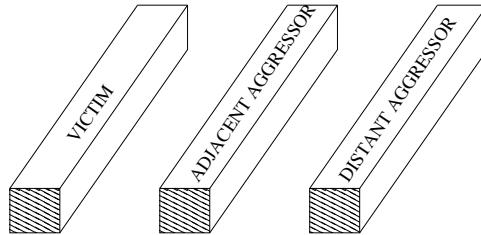


Figure 2.7. Victim line with one adjacent and one distant aggressor

### 2.2.3 Coupled interconnect with two adjacent aggressors

In order to examine the applicability of such an architecture for SoC data buses, the electrical behavior of victim lines having more than a single adjacent aggressor should be thoroughly studied. Typical co-planar bus of parallel lines result in at most two adjacent aggressors per victim line (three-bit bus in Figure 2.9). To these, one could add at most two distant aggressors (five-bit bus in Figure 2.15). Beyond that, further wires cannot exert any significant noise on a

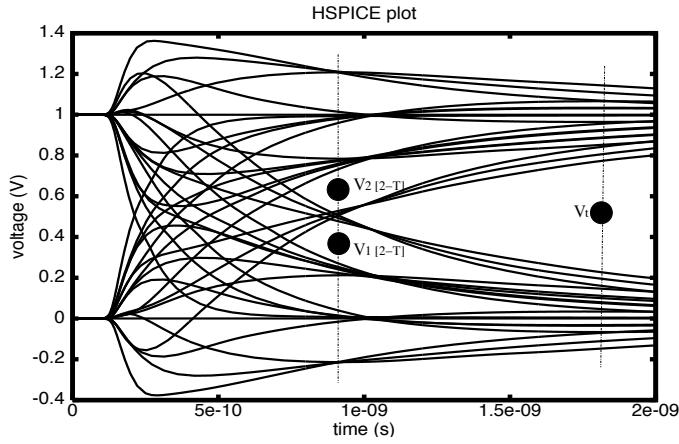


Figure 2.8. All transitions for a victim line with one adjacent and one distant aggressor (IBM 65nm)

victim line and those will be ignored in our analysis.

Following the same guidelines as in the case of a two-bit bus, the slowest transitions are identified based on the previous states of the victim and aggressor wires. Once a potential slow switching is sensed, appropriately low or high threshold voltages are used in order to speed up the read operation. As a consequence, given the characterization of a line with two adjacent aggressors (illustrated in Figures 2.10, 2.11, 2.12 and 2.13), one can design receiving logic of two-thresholds (Table 2.4), three-thresholds (Table 2.5), four-thresholds (Table 2.6) and five-thresholds (Table 2.7). From the previous alternatives, the latter one proves to be the delay optimal configuration (Figure 2.14). When (at most two) distant aggressors are present, the aforementioned multi-threshold logic (for two adjacent aggressors) applies as is. The anticipated electric behavior of such a victim line will slightly differ because of the contribution of the added lines (Figure 2.16).

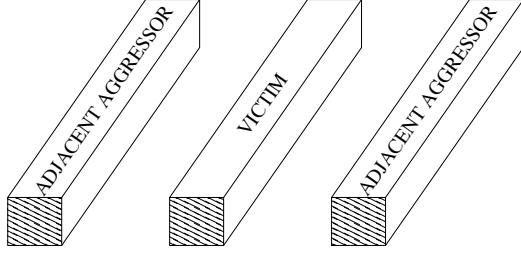


Figure 2.9. Victim line with two adjacent aggressors

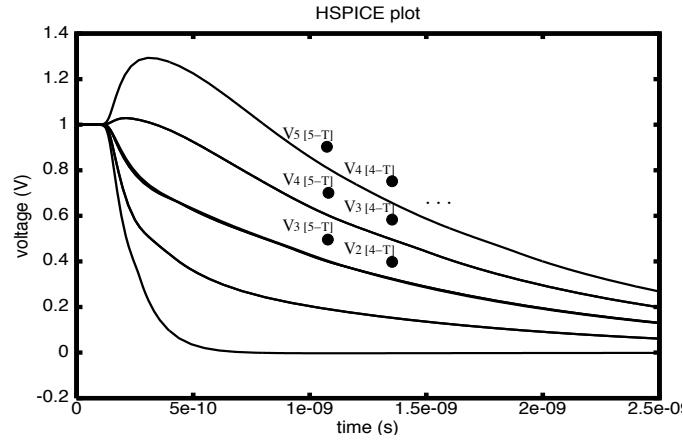


Figure 2.10. Falling transitions for a victim line with two adjacent aggressors (IBM 65nm)

#### 2.2.4 Circuit implementation

We propose a pipeline-based implementation of the new data capturing method as shown in Figure 2.17. The pipeline consists of two stages. Specifically, in the first stage the architecture handles wire initialization and data transmission. In the next stage, the appropriate information is quickly processed in order to calculate the transmitted data. The overall combinational logic introduced by the method consists of the receiver gates with the reset circuitry (in the first stage) and the core lookup logic (in the second stage). The former one performs the reading of the propagated data followed by a reset in the next cycle, whereas the latter one determines the value of the transmitted data. The receivers may be

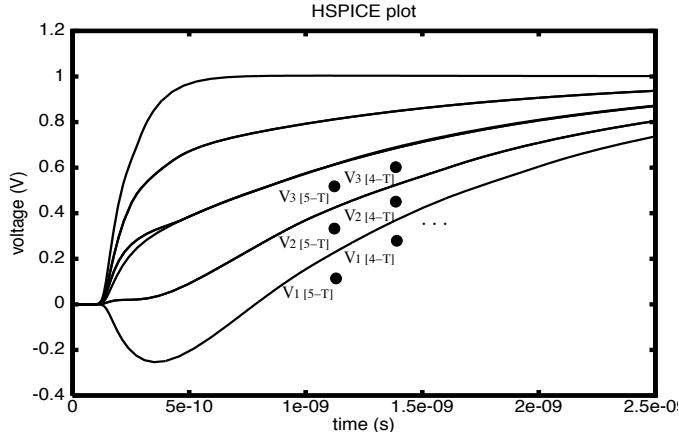


Figure 2.11. Rising transitions for a victim line with two adjacent aggressors (IBM 65nm)

Table 2.4. Two-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm)

<b>Victim Prev.</b>	<b>Aggressors Prev.</b>	<b>Victim Volt. Range</b>	<b>Victim Next</b>
<b>x</b>	<b>xx</b>	$V > V_2 \text{ [2-T]}$	<b>1</b>
<b>1</b>	<b>00</b>	$V < V_2 \text{ [2-T]}$	<b>0</b>
<b>0</b>	<b>11</b>	$V > V_1 \text{ [2-T]}$	<b>1</b>
<b>x</b>	<b>xx</b>	$V < V_1 \text{ [2-T]}$	<b>0</b>

ordinary inverters or buffers operating at different thresholds. The lookup logic is a simple circuit that implements the mapping of Tables 2.2 or 2.3.

In more details, the first stage of the pipeline involves two distinct tasks. First, a necessary wire initialization is triggered before transmitting the data. This resetting task aims to quickly initialize the line voltage to the nearest valid logic level ( $V_{dd}$  or *Ground*). This is achieved by deactivating a tristate buffer at the transmitter end while at the same time activating a tristate buffer at the receiver end. This coordinated undertaking can be done using a single clock. The inverter -

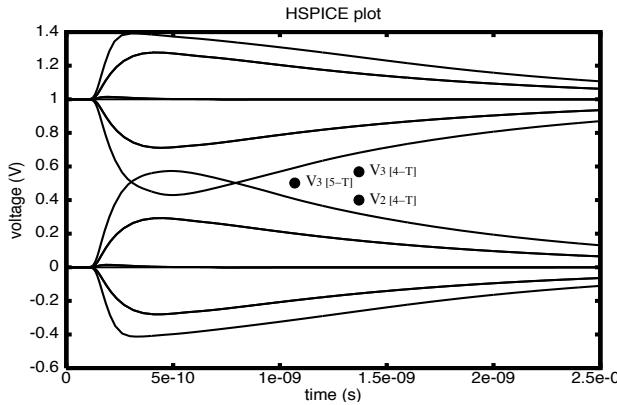


Figure 2.12. Hazards (only) transitions for a victim with two adjacent aggressors (IBM 65nm)

tristate buffer loop at the receiver end pushes the line voltage to the nearest valid logic level (Figure 2.18).

The second task is the data propagation. It is accomplished with the tristate buffer at the transmitter end being active while the one at the receiver end being idle. The data propagation delay depends on factors such as the crosstalk interference between the lines, the number of reference (threshold) voltages used, the desired noise margin, the strength of the driver gates and the delay of the receiver logic. At the end of this task, a snapshot of the receivers output is taken. The lookup logic calculates the next state of the victim under consideration based on the information collected from the receivers and the knowledge of the initial states for both the aggressor and the victim. The computed logic value is subsequently stored into a latch or directly fed into the input port queue of a receiving network component.

The wire initialization step in the first stage of the pipeline will impose some additional hold up time on the overall delay. Since this determines the operational clock frequency, the clock period for the proposed method is obtained as a sum of the time delays to reset ( $D_r$ ), propagate ( $D_p$ ) and read the data at the receivers

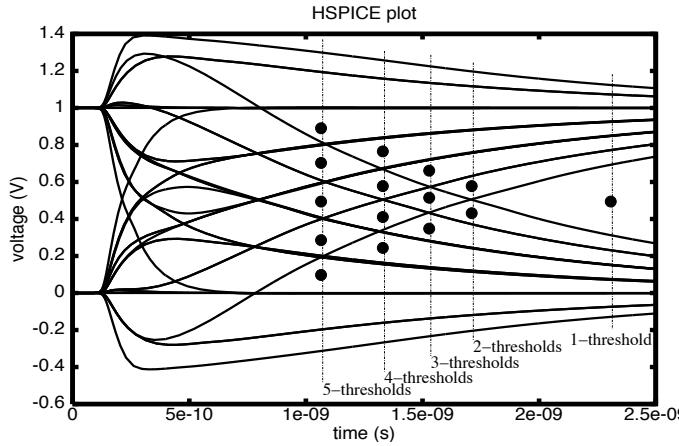


Figure 2.13. All transitions for a victim line with two adjacent aggressors (IBM 65nm)

output ( $D_s$ ). We define  $P_{new}$  the clock period in the proposed method and  $P_{buf}$  to be the clock period required by a conventional repeater buffer based approach. It follows that:

$$P_{new} = D_r + D_p + D_s$$

In Table 2.8, we provide some indications on the performance of the proposed method in comparison with an optimized repeater-based technique for a coupled bus. The results are also illustrated graphically in Figures 2.19 and 2.20. Similar results are also provided for a three-bit (Figure 2.21) and a five-bit (Figure 2.22) buses. The notation  $C_{sub}$  refers to the wire capacitance with respect to the substrate. The reset time delay of the proposed method is measured when the wire voltage is within a 10% margin from its final reset value. Furthermore, both the data-read time delay for the proposed method and the overall time delay for the typical repeater method are measured at 90% of the final voltage value at the respective receivers output. For this evaluation, the delay-optimal three-threshold logic is used for the proposed method.

If  $K_{buf}$  is the number of clock cycles needed to deliver data along a line in a

Table 2.5. Three-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm)

<b>Victim Prev.</b>	<b>Aggressors Prev.</b>	<b>Victim Volt. Range</b>	<b>Victim Next</b>
<b>x</b>	<b>xx</b>	$V > V_3 [3-T]$	<b>1</b>
<b>1</b>	<b>00</b>	$V < V_3 [3-T]$	<b>0</b>
<b>0</b>	<b>1x</b>	$V > V_2 [3-T]$	<b>1</b>
<b>1</b>	<b>0x</b>	$V < V_2 [3-T]$	<b>0</b>
<b>0</b>	<b>11</b>	$V > V_1 [3-T]$	<b>1</b>
<b>x</b>	<b>xx</b>	$V < V_1 [3-T]$	<b>0</b>

typical bus containing repeater buffers, then the number of clock cycles required for completing the same data transfer in the proposed pipeline-based method, denoted by  $K_{new}$ , is:

$$K_{new} = K_{buf} + (S - 1)$$

where  $S$  is the total number of stages in the pipeline (in this case  $S = 2$ ).

Therefore, for a fairly large amount of data  $K_{new} \simeq K_{buf}$ . The experimental results show that  $P_{new} < P_{buf}$  for the lines and the technologies we experimented with. Then it follows that, for a large amount of data, the total time required to complete the transfer is considerably smaller compared to the buffer-based approach, and as a result throughput can be efficiently increased.

### 2.2.5 Resilience and variability of the method

For clock skew, variations and random noise related deviations, a margin enhancement is possible for the proposed method. Specifically and as described previously in Figures 2.2, 2.3 and 2.4, the multi-threshold technique is defined with respect to adequate noise margins between the reference voltages and the upper or

Table 2.6. Four-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm)

<b>Victim Prev.</b>	<b>Aggressors Prev.</b>	<b>Victim Volt. Range</b>	<b>Victim Next</b>
<b>x</b>	<b>xx</b>	$V > V_4 [4-T]$	<b>1</b>
<b>1</b>	<b>00</b>	$V < V_4 [4-T]$	<b>0</b>
<b>0</b>	<b>xx</b>	$V > V_3 [4-T]$	<b>1</b>
<b>1</b>	<b>11</b>	$V > V_3 [4-T]$	<b>1</b>
<b>1</b>	<b>0x</b>	$V < V_3 [4-T]$	<b>0</b>
<b>0</b>	<b>1x</b>	$V > V_2 [4-T]$	<b>1</b>
<b>0</b>	<b>00</b>	$V < V_2 [4-T]$	<b>0</b>
<b>1</b>	<b>xx</b>	$V < V_2 [4-T]$	<b>0</b>
<b>0</b>	<b>11</b>	$V > V_1 [4-T]$	<b>1</b>
<b>x</b>	<b>xx</b>	$V < V_{t1}$	<b>0</b>

lower bounds of slow transitions. Due to deep sub-micron related uncertainty and vulnerability, it is always advisable to assume a margin sufficient to absorb manufacturing irregularities, ambient variations and random noise. The proposed method can be adapted so that it is hardened as much as the circumstances require.

Delaying the sampling time instant ensures that a signal propagating on a coupled wire will increasingly lean towards its intended next state value and hence move away from its upper or lower bound. Thus, the noise margin increases and the robustness of the method is refined as much as needed. Figure 2.23 illustrates the increase in the resilience of the method. An indication on the impact of this adaptation is given in Figure 2.24. In the latter, the sampling instant is delayed by only a small percentage of the clock cycle. When two adjacent aggressors are

Table 2.7. Five-threshold mapping table for a victim line with two adjacent aggressors (IBM 65nm)

<b>Victim Prev.</b>	<b>Aggressors Prev.</b>	<b>Victim Volt. Range</b>	<b>Victim Next</b>
<b>x</b>	<b>xx</b>	$V > V_5 [5-T]$	<b>1</b>
<b>1</b>	<b>00</b>	$V < V_5 [5-T]$	<b>0</b>
<b>1</b>	<b>1x</b>	$V > V_4 [5-T]$	<b>1</b>
<b>1</b>	<b>0x</b>	$V < V_4 [5-T]$	<b>0</b>
<b>1</b>	<b>11</b>	$V > V_3 [5-T]$	<b>1</b>
<b>0</b>	<b>xx</b>	$V > V_3 [5-T]$	<b>1</b>
<b>0</b>	<b>00</b>	$V < V_3 [5-T]$	<b>0</b>
<b>1</b>	<b>xx</b>	$V < V_3 [5-T]$	<b>0</b>
<b>0</b>	<b>1x</b>	$V > V_2 [5-T]$	<b>1</b>
<b>0</b>	<b>0x</b>	$V < V_2 [5-T]$	<b>0</b>
<b>0</b>	<b>11</b>	$V > V_1 [5-T]$	<b>1</b>
<b>x</b>	<b>xx</b>	$V < V_1 [5-T]$	<b>0</b>

present, it is also possible to increase the resilience of such an arrangement by trading off some delay (Figure 2.27). Similar arguments apply in the special case of existing distant neighbors (Figure 2.28).

The addition of reference voltages in the receiving logic is also beneficial for the delay variance exhibited by interconnects operating under coupling. In general, wire crosstalk interference and resistance amplify the impact of the existing variations. As a consequence, slower transitions will most likely exhibit higher variance earlier than faster ones and the spread in performance broadens significantly with time (Figure 2.25). The proposed method defines a sampling instant quite earlier than conventional methodologies. As a consequence, the

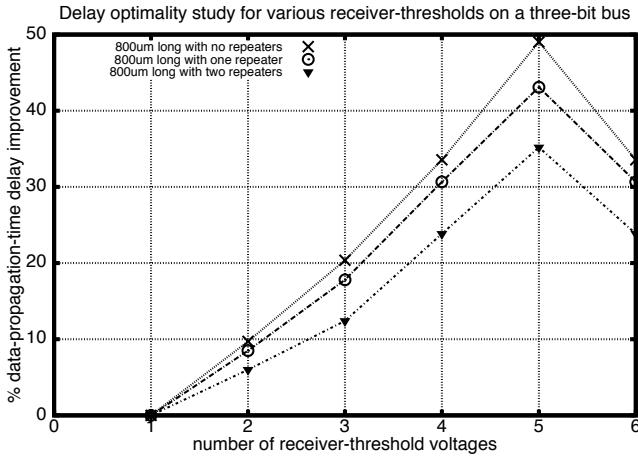


Figure 2.14. Study on delay optimal number of receiver thresholds for a three-bit  $800 \mu m$  long bus (IBM 65nm)

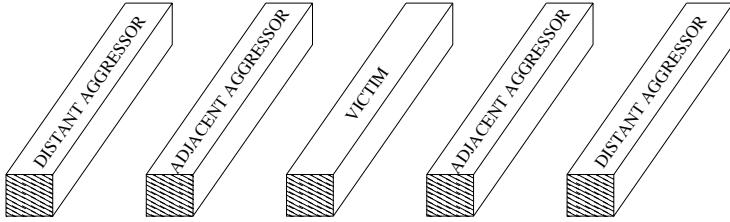


Figure 2.15. Victim line with two adjacent and two distant aggressors

transmitted signal delay does not stretch as much and hence exhibits better variance characteristics.

The results of Figure 2.25 and the plotted standard deviation results in Figure 2.26 are obtained using an Monte Carlo driven HSPICE analysis. The setup involves two identical buses that use a single-threshold and multiple-threshold receivers respectively. In the case of the typical single-threshold receiver, its reference voltage is taken  $V_t = \frac{V_{dd}}{2}$ . Inter-die variations of 8% and  $25mV$  at  $1\sigma$  are assumed on the channel length and threshold voltage respectively of all transistors in our experiments. It is observed for different noise margins a multi-threshold receiving logic manifests a higher tolerance to manufacturing variations than the one in standard single receiver buses.

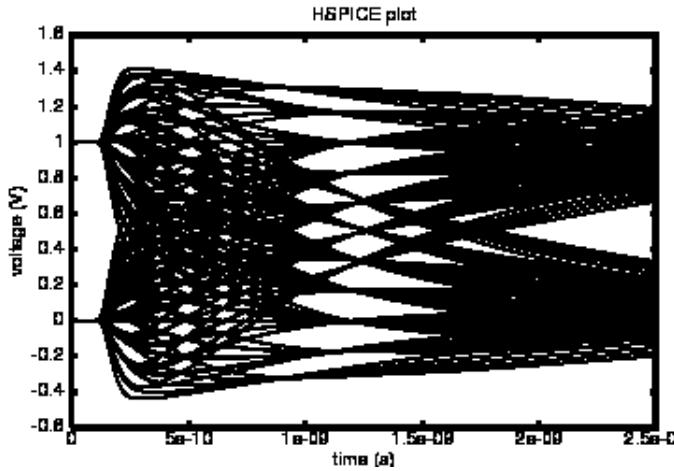


Figure 2.16. All transitions for a victim line with two adjacent and two distant aggressors (IBM 65nm)

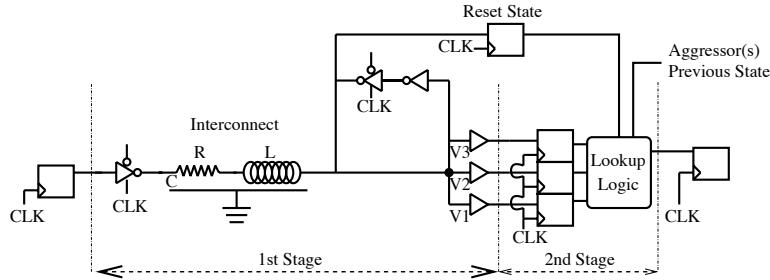


Figure 2.17. Proposed pipeline for data interconnects

### 2.3 EXPERIMENTAL EVALUATION ON GENERALIZED DATA BUSES

We present experimental results in the IBM CMOS 65nm technology for 32-bit metal layer 3 buses of  $250\mu m$ ,  $500\mu m$  and  $750\mu m$  length. A C-language programming tool was developed to synthesize HSPICE circuit netlists, carry out Monte Carlo simulations when needed, extract and elaborate on the experimentally obtained data. The buses were first implemented with the typical repeater-based method. The simulated interconnects for these buses are delay-optimized with respect to wire resistance, inductance and capacitance to substrate. Specifically,

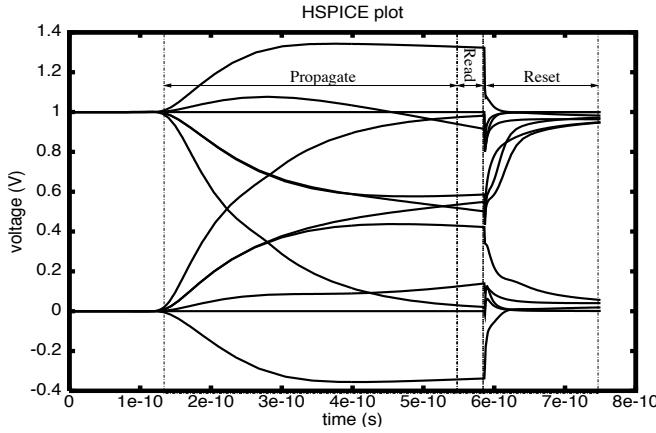


Figure 2.18. Data propagation, read and reset (in next clock cycle) tasks

extensive parametric analyses were performed for picking the optimal number and size of repeater inverters to be inserted for each of the above bus types. Same driver sizes apply for both the repeater-based and the proposed buses. For shielding purposes, highly-resistive RLC-modeled wires were utilized.

The proposed method was implemented by partitioning the 32-bit bus in isolated two-bit neighborhoods as shown in Figure 2.29. The reset and receiving logic circuitry was incorporated in the setup. For the experimental results, a noise margin of  $50mV$  is assumed. We insisted that this arrangement occupies the same area as a typical repeater-based bus structure. In particular, in Figure 2.29, it is assumed that  $\lambda$  is the minimum width of a line in the bus. For simplicity,  $\lambda$  is also taken as the minimum separation distance between two parallel interconnects that does not violate the technology design rules. This assumption is consistent with layout guidelines of modern CMOS processes.

In Figure 2.29,  $C_d$  denotes the coupling capacitance between two signal lines at distance  $d$  in the proposed architecture. Similarly,  $a$  is the separation distance between signal lines and shielding in the partitioned neighborhoods and  $C_a$  is the parasitic capacitance between a signal and a shielding line. On the other hand,  $D$

Table 2.8. Clock period estimation (three-threshold logic) for a line with one adjacent aggressor (IBM 65 nm)

Wire length (um)	200	300	400	500	600	700	800	900	1000
<b>Delay in typical (ps)</b>	252	327	443	543	657	709	822	926	987
<b>Repeater in typical</b>	1	1	2	2	2	3	3	4	5
<b>Repeater in proposed</b>	0	0	0	0	0	0	0	0	0
<b>Propagate in proposed (ps)</b>	166	199	239	289	349	419	499	590	688
<b>Read in proposed (ps)</b>	35	35	35	35	35	35	35	35	35
<b>Reset in proposed (ps)</b>	65	57	58	62	77	96	118	212	217
<b>Delay in proposed (ps)</b>	266	291	332	386	461	550	652	837	940

is the distance between two adjacent data lines in the generic bus and  $C_D$  is the coupling capacitance between those two lines. Equal implementation area in the partitioned and un-partitioned buses implies that:

$$d = 2D - 2a - \lambda$$

Specific coupling intensity levels are used per wire separation distance.

The insertion of shielding lines for partitioning a bus leads to an adjustment of the parasitic capacitance of each victim wire. These new parasitics were also included in the analysis. Specifically, two different topology arrangements were examined for the proposed method and their corresponding capacitance extraction ratios are given in Tables 2.9 and 2.10. For a given technology, the computed values for  $d$  and  $a$  can be sufficiently approximated in the chip layout phase. Specifically, the IBM 65nm CMOS process calls for a minimum width of  $0.1\mu m$  in metal layer 3, whereas a chip is laid out on a  $0.005\mu m$  grid. Hence, it is possible to approximate to a good extent the computed wire placements.

The simulated bus lines for both the proposed and the repeater-based

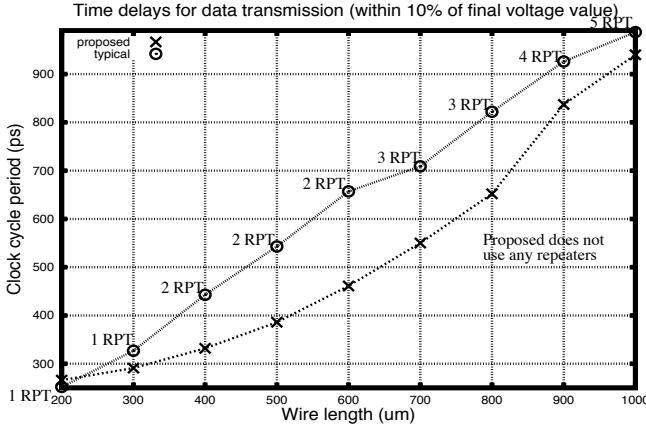


Figure 2.19. Clock periods for proposed (three-threshold logic) and conventional methods in a two-bit bus with Coupling = 10 C<sub>sub</sub> (IBM 65nm)

methods were of minimum width and are modeled as RLC networks in metal layer 3 in both CMOS processes. For the interconnects, a 100% metal coverage is assumed from metal layer 4 (top) and metal layer 2 (bottom). RC network parameters were extracted from the IBM documentation and test data. Wire inductance L was estimated from [62].

When a Monte Carlo driven circuit analysis was required, the statistically significant value of 100 runs per case was assumed. We considered the following independent inter-die variations: 8%, 25mV and 50mV at  $1\sigma$  for the device channel length, the transistor threshold voltage and the power supply voltage ( $V_{dd}$ ) respectively. The assumed variations are traditionally due to lithography and machine imperfections, shifts in doping (carrier) concentrations, oxide thickness variation and simultaneous switching noise.

The following delay estimation method was applied to both the standard repeater-based and the proposed approach. Rising and falling times for the transitions at the driver inputs were both taken to be 50ps. Delay was measured from the time instant for which the voltage is  $\frac{V_{dd}}{2}$  at the driver input to  $\frac{9}{10}V_{dd}$  for a

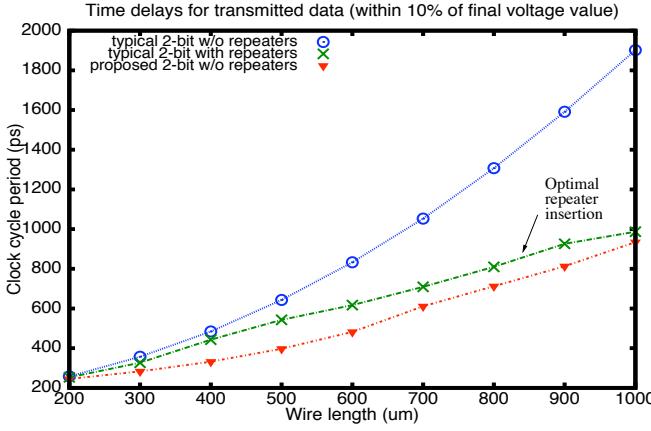


Figure 2.20. Clock periods for proposed (three-threshold logic) and conventional methods in a two-bit bus with Coupling = 10 C<sub>sub</sub> (IBM 65nm)

$0 \rightarrow 1$  transition and to  $\frac{1}{10}V_{dd}$  for a  $1 \rightarrow 0$  transition at the flip-flop input at the receiver-end of the wire.

We compared the performance of the proposed method to the buffer insertion method with respect to the increasing challenges that emerge in deep sub-micron such as: stronger crosstalk, shrinking wire separation distance, amplified wire resistance, sharp  $V_{dd}$  scaling and intensified manufacturing and environmental variations. Figures 2.30 to 2.36 summarize the results of the comparative study. For the typical approach, we also list the number of repeaters used in the bus to optimize its performance.

Figures 2.30 and 2.31 show that in most cases the proposed methodology achieves higher speed with decreasing line separation distance and hence increasing crosstalk intensity. From the slopes of the plots, it is clear that the trend also holds for higher crosstalk than the one assumed in these experiments. The estimated clock period for the proposed technique in Figures 2.30 and 2.31 follows the guidelines of Table 2.8. In particular, the proportionality of the individual contributions of the propagation, reset and read time delays is more or less

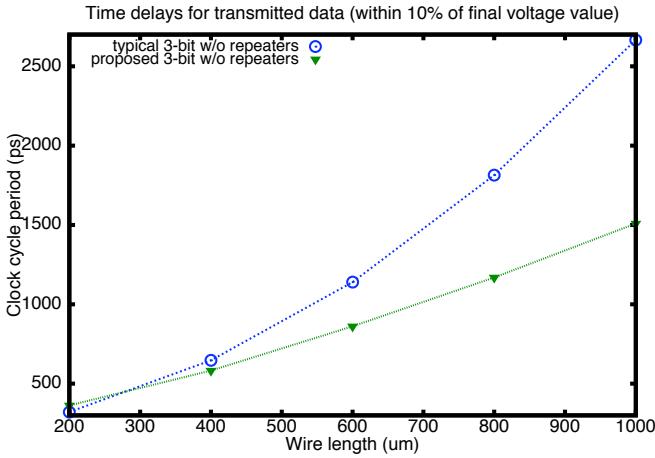


Figure 2.21. Clock periods for proposed (five-threshold logic) and conventional methods in a three-bit bus with Coupling = 10 Csub (IBM 65nm)

equivalent to that of Table 2.8. A study on the performance of the architectures with decreasing  $V_{dd}$  indicates that lower power supply deteriorates the performance of typical repeater-based buses more seriously than the proposed technique (Figure 2.32).

The results in Figures 2.33, 2.34 and 2.35 indicate that the resulting wire delay variance in the proposed method is affected less by coupling and design/ambient irregularities than the ordinary repeater-based buses. This is also confirmed in Figures 2.36 which examines the response of the two topologies to the ever increasing (in the deep sub-micron) coupling and wire resistivity effects. In this last case, repeaters are not used for either method so that the deep sub-micron effects are amplified for more accurate observations and a more transparent analysis. Lastly, HSPICE simulations show that the power consumed by the proposed method is comparable to the power required by the typical approach and in the worst-case it is not increased by more than 5%.

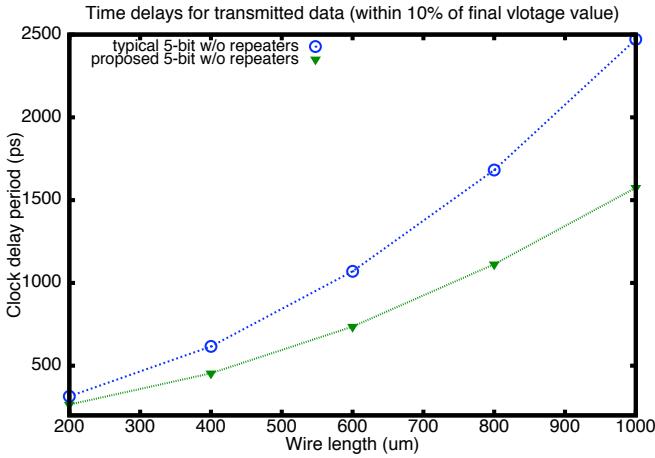


Figure 2.22. Clock periods for proposed (five-threshold logic) and conventional methods in a five-bit bus with Coupling = 10 C<sub>sub</sub> (IBM 65nm)

## 2.4 CONCLUSIONS

A fast data capturing method was presented for high performance bus architectures in deep sub-micron that involves additional reference voltages in the receiving logic. With the proposed research, the increasingly severe impact of restrictions in data bus design and the overhead of repeater-based solutions can be effectively mitigated. Also, variation tolerance of coupled interconnects can be significantly enhanced boosting thus chip yield. Experimental results are given to validate the proposed concept. Current and future work will concentrate on enhancing the combinational lookup logic with fault tolerant circuitry for treating intermittent and unaccounted upsets. Encoding algorithms combined with the proposed research will allow for elimination of certain transition patterns and thus will benefit the speed and resilience of our method. Furthermore, a flexible generalization of the multi-threshold method for wider buses is currently under investigation.

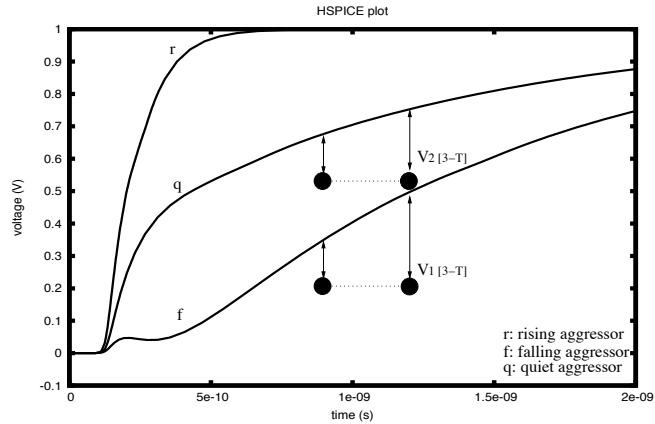


Figure 2.23. Increased noise margin for a victim wire in a two-bit bus (IBM 65nm)

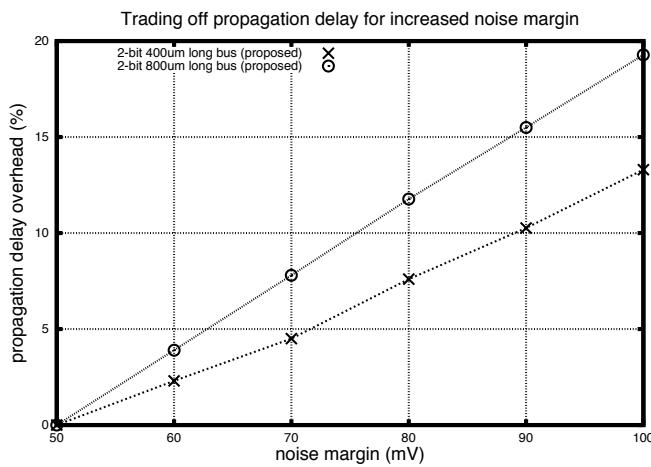


Figure 2.24. Wire delay change with increasing noise margin for two-bit buses using three-threshold logic (IBM 65nm)

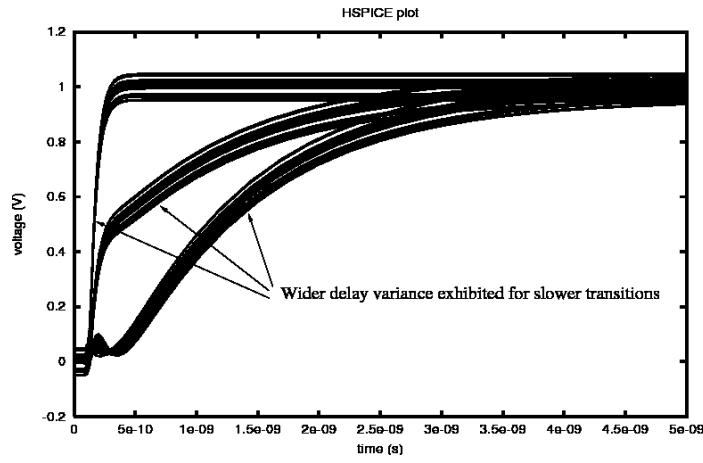


Figure 2.25. Wire delay variance for slow and fast transitions (IBM 65nm)

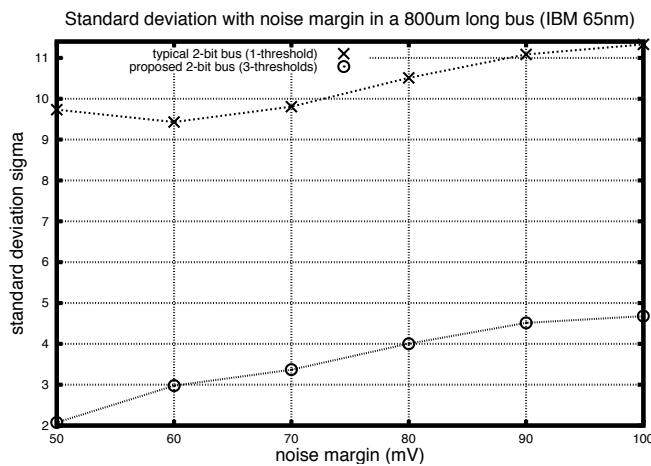


Figure 2.26. Standard deviation of the wire delay for typical and proposed two-bit buses using three-threshold logic with Coupling = 10 C<sub>sub</sub> (IBM 65nm)

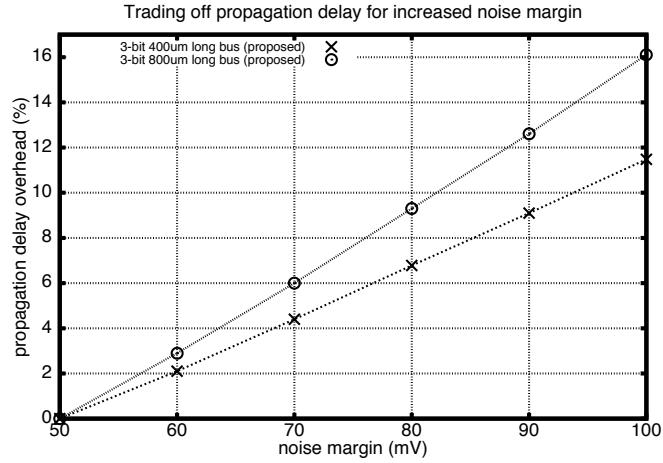


Figure 2.27. Wire delay change with noise margin for three-bit buses with five-threshold logic (IBM 65nm)

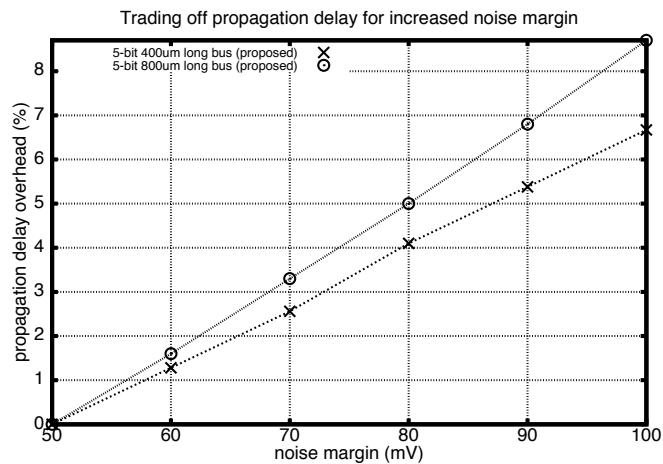


Figure 2.28. Wire delay change with noise margin for five-bit buses with five-threshold logic (IBM 65nm)

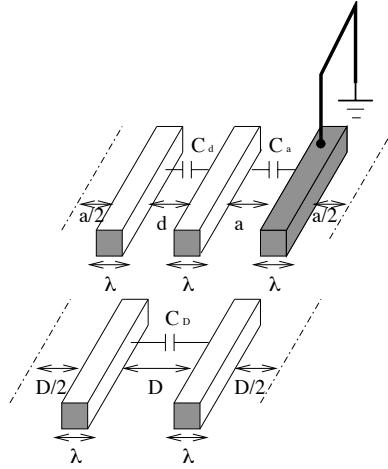


Figure 2.29. A two-bit bus partitioning (top) of a generic bus (bottom)

Table 2.9. Capacitance extraction for a two-bit neighborhood

$d = a$				
$D$	$d$	$a$	$C_d$	$C_a$
$2\lambda$	$\lambda$	$\lambda$	$2C_D$	$2C_D$
$3\lambda$	$1.667\lambda$	$1.667\lambda$	$1.8C_D$	$1.8C_D$
$4\lambda$	$2.334\lambda$	$2.334\lambda$	$1.713C_D$	$1.713C_D$
$5\lambda$	$3\lambda$	$3\lambda$	$1.667C_D$	$1.667C_D$
$6\lambda$	$3.667\lambda$	$3.667\lambda$	$1.636C_D$	$1.636C_D$

Table 2.10. Capacitance extraction for a two-bit neighborhood

$d = 3a$				
$D$	$d$	$a$	$C_d$	$C_a$
$3\lambda$	$3\lambda$	$\lambda$	$C_D$	$3C_D$
$4\lambda$	$4.2\lambda$	$1.4\lambda$	$0.95C_D$	$2.85C_D$
$5\lambda$	$5.4\lambda$	$1.8\lambda$	$0.92C_D$	$2.78C_D$
$6\lambda$	$6.6\lambda$	$2.2\lambda$	$0.90C_D$	$2.72C_D$

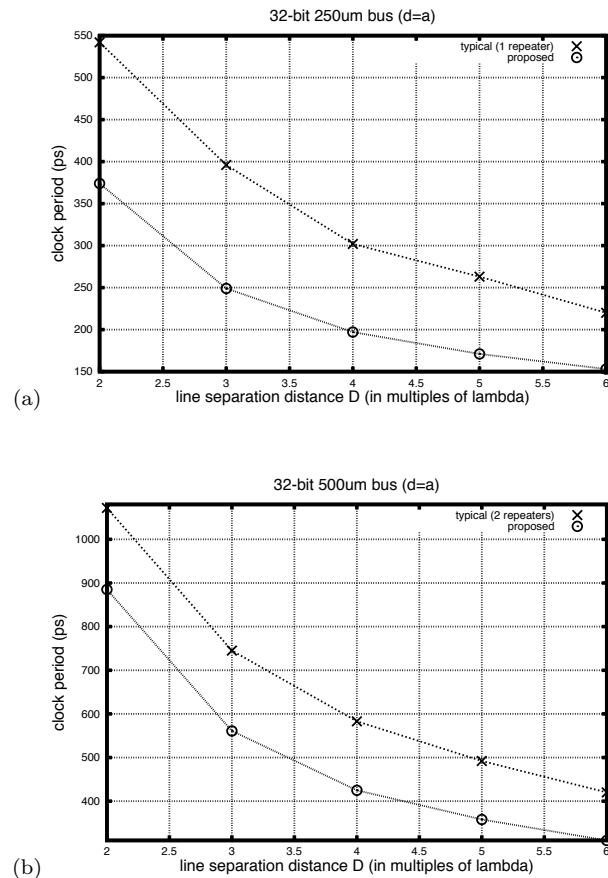


Figure 2.30. Clock period for a 32-bit bus of (a)  $250\mu m$  and (b)  $500\mu m$  length with the proposed two-bit isolated neighborhoods (three-threshold logic) using  $d = a$  (IBM 65nm)

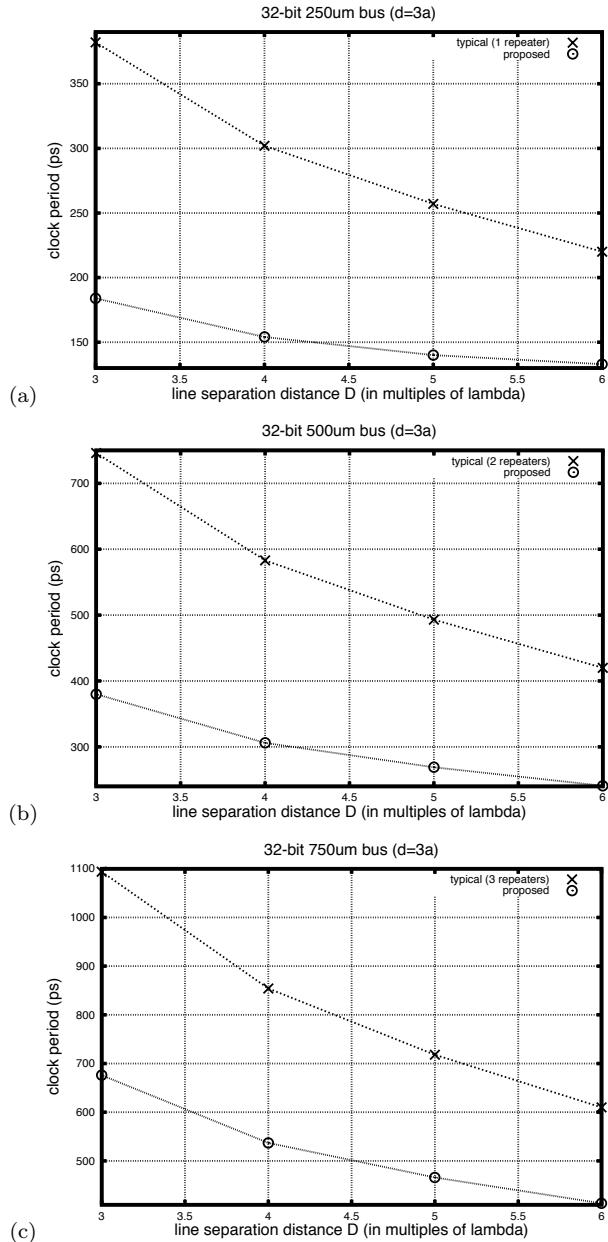


Figure 2.31. Clock period for 32-bit bus of (a)  $250\mu m$ , (b)  $500\mu m$  and (c)  $750\mu m$  length with the proposed two-bit isolated neighborhoods (three-threshold logic) using  $d = 3a$  (IBM 65nm)

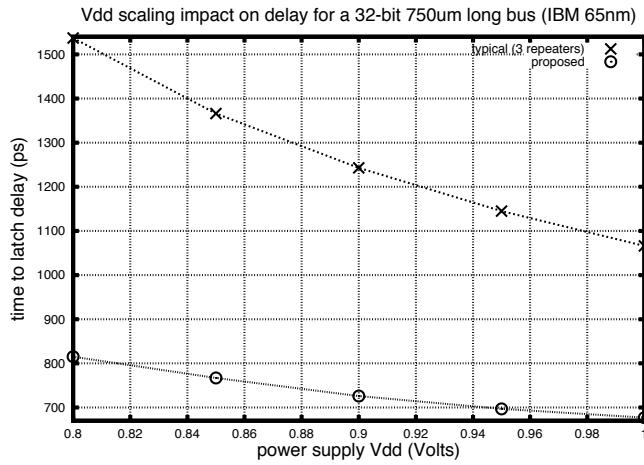


Figure 2.32. Delay change with Vdd scaling for a 32-bit  $750\mu m$  long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using  $D = 3\lambda$  and  $d = 3\lambda = 3a$  (IBM 65nm)

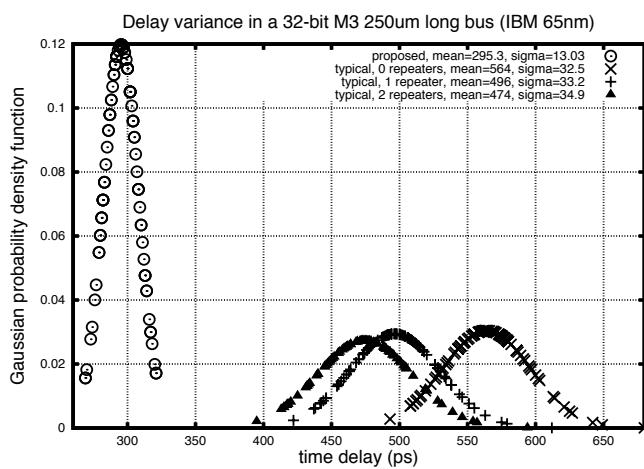


Figure 2.33. Delay variance for a 32-bit  $250\mu m$  long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using  $D = 3\lambda$  and  $d = 3\lambda = 3a$  (IBM 65nm)

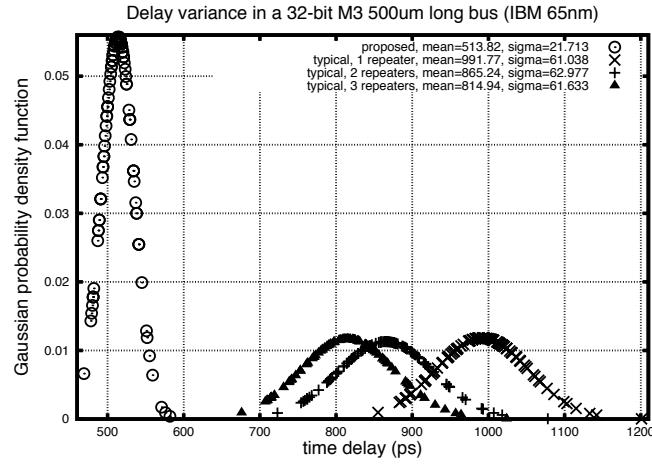


Figure 2.34. Delay variance for a 32-bit  $500\mu m$  long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using  $D = 3\lambda$  and  $d = 3\lambda = 3a$  (IBM 65nm)

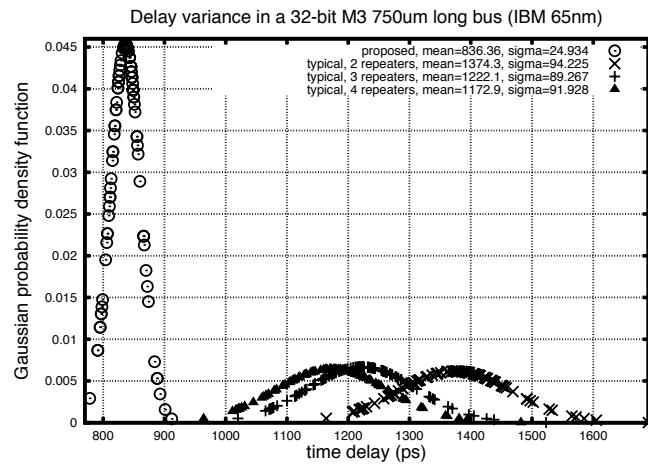
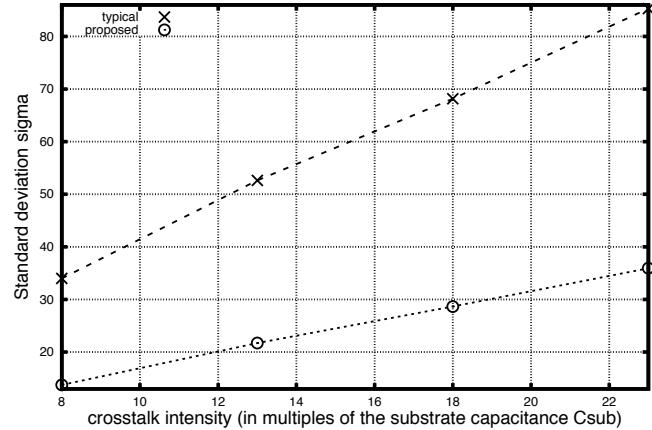


Figure 2.35. Delay variance for a 32-bit  $750\mu m$  long bus with the proposed two-bit isolated neighborhoods (three-threshold logic) using  $D = 3\lambda$  and  $d = 3\lambda = 3a$  (IBM 65nm)

Coupling intensity impact on delay deviation for a 32-bit 500um long bus (IBM 65nm)



Wire length impact on delay deviation for a 32-bit 500um long bus (IBM 65nm)

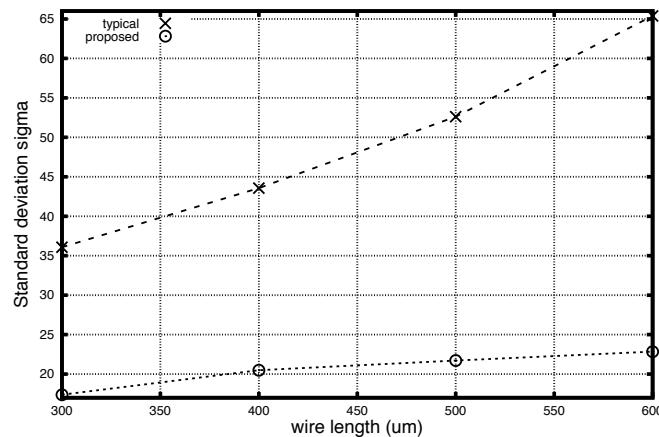


Figure 2.36. Effects of coupling intensity and wire resistance on delay deviation for a 32-bit  $500\mu m$  long bus ( $d = 3a$ ) with the proposed two-bit isolated neighborhoods (three-threshold logic) without any repeaters for the typical method (IBM 65nm)

## CHAPTER 3

### ERROR DETECTION ON BUSES WITH MULTI-THRESHOLD RECEIVING LOGIC

Increasing complexity in SoC applications undermines fast and reliable communication between the cores. Electrical interference, reduced power supply, small noise margins, shrunk internal node capacitance, significant wire resistance and cosmic radiation, among others, complicate the data transfer and frequently cause circuit failures on signal lines.

Thus, on-chip interconnects have become a serious bottleneck for designing and delivering reliable and fast systems [1, 2, 3]. To cope with this, encoding algorithms have been proposed to alleviate signal lines from coupling interference and maintain their high performance [20, 21, 22, 40, 23, 24]. In addition, in order to secure failure free operation, research emphasis is given to the on-line detection of errors with or without error recovery arrangements [13, 63, 65, 66, 67, 69, 70] and with error correcting codes for complete error tolerance [63, 64, 68].

In this chapter, we propose an architecture that allows for data reading with error detection capability on lines which are likely to operate under a potentially wide range of capacitive coupling. The approach is a modification of the architecture of the previous chapter so that it accommodates error detection. Thus, multiple *reference* or *threshold* voltages in the receiving logic of the lines are considered instead of typically one. However, the proposed technique utilizes the additional reference voltages to evaluate whether an error has occurred during the capture of the data. Some additional combinational logic is introduced on the receiver side to accomplish this task.

The mechanism is initially illustrated on a line with one adjacent aggressor. Subsequently, the case of a line with two adjacent aggressors is discussed and it is

shown how to generalize the technique for wide buses.

Detection circuitry on data buses for corrupted data is necessary since errors relating to unaccounted noise and other random events cannot be practically simulated and incorporated in the design phase of data transmission and capture methods. The proposed on-line detection block collects initial state and current-cycle voltage information from the victim line, as well as from its neighboring aggressor(s), and identifies cases in which an erroneous value is sensed at the receiver-end of the wires. In this work the efficiency of the error detection mechanism is evaluated for both single and multiple error occurrences.

A preliminary exploration of buses that use more than one receiver threshold voltage was included in [43] and has been extended in the previous chapter. The primary objective in that work was to increase data throughput. However, the architecture in [43] suffers from limitations in the allowable electrical noise, the data sampling window duration and the tolerated crosstalk intensity. In this chapter, an improved bus architecture alleviated from such restrictions is used. The refined data sampling architecture will be briefly outlined in the following section since it follows the ideas of the previous chapter.

In this work, an *error* is defined as any inconsistency in the reported voltage amplitude during a transition on a line that indicates a violation of its corresponding threshold. This will be physically observed as a stretch of the line voltage across a usually adjacent range typically contained within two threshold voltages (*error adjacency* assumption). This assumption is reasonable and is justified by extensive HSPICE simulations. The errors to be detected are random, occasionally asymmetric (*i.e.*, each error may change the voltage characteristics on a single wire) and, even though they could, they are not always induced by crosstalk interference. Such errors can be generated by electric noise, single transients resulting from collisions of cosmic particles with semiconductor

material, or other random electrical events that may cause failure of the multi-threshold logic to interpret the transmitted information properly.

### 3.1 OVERVIEW OF A MULTI-THRESHOLD RECEIVER METHOD

#### 3.1.1 The fundamentals of the technique

In the following, a data capture method using more than one reference voltages on the receiver side is explained. Worst-case coupling between wires is assumed. Bus lines are modeled as RLC segments in HSPICE. For a victim line, all potential transitions with respect to the switching activity in the aggressor are considered. Since the objective in this section is to provide an overview of a data capturing method, an error-free operation is assumed in the following analysis. The multi-threshold methodology essentially is an expansion or generalization of the traditional single-threshold receiver approach.

Typically, a single receiver inverter of static threshold is used at the receiver end of buses. The characterization in Table 3.1 reflects this mode of operation. The notations  $R$ ,  $Q$  and  $F$  denote *rising* ( $low \rightarrow high$ ), *quiet* ( $low \rightarrow low$  or  $high \rightarrow high$ ) and *falling* ( $high \rightarrow low$ ) transitions respectively on the aggressor. An empty table entry implies that characterization has rendered this case infeasible. It is clear that regardless of the type of transition on the aggressor, a voltage above or below the threshold would translate to a correspondingly *high* or *low* boolean value. It is important to note that the insignificance of the transition on the aggressor for such a data capturing arrangement on the victim leads to the wires' initial conditions (previous logic states) not being taken into consideration.

As in the previous chapter, instead of using one common threshold voltage for all transitions on a coupled data line, several threshold voltages at the receiver-end may as well be used. Each of those thresholds is intended to be used for some appropriate and pre-determined cases only. This is justified and enforced

Table 3.1. Single-threshold characterization table for the victim with respect to the activity on the aggressor

Voltage Ranges	Previous State		Next State
	1	0	
$V > V_t$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V < V_t$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>

by the different electric responses of a line with respect to the activity on an aggressor, especially in the presence of strong coupling. As a result, multiple receiver threshold voltages can be employed and effectively used to read data transmitted along wires.

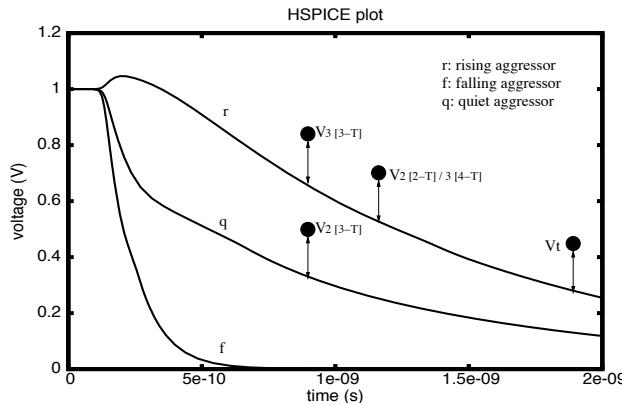


Figure 3.1. Falling (only) transitions for a victim with one adjacent aggressor (IBM 65nm)

For the special case of two-bit lines, a topology of lines having one adjacent aggressor, two, three or four total reference voltages may be on the receiver side. The notations  $V_j [2-T]$ , with  $j \in \{1, 2\}$ ,  $V_j [3-T]$ , with  $j \in \{1, 2, 3\}$  and  $V_j [4-T]$ , with  $j \in \{1, 2, 3, 4\}$ , are used to denote threshold voltages that belong to the 2-tuple,

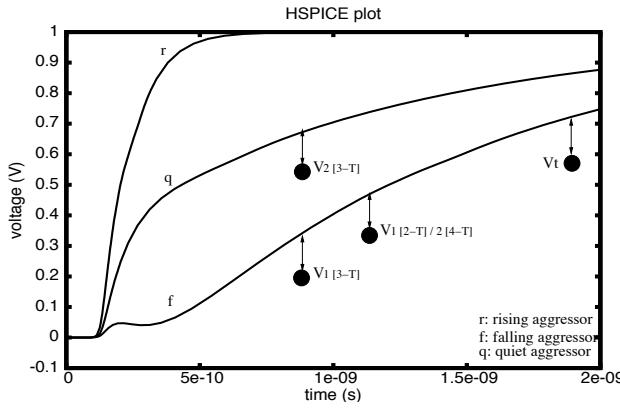


Figure 3.2. Rising (only) transitions for a victim with one adjacent aggressor (IBM 65nm)

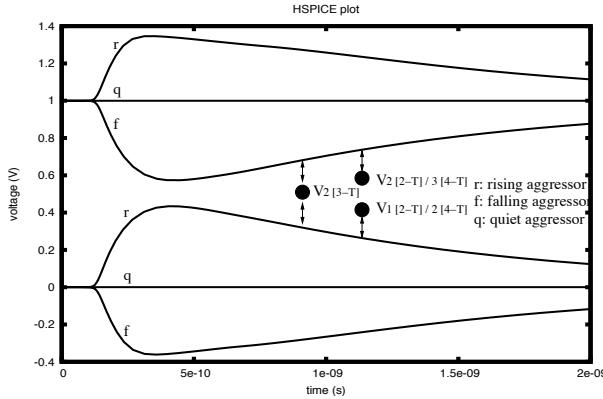


Figure 3.3. Voltage glitches (hazards) only for quiet transitions ( $0 \rightarrow 0$  or  $1 \rightarrow 1$ ) for a line with one adjacent aggressor(IBM 65nm)

3-tuple and 4-tuple of the reference voltages respectively. In particular for three thresholds, a low threshold voltage ( $V_1 [3-T]$  in Figure 3.2) is defined to identify the slowest rising transition. Similarly, a high threshold voltage ( $V_3 [3-T]$  in Figure 3.1) is meant for handling the slowest falling transition on the victim. A third reference voltage ( $V_2 [3-T]$  in Figure 3.3) is employed to distinguish between the appearing voltage glitches on the lines (rising and falling), and to distinguish each of these glitches from the unperturbed -due to a quiet aggressor- transitions (rising and falling respectively). Tables 3.2, 3.3 and 3.4 contain the electrical characterization

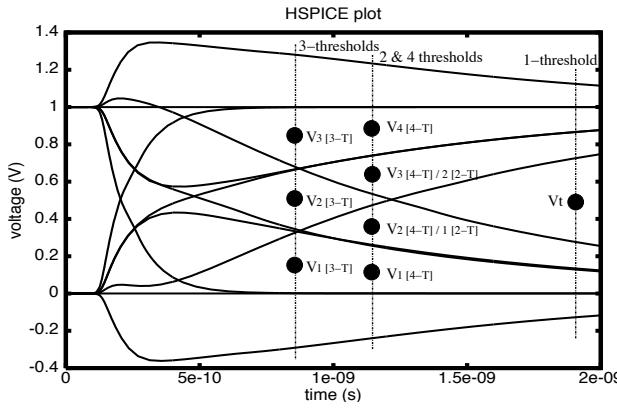


Figure 3.4. All (superimposed) transitions for a victim wire with one adjacent aggressor (IBM 65nm)

results of the victim for two, three and four thresholds respectively within its receiver logic. A characterization table is an alternative means of illustrating the lower and upper bounds of all transitions on a wire, subject to the influence of a concurrent transition on its aggressor.

Along the same guidelines, multiple thresholds can be defined for a victim line having two adjacent aggressors. The characterization of such lines reveal that an even larger number of thresholds can be defined compared to the previous case. In this chapter, although several other cases exist, four and five threshold voltages are utilized and evaluated (Figures 3.5, 3.6, 3.7 and 3.8). The notations  $V_j [4-T]$ , with  $j \in \{1, 2, 3, 4\}$ , and  $V_j [5-T]$ , with  $j \in \{1, 2, 3, 4, 5\}$ , are used to denote threshold voltages that belong to the 4-tuple and 5-tuple of the reference voltages respectively. Tables 3.5 and 3.6 describe the electrical characterization of the victim for four and five thresholds respectively. As before, the notations  $R$ ,  $Q$  and  $F$  denote *rising* ( $low \rightarrow high$ ), *quiet* ( $low \rightarrow low$  or  $high \rightarrow high$ ) and *falling* ( $high \rightarrow low$ ) transitions respectively on the aggressor. The arithmetic value attached to these transition relevant notations state the number of aggressors exhibiting the particular type of switching. An empty table entry implies that

Table 3.2. Two-threshold characterization table for the victim with respect to the activity on the aggressor

Voltage Ranges	Previous State		Next State
	1	0	
$V > V_2 [2-T]$	$R, Q, F$	$R, Q$	<b>1</b>
			<b>0</b>
$V > V_1 [2-T]$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V < V_2 [2-T]$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>
$V < V_1 [2-T]$			<b>1</b>
	$Q, F$	$R, Q, F$	<b>0</b>

HSPICE characterization has rendered this case infeasible.

For a better understanding of a characterization table, it is observed that rising transitions on a victim line are bounded from below and hence the voltage on that line can only be larger than ( $V >$ ) a threshold level that depends on the type of transition on the aggressor. Similarly, falling transitions on the victim are bounded only from above and therefore the voltage on the line can only be smaller than ( $V <$ ) a threshold that depends on the aggressor transition. In the same way, rising and falling voltage glitches are bounded from above and below respectively. Upon data capture completion, a line reset procedure is initiated to quickly bring the line voltage to a discrete logic state ( $V_{dd}$  or *ground*) before the release of the next data on the lines.

Table 3.3. Three-threshold characterization table for the victim with respect to the activity on the aggressor

Voltage Ranges	Previous State		Next State
	1	0	
$V > V_3 [3-T]$	$R, Q$	$R$	<b>1</b>
			<b>0</b>
$V > V_2 [3-T]$	$R, Q, F$	$R, Q$	<b>1</b>
			<b>0</b>
$V > V_1 [3-T]$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V < V_3 [3-T]$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>
$V < V_2 [3-T]$			<b>1</b>
	$Q, F$	$R, Q, F$	<b>0</b>
$V < V_1 [3-T]$			<b>1</b>
	$F$	$Q, F$	<b>0</b>

Table 3.4. Four-threshold characterization table for a victim (with one adjacent aggressor) with respect to the activity on its only aggressor

Voltage Ranges	Previous State		Next State
	1	0	
$V > V_4 [4-T]$	$R, Q$	$R$	<b>1</b>
			<b>0</b>
$V > V_3 [4-T]$	$R, Q, F$	$R, Q$	<b>1</b>
			<b>0</b>
$V > V_2 [4-T]$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V > V_1 [4-T]$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V < V_4 [4-T]$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>
$V < V_3 [4-T]$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>
$V < V_2 [4-T]$			<b>1</b>
	$Q, F$	$R, Q, F$	<b>0</b>
$V < V_1 [4-T]$			<b>1</b>
	$F$	$Q, F$	<b>0</b>

Table 3.5. Four-threshold characterization table for the victim with respect to the activity on the two aggressors

Voltage Ranges	Previous State		Next State
	1	0	
$V > V_4 [4-T]$	$2R, 1R, Q, 1F$	$2R, 1R$	<b>1</b>
			<b>0</b>
$V > V_3 [4-T]$	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q$	<b>1</b>
			<b>0</b>
$V > V_2 [4-T]$	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q, 1F$	<b>1</b>
			<b>0</b>
$V > V_1 [4-T]$	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>1</b>
			<b>0</b>
$V < V_4 [4-T]$			<b>1</b>
	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>0</b>
$V < V_3 [4-T]$			<b>1</b>
	$1R, Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>0</b>
$V < V_2 [4-T]$			<b>1</b>
	$Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>0</b>
$V < V_1 [4-T]$			<b>1</b>
	$1F, 2F$	$1R, Q, 1F, 2F$	<b>0</b>

Table 3.6. Five-threshold characterization table for the victim with respect to the activity on the two aggressors

Voltage Ranges	Previous State		Next State
	<b>1</b>	<b>0</b>	
$V > V_5 [5-T]$	$2R, 1R, Q$	$2R$	<b>1</b>
			<b>0</b>
$V > V_4 [5-T]$	$2R, 1R, Q, 1F$	$2R, 1R$	<b>1</b>
			<b>0</b>
$V > V_3 [5-T]$	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q$	<b>1</b>
			<b>0</b>
$V > V_2 [5-T]$	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q, 1F$	<b>1</b>
			<b>0</b>
$V > V_1 [5-T]$	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>1</b>
			<b>0</b>
$V < V_5 [5-T]$			<b>1</b>
	$2R, 1R, Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>0</b>
$V < V_4 [5-T]$			<b>1</b>
	$1R, Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>0</b>
$V < V_3 [5-T]$			<b>1</b>
	$Q, 1F, 2F$	$2R, 1R, Q, 1F, 2F$	<b>0</b>
$V < V_2 [5-T]$			<b>1</b>
	$1F, 2F$	$1R, Q, 1F, 2F$	<b>0</b>
$V < V_1 [5-T]$			<b>1</b>
	$2F$	$Q, 1F, 2F$	<b>0</b>

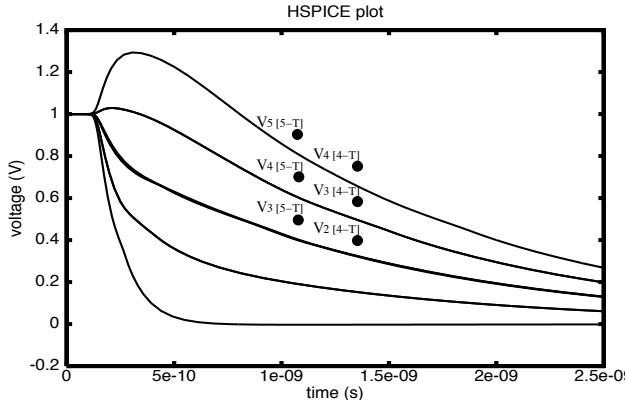


Figure 3.5. Falling (only) transitions for a victim with two adjacent aggressors (IBM 65nm)

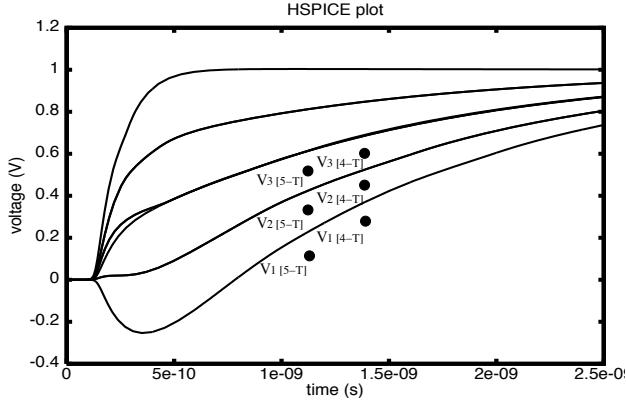


Figure 3.6. Rising (only) transitions for a victim with two adjacent aggressors (IBM 65nm)

## 3.2 ON-LINE DETECTION OF ERRORS FOR LINES WITH ONE ADJACENT AGGRESSOR

### 3.2.1 Overview

It was shown previously how the multi-threshold receiver logic accommodates reading data by utilizing additional reference voltages and stored binary values representing the previous states on the wires. The objective of this section is to capitalize on existing features of the methodology for online error

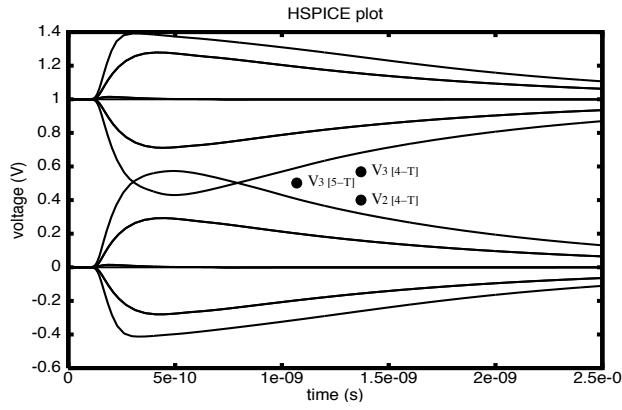


Figure 3.7. Voltage glitches (hazards) only for a victim with two adjacent aggressors (IBM 65nm)

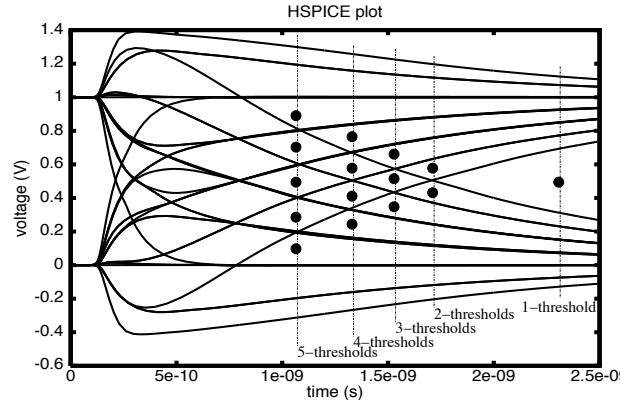


Figure 3.8. All (superimposed) transitions for a victim with two adjacent aggressors (IBM 65nm)

detection. Specifically, the characterized voltage ranges under the error-free (and for a wide range of coupling) operation of the wires can be used to detect the presence of an intermittent error. For two coupled isolated data lines (Figure 3.9), at least two distinctive receiver thresholds are required for basic detection capability (single occurrence detection). In general for a data bus, a minimum of  $(\max. \text{ number of aggressors in the topology per line}) \times 2$  threshold voltages are needed for single error detection.

The intensity of the coupling between the two lines will determine, as far as

the wire voltage is concerned, how far above or below the corresponding receiver-thresholds the error will essentially be. The symmetry and mutual dependence of the two lines derived from the topology allows for defining a correlation between the interconnects. This correlation partitions and narrows down the single-bounded error-free ranges into confined interdependent sub-ranges. As a result, a handful of distinctive error-free patterns evolve for each initial state of the bus and hence a error detection mechanism can be subsequently developed.

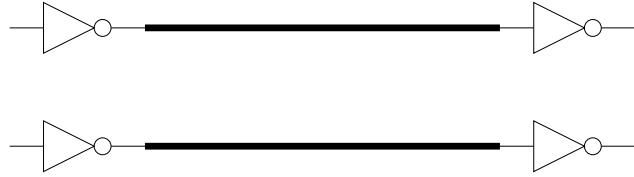


Figure 3.9. Example of bus line with one adjacent aggressor

### 3.2.2 Error-free characterization of lines

In the following, it is assumed that three reference voltages are available on the receiver side but the same principles apply for two threshold voltages as well. The two lines of the topology will be referred to as  $l_1$  and  $l_2$ . Given that the initial conditions of the wires define the sub-set of potential transitions in the next clock cycle, pairs of legitimate voltage ranges (each pair corresponds to a likely transition case) can be mapped to the initial states of the two lines.

For instance, when both lines are discharged ( $l_1 \rightarrow low$ ,  $l_2 \rightarrow low$ ) before the new data are transmitted, only rising or quiet transitions can occur on the wires. As a consequence, four likely transition scenarios exist: (a) both  $l_1$  and  $l_2$  stay quiet ( $low \rightarrow low$ ), (b)  $l_1$  is quiet ( $low \rightarrow low$ ) while  $l_2$  rises ( $low \rightarrow high$ ), (c)  $l_1$  experiences a rising transition ( $low \rightarrow high$ ) whereas  $l_2$  stays quiet ( $low \rightarrow low$ ) and (d) both lines undergo rising transitions ( $low \rightarrow high$ ).

When instead, line  $l_1$  is initially discharged and line  $l_2$  is fully charged

$(l_1 \rightarrow low, l_2 \rightarrow high)$ , the likely next cycle transitions are: (e)  $l_1$  stays quiet ( $low \rightarrow low$ ) while  $l_2$  has a falling transition ( $high \rightarrow low$ ), (f) both remain quiet ( $low \rightarrow low$  and  $high \rightarrow high$ ), (g)  $l_1$  has a rising ( $low \rightarrow high$ ) and  $l_2$  has a falling ( $high \rightarrow low$ ) transition and (h)  $l_1$  undergoes a rising ( $low \rightarrow high$ ) transition while  $l_2$  remains quiet ( $high \rightarrow high$ ).

To find the expected range, the intended transition on the victim is resolved with respect to the switching activity of the aggressor line (Table 3.3). A falling transition on the victim will be bounded from above and a rising transition will be bounded from below. Similarly, rising and falling hazards will be bounded from top and bottom respectively. Considering the type of transition on the aggressor, the overlap of the legitimate voltage ranges on the victim line is identified as the expected range. Determining the overlap of the characterized voltage ranges guarantees that the tightest lower or upper bound is presumed for a given transition. Since the data is obtained from actual electrical simulations, the mutual coupling effects of the victim line on its aggressor are already incorporated in the analysis.

For illustration purposes, assume a rising ( $low \rightarrow high$ ) transition in the victim, while the aggressor line stays quiet. From Table 3.3 it is obtained that a rising transition on the victim, while the aggressor remains quiet, yields the following legitimate voltage ranges: (i)  $V_{victim} > V_1|_{[3-T]}$  and (ii)  $V_{victim} > V_2|_{[3-T]}$  (rising transition bounded from below). Since  $V_2|_{[3-T]} > V_1|_{[3-T]}$ , it is safe to eliminate case (i) and conclude that the anticipated voltage range is  $V_{victim} > V_2|_{[3-T]}$ . Using similar considerations, the error-free voltage ranges are summarized in Table 3.7. In particular, for the previously defined cases we have the following analysis:

- (a)  $l_1: Q, l_2: Q \Rightarrow V_{l1} < V_1|_{[3-T]}, V_{l2} < V_1|_{[3-T]}$

- (b)  $l_1: Q, l_2: R \Rightarrow V_{l1} < V_{l2} [3-T], V_{l2} > V_{l1} [3-T]$
- (c)  $l_1: R, l_2: Q \Rightarrow V_{l1} > V_{l2} [3-T], V_{l2} < V_{l1} [3-T]$
- (d)  $l_1: R, l_2: R \Rightarrow V_{l1} > V_{l2} [3-T], V_{l2} > V_{l1} [3-T]$
- (e)  $l_1: Q, l_2: F \Rightarrow V_{l1} < V_{l2} [3-T], V_{l2} < V_{l1} [3-T]$
- (f)  $l_1: Q, l_2: Q \Rightarrow V_{l1} < V_{l2} [3-T], V_{l2} > V_{l1} [3-T]$
- (g)  $l_1: R, l_2: F \Rightarrow V_{l1} > V_{l2} [3-T], V_{l2} < V_{l1} [3-T]$
- (h)  $l_1: R, l_2: Q \Rightarrow V_{l1} > V_{l2} [3-T], V_{l2} > V_{l1} [3-T]$

The previous conclusions on the error-free ranges along with the remaining ones for all possible initial conditions (*i.e.*, cases (i) to (p)) are labeled in Table 3.7 and are also depicted graphically in Figure 3.10.

### 3.2.3 Correlation of coupled lines

The previous characterization of the error free ranges was performed for a wide range of coupling intensity varying from practically nonexistent to worst-case. It is obvious that because of symmetry and mutual dependence of the coupled lines, a potential fluctuation of the coupling intensity (a realistic possibility for post-fabricated integrated circuits) would affect the coupled lines equally. For instance, a smaller or larger coupling between a line experiencing a rising transition and one that stays quiet at *low* (*i.e.*, case (c) in Figure 3.10) would cause a faster or slower transition on the first line and a glitch (or else *hazard*) of smaller or larger magnitude on the second one respectively. As a consequence, the voltage values observed on the two lines will move closer or further apart more or less by the same amount. Similarly, when both lines experience opposing transitions (*i.e.*, cases (g) and (l) in Figure 3.10).

Therefore, it is possible to define discretized correlations between smaller intervals (designated by the existing receiver thresholds) within the error-free ranges that will still reflect error-free behavior. Such a granularization of the long ranges into shorter sub-ranges makes possible to capture the anticipated unerring operation of the bus utilizing tighter bounds. Any potential variation in the coupling would not endanger the correctness of the scheme as long as the intensity level does not exceed the worst-case coupling used during characterization. The tight bounds do not introduce any restrictions or constraints, since under error-free operational conditions there will always be a pair of correlated sub-ranges rendering the reported data valid and acceptable.

For illustration purposes, the correlated sub-ranges are validated with HSPICE and are provided in Figure 3.11. In the latter, two sub-ranges having the same label are correlated. This entails that when the reported pair of sub-ranges for the two (victim and aggressor) lines matches exactly one acceptable case of correlated sub-ranges for the particular initial state, then a reading operation can be done. The next states of the two lines are derived from the particular matching pair. For instance, for initial conditions  $l_1 \rightarrow low$  and  $l_2 \rightarrow high$ , assume that the reported pair of sub-ranges is  $V_{1 [3-T]} < V_{l1} < V_{2 [3-T]}$  and  $V_{2 [3-T]} < V_{l2} < V_{3 [3-T]}$ . This is case (g) in Figure 3.11 and hence the obtained next states are  $l_1 \rightarrow high$  and  $l_2 \rightarrow low$ .

### 3.2.4 On-line detection of single error occurrences

When a hazardous discrepancy or perturbation occurs, this leads to one of the lines reporting an invalid sub-range. Having already established those pairs of correlated sub-ranges that are legitimate under the given initial conditions, the error can be therefore immediately detected.

First, the detection methodology is assessed for the particular initial state in

which both lines are discharged. As explained and illustrated before (Figure 3.11 (a), (b), (c) and (d)), there are four potential scenarios for the next clock cycle. Assuming the case in Figure 3.11-(a) to be the intended transition (both lines remain discharged with their voltage below  $V_1 [3-T]$ ), two potential error cases are defined that need to be detected. These occur when the sensed voltage for one of the two lines belongs to the top adjacent range ( $V_1 [3-T] < V_{l1} < V_2 [3-T]$  or  $V_1 [3-T] < V_{l2} < V_2 [3-T]$  shown in Figures 3.12-(a1) and 3.12-(a2) respectively). In the same figures the correlated sub-ranges are also seen. The sensed erroneous voltage (on either line) is compared against all the legitimate error cases (Figure 3.11) in order to determine a match. For instance, assuming that the captured pair of ranges is  $V_1 [3-T] < V_{l1} < V_2 [3-T]$  and  $V_{l2} < V_1 [3-T]$ , there is no case in Figure 3.11 that is in agreement with the reported pair. Hence, the measured duplet is labeled erroneous and the occurrence is immediately detected. A similarly approach applies to these errors that belong to the remaining cases, as illustrated in Figures 3.12 and 3.13.

Next we consider the initial state where one line is initially discharged and the other one is fully charged (Figure 3.11 (e), (f), (g) and (h)). A similar approach applies. Assuming the case in Figure 3.11-(e) to be the intended transition (*i.e.*, the next logic state for both lines is *low*), we define two potential errors to be detected. These occur when the sensed voltage belongs to the top adjacent ranges for one of the two lines ( $V_1 [3-T] < V_{l1} < V_2 [3-T]$  or  $V_2 [3-T] < V_{l2} < V_3 [3-T]$  shown in Figures 3.14-(e1) and 3.14-(e2) respectively). For illustration purposes, it is presumed that the erroneous pair of sensed voltages at the receiver-end of the lines is  $V_1 [3-T] < V_{l1} < V_2 [3-T]$  and  $V_{l2} < V_2 [3-T]$  (*i.e.*, line  $l_1$  carries a error as shown in Figure 3.14-(e1)). This pair of voltages is compared against the error-free sub-range pairs (encapsulated in Figure 3.11) in order to determine a match. As before, no match is identified and hence the error is

detected. A similar approach applies to those error occurrences that belong to the remaining cases illustrated in Figures 3.14 and 3.15.

### 3.2.5 On-line detection of multiple error occurrences

The previous three-threshold logic is modified to detect multiple error occurrences. Two different orthogonal approaches can be employed to cope with multiple errors: *(i)* introduce clock delay, *(ii)* use more threshold voltages. Both enhance significantly the detection capability of the method in terms of both magnitude and number of error occurrences. The more threshold voltages are utilized, the higher the amplitude of a detectable error can be. Thus the approach will tolerate a higher error effect. Similarly, the more clock delay is introduced, the more simultaneous errors can be detected. The previous chapter demonstrated that a multi-threshold receiving logic results in a faster arrangement than the typical single-threshold receivers. As a consequence, the delay gained by using multiple thresholds can be traded off in order to harden data lines with error tolerance. These observations hold for a larger number of aggressors per victim line as well.

To illustrate the above premises on the two lines, assume that under the initial conditions  $l_1 \rightarrow low$  and  $l_2 \rightarrow high$ , errors occur simultaneously on both wires. Without the loss of generality, it is considered that those are the ones that are separately depicted in Figures 3.14-(e1) and 3.14-(e2). Since the measured voltages for the lines are  $V_1 [3-T] < V_{l1} < V_2 [3-T]$  and  $V_2 [3-T] < V_{l2} < V_3 [3-T]$ , this appears to be the legitimate case for which the two lines perform opposite polarity transitions (shown in Figure 3.11-(g)) and the errors can therefore be masked. However, if the data read time is appropriately delayed, then the resulting error-free characterization for the new capture time instant (Figure 3.16) does not recognize the reported voltage duplet as valid. An exhaustive examination of all remaining cases yields that analogous clock delay resolves potential conflicts or

maskings of multiple errors and hence detection of such occurrences can be accomplished in a straightforward fashion.

A similar analysis can be performed when an additional threshold voltage is employed at the receiver-end of the lines. When the same errors occur simultaneously on the two lines as before ( $V_1 [3-T] < V_{l1} < V_2 [3-T]$  and  $V_2 [3-T] < V_{l2} < V_3 [3-T]$ ), these will also be detected under a four-threshold receiving logic characterization (Figure 3.17-(g)). One may argue that the introduction of an added threshold voltage partitions the voltage axis in more voltage ranges, reducing thus the span of each range when compared to a case of less thresholds. However, analysis actually shows that a four-threshold logic can also detect a single error of double-range magnitude, and as a consequence the operational margin of such an architecture is rather enhanced. The conclusions from the above analysis on single and multiple erroneous occurrences are summarized in Table 3.8.

### 3.3 ON-LINE ERROR DETECTION FOR A LINE WITH TWO ADJACENT AGGRESSORS

#### 3.3.1 Preliminaries

When a victim line has two adjacent aggressors (*i.e.*, as in the three-bit bus of Figure 3.18), the inherent asymmetry of the topology prevents us from defining correlations among the different error-free sub-ranges for error detection purposes. This is because each voltage sub-range for a line with one aggressor (as the edge lines of a three-bit bus are) cannot be strictly and always correlated with a voltage sub-range for a line with two adjacent aggressors (*i.e.*, middle line of a three-bit bus). The same amount of variance in the intensity of the post-fabrication coupling would cause disproportionate voltage shifts to the aforementioned

(middle and edge) wires and hence such correlation cannot be safely defined. As before, characterized voltage ranges under error-free (and for a wide range of coupling) operation of lines provide the valid voltage patterns for each initial state of the data bus. These can ultimately be used for matching purposes in order to detect the presence of an error.

The characterization tables resulting from an HSPICE-driven analysis are used to extract and depict more clearly the patterns of the error-free ranges for each set of initial states. This information (that is specifically contained in Tables 3.4, 3.5, 3.6 in this chapter as well as in Table 5.8 in the Appendix) can be easily used to produce the anticipated electrical behavior of the lines (Figures 3.19 in this chapter and in Figures 5.9, 5.10, and 5.11 in the Appendix using four thresholds). The appropriate lower and upper bounds are determined based on the intended transition of the victim and the type of switching occurring on the aggressor(s). Once all error-free ranges are characterized, they are used to evaluate the potential and the efficiency of the detection method for different kinds of potential errors. The lines of the topology will be referred to as  $l_1$ ,  $l_2$  and  $l_3$ .

### 3.3.2 On-line detection of single error occurrences

In the presence of a single erroneous occurrence during data capture in the examined bus of lines, matching against pre-characterized error-free wire electric behavior (under the given previous states) can reveal the existence of the error. Four-thresholds will be used for demonstration but a similar approach applies for five-threshold receivers. Due to the larger number of lines involved, more candidate next states exist. For instance, when the initial conditions of the lines in such a topology are  $l_1 \rightarrow low$ ,  $l_2 \rightarrow high$  and  $l_3 \rightarrow low$ , eight different possibilities exist as far as the next state is concerned (displayed in Figure 3.19).

Upon data capture, a voltage triplet is reported to the error detection

module. If any of the eight candidate error-free next states matches the reported set of data, then the received information is labeled as uncorrupted and the next states are immediately retrieved from the matching case. If instead no match is found, an error is detected and the received data are disregarded. To illustrate the previous decision flow, assume the previous initial conditions on the lines and that the intended next states are all *low*. Furthermore, it is taken that the read voltage information is  $V_1 [4-T] < V_{l1} < V_2 [4-T]$ ,  $V_1 [4-T] < V_{l2} < V_2 [4-T]$  and  $V_{l3} < V_1 [4-T]$  (*i.e.*, line  $l_1$  carries an error). Comparing these against the error-free ranges in Figure 3.19, one can immediately see that since  $V_1 [4-T] < V_{l1} < V_2 [4-T]$  the reported voltages are disqualified from further consideration and so the error is detected.

### 3.3.3 On-line detection of multiple error occurrences

In the realistic case that more than a single error occurs, the multi-threshold logic can still adapt and provide tolerance to such events. As previously, two separate approaches can be employed to handle multiple errors: (*i*) introduce clock delay, (*ii*) use a higher number of threshold voltages (subject to an upper limit). These modifications are able to increase considerably the detection capability of the method in terms of both magnitude and number of occurrences. More threshold voltages can detect higher error effects (errors of higher voltage amplitude), whereas introducing clock delay leads to the identification of multiple simultaneous errors. In the following, error detection is discussed when a delayed four-threshold and a delayed five-threshold schemes are utilized.

For initial conditions  $l_1 \rightarrow low$ ,  $l_2 \rightarrow high$ ,  $l_3 \rightarrow low$  and presuming a four-threshold based receiving logic, the ideally anticipated voltage values on the lines for all possible next states -after delaying the clock- are expressed in ranges and displayed in Figure 3.20. If all lines prove to be vulnerable to errors (as

portrayed in Figure 3.20), then the erroneous voltage triplet

$V_2 [4-T] < V_{l1} < V_3 [4-T]$ ,  $V_3 [4-T] < V_{l2} < V_4 [4-T]$ ,  $V_2 [4-T] < V_{l3} < V_3 [4-T]$  will be identified as corrupted. This is seen in the same figure where even though the errors on the edge lines match with other valid next states and are masked, the middle one does not and this essentially leads to the identification of the error. Note that under an un-delayed four-threshold logic, all of the previous errors are masked (refer to Figure 3.19). A similar analysis follows in the case of a five-threshold based receiver circuit (Figure 3.20).

In a realistic wide data bus that is traditionally implemented as a co-planar parallel line topology, every wire (besides the top and bottom ones) has two adjacent aggressors. The distant aggressors may contribute to the overall crosstalk imposed on a victim line, but this contribution can be effectively offset by introducing a very small delay on the clock. As a result, the previously explained methodology can be implemented in triplets of wires and thus provide credible error protection for a wide bus.

### 3.4 EXPERIMENTAL RESULTS AND CONCLUSIONS

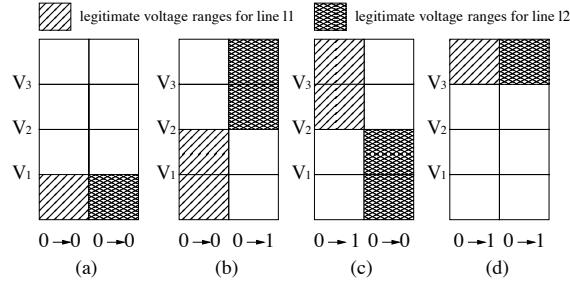
The method has been implemented and evaluated using HSPICE. The main focus is to determine delay characteristics versus an increased error detection capability. For this purpose, we experimented with a three-bit bus in which any adjacent-range error(s) may occur. A five-threshold logic is presumed for this topology.

The experiments have shown a delay tradeoff (compared to the delay optimal five-threshold setup) from 13% to 28% for lines of length between  $200\mu m$  and  $1000\mu m$  (Figure 3.21). Such delay penalty can be mitigated with the insertion of repeaters or through appropriate driver scaling. Current investigation objectives are error recovery mechanisms when single or multiple errors are present.

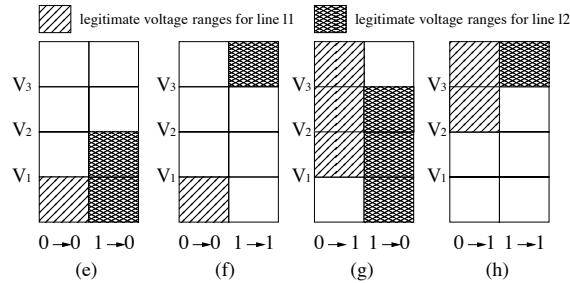
Table 3.7. Error-free voltage ranges for a line with one adjacent aggressor using three-threshold receiving logic

$l_{1(previous)}$	$l_{2(previous)}$	Valid voltage ranges ( $l_1, l_2$ )
<b>0</b>	<b>0</b>	<ul style="list-style-type: none"> <li>(a) <math>V_{l1} &lt; V_1 [3-T], V_{l2} &lt; V_1 [3-T]</math> or</li> <li>(b) <math>V_{l1} &lt; V_2 [3-T], V_{l2} &gt; V_2 [3-T]</math> or</li> <li>(c) <math>V_{l1} &gt; V_2 [3-T], V_{l2} &lt; V_2 [3-T]</math> or</li> <li>(d) <math>V_{l1} &gt; V_3 [3-T], V_{l2} &gt; V_3 [3-T]</math></li> </ul>
<b>0</b>	<b>1</b>	<ul style="list-style-type: none"> <li>(e) <math>V_{l1} &lt; V_1 [3-T], V_{l2} &lt; V_2 [3-T]</math> or</li> <li>(f) <math>V_{l1} &lt; V_1 [3-T], V_{l2} &gt; V_3 [3-T]</math> or</li> <li>(g) <math>V_{l1} &gt; V_1 [3-T], V_{l2} &lt; V_3 [3-T]</math> or</li> <li>(h) <math>V_{l1} &gt; V_2 [3-T], V_{l2} &gt; V_3 [3-T]</math></li> </ul>
<b>1</b>	<b>0</b>	<ul style="list-style-type: none"> <li>(i) <math>V_{l1} &gt; V_3 [3-T], V_{l2} &lt; V_1 [3-T]</math> or</li> <li>(j) <math>V_{l1} &gt; V_3 [3-T], V_{l2} &gt; V_2 [3-T]</math> or</li> <li>(k) <math>V_{l1} &lt; V_2 [3-T], V_{l2} &lt; V_1 [3-T]</math> or</li> <li>(l) <math>V_{l1} &lt; V_3 [3-T], V_{l2} &gt; V_1 [3-T]</math></li> </ul>
<b>1</b>	<b>1</b>	<ul style="list-style-type: none"> <li>(m) <math>V_{l1} &gt; V_2 [3-T], V_{l2} &lt; V_2 [3-T]</math> or</li> <li>(n) <math>V_{l1} &gt; V_3 [3-T], V_{l2} &gt; V_3 [3-T]</math> or</li> <li>(o) <math>V_{l1} &lt; V_1 [3-T], V_{l2} &lt; V_1 [3-T]</math> or</li> <li>(p) <math>V_{l1} &lt; V_2 [3-T], V_{l2} &gt; V_2 [3-T]</math></li> </ul>

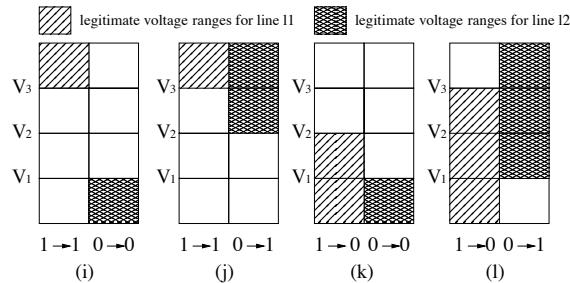
Initial Conditions:  $(l_1, l_2) = (0, 0)$



Initial Conditions:  $(l_1, l_2) = (0, 1)$



Initial Conditions:  $(l_1, l_2) = (1, 0)$



Initial Conditions:  $(l_1, l_2) = (1, 1)$

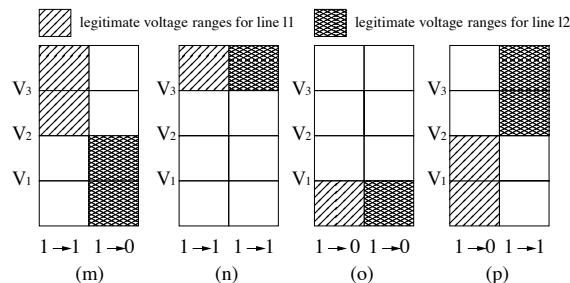
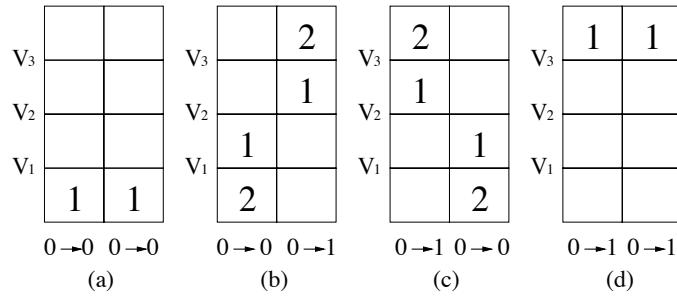
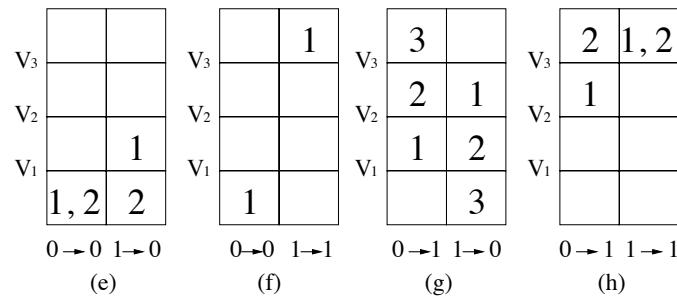


Figure 3.10. Error-free characterization of ranges for all possible initial conditions of a line with one adjacent aggressor

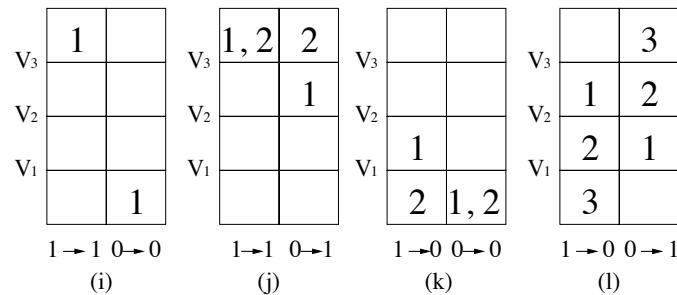
Initial Conditions:  $(l_1, l_2) = (0, 0)$



Initial Conditions:  $(l_1, l_2) = (0, 1)$



Initial Conditions:  $(l_1, l_2) = (1, 0)$



Initial Conditions:  $(l_1, l_2) = (1, 1)$

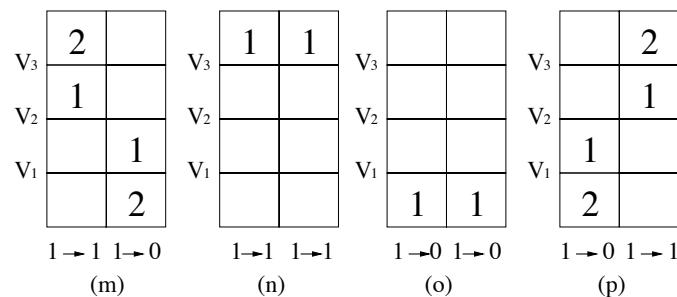


Figure 3.11. Correlated sub-ranges within the error-free ranges for all possible initial conditions on a line with one adjacent aggressor

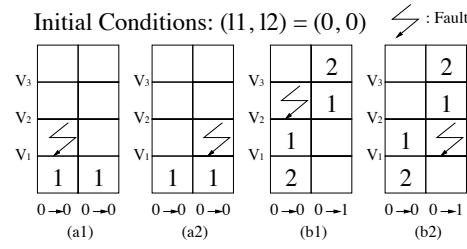


Figure 3.12. Likely errors detected in the topology of a line with one adjacent aggressor

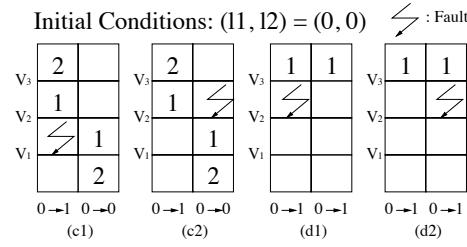


Figure 3.13. Likely errors detected in the topology of a line with one adjacent aggressor

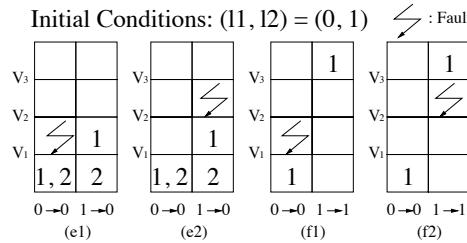


Figure 3.14. Likely errors detected in the topology of a line with one adjacent aggressor

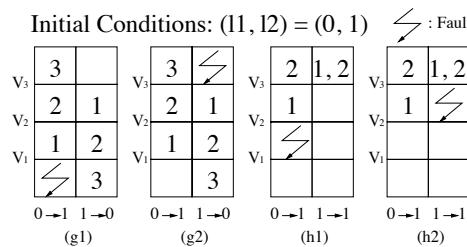


Figure 3.15. Likely errors detected in the topology of a line with one adjacent aggressor

Initial Conditions:  $(l_1, l_2) = (0, 1)$

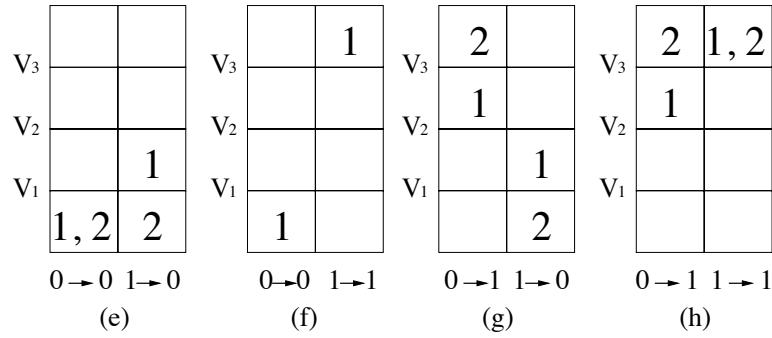


Figure 3.16. Correlated sub-ranges for a delayed three-threshold logic in a topology of lines with one adjacent aggressor

Initial Conditions:  $(l_1, l_2) = (0, 1)$

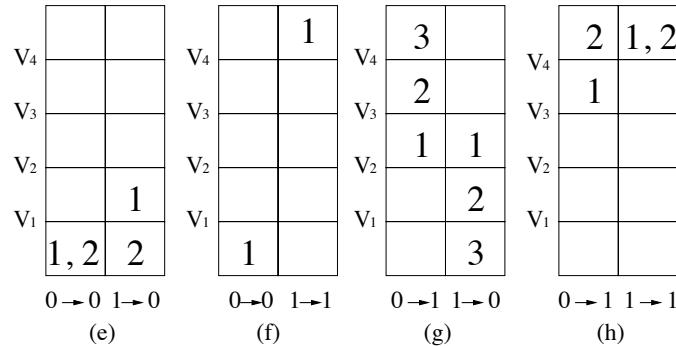


Figure 3.17. Correlated sub-ranges for a four-threshold logic in a topology of lines with one adjacent aggressor

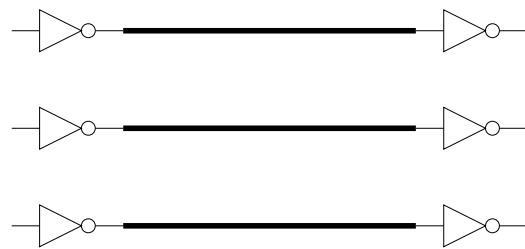


Figure 3.18. Example of bus line with two adjacent aggressors

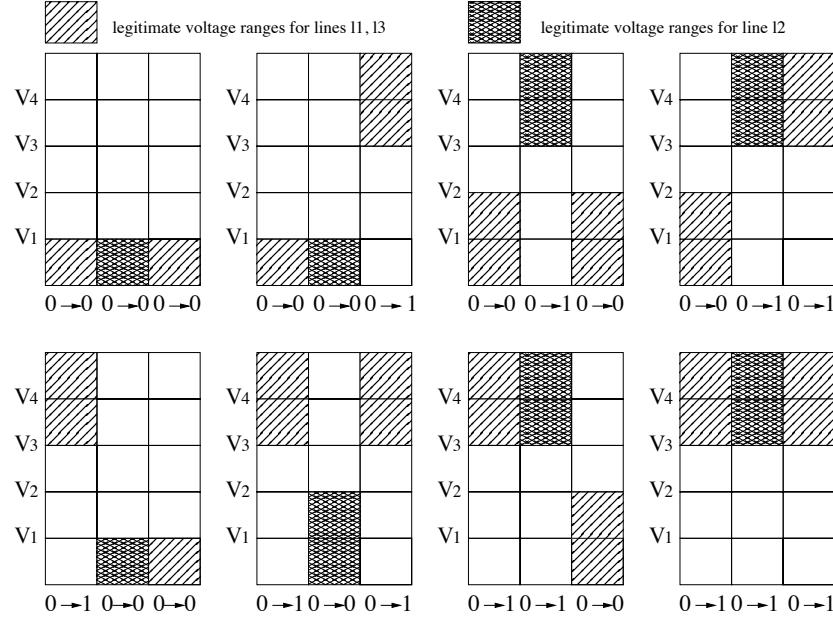
Table 3.8. Error detection when clock delay and additional threshold voltages are introduced in a topology of lines with only one adjacent aggressor

Type of Receiver Logic	Error Detection Efficiency
2-thresholds (earliest)	Detects one single-range error with correlated sub-ranges
2-thresholds (delayed)	Detects two single-range errors with correlated sub-ranges
3-thresholds (delay-optimal)	Detects one single-range error with correlated sub-ranges
3-thresholds (delayed)	Detects two single-range errors with correlated sub-ranges
4-thresholds (as fast as 2-thresholds)	Detects two single-range OR one double-range errors with correlated sub-ranges

Table 3.9. Error detection when clock delay is introduced in a topology of a line with two adjacent aggressors (using four-thresholds)

Type of Receiver Logic	error Detection Efficiency
4-thresholds (earliest)	Detects one single-range error
4-thresholds (as fast as 3 thresholds)	Detects three single-range errors
4-thresholds (as fast as 2 thresholds)	Detects three single-range errors
4-thresholds (as fast as 1 threshold)	Detects three single-range errors

Initial Conditions:  $(l_1, l_2, l_3) = (0, 0, 0)$



Initial Conditions:  $(l_1, l_2, l_3) = (0, 1, 0)$

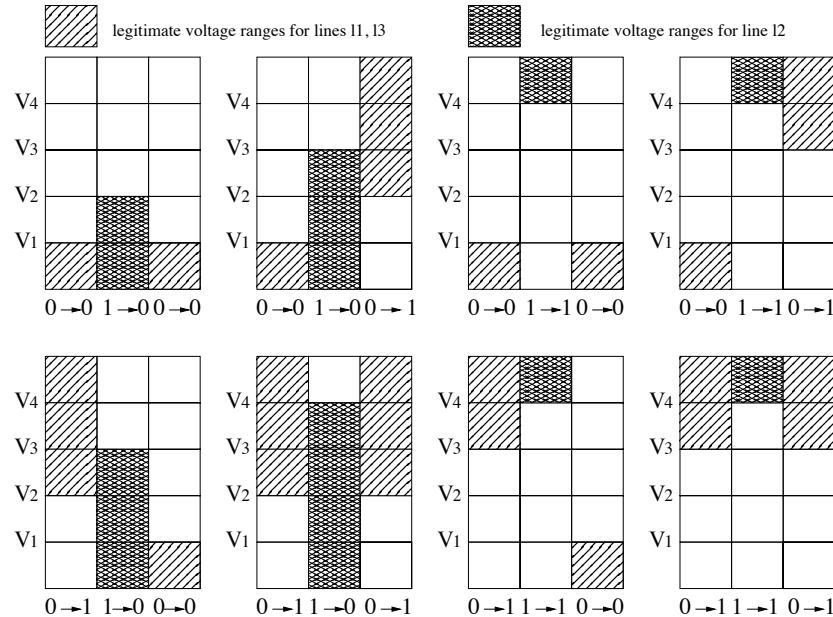


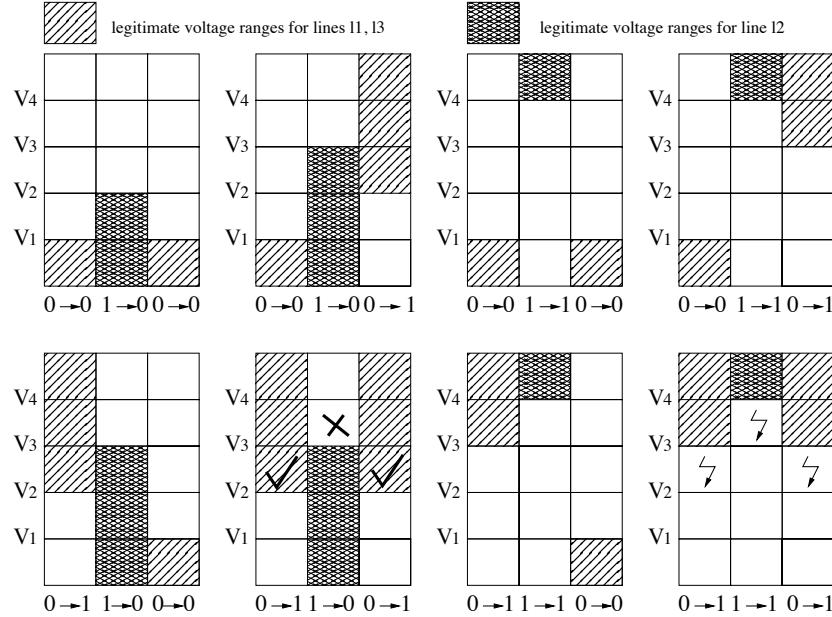
Figure 3.19. Error-free characterization of ranges for a line with two adjacent aggressors using four-thresholds

Table 3.10. Error detection when clock delay is introduced in a topology of a line with two adjacent aggressors (using five-thresholds)

Type of Receiver Logic	error Detection Efficiency
5-thresholds (delay-optimal)	Detects one single-range OR one double-range errors
5-thresholds (as fast as 4 thresholds)	Detects three single-range OR one double-range errors
5-thresholds (as fast as 3 thresholds)	Detects two single-range AND one double-range errors
5-thresholds (as fast as 2 thresholds)	Detects one single-range AND two double-range errors
5-thresholds (as fast as 1 threshold)	Detects three double-range errors

error detection with four-threshold receiver logic performing delayed data capture

Initial Conditions:  $(l1, l2, l3) = (0, 1, 0)$



error detection with five-threshold receiver logic performing delayed data capture

Initial Conditions:  $(l1, l2, l3) = (0, 1, 0)$

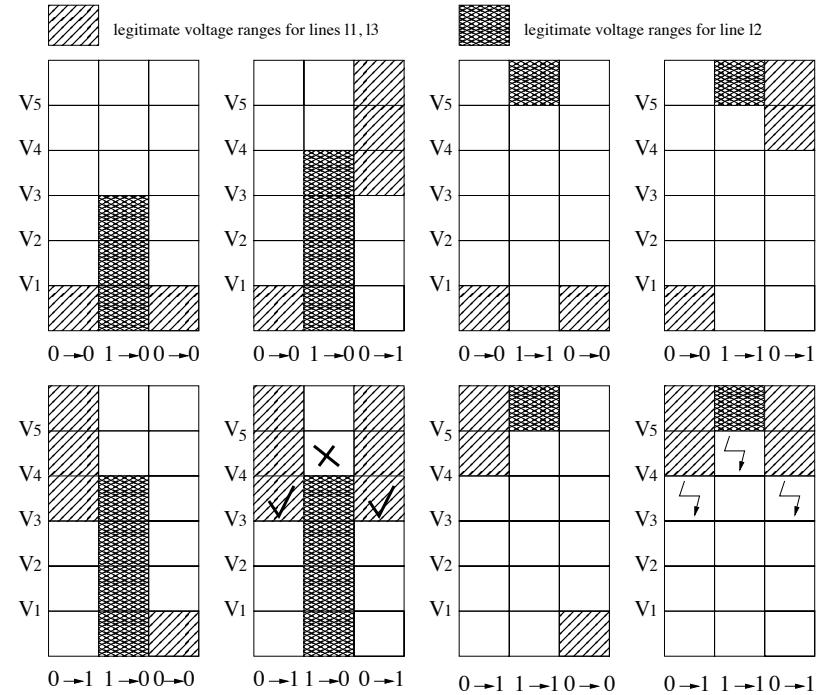


Figure 3.20. Detection of adjacent errors by introducing clock delay for a line with two adjacent aggressors (using four and five thresholds)

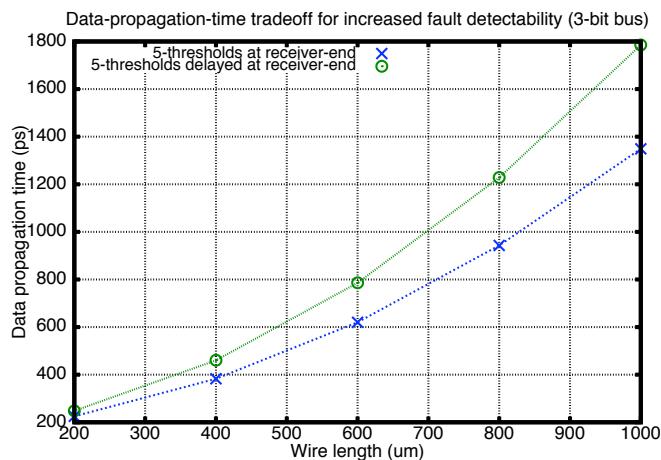


Figure 3.21. Delay tradeoff for increased error detection capabilities on a three-bit bus with a five threshold receiver logic to detect any (*i.e.*, three) simultaneous adjacent-range errors

## CHAPTER 4

### A DYNAMICALLY ADAPTIVE CIRCUIT FOR HAZARD TOLERANCE ON DATA BUSES

This chapter focuses on a special case of errors on data buses. In deep sub-micron, technology scaling and manufacturing defects have increased circuit vulnerability due to crosstalk interference between the different components and especially between parallel data wires. This interference is the direct result of parasitic coupling. The latter may induce additional delay on transmitted data and generate brief electric voltage glitches or hazards on a line. When a hazard appears within the clock sampling window at the receiver-end of a bus line, it will most certainly result in a functional error. Existing literature suggests and optimizes methods to deal with these issues, such as the well-known repeater insertion [27, 28, 29, 26], interconnect tuning [11] and dynamic shielding [12].

Extensive work has also been done on data encoding in order to manipulate the electric transitions on the wires to the designer's advantage. Encoding schemes for crosstalk, such as Hamming and Dual-Rail codes [20, 21], self-shielding code [22, 40], transition code [23], odd/even bus invert code [42] and predefined codeword generation [24] have been proposed. Other encoding schemes minimize coupled transitions on lines [45, 46, 41], reduce transitions on lines [35], minimize dynamic [47], static [36], peak [44] and leakage power [37].

The encoding algorithm in [40] generates a data-induced shielding as it only allows transitions for every other line. As a consequence, a bus line experiencing a transition is shielded by its adjacent neighbors that do not change their logic (and hence electrical) states with respect to the previous clock cycle. However, this means that the passive lines that are originated from such an encoding type will be between electrically active interconnects and will most likely be vulnerable to a

possible coordinated aggression. One such event can potentially create bad-case hazards that might put the proper operation of the circuit at stake.

In this paper, we assume that the data are encoded as in [40], and the focus is to alleviate a victim bus line from such hazards. We propose a novel circuitry for the receiving end of the bus. In particular, a receiver-end correcting mechanism is devised to treat coupling-induced glitches that appear on bus lines by dynamically adjusting its threshold voltage. Wire delay and dissipated power are also taken into consideration in the analysis. The proposed methodology is compared with the known from the open literature approach of repeater insertion in its optimized form.

#### 4.1 THE PROPOSED ADAPTIVE CIRCUIT

The proposed method minimizes crosstalk induced glitches through dynamically adjusting the threshold of the receiving gate so that the hazard is ignored. A rising glitch on the victim line would require an upwards adjustment of the threshold voltage in the receiver. Along the same principles, a falling glitch on the victim line would require a downwards adjustment of the receiver threshold voltage. This is achieved by synchronizing the adaptive circuit with the aggressor line so that it operates when the aggressor threatens the validity of the data on the victim line.

The proposed circuit to accomplish the threshold adjusting effect is shown in Figure 4.1. The method introduces a circuit at the receiver-end of the line that interacts with the aggressor(s), whereas a small size inverter is preemptively used for treating very narrow hazards at low cost. Signals  $a_d$  and  $a_i$  are the delayed and inverted versions of the aggressor(s) signal(s). These two signals are required and they are generated by using one and two minimum size inverters in series that produce the inverted and delayed signal respectively (Figure 4.2). The additional

capacitive load caused by these inverters is minimal and hence the impact on the aggressors is very small.

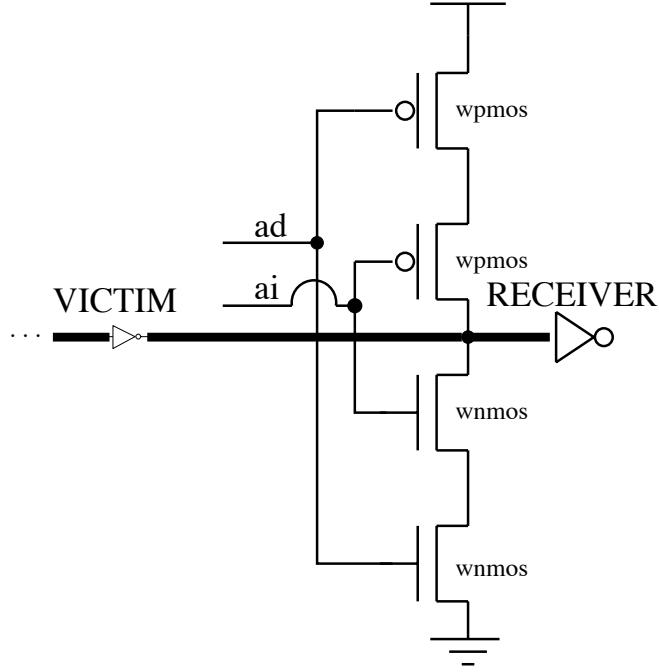


Figure 4.1. The proposed adaptive circuit

In the following, the basic operation and contribution of the adaptive circuit is explained. A rising transition on the aggressor line is considered while the victim line remains quiet at low logic. This will result in a rising glitch on the victim. For a rising transition on the aggressor, the generated inverted signal will be a falling transition. The delay of the aggressor signal is adjusted so that it is also held at low logic for some time interval  $D_t$  as shown in Figure 4.3. As a result, the pMOS network in Figure 4.1 will conduct, shifting thus the threshold voltage of the circuitry at the receiver-end of the line upwards and as a result the rising glitch on the victim line is ignored.

Similarly in the case of a falling transition on the aggressor while the victim remains quiet at high logic. In this case, a falling hazard is generated on the victim line. The delayed version of the aggressor signal is adjusted so that they it

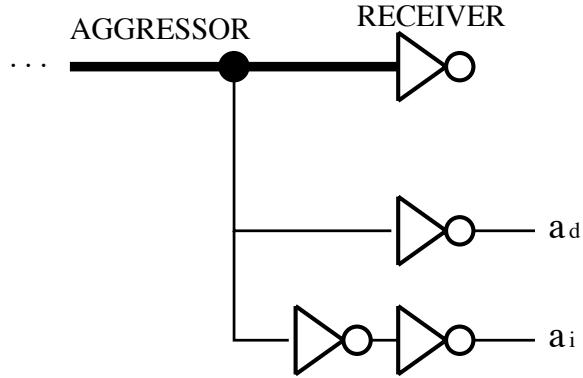


Figure 4.2. Generation of the control signals for the adaptive circuit

is held at high logic for a specific time interval. As anticipated, the nMOS network in Figure 4.1 will conduct, shifting thus the threshold voltage of the circuitry at the receiver-end of the line downwards and so, the falling glitch on the victim line is not propagated to the output of the receiver. A DC analysis illustrated in Figure 4.4 describes the threshold adaptation exhibited by the proposed circuit.

## 4.2 OPTIMIZATION OF REPEATER-BASED CONFIGURATIONS

In order to evaluate the overall performance of the proposed adaptive circuit, a traditional repeater-based technique is incorporated for a comparative analysis. For this purpose, delay-optimal repeater arrangements are appropriately selected and separately optimized. The following bus configurations are defined: (*I*) no repeaters, (*II*) one repeater, (*III*) two repeaters, (*IV*) three repeaters and (*V*) four repeaters. In each circuit configuration, all inverters are of identical size and they all scale up or down uniformly as it is also found in existing literature [27, 28, 26]. The objective is to identify the optimum transistor sizes for each of the above arrangements with respect to the exhibited wire delay. This is done because such topologies are anticipated to be optimized when used in high performance

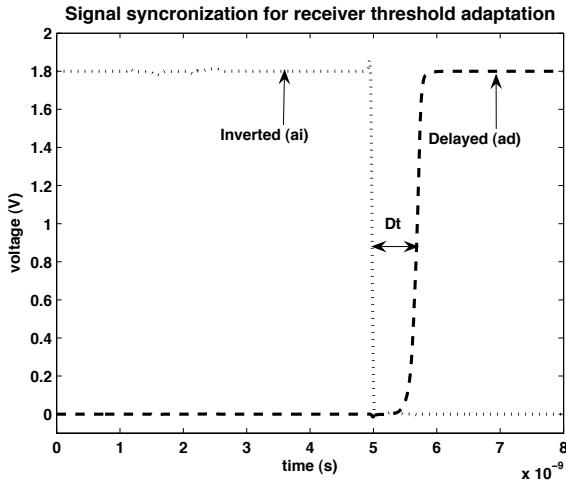


Figure 4.3. Illustrating the synchronization of the control signals in the adaptive circuit

circuits. The optimization is performed experimentally by running parametric analyses (on metal layer 3 lines in the TSMC  $0.18\mu m$  CMOS technology).

### 4.3 EXPERIMENTAL EVALUATION

For the experimental setup, an Assura-extracted  $500\mu m$  long and  $0.28\mu m$  wide metal layer 3 line is used in the TSMC  $0.18\mu m$  CMOS process. In agreement with encoding schemes that could produce worst-case harmful hazards, two identical aggressors are considered to impose simultaneously on an electrically passive victim line. The considered circuit setup involves severe coupling, as it is the case in current and future deep sub-micron technologies. For a comprehensive evaluation, experimental results on all the repeater-based configurations of the previous section are given along with the findings for the proposed circuit. It is reminded that each one of the traditional bus configurations is delay-optimized by appropriately sizing the repeaters (as illustrated in Figure 4.5).

The proposed method appears to be very effective in treating crosstalk induced glitches on a victim line and it is more suitable for short and medium

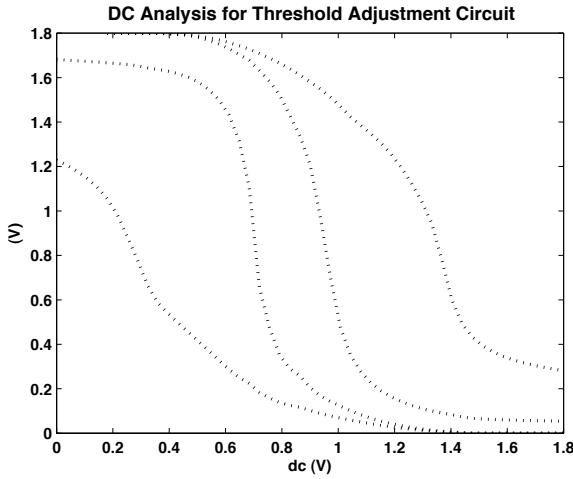


Figure 4.4. DC Analysis for the proposed hazard removal circuit

length interconnects. For longer lines, a traditional repeater insertion seems like a more appropriate design strategy due to a significantly larger wire resistance and accumulated capacitive load on the line. The presented method is convenient for glitches over  $400ps$  duration. On the other hand, for narrow glitches up to  $150\text{-}200ps$  width, a small receiver inverter (*i.e.*, with slow response) can be inserted in front of the main receiver inverter. So, it is possible to cut off successfully the incoming glitch with a small delay penalty ( $75ps$ ).

To address both the crosstalk mitigation efficiency along with the corresponding performance manifested by the investigated circuits, delay and power measurements are taken. The transmitted signal on the victim line is such that both the delay of an active transition ( $low \rightarrow high$  or  $high \rightarrow low$ , while the aggressors remain electrically quiet according to the referenced encoding scheme in [40]) and the hazard treatment capability are evaluated simultaneously in two consecutive cycles (Figure 4.6). The bus delay and the dissipated power are recorded for each configuration as a function of the exhibited crosstalk amplitude. The measured power for each topology includes the power consumed by both the

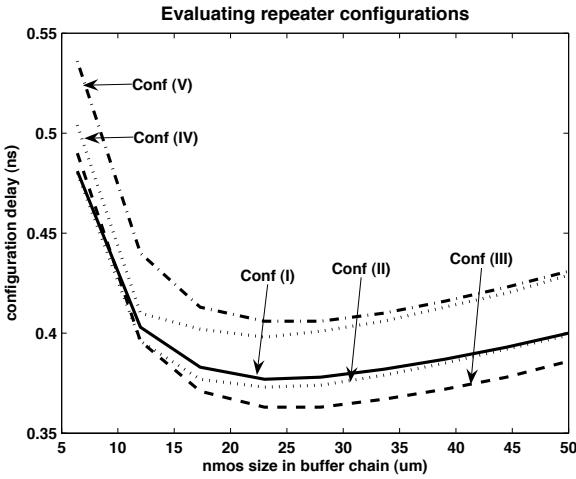


Figure 4.5. Experimental optimization of typical repeater-based configurations

aggressors and the victim.

For the adaptive circuit, the glitch amplitude at the output of the receiver is reduced by 95% when relatively small transistors are used, exhibiting a delay of  $460ps$  and with a  $10.206pW$  power consumption (Figure 4.7). For a fair comparison, a continuous uniform scaling is performed on the typical buses (and after optimizing each one separately for smallest delay) until the glitch amplitude at the receiver output indicates that a 95% reduction is also achieved. For each repeater-based bus, a waveform of the signal before (dotted line) and after (solid line) the scaling is provided (Figures 4.8, 4.10, 4.12, 4.14 and 4.16). Also, delay and power measurements for all cases are given as functions of the achieved crosstalk reduction (Figure 4.9, 4.11, 4.13, 4.15, 4.17).

In detail for the several repeater-driven arrangements, a 95% crosstalk reduction in bus configuration (*I*) requires a scale up of the involved repeater logic by  $8.5\times$ . The delay ( $631ps$ ) is increased by 37% and the required power ( $60.849pW$ ) has also increased by 496% in comparison with the proposed circuit. For the delay optimized configuration (*II*), and for the same order of glitch

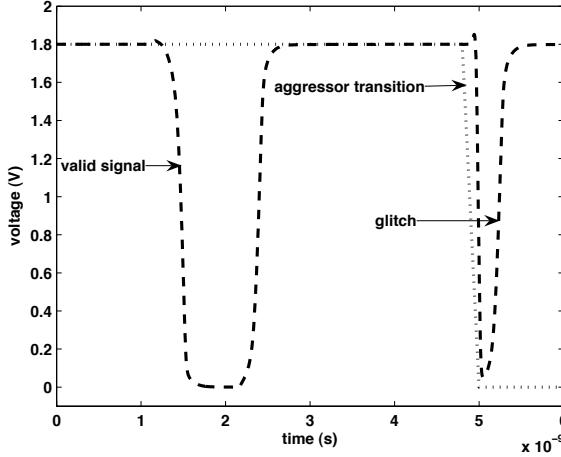


Figure 4.6. Transmitted signal arrangement for evaluating hazard removal efficiency and potential delay degradation for legitimate signals

amplitude reduction, a scale up of the repeaters by  $4.5\times$  is required. The delay ( $511ps$ ) is increased by 11% and the power ( $34.893pW$ ) has increased by 240% compared to the adaptive circuit. For configuration (III) a  $5\times$  scale up is required. The delay ( $554ps$ ) has shifted upwards by 20% and the power dissipation ( $53.064pW$ ) is larger by 419%. Arrangement (IV) requires a scaling of about  $4\times$ . The delay ( $515ps$ ) has increased by 12% and the power ( $36.297pW$ ) has also increased by 255%. Lastly, for topology (V) a  $5.8\times$  scale up is performed. The delay ( $613ps$ ) has increased by 33% and the needed power ( $64.242pW$ ) is up by 529%.

For different levels of crosstalk alleviation (from 75% ( $\simeq V_t$ ) to 95% ( $\simeq \frac{V_t}{4}$ )), the delay and the dissipated power are recorded in Figures 4.19 and 4.18. As far as the dissipated power is concerned, the proposed method outperforms the traditional methodology. For a limited glitch reduction ( $V_t$ ), the delay of the repeater buses is smaller but that is not enough to guarantee proper circuit operation due to the significant amplitude of the hazard.

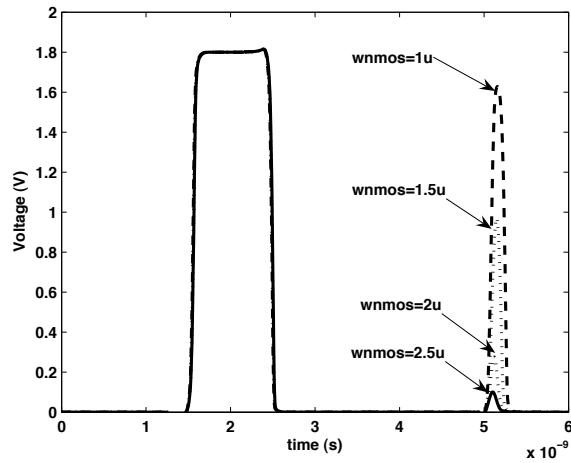


Figure 4.7. Hazard removal and delay characteristics for the proposed circuit

#### 4.4 CONCLUSIONS

A method to be used in combination with an existing encoding scheme is presented. Experimental results indicate high efficiency in treating hazards without imposing on the bus delay and the dissipated power, whereas repeater insertion is avoided.

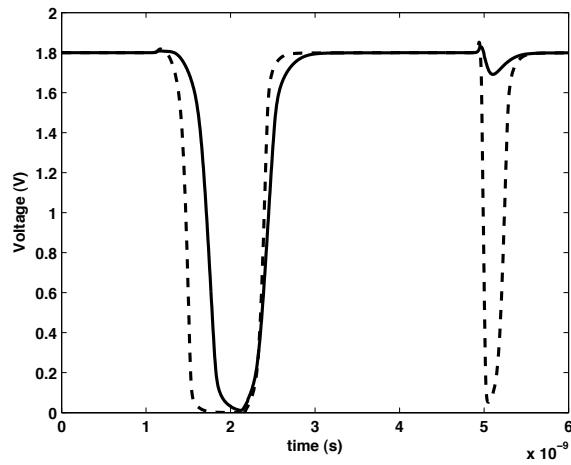


Figure 4.8. Transmitted signal for Configuration (I)

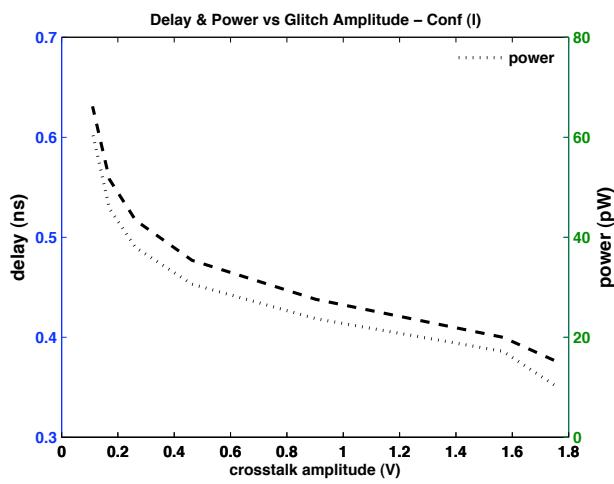


Figure 4.9. Delay and dissipated power for Configuration (I)

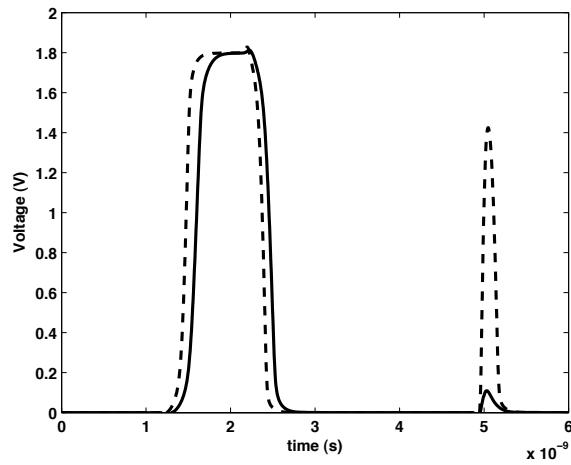


Figure 4.10. Transmitted signal for Configuration (II)

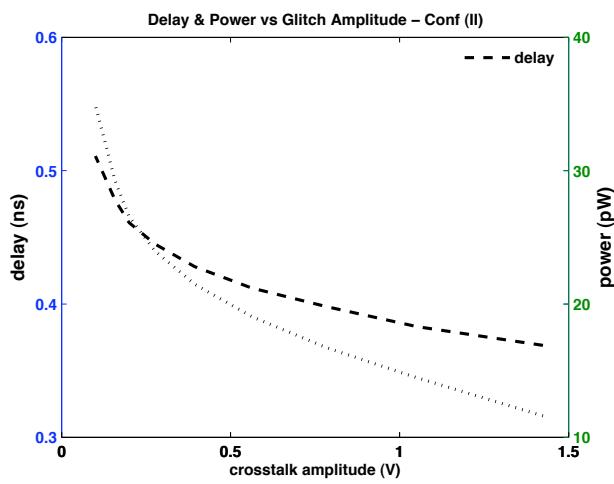


Figure 4.11. Delay and dissipated power for Configuration (II)

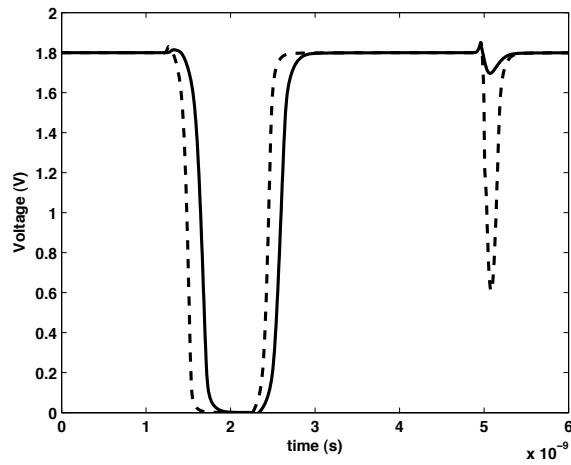


Figure 4.12. Transmitted signal for Configuration (III)

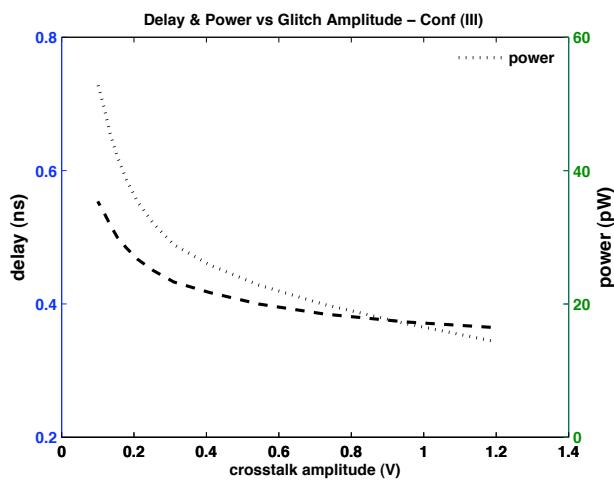


Figure 4.13. Delay and dissipated power for Configuration (III)

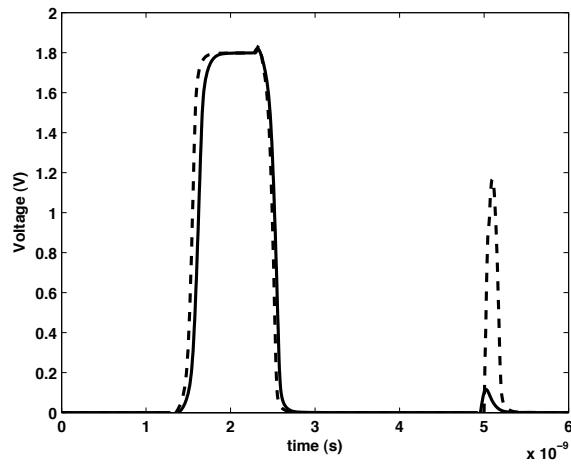


Figure 4.14. Transmitted signal for Configuration (IV)

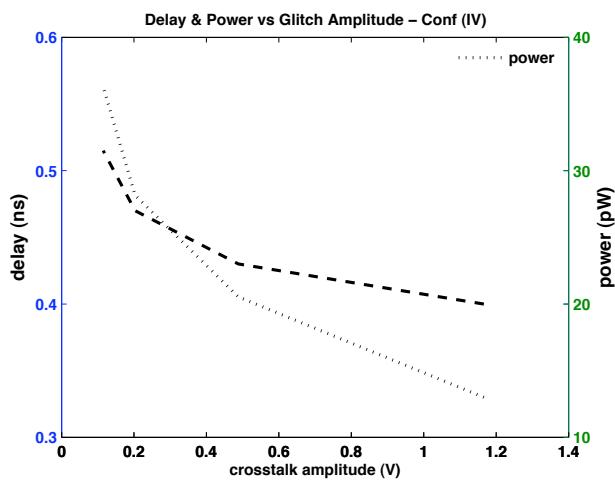


Figure 4.15. Delay and dissipated power for Configuration (IV)

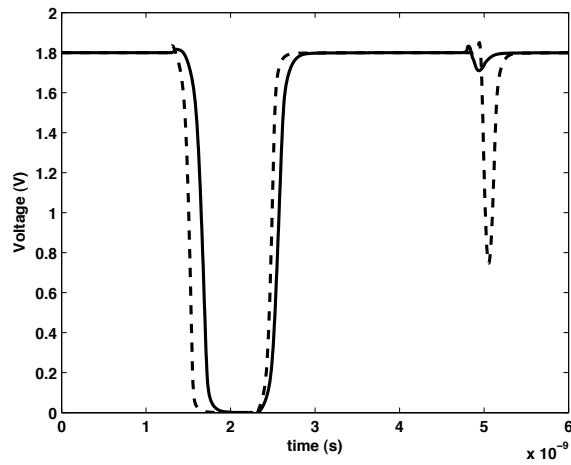


Figure 4.16. Transmitted signal for Configuration (V)

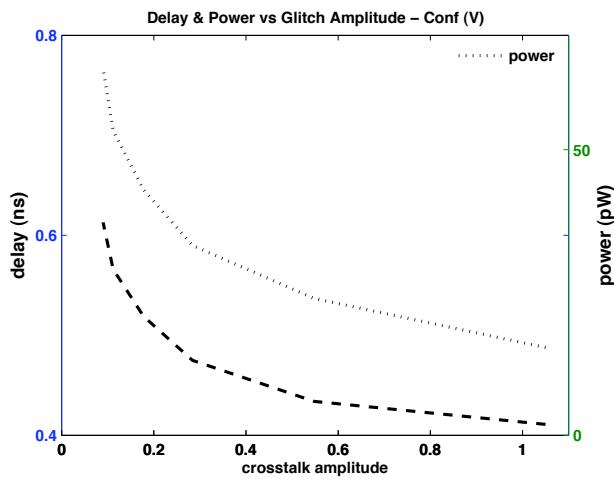


Figure 4.17. Delay and dissipated power for Configuration (V)

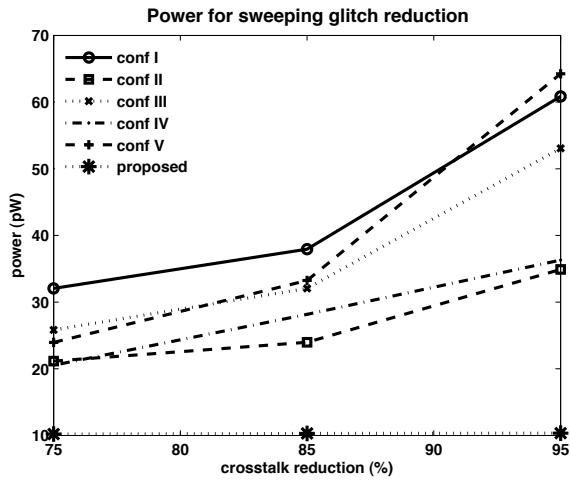


Figure 4.18. Power sweep for different levels of crosstalk reduction

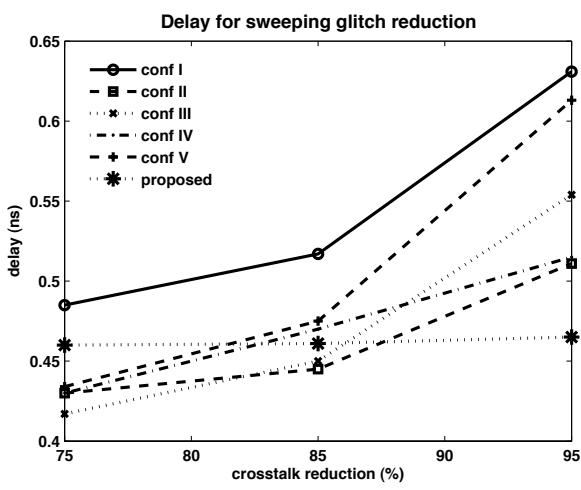


Figure 4.19. Delay sweep for different levels of crosstalk reduction

## CHAPTER 5

### SINGLE TRANSIENT EFFECTS IN COMBINATIONAL LOGIC AND A HARDENING METHODOLOGY

Scaling of semiconductor devices, shorter pipeline length and reduction in power supply have all contributed significantly to an increased vulnerability of modern integrated circuits to random electrical transient events. Inherent protective mechanisms that logic traditionally held against random noise have weakened significantly and, as a consequence, transient faults have become increasingly observable in combinational logic. Such events are referred to as Single Event Transients (SETs) and are mainly caused by cosmic radiation (either high-energy or low-energy neutrons) or packaging impurities in the form of alpha particles [71, 72].

Some of those particle-semiconductor interaction effects are described in [80] and may lead to permanent failure or even destruction of a device. These destructive effects can be classified into single-event latchup (SEL), single-event burnout (SEB), single-event gate rupture (SEGR) and single-event snapback (SESB). Usually though, when such particles collide with semiconductor material nearby or through a reverse-biased pn-junction, they produce electron-hole pairs. The latter will likely forward-bias the junction and generate a voltage-glitch at that junction node [71].

Robust electrical transients of that type are expected to erroneously trigger logic gates. The extent of impact that an ionizing particle will have on integrated circuits depends on its linear energy transfer (LET) measured in  $Me - cm^2/mg$ . It is reported that for every 3.6 eV of energy released by the ion, one electron-hole pair is produced. It is highly likely that if there is enough charge to generate a voltage wave in pipelined logic, that may eventually flip the logic state of a

memory cell. This phenomenon is termed Single Event Upset (SEU) or soft error [72]. The minimal charge that needs to be deposited across a junction for flipping the logic state of that node is defined as the critical charge  $Q_{crit}$ . A really bad-case of particle strike may also produce a multi-bit upset (MBU) instead of a more probable single-bit upset (SBU). A popular metric for quantifying the severeness of soft-error occurrence in a circuit or of the Soft Error Rate (SER), is the Failure in Time (FIT). The latter corresponds to a failure in  $10^9$  device hours [71, 72]. A typical SER for a hardened circuit is in the 50-200 FIT range. Without any fault-tolerant techniques, SER may exceed 50,000 FIT per chip [72].

Although designers in the past targeted enforcing memory array tolerance, combinational logic tolerance emerges as the new challenge in modern and future technologies. It is anticipated that in the  $45\text{ nm}$ , most of the temporary functional failures in memory elements will be associated with transient faults originated in the combinational logic [103]. For the latter, inherent defense mechanisms such as *logical masking* (elimination of transients along unsensitizable paths), *electrical masking* (attenuation of transients due to non-saturated transistor devices) and *temporal masking* (disregard of transient effects arrived at the inputs of memory elements at other than the designated sampling-window time range), are unable to increase circuit tolerance as effectively as in the past [123]. Hence, transient propagation in combinational logic becomes a serious concern.

In general, one could distinguish between a statistical and a worst-case transient propagation approach, each having a different objective. The former relies on logic simulation to define sets of sensitizable paths, adopting thus a Monte Carlo driven perspective. Such an approach aims to determine averaged metrics such as rate and probability. However, this methodology requires an extremely high number of input vectors for a worst-case type of analysis, entailing prohibitive cost.

In order to assess the worst-case impact of potential transients on the circuit operation, this research aims to estimate an upper bound for certain electrical characteristics of those transients using a static instead of a dynamic analysis. Such information could be used to refine existing treatment and mitigating methods. In this way, only the minimally needed overhead is imposed on the circuit and excessive redundancy is avoided. In particular, more effective latch hardening, gate sizing and sampling-window adjustments are possible. For this research and based on existing literature [85, 88, 90] among others, we have determined the transient duration and its associated magnitude as the most essential metrics in our effort to evaluate the impact of such hazzards in logic. These metrics for certain transients can be dangerously amplified by the interaction with other existing correlated or independent transients. The objective of the proposed research is to approximate the effect of this interaction.

The rest of this chapter is as follows: The following section reviews existing literature on single event transients and emphasizes the criticality of the problem. In Section 4.2, the employed static propagation of transients is discussed. Furthermore, a quick and accurate method of statistical characterization for a logic gate is presented for multiple input transients per gate. The custom generated standard cells provide the realistic amount of electrical attenuation or electrical enhancement of transients when propagated through logic gates. In the same section, methods to sensitize paths in combinational logic are also briefly discussed. In Section 4.3, a hardening methodology for combinational logic based on filter circuits (C-elements) and optimization methods for performance purposes are proposed. Experimental results are also given in both Sections 4.2 and 4.3 to demonstrate the benefits of the proposed research in different die sizes and the chapter is concluded in Section 4.4.

## 5.1 BACKGROUND RESEARCH IN SINGLE EVENT TRANSIENT EFFECTS

In this section, we review literature relevant to the increasing soft error susceptibility of circuits with technology scaling and increasing challenges of new emerging trends and design paradigms. Then, we review published work on ways of modeling and simulating single event transient generation and propagation through logic. Fault-tolerant design approaches are also referenced.

Continuous technology scaling makes memory and combinational circuits more vulnerable to random transient events. This is due to considerable reduction of the critical charge  $Q_{crit}$  from one process-generation to the next. From an early study conducted by IBM [75], it is evident that a wide range of energies is encountered in ionized particles. As anticipated, lower energy particles can appear more frequently than higher energy ones. An estimate reports that one order of magnitude difference in energy corresponds to two orders of magnitude in larger flux [75]. As a consequence, smaller technologies must cope with a lot more transient events than their predecessor ones.

In [77], effects of device scaling and superpipelining are studied through a model capturing electrical and latching-window masking. Results show an increase of SER in logic by 9 orders of magnitude in 20 years, from  $600\text{ nm}$  to  $50\text{ nm}$  processes and from 16FO4 to 6FO4 clock periods. This renders SER of combinational logic comparable to SER of unprotected memory. In [87] a test chip is used to evaluate SETs in  $0.25$  and  $0.18\text{ }\mu\text{m}$  processes. The main goal is to derive the contribution of SETs to the total error rate. Linear increase of SER versus frequency of operation is observed. The probability to capture an SET at latches increases linearly with frequency as a consequence.

In [74], it is reported that the SER on SRAM was predicted to increase at most linearly with decreasing feature size. Another parameter introduced in

nano-scale designs is performance variation. In [81] it is explained how design variability affects tolerance of a circuit to random transients. The authors examine how the static (inter and intra die), dynamic (power supply and temperature) and aging-caused variations affect the soft error rate. The latter was found to vary up to 41% in the case of threshold deviations due to intra-die variations. Power supply variations yielded a maximum SER variation of 24.85% on several tested circuits. Furthermore, temperature had the strongest impact on SER variation based on the experiments.

Another interesting concern among researchers in both academia and industry is the appearance of multiple erroneous events caused by a single particle strike [82, 83]. Reduced critical charge  $Q_{crit}$  and existence of heavy ions at sea level [75] render this a realistic scenario. In [82] the authors examine the likelihood that a single particle strike can occur in such a way to induce multiple transients in the circuit. Then they show that such an event can generate bi-directional errors at the circuit output and compute the probability of such a case. It is derived that such a probability can be around 20% for some benchmark circuits. In [83], multi-bit failures in SRAM cells are analyzed in  $90\text{ nm}$  and  $130\text{ nm}$  technologies.

Except for the aforementioned challenges resulting directly from process size, new SoC design paradigms complicate traditional fault-tolerant tactics. For instance, it is mentioned in [76] that IP re-use for several applications entails different modes of operation for embedded cores of a SoC. As a result, the existing fault-tolerant circuitry on each core might have an undesirable impact on some of the applications which the core is intended to carry through. As a result, transient tolerance may vary for a core depending on the application. In [79], it is emphasized that soft error protection mechanisms must be designed for SoCs since this is the current direction that industry is heading towards. Including protection for all cores entails prohibitive cost and hence a selected number of cores is

recommended to be guarded against SEUs. This is done with different techniques depending on the nature of the core and the SoC application.

Another potential effect of the cosmic ray radiation involves clock jitter and race. The latter is defined as the racing of data following false clock edge [78]. Their results show that a scheme incorporating hardened pulse latches and a hardened pulse generator improves SER by 20x with little overhead.

In order to efficiently rectify increasing vulnerability of integrated circuits to transient events, the latter must be thoroughly studied and effectively modeled. Accurate modeling provides invaluable assistance in efforts to develop fault tolerant techniques for circuits. A typical approach to model the electric current resulting from a particle strike is based on the independent double-exponential current pulse. This model is considered to have the advantage of capturing both drift and diffusion of charge carriers during the ion collision [97].

Recent published work [84] contests the accuracy of such a model and instead propose a modified current source model for a particle strike. They claim that traditional models assume constant voltage across the conducting pn-junction, whereas drift of carriers during re-combination is expected to change the potential difference across that junction. Their new model predicts wider transients which can propagate with less attenuation through the combinational logic. As a result, a higher soft error rate is calculated with this technique, hinting that methods modeling the particle strike with the typical double-exponential current source model are rather optimistic.

Concerns about likely inaccurate modeling of transients prompted the design of a test circuit structure for measuring the width of transients [85]. Due to the fact that for a particle of a given level of energy, the transient duration depends not only on the struck structure characteristics but also on how the particle is positioned with respect to the struck node, a distribution of pulse widths is

obtained instead of single pulse duration. In [86], by practically eliminating SEUs in the memory cells via the usage of radiation-hardened latches, the authors study the width of the transients originating in combinational logic chains. Test chips were fabricated in 0.25 and 0.18  $\mu m$  processes. Their results indicate the importance of the duration of an SET over its magnitude. If the SET is too narrow, then not enough charge is deposited to cause a hazardous transient and the latter is attenuated early irrespective of its potential amplitude. It was found that in bulk silicon for an ionizing particle with an LET of 100  $MeV - cm^2/mg$  (heavy-ion), transients of approximately 2  $ns$  can be created.

In [88] another fabricated test structure (in 0.18  $\mu m$ ) is presented for the purpose of measuring SET widths in logic. The transient pulse width was found to increase linearly with increasing LET. The measured width range is 350  $ps$  to 1.3  $ns$  for an LET range of 11.5 to 64  $MeV - cm^2/mg$  respectively. In [89] they study characteristics of transient shapes and their propagation in inverter chains. Their results indicate that hazardous SETs can be generated following strikes of particles of low energy and propagate unattenuated. In bulk technologies they found SETs to be unexpectedly wide ( $\simeq 1 ns$  for a 70  $MeV - cm^2/mg$  LET), especially for heavy-ions. Note that the authors define the width as the duration of the transient in the struck device itself. They have found that to be 15-20% narrower than the width of the transient at the output node of the struck cell.

In [90], as previously in [85], a wide distribution of potential transient durations is observed instead of a single discrete width. In [91], the authors propose new circuit modeling approaches correlating transient width with charge collection. It is confirmed experimentally that the latter determines the generated transient duration following a heavy-ion collision with semiconductor material. The traditional double-exponential independent current source model is rather inadequate in comparison with the experimental data in this paper. They propose

a new model incorporating SPICE analysis and 3-D device physics simulations.

Their model also provides SETs wider than previously believed.

Other work focuses on devising accurate methods for propagating a single event transient through logic. In [93], circuits were fabricated for studying attenuation of transient pulses in combinational logic. Their results indicate that even though transients resulting from particle strikes clearly do attenuate, their measurements are inconclusive as to whether the circuit SER depends largely on attenuation effects. In [94] a mathematical model calibrated with HSPICE is presented for simulating single event transient propagation in combinational logic with a 10% error. Another mathematical model is proposed in [104]. This requires HSPICE pre-characterized libraries for the gate delay and output slope with respect to input slopes, output capacitive loads, input rise and fall times.

A method for simulating transient propagation based on linear RC gate modeling is presented in [95]. The ion-induced transient is modeled with another linear model. A test structure is designed to calibrate the model and yield results within 10% error compared to SPICE-extracted results. In [96], an analytical waveform modeling is presented that is based on the Weibull cumulative distribution function. The Weibull function accommodates capturing various waveforms. They present an algorithm for converting a random shape obtained by circuit simulation to the new Weibull model. Their work includes cell characterization for Weibull-centric circuit simulations. The obtained results are within 5% error.

The increasing risk of circuit failure due to single event transients, as seen from the referenced experimental data, urges industry and academia to develop accurate SER modeling techniques. In [98], measurement techniques for SER modeling purposes are reviewed in sub-90 nm technologies. They also make a comparison study on published neutron and alpha-particle SER scaling reports. A

circuit-level SER model based on considerations of device-level simulations is illustrated in [99]. Particle test measurements for calibration purposes, statistical simulations and analytical SER modeling comprise this new technique that can be embedded in circuit simulators for use. The FASER tool in [100] is based on static timing analysis, where pre-characterized models for gates are utilized and path sensitization is achieved by the usage of binary decision diagrams. The average error across the simulated benchmark circuits is 12%, but could be reduced by a more accurate library characterization.

In [101], a soft-error tolerance analysis tool of deep-submicron combinational circuits is presented (ASERTA). It uses SPICE-based library characterization, an exponential current source model for the particle strike and Monte Carlo logic simulation. Based on this tool, soft-error tolerance optimization is performed through an optimal selection of supply voltages, threshold voltages, gate sizes and optimal capacitive loads at the primary outputs.

The SERA tool in [102] is a blend of probability theory, graph theory, circuit and fault simulation. It is within 95 % accurate and is five orders of magnitude faster than typical Monte Carlo driven methods. A method of parameterized descriptors is proposed in [103]. They utilize the Weibull function for representing particle strikes in devices. They also introduce a descriptor object containing correlation information between transient waveforms and associated rate distribution functions. The algorithm has a linear complexity since a simple topological traversal is performed while these descriptors are injected, propagated and merged. In [104], two components of the SEAT toolset are introduced and validated. SEAT-DA models single event transients at the device level, whereas SEAT-LA operates at the gate level. The average error of this technique is reported to be 6.5 %.

An infamous SER analysis tool is the IBM SEMM [105, 106]. The first of

these two [105] calculates the SER of semiconductor chips based on information about the design and the radiation conditions of the ambient. A Monte Carlo method is used to simulate a large number of radiation events. However, this tool was developed for bipolar technologies. SEMM2 was recently built to handle CMOS technologies [106]. Lastly, the proposed MARS-C tool in [107] uses binary decision diagrams and algebraic decision diagrams. The utilized data structure facilitates SER analysis in case of multiple hits at different nodes in the circuit because of inherent masking mechanisms. Results obtained are within 7 % error.

Soft-error tolerant techniques include logic hardening [108, 109, 110, 111], tolerant flip-flop design and latch hardening [112, 113, 115, 116, 117, 118, 119, 120], layout and device level approaches [121, 122], circuit level methods [123, 124, 125, 126, 127] and time redundancy [123, 128].

## 5.2 TRANSIENT PROPAGATION IN COMBINATIONAL LOGIC

### 5.2.1 Static transient propagation

Logic delay uncertainty, that is typically produced by process, environmental and manufacturing variations, will most likely alter the timing characteristics (arrival, duration, rise and fall times) of propagated transients through logic. As a consequence of that, transients on re-convergent paths that interact at the inputs of a gate cannot be properly addressed by dynamic, time-specific SPICE simulations. A deterministic scenario may end up being too optimistic if the assumed timing specifications don't capture the possibility of a worst-case event.

When, for instance, transients (that may be caused by the same or by different heavy energy ions) meet at the inputs of a gate, their combined effect will vary depending on the time of arrival and the polarity of the transients (controlling or non-controlling) with respect to the type of the gate (Figures 5.1 and 5.2). For this reason, it is more prudent to employ a static approach that uses an estimated

time interval of arrival and subside for each transient, along with its corresponding minimum and maximum values for slope and duration.

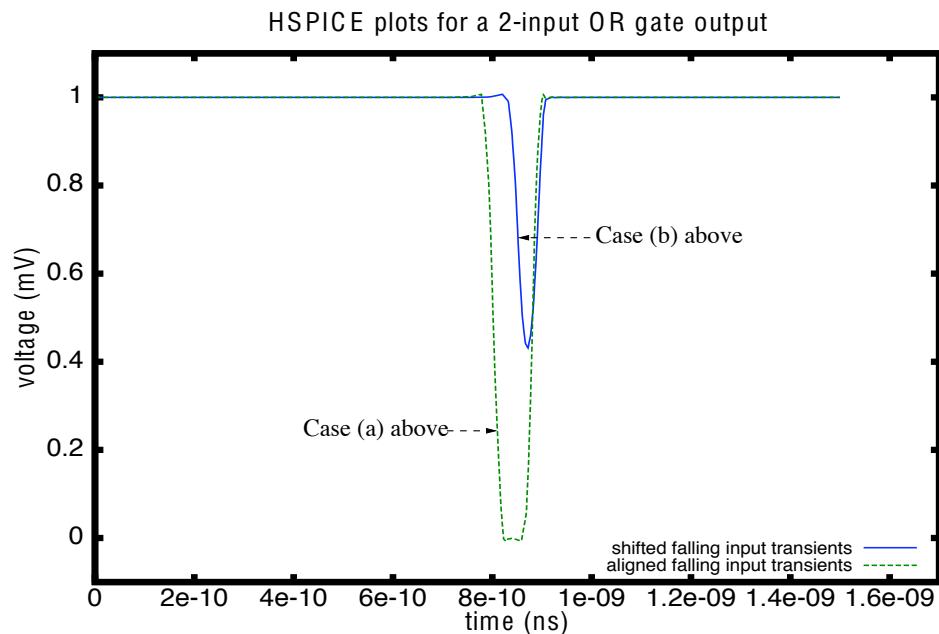
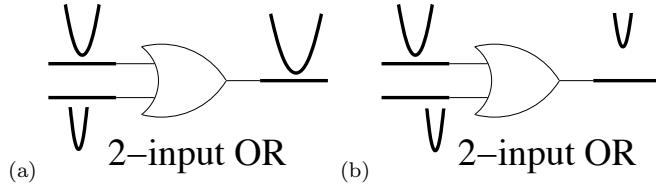


Figure 5.1. Resulting output transient for a gate when non-controlling value transients appear at its inputs

Lower (earliest) and upper (latest) time bounds that capture variation-related mismatches and delay uncertainty are assigned to each transient (Figure 5.3). For a group of interacting transients at the inputs of a gate, the different lower and upper bounds define multiple time sub-intervals with a high number of transient-interaction possibilities that increase exponentially in the next logic levels. Since such a setup leads to an incredibly difficult computational task, it is pessimistically assumed that all transients are defined (*i.e.*, may appear and subside) within the same time interval. The latter is defined by the earliest arrival

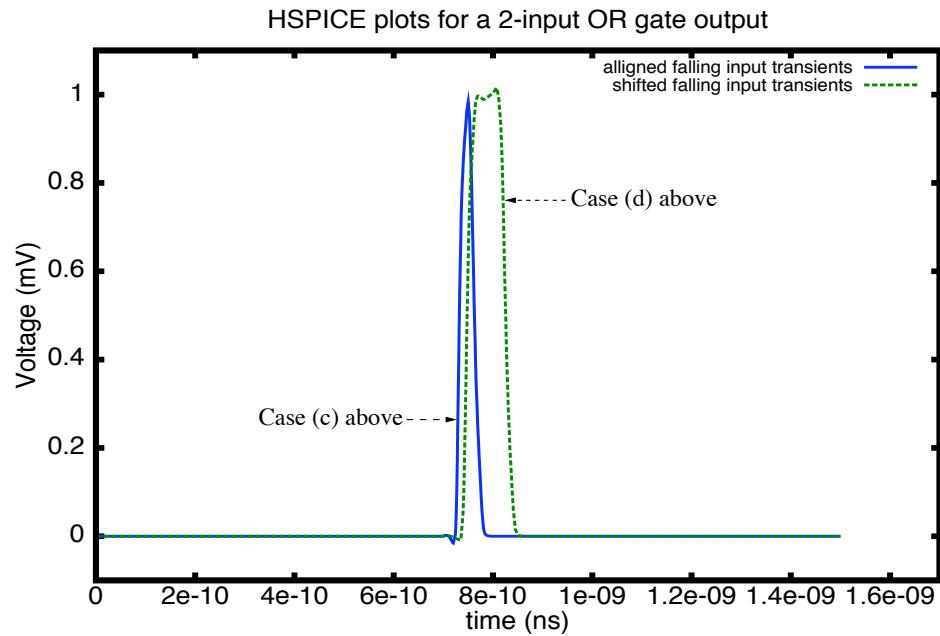
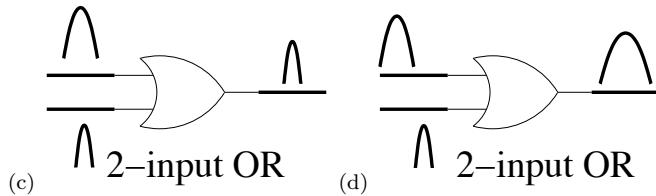


Figure 5.2. Resulting output transient for a gate when controlling value transients appear at its inputs

and latest subside time bounds determined by all interacting input transients. In order to support and avoid undermining a conservative analysis, opposite polarity transients are not allowed to interact and they are instead propagated independently.

Within the common defined interval, all transients are assumed that can occur at any time. Therefore, by taking into consideration the controlling or non-controlling polarity of the input transients with respect to the gate, the worst-case (maximum) values for the slope, delay and duration of the output transient can be computed. In order to do that with SPICE-driven accuracy, libraries of standard cells must be efficiently characterized to accommodate such a

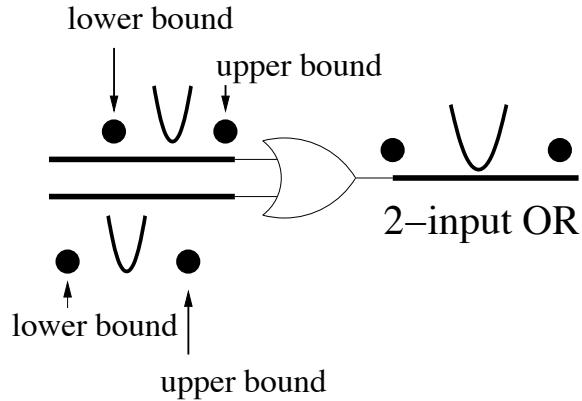


Figure 5.3. Static propagation of transients

type of analysis.

### 5.2.2 Electrical characterization for static analysis

The ever increasing unpredictability of essential electrical and time parameters for transients requires a discretized, parametric, statistical characterization of standard cells with respect to transient arrival time, slope and duration. In essence, transient interaction must be evaluated and recorded for different combinations of the aforementioned parameters (arrival time, slope and duration) and for a varying number of interacting input transients. Considering the different types of available gates and the various possibilities as far as fan-in (input) and fan-out (output) lines in a gate, the task appears to be computationally demanding and extremely time consuming. In existing literature and probably due to this reason, only single transient standard cell libraries are found.

For instance, assuming three distinctive input transients with each being characterized in a discretized fashion by two slope, two duration and two likely arrival time instants and for different polarities, a total of 4,096 HSPICE simulation runs ( $2^3 \times 2^3 \times 2^3 \times 2^3$ ) are needed for an exhaustive characterization.

This would approximately require 10.25 hours of run time on a Sun SPARC Blade 1000 dual processor (750MHz) workstation. It is noted that the above is for a single type of gate and for a specific number of fan-in and fan-out lines only. Therefore, to characterize all possible gate types for different number of fan-in lines, fan-out lines and interacting transients would be a prohibitively time-consuming task. Furthermore, simulations show that the resulting error of this approach due to the small number of discrete levels for each transient parameter (slope, duration, etc.) is unacceptable.

It is also important to note that a statistical cell library characterization for static analysis entails a Monte Carlo driven HSPICE analysis. Therefore, in reality every characterization run consists of multiple such HSPICE runs and this ultimately blows up the overall number of runs needed. Since the total characterization cost is exponential to the number of transients, creating efficient cell libraries for static analysis is a cumbersome task with many challenges and obstacles that must be intelligently resolved and overcome.

Keeping in mind that such a type of analysis is by definition a conservative and pessimistic approach, a standard cell library (in a particular technology) is targeted for this research that is always slightly pessimistic. This is achieved by collapsing sets of cases in fewer approximate occurrences. In this way it is possible to use more discrete levels for each transient parameter (slope, duration and arrival time or equivalently time-shift between transients) for better accuracy. An output transient is characterized by estimating minimum and maximum values for slope, duration and delay. In the following, it is briefly explained how the proposed characterization is performed. The latter is constant in the number of transients (instead of exponential) and polynomial in the number of discrete parameter levels used per transient parameter.

For a reliable and fast library extraction within a static analysis framework,

the ultimate goal is to considerably reduce the number of existing combinations that need to be simulated. In order to do that, the conventional expectation that the input transients would be distinctive instances with their own particular slope and duration needs to be released. On the contrary, it is presumed that the appearing transients at the inputs of a gate would be identical instances and thus a large number of parameter combinations is collapsed together and approximated by fewer cases. As a direct result of this simplification, the overall characterization time is significantly reduced.

The complexity of the extraction flow is constant in the number of input transients under this setup, although polynomial in the number of discrete levels assigned to each electrical parameter. To increase accuracy, a higher granularity for those parameters is utilized compared to a previous example. For instance, in the case of a chain of (five) inverters and for a dynamic simulation, a higher granularity provides a satisfactory approximation of the electric response of gates (Figure 5.4). Similarly for other benchmark circuits (in Appendix).

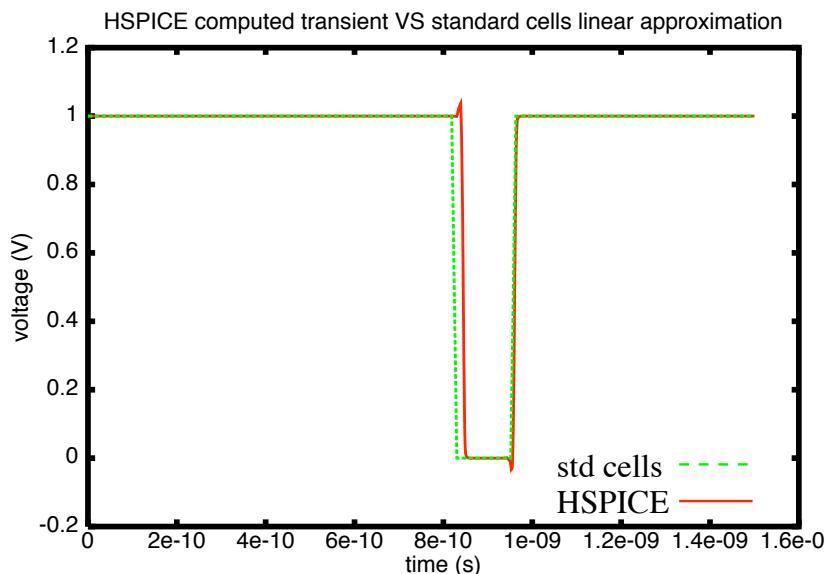


Figure 5.4. Approximation of a five inverter chain with standard cells

During the circuit (graph) analysis and the propagation of hazards using such type of cell library, all input transients will eventually need to be mapped to identical occurrences. So, it is next to be determined how this mapping will take place. Taking into account the inherent pessimism of a static approach, the individual electrical parameters are evaluated and the worst-case of those are assigned to a new set of transient occurrences replacing the original ones (Figure 5.5). This entails that the instances obtained from this mapping will have the steepest slope, largest duration and smallest or largest time-shift between transients (depending on the controlling nature of the transients with respect to the gate type) that can be accommodated in the interval between the earliest and latest time bounds. Transient occurrences of extremely low slope and duration are disregarded from the analysis in order to mitigate the introduced pessimism.

By appropriately positioning the identical input transients (as indicated in Figure 5.6), a minimum-maximum characterization is performed with HSPICE. As hinted before, this cell extraction is Monte Carlo driven. Each of the output transient parameters (slope, duration and delay) is dependent upon the potential input transient arrival time (determined by the design and other variations of the previous logic levels) and the likely response of the current gate (determined by its own variations). To allow for realistic deviations in the transient electrical parameters obtained by this analysis, manufacturing and environmental variations are employed. Specifically, the following variations having typical Gaussian distributions are used to characterize the cells: 15% at  $3\sigma$  for all of the following: transistor channel length, transistor threshold voltage and chip power supply ( $V_{dd}$ ) in the IBM 65nm and the TSMC 180nm technologies. The obtained minimum-maximum parameter characteristics are subject to the successful statistical coverage of the corner cases by HSPICE.

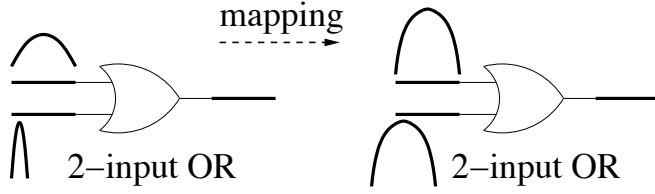


Figure 5.5. Input transient mapping to a slightly worse case of electrical parameters in the cell library

### 5.2.3 Logical masking

Logical masking or path sensitization in essence confines the propagation of transients only on specific paths that can allow hazards to be propagated to a primary output line. Traditionally, path sensitization is performed using a Monte Carlo input vector application [103]. This basically means that input vectors are repeatedly applied at the primary inputs of the circuit in order to reveal some of those hidden sensitizable paths. However, this is highly inefficient and unreliable for large circuits and thus leads to inconclusive results. Often, sensitizable paths are exhaustively stored in complex data structures [132]. This is an accurate approach although expensive and resource-demanding. In the proposed work, it is considered that all paths are sensitizable. Even though this constitutes a pessimistic approach, it guarantees a faster transient processing with an upper bound for the worst-case transient effects.

### 5.2.4 Experimental evaluation

The proposed approach is evaluated on a series of ISCAS85 and ISCAS89 benchmark circuits. Indicative results are taken on the worst-case duration of all observed transients at primary circuit outputs. The results of the proposed static analysis are compared against the outcome of dynamic analyses for the same experimental setup in each technology (Table 5.1 and 5.2). For the dynamic

SPICE-approximate evaluation in particular, results are taken both with and without path sensitization. For the proposed approach, it is assumed that all paths in logic are sensitizable. The experiments were performed in the  $65nm$  and  $180nm$  CMOS technologies in order to make additional observations on how technology scaling effects impact the observed transient effects in the presence of multiple hazards (Table 5.3 and 5.4).

Table 5.1. Electrical parameters of the injected transients used in the experimental setup ( $V_{dd} = 1 V$ ) (IBM 65nm)

Transient ID	Peak Amplitude	Duration	Slope
Transient 1	850 mV	150 ps	6 mV/ps
Transient 2	750 mV	150 ps	5 mV/ps
Transient 3	800 mV	50 ps	16 mV/ps
Transient 4	1000 mV	125 ps	40 mV/ps

Table 5.2. Electrical parameters of the injected transients used in the experimental setup ( $V_{dd} = 1.8 V$ ) (TSMC 180nm)

Transient ID	Peak Amplitude	Duration	Slope
Transient 1	1250 mV	150 ps	8. mV/ps
Transient 2	1150 mV	150 ps	7 mV/ps
Transient 3	1200 mV	50 ps	24 mV/ps
Transient 4	1600 mV	125 ps	64 mV/ps

The experimental evaluation shows that conventional dynamic approaches underestimate the potential transient effects in a circuit because of lack of delay uncertainty in the analysis. On the contrary, a static approach incorporates flexibility in the arrival time, slope and duration of propagated transients that

allows it to capture their likely severe effects in a circuit. Another observation is that for previous generations ( $180nm$  in Table 5.4), transient duration boosting occurred less frequently and less aggressively compared to newer technologies ( $65nm$  in Table 5.3). The results also show that, even though for single transient analysis it is anticipated that a Monte Carlo driven simulation will produce more optimistic results (due to path sensitization) than an analysis where all paths are assumed sensitized, this is not the case when multiple transients are involved (benchmark circuits  $c7552$ ,  $s38584$  in Table 5.3 and  $c1908$  in Table 5.4).

Table 5.3. Worst-case transient duration at primary outputs measured by static, dynamic sensitized and dynamic unsensitized analyses with the custom generated standard cells (IBM 65nm)

Benchmarks	Static (ps)	Monte Carlo Dynamic (ps)	Dynamic (w/o log. mask.) (ps)
c880	189	135	135
c1908	186	140	182
c2670	489	150.5	150.5
c3540	489	135	146
c7552	407	135	133
s9234	440	133	185
s15850	489	135	135
s38584	356	229	139

### 5.3 A LOGIC HARDENING TECHNIQUE USING C-ELEMENTS

#### 5.3.1 C-element overview

In this section it is explored how to eradicate transient effects at the primary outputs of a circuit with the least possible overhead. In particular, the use of

Table 5.4. Worst-case transient duration at primary outputs measured by static, dynamic sensitized and dynamic unsensitized analyses with the custom generated standard cells (TSMC 180nm)

Benchmarks	Static (ps)	Monte Carlo Dynamic (ps)	Dynamic (w/o log. mask.) (ps)
c880	184	171	171
c1908	145	136	108
c2670	482	150.5	150.5
c3540	125	125	125
c7552	139	121	121
s9234	482	0	0
s15850	182	123	123
s38584	182	146	146

hazard filter circuits is examined for that purpose. Such a popular circuit is the well-known *C-element* (Figure 5.7). The latter's main objective and traditional use is to prevent pulses of up to a specific duration from propagating to its output node. This particular duration is known as the C-element's *threshold*.

The basic operation of this circuit is as follows: an incoming pulse is delayed at one of the inputs by a delay unit for an amount of time equal to the threshold of the element, whereas the same pulse is immediately applied to the other input of the circuit. As a consequence, the transistor located further from the output node (either pmos or nmos depending on the incoming pulse polarity) conducts and rapidly opens a path to either the power supply or the ground network. The device closer to the output line is controlled by the delayed signal. If the pulse has duration less than the threshold of the element (equal to unit delay), then both transistors will open at different times and hence the pulse is not propagated to

the output. If the duration of the pulse is larger than the element's threshold, then the transistors will conduct for a common period of time allowing thus the full or attenuated propagation of the incoming signal. The associated delays of a minimum size C-element circuit for different thresholds are given in Table 5.5.

Table 5.5. Delay overhead for a minimum-size C-element circuit

Threshold (ps)	Delay (ps) (65nm)	Delay (ps) (180nm)
100	110	247
200	211	352
400	411	533
800	810	892

Upon computing an upper bound for the worst-case transient effects that may be observed at a primary output line (by using the static analysis of the previous section), the C-element circuit can be effectively used to block outgoing hazards. By doing this, it is possible to protect the memory elements or other sensitive components that read data off the combinational logic primary outputs. The static analysis approach that was presented previously can provide with the appropriate C-element size and threshold in order to protect the digital circuit of interest.

Besides a crude placement of filters at the primary outputs of a circuit, a distributed placement within the main circuit is also possible. In fact, an internal placement of filters is anticipated to reduce the delay overhead that is imposed by the direct and unrefined insertion of C-elements in the circuit outputs. In the following, different algorithms are investigated and evaluated for identifying efficient insertion methods of such protective circuitry.

### 5.3.2 C-element insertion heuristics

In the following, several investigated algorithms for filter insertion within a circuit are presented. After executing each heuristic, a graph updating subroutine is called in order to adjust the existing transients to the current circuit setup (Algorithm 1). In the following, the processed circuit gate and the filter threshold are denoted with  $G$  and  $\theta$  respectively. The studied algorithms for filter placement at a gate's inputs are: on all gate inputs conveying transients (*All*), on gate input(s) with the strongest transient (*Max*), on gate input(s) with the weakest transient (*Min*), on gate input(s) with most transients (*Most*) and on gate input(s) with the least critical path delay (*Least Critical*). For the latter, an additional subroutine for critical path delay computation based on a depth first search traversal of a graph is utilized. The above heuristics are described in Algorithms 2 to 6.

**Input:** Circuit graph

**Output:** Updated circuit graph

```
1 foreach gate  $G$  do
2     retrieve all input transient(s)  $w_i$ ;
3     compute gate  $G$  response;
4     assign  $G$  output transient(s)  $w_o$ ;
5 end
```

**Algorithm 1:** Update Graph Subroutine

### 5.3.3 Experimental evaluation

For the same benchmark circuits for which a static transient propagation was applied in the previous section, indicative experimental results are taken with the

**Input:** Gate  $G$  with at least one transient at inputs

**Output:** Gate  $G$  with filter circuits at all input with hazards

```
1 retrieve all input transient(s)  $w_i$ ;  
2 compute gate  $G$  response;  
3 if at least one  $G$  output transient  $w_o$  has duration > filter threshold  $\theta$   
then  
4   for all  $G$  input lines each having at least one transient do  
5     insert filter circuit on this input line;  
6     eliminate all transients for this line;  
7   end  
8   eliminate all gate  $G$  output transients;  
9   Update Graph Subroutine;  
10 end
```

**Algorithm 2:** Filter insertion for all  $G$  input lines with transients

**Input:** Gate  $G$  with at least one transient at inputs

**Output:** Gate  $G$  with filter circuit(s) at input lines

- 1 retrieve all input transient(s)  $w_i$ ;
- 2 compute gate  $G$  response;
- 3 **while** at least one  $G$  output transient  $w_o$  has duration  $> \text{filter threshold } \theta$  **do**
- 4     identify input line with max duration transient;
- 5     insert filter circuit on this input line;
- 6     eliminate all transients for this line;
- 7     retrieve all input transient(s)  $w_i$ ;
- 8     compute gate  $G$  response;
- 9     assign  $G$  output transient(s)  $w_o$ ;
- 10 **end**
- 11 Update Graph Subroutine;

**Algorithm 3:** Filter insertion for  $G$  input line with max duration transient

**Input:** Gate  $G$  with at least one transient at inputs

**Output:** Gate  $G$  with filter circuit(s) at input lines

- 1 retrieve all input transient(s)  $w_i$ ;
- 2 compute gate  $G$  response;
- 3 **while** *at least one G output transient  $w_o$  has duration > filter threshold  $\theta$*  **do**
- 4     identify input line with min duration transient;
- 5     insert filter circuit on this input line;
- 6     eliminate all transients for this line;
- 7     retrieve all input transient(s)  $w_i$ ;
- 8     compute gate  $G$  response;
- 9     assign  $G$  output transient(s)  $w_o$ ;
- 10 **end**
- 11 Update Graph Subroutine;

**Algorithm 4:** Filter insertion for  $G$  input line with min duration transient

**Input:** Gate  $G$  with at least one transient at inputs

**Output:** Gate  $G$  with filter circuit(s) at input lines

- 1 retrieve all input transient(s)  $w_i$ ;
- 2 compute gate  $G$  response;
- 3 **while** *at least one G output transient  $w_o$  has duration > filter threshold  $\theta$*  **do**
- 4     identify input line with most transients;
- 5     insert filter circuit on this input line;
- 6     eliminate all transients for this line;
- 7     retrieve all input transient(s)  $w_i$ ;
- 8     compute gate  $G$  response;
- 9     assign  $G$  output transient(s)  $w_o$ ;
- 10 **end**
- 11 Update Graph Subroutine;

**Algorithm 5:** Filter insertion for  $G$  input line with most transients

**Input:** Gate  $G$  with at least one transient at inputs

**Output:** Gate  $G$  with filter circuit(s) at input lines

- 1 retrieve all input transient(s)  $w_i$ ;
- 2 compute gate  $G$  response;
- 3 **while** *at least one G output transient  $w_o$  has duration > filter threshold  $\theta$*  **do**
- 4     identify input line with least critical delay;
- 5     insert filter circuit on this input line;
- 6     eliminate all transients for this line;
- 7     retrieve all input transient(s)  $w_i$ ;
- 8     compute gate  $G$  response;
- 9     assign  $G$  output transient(s)  $w_o$ ;
- 10 **end**
- 11 Update Graph Subroutine;

**Algorithm 6:** Filter insertion for  $G$  input line with least critical delay

introduced heuristics. For each of the above heuristics, the area (overall number of filter circuits) for the internal and primary output lines are reported in the  $65nm$  and  $180nm$  CMOS processes (Table 5.6 and 5.7). Also, the critical path delay overhead imposed by the filter circuits for each heuristic is calculated. The computation of the introduced delay is accomplished by performing a typical depth first search traversal in the investigated circuit graph.

From the results in Tables 5.6 and 5.7, it is indicated that the most filter circuits are usually required when gate input lines with the least critical path delay are engaged. This is a sensible result since area and hardware are traded off for less delay degradation. In some cases, even a greedy approach (*i.e., All*) involves less C-element circuits but that comes with a higher delay penalty. Another interesting result is that when this greedy heuristic (*i.e., All*) is followed, then the minimum number of filters at the primary outputs is usually obtained (Table 5.6).

#### 5.4 CONCLUSIONS

Multiple transient interaction, either due to re-convergent paths or to more than one simultaneous source sites in a circuit, is frequently underestimated in combinational logic. The ever increasing gate delay uncertainty and circuit node vulnerability to heavy energy ions give rise to frequent circuit failures. This research attempts to estimate an upper bound in the potential worst-case transient effects in logic and provides a first step towards developing methods for radiation tolerant combinational logic.

Table 5.6. Filter insertion with the several investigated heuristics (IBM 65nm)

Benchmarks	All heur.	Max heur.	Min heur.	Most heur.	Least Critical
c880	0 in	0 in	0 in	0 in	0 in
	9 PO	9 PO	9 PO	9 PO	9 PO
	9 Total	9 Total	9 Total	9 Total	9 Total
c2670	7 in	3 in	3 in	3 in	7 in
	9 PO	9 PO	9 PO	9 PO	8 PO
	16 Total	12 Total	12 Total	12 Total	15 Total
c3540	31 in	20 in	26 in	23 in	26 in
	0 PO	15 PO	14 PO	15 PO	13 PO
	31 Total	35 Total	40 Total	38 Total	39 Total
s9234	10 in	8 in	8 in	8 in	9 in
	12 PO	14 PO	18 PO	14 PO	18 PO
	22 Total	22 Total	26 Total	22 Total	27 Total

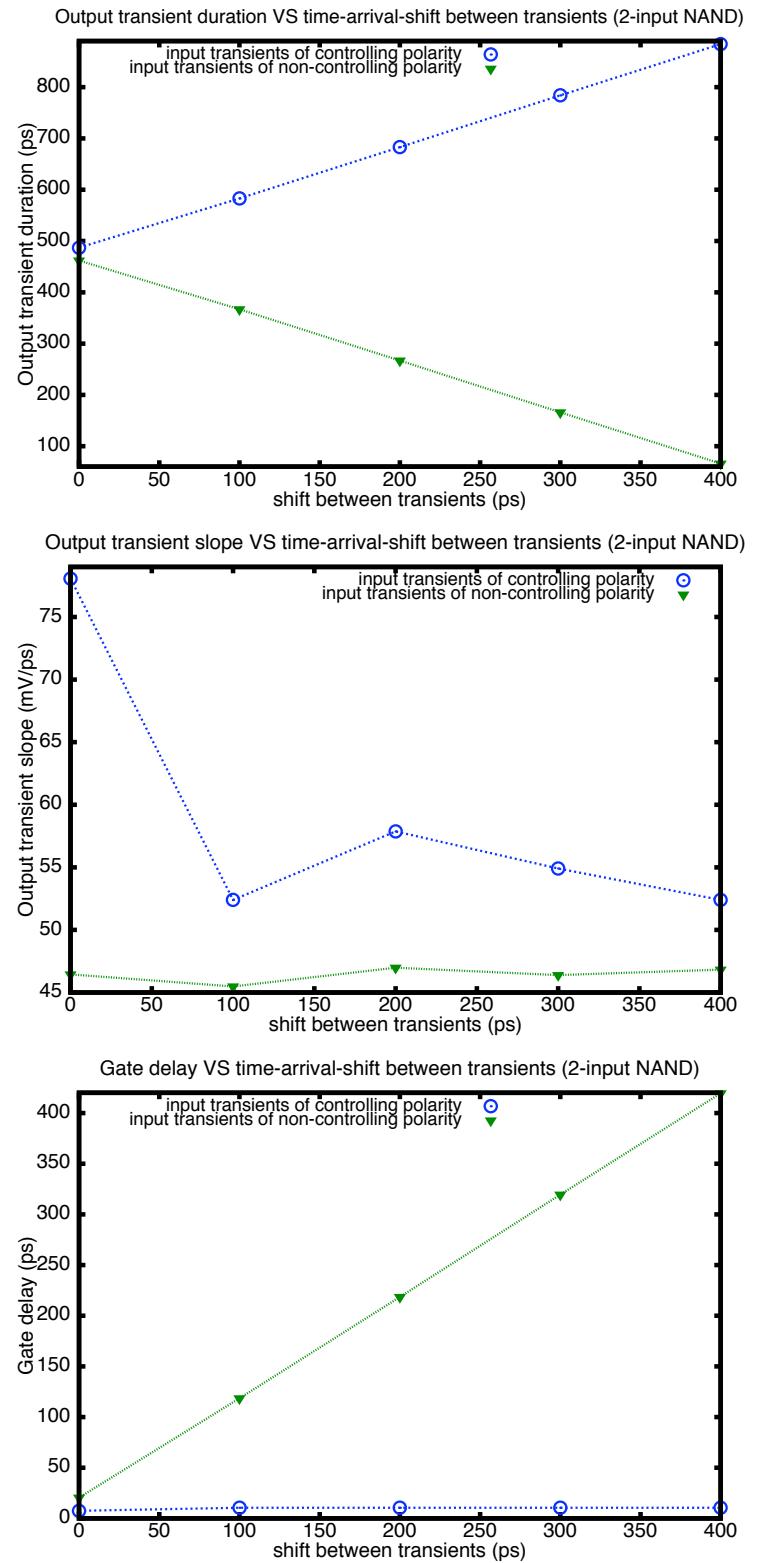


Figure 5.6. Min-max output transient characterization for different arrival times of the input transients

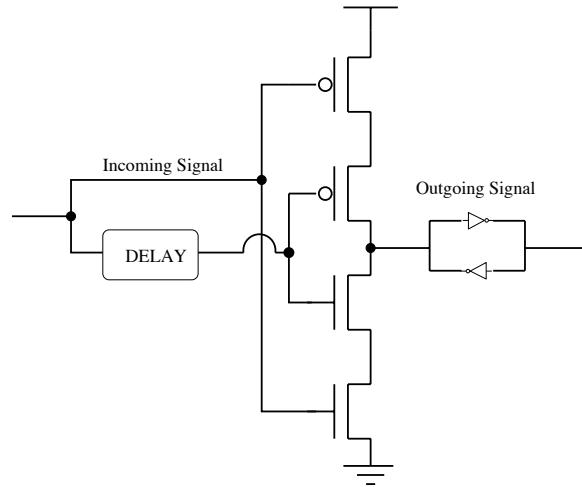


Figure 5.7. The basic C-element circuit

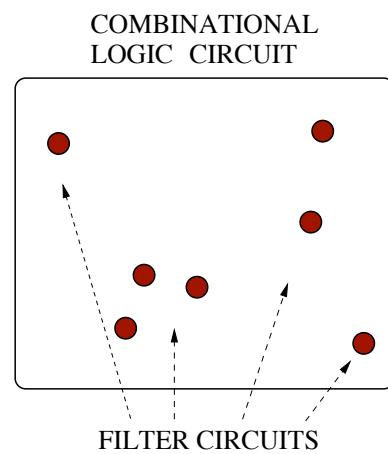


Figure 5.8. Illustration of C-element placement for protection

Table 5.7. Filter insertion with the several investigated heuristics (TSMC 180nm)

Benchmarks	All heur.	Max heur.	Min heur.	Most heur.	Least Critical
c880	0 in	0 in	0 in	0 in	0 in
	2 PO	2 PO	2 PO	2 PO	2 PO
	2 Total	2 Total	2 Total	2 Total	2 Total
c2670	5 in	3 in	4 in	3 in	6 in
	6 PO	6 PO	6 PO	6 PO	6 PO
	11 Total	9 Total	10 Total	9 Total	12 Total
c3540	2 in	1 in	1 in	1 in	1 in
	1 PO	1 PO	1 PO	1 PO	1 PO
	3 Total	2 Total	2 Total	2 Total	2 Total
s9234	15 in	7 in	7 in	7 in	17 in
	1 PO	1 PO	1 PO	1 PO	8 PO
	16 Total	8 Total	8 Total	8 Total	25 Total

## REFERENCES

- [1] J. D. Meindl, "Interconnect Opportunities For Gigascale Integration", *Micro IEEE*, Vol. 3 Issue 3, May-June 2003, pp. 28-35
- [2] A. Deutsch et al., "On-Chip Wiring Design Challenges for Gigahertz Operation", *Proceedings of the IEEE*, Vol. 89, Issue 4, April 2001, pp. 529-555
- [3] A. Sinha, S. K. Gupta, M. A. Breuer, "Validation and Test Issues Related to Noise Induced by Parasitic Inductances of VLSI Interconnects", *IEEE Trans. Advanced Packaging*, Vol. 25, Issue 3, August 2002, pp. 329-339
- [4] D. Deschacht, A. Lopez, "Crosstalk evaluation: the influence of inductance and routing orientation", *Proc. Int. Conf. Microelectronics*, 2004, pp. 185-188
- [5] R. Hossain, F. Viglione, M. Cavalli, "Designing Fast On-Chip Interconnects for Deep Submicrometer Technologies", *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 11, April 2003, pp. 276-280
- [6] T. Sakurai, "Closed-Form Expressions for Interconnection Delay, Coupling, and Crosstalk in VLSI's", *IEEE Trans. Electron Devices*, Vol. 40, Issue 1, January 1993, pp. 118-124
- [7] P. Heydari, M. Pedram, "Capacitive Coupling Noise in High-Speed VLSI Circuits", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 24, Issue 3, March 2005, pp. 478-488
- [8] M. Kuhlmann, S. S. Sapatnekar, "Exact and Efficient Crosstalk Estimation", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 20, Issue 7, July 2001, pp. 858-866
- [9] A. B. Kahng, S. Muddu, "An Analytical Delay Model for RLC Interconnects", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 16, Issue 12, December 1997, pp. 1507-1514
- [10] R. Venkatesan, J. A. Davis, J. D. Meindl, "Time Delay, Crosstalk and Repeater

- Insertion Models For High Performance SOC's", *IEEE Int. ASIC/SOC Conf.*, September 2002, pp. 404-408
- [11] A. B. Kahng, S. Muddu, E. Sarto, and R. Sharma, "Interconnect Tuning Strategies for High-Performance ICs", *Proc. Design Automation and Test in Europe*, February 1998, pp. 471-478
- [12] K. Agarwal, D. Sylvester, D. Blaauw, "Dynamic Clamping: On-Chip Dynamic Shielding and Termination for High-Speed RLC Buses", *Int. Symp. System-on-Chip*, November 2003, pp. 97-100
- [13] C. Metra, M. Favalli, B. Ricco, "Self-Checking Detection and Diagnosis of Transient, Delay, and Crosstalk Faults Affecting Bus Lines", *IEEE Trans. Computers*, Vol. 49, Issue 6, June 2000, pp. 560-574
- [14] N. Hanchate, N. Ranganathan, "Simultaneous Interconnect Delay and Crosstalk Noise Optimization through Gate Sizing Using Game Theory", *IEEE Trans. Computers*, Vol. 55, Issue 8, August 2006
- [15] O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization", *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 5, Issue 4, December 1997, pp. 465-472
- [16] T. Xue, E. Kuh, D. Wang, "Post Global Routing Crosstalk Synthesis", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 16, Issue 12, December 1997, pp. 1418-1430
- [17] P. Saxena, C. L. Liu, "A Postprocessing Algorithm for Crosstalk-Driven Wire Perturbation", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 19, Issue 6, June 2000, pp. 691-702
- [18] A. Vittal, M. Marek-Sadowska, "Crosstalk Reduction for VLSI", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 16, Issue 3, March 1997, pp. 290-298
- [19] D. Rossi, A. K. Nieuwland, C. Metra, "Simultaneous Switching Noise: The Re-

- lation between Bus Layout and Coding”, *IEEE Design and Test of Computers*, Vol. 25, Issue 1, January-February 2008, pp. 76-86
- [20] D. Rossi, A. K. Nieuwland, A. Katoch, C. Metra, ”Exploiting ECC Redundancy to Minimize Crosstalk Impact”, *IEEE Design and Test of Computers*, Vol. 22, Issue 1, January 2005, pp. 59-70
- [21] D. Rossi, A. K. Nieuwland, A. Katoch, C. Metra, ”New ECC for Crosstalk Impact Minimization”, *IEEE Design and Test of Computers*, Vol. 22, Issue 4, July-August 2005, pp. 340-348
- [22] B. Victor, K. Keutzer, ”Bus Encoding to Prevent Crosstalk Delay”, *IEEE Int. Conf. Computer Aided Design*, November 2001, pp. 57-63
- [23] M. Anders, N. Rai, R. K. Krishnamurthy, S. Borkar, ”A Transition-Encoded Dynamic Bus Technique for High-Performance Interconnects”, *IEEE Journal Solid-State Circuits*, Vol. 38, Issue 5, May 2003, pp. 709-714
- [24] R. Ayoub, A. Orailoglu, ”A Unified Transformational Approach for Reductions in Fault Vulnerability, Power, and Crosstalk Noise & Delay on Processor Buses”, *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Vol. 2, January 2005, pp. 729-734
- [25] D. Pamunuwa, H. Tenhunen, ”Repeater Insertion To Minimise Delay in Coupled Interconnects”, *Int. Conf. VLSI Design*, January 2001, pp. 513-517
- [26] Pamunuwa, D., Tenhunen H., On dynamic delay and repeater insertion in distributed capacitively coupled interconnects”, *Proceedings International Symposium on Quality Electronic Design*, 18-21 March 2002, pp. 240-245
- [27] Y. I. Ismail, E. Friedman, ”Optimum Repeater Insertion Based on a CMOS Delay Model for On-Chip RLC interconnect”, *IEEE Int. ASIC Conf.*, September 1998, pp. 369-373
- [28] Guoqing Chen, Friedman E.G., Low-power repeaters driving RC and RLC interconnects with delay and bandwidth constraints, *IEEE Trans. Very Large*

*Scale Integration (VLSI) Systems*, Vol. 14, Feb. 2006, pp. 161-172

- [29] C. J. Alpert, A. Devgan, S. T. Quay, "Buffer Insertion for Noise and Delay Optimization", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 18, Issue 11, November 1999, 1633-1645
- [30] M. N. Skoufis, H. Wang, T. Haniotakis, S. Tragoudas, "Glitch control with Dynamic Receiver Threshold Adjustment", *Int. Symp. Quality Electronic Design (ISQED)*, March 2007, pp. 410-415
- [31] I.H.R. Jiang, Y.W. Chang, and J. Y. Jou, "Crosstalk driven interconnect optimization by simultaneous gate and wire sizing," *IEEE Trans. Computer-Aided Design*, Vol. 19, Sept. 2000, pp. 999-1010
- [32] Tong Xiao, M. Marek-Sadowska, "Gate sizing to eliminate crosstalk induced timing violation," *Proc. Int. Conf. Computer Design*, 23-26 Sept. 2001, pp. 186 - 191
- [33] Chung-Ping Chen; N. Menezes, "Noise-aware repeater insertion and wire sizing for on-chip interconnect using hierarchical moment-matching," *Proc. Design Automation Conference*, 21-25 June 1999, pp. 502 - 506
- [34] P. Saxena, N. Menezes, P. Cocchini, D.A. Kirkpatrick, "Repeater scaling and its impact on CAD," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol 23, No. 4, April 2004, pp. 451 - 463
- [35] U. Narayanan, Ki-Seok Chung, Taewhan Kim, "Enhanced bus invert encodings for low-power," *IEEE Int. Symp. on Circuits and Systems*, Vol. 5, 26-29 May 2002, pp. 25 - 28
- [36] Naehyuck Chang, Kwanho Kim, Jinsung Cho, "Bus encoding for low-power high-performance memory systems," *Proc. in Design Automation Conference*, June 5-9 2000, pp. 800 - 805
- [37] R.R. Rao, H.S. Deogun, D. Blaauw, D. Sylvester, "Bus encoding for total power reduction using a leakage-aware buffer configuration", *IEEE Trans. on Very*

*Large Scale Integration (VLSI) Systems*, Vol. 13, No. 12, Dec. 2005, pp. 1376 - 1383

- [38] Youngsoo Shin, T. Sakurai, "Coupling-driven bus design for low-power application-specific systems," *Proc. Design Automation Conference*, 2001, pp. 750 - 753
- [39] W.J. Daly, B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks", *Proc. in Design Automation Conference*, June 18-22 2001
- [40] P. Subrahmanyam, R. Manimegalai, V. Kamakoti, "A Bus Encoding Technique for Power and Cross-talk Minimization", *Proc. Int. Conf. VLSI Design*, 2004, pp. 443-448
- [41] M. R. Stan, W. P. Burleson, "Bus-Invert Coding for Low-Power I/O", *IEEE Trans. VLSI Systems*, Vol. 3, No. 1, March 1995, pp. 49-58
- [42] Y. Zhang, J. Lach, K. Skadron, M. R. Stan, "Odd/Even Bus Invert with Two-Phase Transfer for Buses with Coupling", *Proc. Int. Symp. Low Power Electronics and Design (ISLPED)*, 2002, pp. 80-83
- [43] M. N. Skoufis, K. Karmarkar, T. Haniotakis, S. Tragoudas, "A High-Performance Bus Architecture for Strongly Coupled Interconnects", *Proc. Int. Symp. Quality Electronic Design*, March 2008, pp. 407-410
- [44] H. Kaul, D. Sylvester, M. Anders, R. Krishnamurthy, "Spatial Encoding Circuit Techniques for Peak Power Reduction of On-Chip High-Performance Buses", *Proc. Int. Symp. Low Power Electronics and Design*, August 2004, pp. 194-199
- [45] K-W Kim, K-H Baek, N. Shanbhag, C.L. Liu, S-M Kang, "Coupling-Driven Signal Encoding Scheme for Low-Power Interface Design", *IEEE/ACM Int. Conf. Computer Aided Design*, November 2000, pp. 318-321
- [46] K-H Baek, K-W Kim, S-M Kang, "A Low Energy Encoding Technique for Reduction of Coupling Effects in SoC Interconnects", *Proc. Midwest Symp. Circuits and Systems*, Vol.1, August 2000, pp. 80-83

- [47] C-G Lyuh, T. Kim, "Low Power Bus Encoding With Crosstalk Delay Elimination", *Int. ASIC/SOC Conf.*, September 2002, pp. 389-393
- [48] Y. Ran, M. Marek-Sadowska, "Crosstalk Noise in FPGAs", *Proc. Design Automation Conference*, June 2003, pp. 944-949
- [49] C. Duan, A. Tirumala, S. P. Khatri, "Analysis and Avoidance of Cross-talk in On-Chip Buses", *Hot Interconnects Conf.*, August 2001, pp. 133-138
- [50] S. Srinivasaraghavan, W. Burleson, "Interconnect Effort - A Unification of Repeater Insertion and Logical Effort", *IEEE Proc. Ann. Symp. VLSI*, February 2003, pp. 55-61
- [51] M. Agarwal, K. Agarwal, D. Sylvester, D. Blaauw, "Statistical Modeling of Cross-Coupling Effects in VLSI Interconnects", *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, January 2005, Vol. 1, pp. 503-506
- [52] M. Mondal, K. Mohanram, Y. Massoud, "Parameter-Variation-Aware Analysis for Noise Robustness", *Proc. Int. Symp. Quality Electronic Design*, March 2007, pp. 655-659
- [53] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, V. De, "Parameter Variations and Impact on Circuits and Microarchitecture", *Proc. Design Automation Conference*, June 2003, pp. 338-342
- [54] J. Tschanz, K. Bowman, V. De, "Variation-Tolerant Circuits: Circuit Solutions and Techniques", *Proc. Design Automation Conference*, June 2005, pp. 762-763
- [55] O. Neiroukh, X. Song, "Improving the Process-Variation Tolerance of Digital Circuits Using Gate Sizing and Statistical Techniques", *Proc. Design, Automation and Test in Europe (DATE)*, 2005, Vol. 1, pp. 294-299
- [56] C. H. Kim, S. Hsu, R. Krishnamurthy, S. Borkar, K. Roy, "Self Calibrating Circuit Design for Variation Tolerant VLSI Systems", *IEEE Int. Symp. On-Line Testing (IOLTS)*, July 2005, pp. 100-105
- [57] J. Kil, J. Gu, C. H. Kim, "A High-Speed Variation-Tolerant Interconnect Tech-

- nique for Sub-Threshold Circuits Using Capacitive Boosting”, *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 16, No. 4, April 2008, pp. 456-465
- [58] A. B. Kahng, B. Liu, and X. Xu, ”Statistical Timing Analysis in the Presence of Signal-Integrity Effects,” *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, Vol. 26, No. 10, October 2007, pp. 1873-1877
- [59] E. Nigussie, S. Tuuna, J. Plosila, J. Isoaho, ”Analysis of Crosstalk and Process Variations Effects on On-Chip Interconnects,” *Int. Symp. System-on-Chip*, November 2006, pp. 1-4
- [60] E. Demircan, ”Effects of Interconnect Process Variations on Signal Integrity,” *Int. Conf. SOC*, September 2006, pp. 281-284
- [61] R. Tayade, S. Sundereswaran, J. Abraham, ”Small-Delay Defect Detection in the Presence of Process Variations”, *Proc. Int. Symp. Quality Electronic Design*, March 2007, pp. 711-716
- [62] Arizona State University - Predictive Technology Model at  
<http://www.eas.asu.edu/> ptm/
- [63] A. K. Nieuiland, A. Katoch, D. Rossi, C. Metra, ”Coding Techniques for Low Switching Noise in Fault Tolerant Busses,” *Proc. Int. On-Line Testing Symposium*, July 2005, pp. 183-189
- [64] L. Di Silvio, D. Rossi, C. Metra, ”Crosstalk Effect Minimization for Encoded Busses,” *Proc. Int. On-Line Testing Symposium*, July 2003, pp. 214-218
- [65] R. Kothe, C. Galke, H. T. Vierhaus, ”A Multi-Purpose Concept for SoC Self Test Including Diagnostic Features,” *Proc. Int. On-Line Testing Symposium*, July 2005, pp. 241-246
- [66] C. Metra, M. Favalli, B. Ricco, ”On-Line Testing Scheme for Clock’s Faults,” *International Test Conference (ITC)*, 1997, pp. 587-596
- [67] Y. Zhao, L. Chen, S. Dey, ”On-line Testing of Multi-source Noise-induced Er-

- rors on the Interconnects and Buses of System-on-Chips," *International Test Conference (ITC)*, 2002, pp. 491-499
- [68] N. Venkateswaran, S. Balaji, V. Sridhar, "Fault Tolerant Bus Architecture for Deep Submicron Based Processors," *ACM SIGARCH Comp. Architecture News*, Vol. 33, No. 1, March 2005, pp. 148-155
- [69] D. Rossi, C. Steiner, C. Metra, "Analysis of the Impact of Bus Implemented EDCs on On-Chip SSN," *Design Automation and Test in Europe*, 2006
- [70] D. Bertozzi, L. Benini, G. De Michelli, "Error Control Schemes for On-Chip Communication Links: The Energy-Reliability Tradeoff," *IEEE Trans. Computer-Aided Des. of Integ. Circ. and Syst. (TCAD)*, Vol. 24, No. 6, June 2005, pp. 818-831
- [71] R. C. Baumann, "Soft Errors in Advanced Computer Systems," *IEEE Design and Test of Computers*, Vol. 22, No 3, 2005, pp.258-265
- [72] R. C. Baumann, "Radiation-induced Soft Errors in Advanced Semiconductor technologies," *IEEE Trans. Device and Materials Reliability*, Vol. 5, No 3, 2005, pp. 305-316
- [73] R. D. Schrimpf, P. Eaton, J. Benedetto, T. Turflinger, "Modeling and Verification of Single Event Transients in Deep Submicron Technologies", *IEEE Int. Symp. Reliability Physics*, 2004, pp. 673-674
- [74] P. Hazucha, C. Svensson, "Impact of CMOS technology Scaling on the Atmospheric Neutron Soft Error Rate," *IEEE Trans. Nuclear Science*, Vol. 47, No 6, 2000, pp. 2586-2594
- [75] J. F. Ziegler, "Terrestrial Cosmic Rays," *IBM J. Res. Develop.*, Vol. 40, No 1, pp. 19-39
- [76] S. Mitra, T. Karnik, N. Seifert, M. Zhang, "Logic Soft Errors in Sub-65nm technologies Design and CAD Challenges," *Proc. Design Automation Conference (DAC)*, 2005, pp. 2-4

- [77] P. Shivakumar, M.Kistler, S. W.Keckler, D.Burger, and L.Alvisi, "Modeling the Effect of Technology Trends On the Soft Error Rate of Combinational Logic," *Proc. International Conference on Dependable Systems and Networks*, 2002, pp. 389-398
- [78] N. Seifert *et al.*, "Radiation-Induced Clock Jitter and Race," *Proc. 43rd Annual IEEE Int. Reliability Physics Symp.* , 2005, pp. 215-222
- [79] Y. Zorian, V. A. Vardanian, K. Aleksanyan, K. Amirkhanyan, "Impact of Soft Error Challenge on SoC Design," *Proc. IEEE Int. On-Line Testing Symp. (IOLTS)*, 2005, pp. 63-68
- [80] F. W. Sexton, "Destructive Single-Event Effects in Semiconductor Devices and ICs," *IEEE Trans. Nuclear Science*, Vol. 50, No 3, 2003, pp. 603-621
- [81] K. Ramakrishnan, R. Rajaraman, S. Suresh, N. Vijaykrishnan, Y. Xie, M. J. Irwin, "Variation Impact on SER of Combinational Circuits," *Proc. Int. Symp. Quality Electronic Design (ISQED)*, 2007, pp. 911-916
- [82] D. Rossi, M. Omaña, F. Toma, C. Metra, "Multiple Transient Faults in Logic: An Issue for Next Generation ICs?," *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, 2005, pp. 352-360
- [83] J. Maiz, S. Hareland, K. Zhang, P. Armstrong, "Characterization of Multi-bit Soft Error Events in Advanced SRAMs," *Tech. Dig. IEEE Int. Electron Device Meeting (IEDM)*, 2003, pp. 21.4.1-21.4.4
- [84] S. Hellebrand, C. G. Zoellin, S. L. Torsten, "A Refined Electrical Model for Particle Strikes and its Impact on SEU Prediction," *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, 2007, pp. 50-58
- [85] M. Nicolaidis, R. Perez, "Measuring the Width of Transient Pulses Induced by Ionizing Radiation," *Proc. 41st Annual IEEE Int. Reliability Physics Symp.* , 2003, pp. 56-59
- [86] M. J. Gadlage *et al.*, "Single Event Transient Pulsewidths in Digital Microcir-

- cuits," *IEEE Trans. Nuclear Science*, Vol. 51, No 6, 2004, pp. 3285-3290
- [87] J. Benedetto *et al.*, "Heavy Ion-Induced Digital Single-Event Transients in Deep Submicron Processes," *IEEE Trans. Nuclear Science*, Vol. 51, No 6, 2004, pp. 3480-3485
- [88] P. Eaton *et al.*, "Single Event Transient Pulsewidth Measurements Using a Variable Temporal Latch Technique," *IEEE Trans. Nuclear Science*, Vol. 51, No 6, 2004, pp. 3365-3368
- [89] P. E. Dodd, M. R. Shaneyfelt, J. A. Felix, J. R. Schwank, "Production and Propagation of Single-Event Transients in High-Speed Digital Logic ICs," *IEEE Trans. Nuclear Science*, Vol. 51, No 6, 2004, pp. 3278-3284
- [90] J. M. Benedetto *et al.*, "Variation of Digital SET Widths and the Implications for Single Event Hardening of Advanced CMOS Processes," *IEEE Trans. Nuclear Science*, Vol. 52, No 6, 2005, pp. 2114-2119
- [91] D. G. Mavis, P. H. Eaton, "SEU and SET Modeling and Mitigation in Deep Sub-micron Technologies," *Proc. 45th Annual IEEE Int. Reliability Physics Symp.*, 2007, pp. 293-305
- [92] D. G. Mavis, P. H. Eaton, "Soft error rate mitigation techniques for modern microcircuits", *Proc. Int. Reliability Physics Symp.*, 2002, pp. 216-225
- [93] M. P. Baze, S. P. Buchner, "Attenuation of Single Event Induced Pulses in CMOS Combinational Logic," *IEEE Trans. Nuclear Science*, Vol. 44, No. 6, 1997, pp. 2217-2223
- [94] M. Omaña, G. Papasso, D. Rossi, C. Metra, "A Model for Transient Fault Propagation in Combinational Logic," *Proc. Int. On-Line Testing Symposium (IOLTS'03)*, 2003, pp. 111-115
- [95] K. Mohanram, "Simulation of Transients Caused by Single-Event Upsets in Combinational Logic," *Proc. Int. Test Conf. (ITC)*, 2005.
- [96] C. S. Amin, F. Dartu, Y. I. Ismail, "Weibull-Based Analytical Waveform

- Model," *IEEE Trans. Computer-Aided Design Integrated Circuits and Systems*, Vol. 24, No. 8, 2005, pp. 1156-1168
- [97] P. E. Dodd, L. W. Massengill, "Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics," *IEEE Trans. Nuclear Science*, Vol. 50, No. 3, 2003, pp. 583-602
- [98] T. Karnik, P. Hazucha, J. Patel, "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes," *IEEE Trans. Depend. and Sec. Comp.*, Vol. 1, No. 2, 2004, pp. 128-143
- [99] S. V. Walstra, C. Dai, "Circuit-Level Modeling of Soft Errors in Integrated Circuits," *IEEE Trans. Dev. and Mater. Rel.*, Vol. 5, No. 3, 2005, pp. 358-364
- [100] B. Zhang, W-S Wang, M. Orshansky, "FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based Designs," *Proc. Int. Symp. Quality Electronic Design (ISQED)*, 2006
- [101] Y. S. Dhillon, A. U. Diril, A. Chatterjee, A. D. Singh, "Analysis and Optimization of Nanometer CMOS Circuits for Soft-Error Tolerance," *IEEE Trans. VLSI Systems*, Vol. 14, No. 5, 2006, pp. 514-524
- [102] M. Zhang, N. Shanbhag, "Soft-Error-Rate-Analysis (SERA) Methodology," *IEEE Trans. Computer-Aided Design Integrated Circuits and Systems*, Vol. 25, No. 10, Oct. 2006
- [103] R. R. Rao, K. Chopra, D. Blaauw, D. M. Sylvester, "Computing the Soft Error Rate of a Combinational Logic Circuit Using Parameterized Descriptors," *IEEE Trans. Computer-Aided Design Integrated Circuits and Systems*, Vol. 26, No. 3, 2007, pp. 468-479
- [104] R. Ramanarayanan *et al.*, "Modeling Soft Errors at the Device and Logic Levels for Combinational Circuits," *IEEE Trans. Depend. and Sec. Comp.*, Vol. 5, No. 1, 2008
- [105] P. C. Murley, G. R. Srinivasan, "Soft-error Monte Carlo modeling program,

- SEMM," *IBM J. Res. Develop.*, Vol. 40, No. 1, 1996, pp. 109-118
- [106] H. H. K. Tang, "SEMM-2: A new generation of single-event-effect modeling tools," *IBM J. Res. Develop.*, Vol. 52, No. 3, 2008, pp. 233-244
- [107] N. Miskov-Zivanov, D. Marculescu, "MARS-C: Modeling and Reduction of Soft Errors in Combinational Circuits," *Proc. Design Automation Conference (DAC)*, 2006, pp. 767-772
- [108] C. Zhao, X. Bai, S. Dey, "A Scalable Soft Spot Analysis Methodology for Compound noise Effects in Nano-meter Circuits," *Proc. Design Automation Conf. (DAC)*, 2004, pp. 894-899
- [109] R. R. Rao, D. Blaauw, D. Sylvester, "Soft Error Reduction in Combinational Logic Using Gate Resizing and Flipflop Selection," *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, 2006, pp. 502-509
- [110] Q. Zhou, K. Mohanram, "Gate Sizing to Radiation Harden Combinational Logic," *IEEE Trans. Computer-Aided Design Integrated Circuits and Systems*, Vol. 25, No. 1, 2006, pp. 155-166
- [111] Q. Zhou, K. Mohanram, "Cost-Effective Radiation Hardening Technique for Combinational Logic," *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, 2004, pp. 100-106
- [112] P. Hazucha *et al.*, "Measurements and Analysis of SER-Tolerant Latch in a 90-nm Dual-VT CMOS Process," *IEEE Journal Solid State Circuits*, Vol. 39, No. 9, 2004, pp. 1536-1543
- [113] M. Zhang *et al.*, "Sequential Element Design With Built-In Soft Error Resilience," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 14, No. 12, 2006, pp. 1368-1378
- [114] M. Omaña, D. Rossi, C. Metra, "Latch Susceptibility to Transient Faults and New Hardening Approach," *IEEE Trans. Computers*, Vol. 56, No. 9, pp 1255-1268

- [115] M. Zhang *et al.*, "Design for Resilience to Soft Errors and Variations," *Proc. IEEE Int. Symp. On-Line Testing*, 2007, pp. 23-28
- [116] M. Fazeli, A. Patooghy, S. G. Miremadi, A. Ejlali, "Feedback Redundancy: A Power Efficient SEU-Tolerant Latch Design for Deep Sub-Micron Technologies," *Proc. Int. Conf. Dependable Systems and Networks (DSN)*, 2007, pp. 276-285
- [117] A. Goel, S. Bhunia, H. Mahmoodi, K. Roy, "Low-Overhead Design of Soft-Error-Tolerant Scan Flip-Flops with Enhanced-Scan Capability," *Proc. Asia South Pacific Design Automation Conf.*, 2006, pp. 665-670
- [118] Y. Sasaki, K. Namba, H. Ito, "Soft Error Masking Circuit and Latch Using Schmitt Trigger Circuit," *Proc. IEEE Int. Symp. Defect Fault-Tolerance in VLSI Systems (DFT)*, 2006, pp. 327-335
- [119] R. Naseer, J. Draper, "DF-DICE: A Scalable Solution for Soft Error Tolerant Circuit Design," *Proc. Int. Symp. Circuits and Systems (ISCAS)*, 2006
- [120] V. Joshi, R. R. Rao, D. Blaauw, D. Sylvester, "Logic SER Reduction through Flipflop Redesign," *Proc. Int. Symp. Quality Electronic Design (ISQED)*, 2006
- [121] A. K. Nieuwland, S. Jasarevic, G. Jerin, "Combinational Logic Soft Error Analysis and Protection," *Proc. IEEE Int. On-Line Testing (IOLTS)*, 2006
- [122] T. Karnik *et al.*, "Impact of Body Bias on Alpha- and Neutron-Induced Soft Error Rates of Flip-flops," *Proc. VLSI Circuits Symp.*, 2004, pp. 324-325
- [123] M. Nicolaidis, "Design for Soft Error Mitigation," *IEEE Trans. Device and Materials Reliability*, Vol. 5, No. 3, 2005, pp. 405-418
- [124] S. Almukhaizim, Y. Makris, "Soft Error Mitigation Through Selective Addition of Functionally Redundant Wires," *IEEE Trans. Reliability*, Vol. 57, No. 1, 2008, pp. 23-31
- [125] M. Zhang, N. R. Shanbhag, "Dual-Sampling Skewed CMOS Design for Soft-Error Tolerance," *IEEE Trans. Circuits and Systems*, Vol. 53, No. 12, 2006, pp. 1461-1465

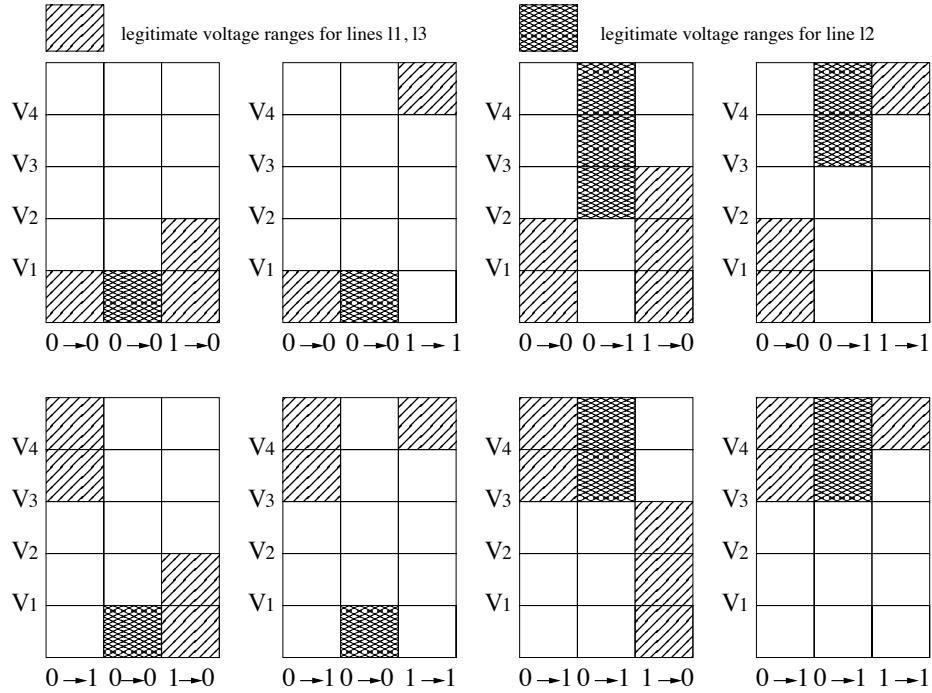
- [126] Q. Zhou, M. R. Choudhury, K. Mohanram, "Design Optimization for Robustness to Single-Event Upsets," *Proc. VLSI Test Symposium (VTS)*, 2006
- [127] H. S. Deogun, D. Sylvester, D. Blaauw, "Gate-Level Mitigation Techniques for Neutron-Induced Soft Error Rate," *Proc. Int. Symp. Quality Electronic Design*, 2005
- [128] M. Nicolaidis, "Time Redundancy Based Soft-Error Tolerance to Rescue Nanometer technologies," *Proc. VLSI Test Symp.*, 1999, pp. 86-94
- [129] K.-T. Cheng and H.C.Chen, "Delay Testing for non-robust Untestable Circuits", *Proc. International Test Conference (ITC)*, 1993, pp. 954-961
- [130] A. Krstic and K. T. Cheng, "Delay Fault Testing for VLSI Circuits," *norwell, MA: Kluwer*, 1998
- [131] S-I. Minato, "Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems", *Proc. Design Automation Conference (DAC)*, 1993, pp. 272-277
- [132] S. Padmanaban, S. Tragoudas, "Efficient Identification of (Critical) Testable Path Delay Faults Using Decision Diagrams," *IEEE Trans. Computer-Aided Design of Integrated Circuits Systems*, Vol. 24, No. 1, 2005
- [133] G. L. Smith, "Model for Delay Faults Based upon Paths," *Proc. International Test Conference (ITC)*, 1985, pp. 342-349
- [134] K. Heragu, J. H. Patel, and V. D. Agrawal, "Fast Identification of Untestable Delay Faults using Implications," *Proc. Int. Conf. Computer-Aided Design*, 1997, pp. 642-647

## **APPENDICES**

Table 5.8. Five-threshold characterization table for a victim (with one adjacent aggressor) with respect to the activity on its only aggressor

Voltage Ranges	Previous State		Next State
	1	0	
$V > V_5 [5-T]$	$R, Q$	$R$	<b>1</b>
			<b>0</b>
$V > V_4 [5-T]$	$R, Q, F$	$R, Q$	<b>1</b>
			<b>0</b>
$V > V_3 [5-T]$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V > V_2 [5-T]$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V > V_1 [5-T]$	$R, Q, F$	$R, Q, F$	<b>1</b>
			<b>0</b>
$V < V_5 [5-T]$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>
$V < V_4 [5-T]$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>
$V < V_3 [5-T]$			<b>1</b>
	$R, Q, F$	$R, Q, F$	<b>0</b>
$V < V_2 [5-T]$			<b>1</b>
	$Q, F$	$R, Q, F$	<b>0</b>
$V < V_1 [5-T]$			<b>1</b>
	$F$	$Q, F$	<b>0</b>

Initial Conditions:  $(l_1, l_2, l_3) = (0, 0, 1)$



Initial Conditions:  $(l_1, l_2, l_3) = (0, 1, 1)$

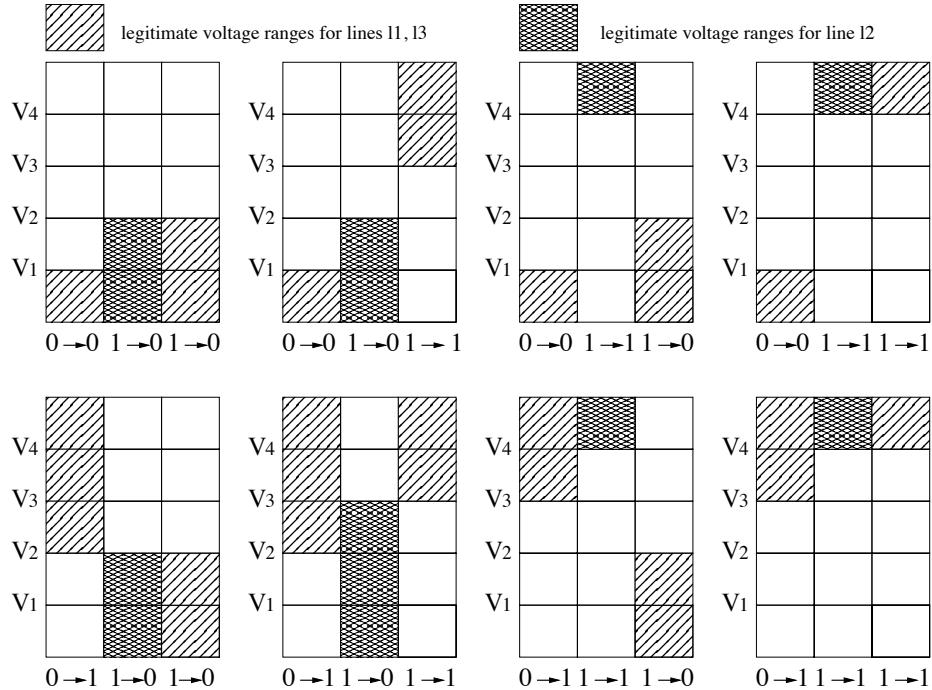
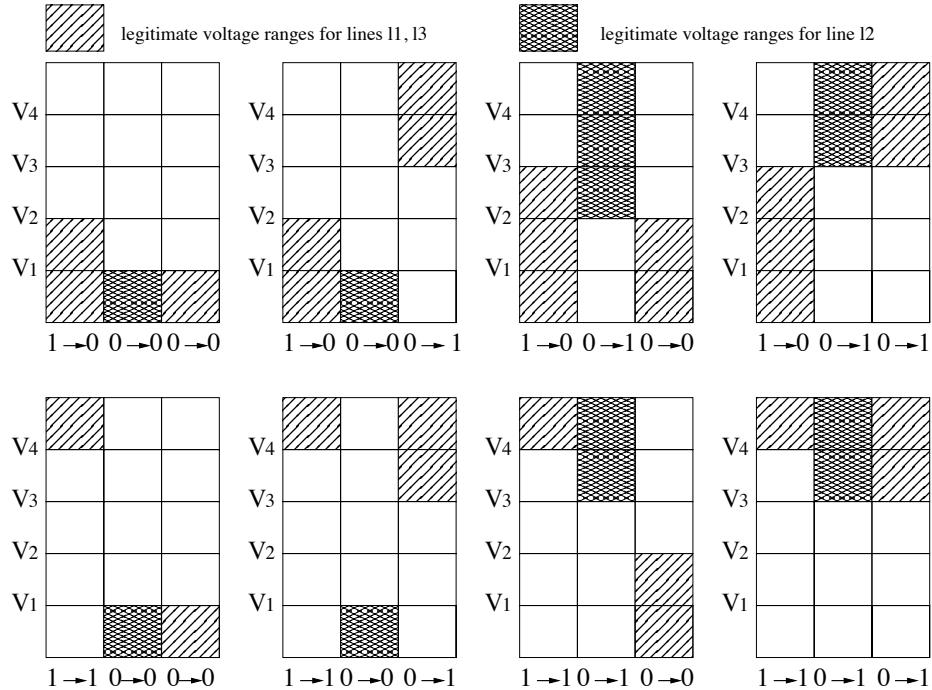


Figure 5.9. Error-free characterization of ranges in selected cases for a line with two adjacent aggressors

Initial Conditions:  $(l_1, l_2, l_3) = (1, 0, 0)$



Initial Conditions:  $(l_1, l_2, l_3) = (1, 0, 1)$

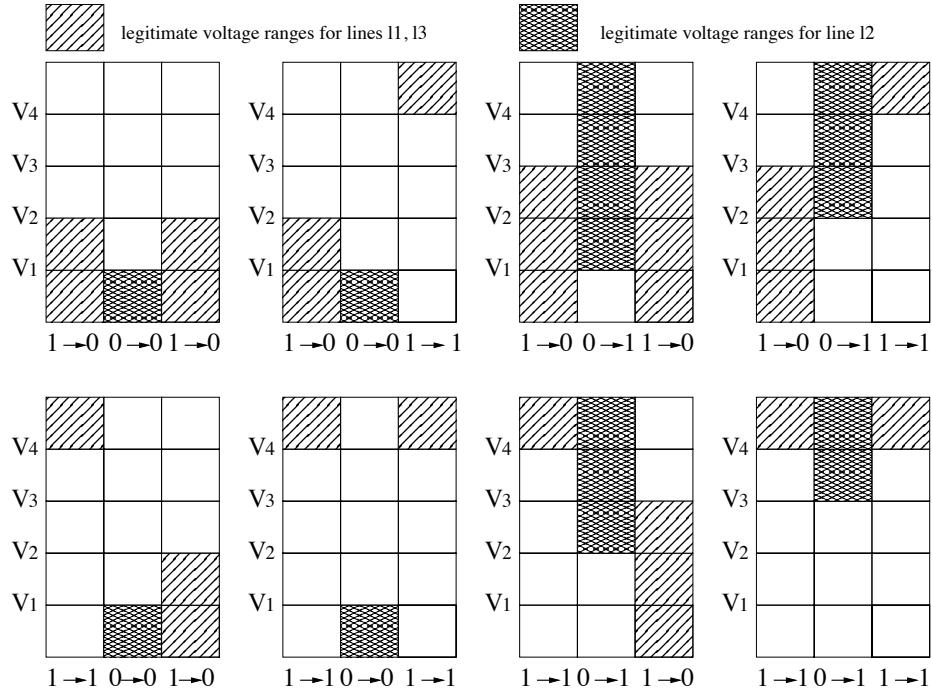
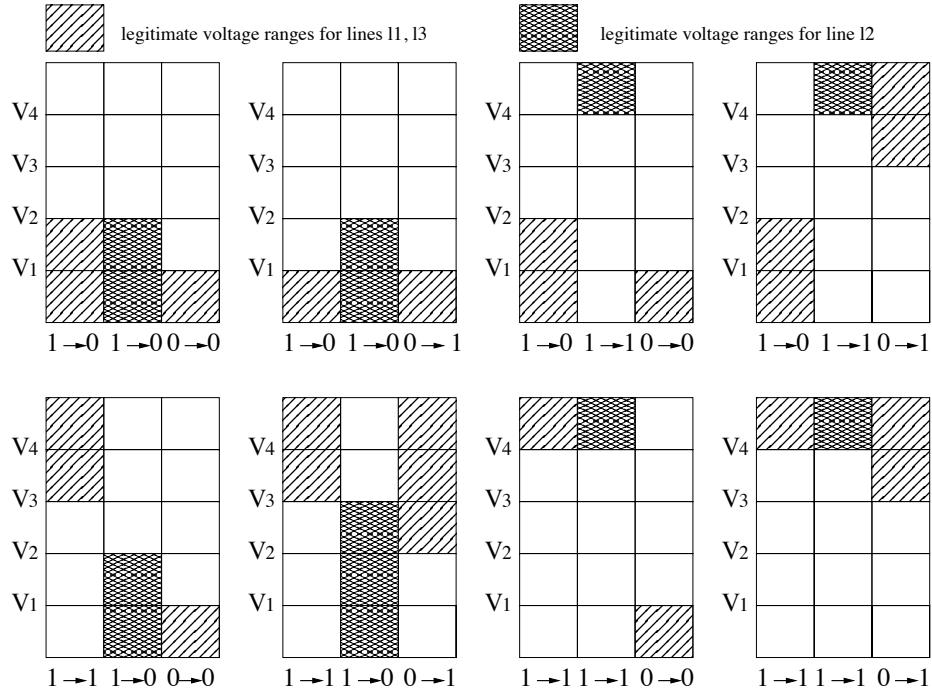


Figure 5.10. Error-free characterization of ranges in selected cases for a line with two adjacent aggressors

Initial Conditions:  $(l_1, l_2, l_3) = (1, 1, 0)$



Initial Conditions:  $(l_1, l_2, l_3) = (1, 1, 1)$

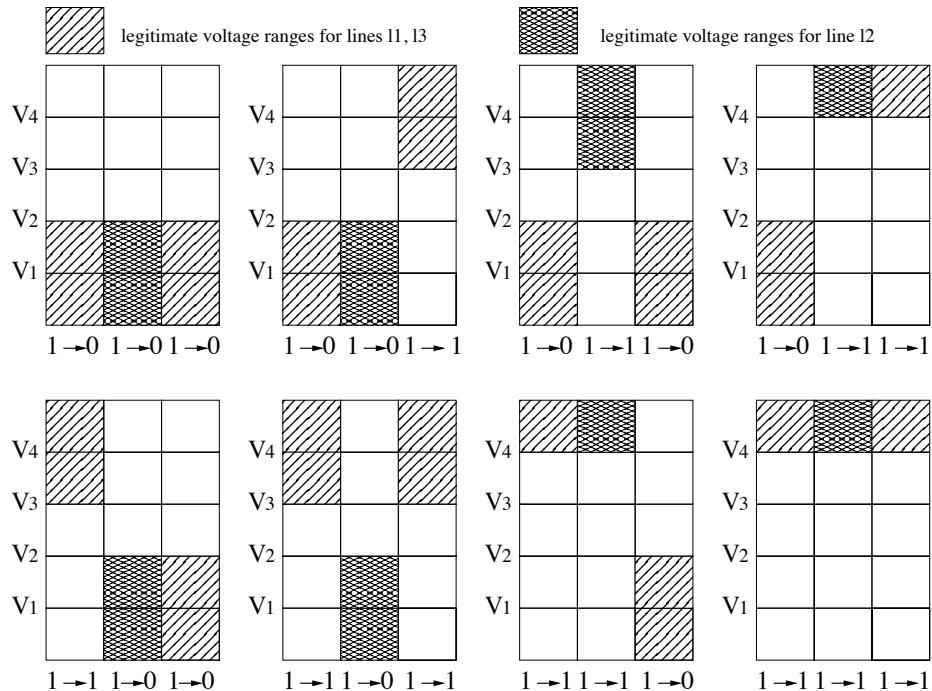


Figure 5.11. Error-free characterization of ranges in selected cases for a line with two adjacent aggressors

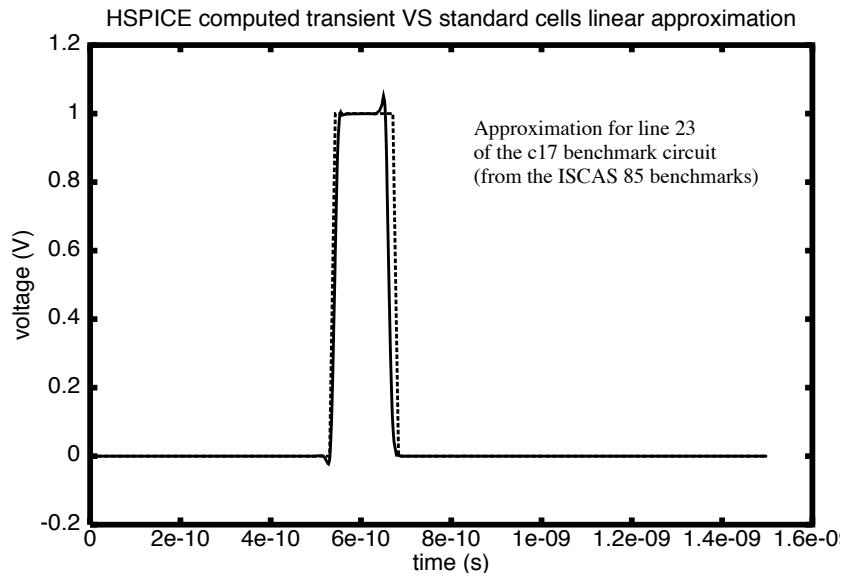


Figure 5.12. Output transient plot given by both HSPICE and custom standard cells for a primary output line in the c17 benchmark circuit. Transient stimulus is applied at a primary input.

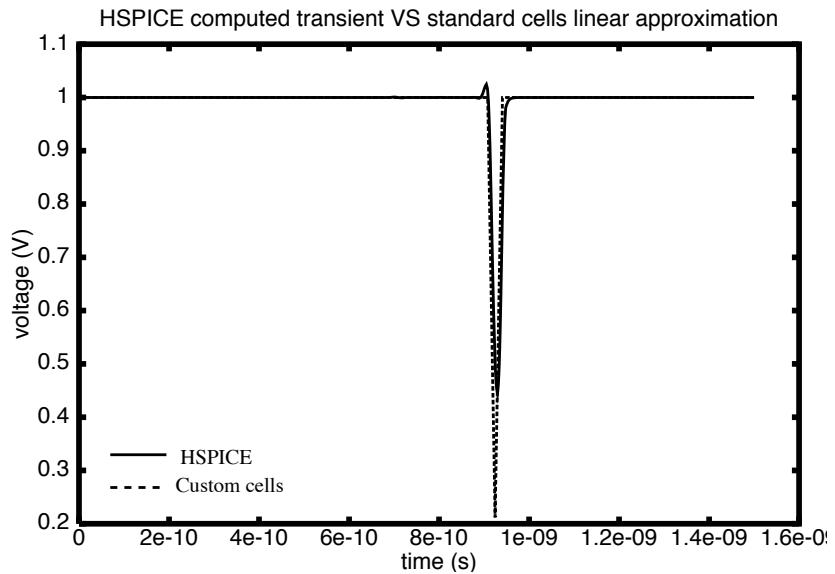


Figure 5.13. Output transient plot for a dynamic simulation given by both HSPICE and the custom cells for an internal line in a benchmark circuit (*s9234*). Random transients throughout the circuit are injected.

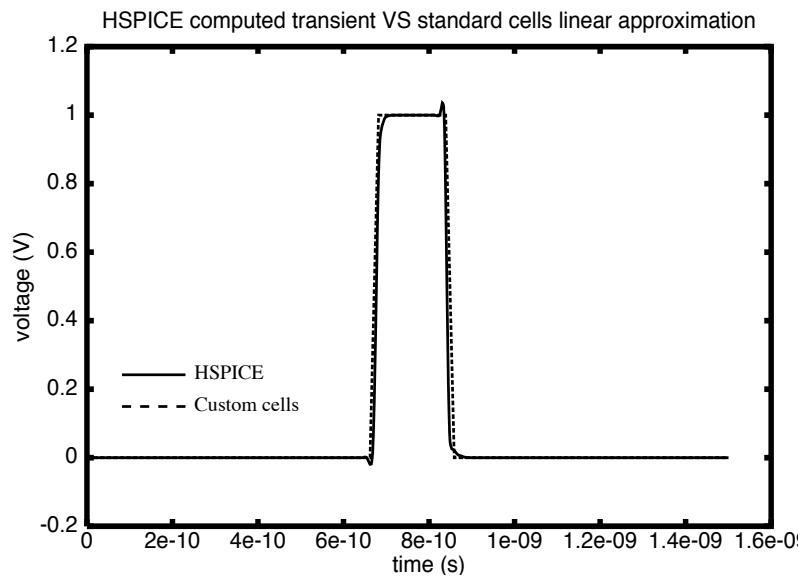


Figure 5.14. Output transient plot for a dynamic simulation given by both HSPICE and the custom cells for an internal line in a benchmark circuit (*s38584*). Random transients throughout the circuit are injected.

## VITA

Graduate School  
Southern Illinois University

MICHAEL N SKOUFIS

Date of Birth: July 29, 1974

1962 EVERGREEN TERRACE DR E #2, CARBONDALE, ILLINOIS 62901

Southern Illinois University at Carbondale  
Master's of Science, Electrical Engineering, December 2002

University of Illinois at Chicago  
Bachelor of Science, Computer Engineering, May 2001

Dissertation Title:

Coping With Delays and Hazards in Buses and Random Logic in Deep Sub-Micron

Major Professor: Dr. S. Tragoudas

Publications:

M. N. Skoufis, H. Wang, T. Haniotakis, S. Tragoudas, "Glitch Control with Dynamic Receiver Threshold Adjustment", *Int. Symp. Quality Electronic Design (ISQED)*, March 2007, pp. 410-415

M. N. Skoufis, K. Karmarkar, T. Haniotakis, S. Tragoudas, "A High-Performance Bus Architecture for Strongly Coupled Interconnects", *Int. Symp. Quality Electronic Design (ISQED)*, March 2008, pp. 407-410

S. Gangadhar, M. N. Skoufis, S. Tragoudas, "Propagation of Transients Along Sensitizable Paths", *Int. Symp. On-Line Testing (IOLTS)*, July 2008, pp. 129-134

M. N. Skoufis, H. Wang, S. Tragoudas, "A Method to Cope with Soft Errors", *Proc. Int. WSEAS Circuits Conf.*, Vol 11, 2007

M. N. Skoufis *et.al.*, "A Data Capturing Method for Buses on Chip", (*under second review in the IEEE Trans. on Circuits and Systems*)