

NOISE ANALYSIS OF FLEXING CROSSBARS UNDER THE VICTIM-AGGRESSOR MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Sertaç Erdemir
June, 2015

NOISE ANALYSIS OF FLEXING CROSSBARS UNDER THE
VICTIM-AGGRESSOR MODEL

By Sertaç Erdemir

June, 2015

We certify that we have read this thesis and that in our opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ezhan Karaşan (Advisor)

Prof. Dr. Ahmet Yavuz Oruç

Prof. Dr. Arif Bülent Özgüler

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

NOISE ANALYSIS OF FLEXING CROSSBARS UNDER THE VICTIM-AGGRESSOR MODEL

Sertaç Erdemir

M.S. in Electrical and Electronics Engineering

Advisor: Prof. Dr. Ezhan Karaşan

June, 2015

This study investigates the effects of crosstalk noise on flexing crossbars, and proposes an efficient method for estimation. The estimation method is also applicable to other submicron VLSI circuits. Circuit theory is utilized to estimate crosstalk emergence due to coupling effects and means of crosstalk reduction are investigated. Peak crosstalk noise amplitude, occurrence time, and time domain waveform are represented in closed form expressions. This research also introduces an empirical approach to compute the best case victim-aggressor alignment that minimizes the crosstalk noise on victim lines. In addition, it suggests a geometric approach reducing the adverse effects of crosstalk noise on flexing crossbars. Delay and signal quality for varied lengths of interconnect wires on interconnection networks using lossy transmission line theory are analyzed and examined in detail. Furthermore, crossbar networks are compared with other interconnection networks in terms of power consumptions.

Keywords: Flexing crossbars, crosstalk, noise analysis, power consumption.

ÖZET

ESNEK ÇAPRAZLAYICI ANAHTARLARIN ETKİLENEN-ETKİYEN MODELİ ALTINDA GÜRÜLTÜ ANALİZİ

Sertaç Erdemir

Elektrik ve Elektronik Mühendisliği, Yüksek Lisans

Tez Danışmanı: Prof. Dr. Ezhan Kardeş

Haziran, 2015

Bu çalışma, esnek çaprazlayıcı anahtarlarda çapraz karışma gürültüsünün etkilerini incelemekte ve kestirim için verimli bir metot önermektedir. Kestirim metodu diğer VLSI devrelerine de uygulanabilmektedir. Kuplaj etkisinden kaynaklanan çapraz karışmanın kestiriminde, devre teorisinden yararlanılmış ve çapraz karışma azalımının etkileri incelenmiştir. Çapraz karışma gürültüsünün tepe genliği, oluşma zamanı ve zaman düzleminde dalga formu kapalı formda ifadelerle gösterilmiştir. Ayrıca bu araştırma, etkilenen sinyal yollarında çapraz karışma gürültüsünü minimize eden en iyi etkilenen-etkiyen yerleşimini hesaplamak için deneysel bir yaklaşım ileri sürmüştür. Ek olarak, esnek çaprazlayıcı anahtarlarda çapraz karışma gürültüsünün etkilerini azaltan geometrik bir yaklaşım önerilmiştir. Araba bağlantı anahtarları için çeşitli uzunluklardaki araba bağlantı yollarındaki yayılım gecikmesi ve sinyal kalitesi kayıplı iletim hatları teorisi kullanılarak analiz edilmiş ve detaylıca incelenmiştir. Ayrıca çaprazlayıcı anahtarlar, diğer araba bağlantı anahtarlarıyla güç tüketimi açısından karşılaştırılmıştır.

Anahtar sözcükler: Esnek çaprazlayıcı anahtarlar, çapraz karışma, hata analizi, güç tüketimi.

Acknowledgement

First and foremost, I owe my deepest gratitude to my brother, Aytaç Erdemir, for his tremendous support during my entire life. There are no words that express the depth of my gratitude for everything he has ever done for me. Without his help, it would be almost impossible to achieve anything that is already achieved with ease.

I am also thankful to my parents for the unceasing encouragement, endless support and attention. I am glad that I could achieve this degree to make them happy.

I want to express my sincere gratitude to Prof. Dr. Ahmet Yavuz Oruç and Prof. Dr. Ezhan Karaşan for their invaluable advices, guidance and insight throughout the study. I gratefully acknowledge their precious comments, criticism and encouragements.

I would like to acknowledge Prof. Dr. Arif Bülent Özgüler for reading and commenting on this thesis.

I place on record, my sincere thanks to ROKETSAN Inc. for their support and understanding.

Finally, I express my special thanks to Scientific and Technical Research Council of Turkey (TÜBİTAK) for their financial support.

Contents

1	Introduction	1
2	Literature Review	5
2.1	Interconnection Networks	6
2.2	Lumped Circuit Models of Interconnection Networks	10
3	On-Chip Interconnection Networks	14
3.1	Elementary Switching Structures	17
3.2	Binary Tree Switching Structures	18
3.3	Crossbar Switches	19
3.4	Flexing Crossbar Switches	20
3.5	Physical Realizations of Crossbar Switches	21
4	Interconnection Network Modeling	24
4.1	Transmission Line Model for Interconnect Wires	25
4.1.1	The Transmission Matrix	28
4.1.2	Delay and Signal Quality Analysis of Interconnect Wires	30
4.2	Lumped Circuit Models	35
4.3	Pass Transistors and Transmission Gates	39
5	Power Consumption and Crosstalk Analysis on Interconnection Networks	44
5.1	Power Consumption in Interconnection Networks	44
5.1.1	The Grid Model for Crossbar Switches	48
5.1.2	Flexing Crossbar Interconnection Network	48
5.1.3	Baseline Interconnection Network	50

5.1.4	Fully-Connected Interconnection Network	51
5.1.5	Power Consumption Analysis Results	52
5.2	Noise in Interconnection Networks	53
5.3	Crosstalk Noise in Flexing Crossbars	54
6	Conclusion	78
A	Code	89

List of Figures

3.1	Overview of a real time processing system.	15
3.2	General real time processing system model.	15
3.3	Block diagram of an interconnection network.	15
3.4	A topological classification of interconnection networks.	16
3.5	On-off and 2×2 elementary switches.	18
3.6	A 4×4 binary tree switch. Retrieved from [1].	19
3.7	A 4×4 crossbar switch. Retrieved from [1].	20
3.8	A 4×4 flexing crossbar. Retrieved from [1].	20
3.9	A 2-level binary tree crossbar switch with direct outputs and its crossbar realization. Retrieved from [1]	21
3.10	A 4×4 crossbar network made by mechanical switches. Retrieved from [2].	22
3.11	A 4×4 crossbar switch with direct links realized by N-type MOSFETs. Retrieved from [1].	22
4.1	Schematic representation of a transmission line as two parallel lines.	25
4.2	General transmission line model.	26
4.3	A two-port network and transmission matrix of it.	28
4.4	Transmission line model of an interconnect without IC packaging.	31
4.5	Transmission line model of an interconnect with IC packaging. . .	31
4.6	Frequency, step and impulse responses of 1, 3, and 5 cm interconnect links without IC packaging.	33
4.7	Frequency, step and impulse responses of 1, 3, and 5 cm interconnect links with IC package models at either end.	34
4.8	An elementary switch can be modeled as a serial resistor.	35

4.9	Interconnect wire geometry.	36
4.10	Depiction of $L - model$, $\pi - model$, and $T - model$ approximations to distributed RC circuit.	36
4.11	Capacitive effects on interconnect wires that have length l , width w , thickness t , and dielectric height h	38
4.12	Methodologies to avoid the low output voltage problem of pass transistor circuits.	40
4.13	Simulation of a pass transistor.	41
4.14	Simulation of a transmission gate.	41
4.15	A transmission gate circuit and its RC equivalent.	42
4.16	RC model simulation of a transmission gate.	42
5.1	First order RC Network	46
5.2	Transmission gate and pass transistor circuits.	46
5.3	Thompson grid model of a 2×2 crossbar network.	48
5.4	Thompson grid model of a 2×2 flexing crossbar network.	49
5.5	An 8×8 Baseline network	50
5.6	Thompson grids on a Baseline network	50
5.7	Fully connected network	51
5.8	Power consumption under full traffic throughput	53
5.9	Schematic representation of N capacitively coupled interconnect wires.	55
5.10	Capacitive coupling between parallel wires and the equivalent circuit.	56
5.11	Schematic representation of capacitively coupled aggressor and victim lines.	58
5.12	Schematic representation of capacitively coupled two segmented aggressor and victim lines.	60
5.13	Output voltages at the far end of aggressor ($V_{out} = V_{12}$) and victim ($V_{crosstalk} = V_{22}$) lines	61
5.14	HSPICE and Simulink simulations give the same crosstalk noise waveforms.	62
5.15	Equivalent circuit constructed to calculate the time constant of the i^{th} node of the victim line. Retrieved from [3]	63

5.16	Relation between maximum crosstalk noise on the far end of victim line and input signal rise time. $R_1 = R_2 = 100 \Omega$, $C_1 = 50 fF$, $C_2 = 60 fF$, $C_c = 30 fF$, $V_{DD} = 2 V$	66
5.17	Schematic representation of capacitively coupled two segmented aggressor and victim lines.	67
5.18	Comparison of the crosstalk noise computed by MATLAB Simulink and the methods proposed in [3–6].	68
5.19	Crosstalk noise comparison of two different topologies' victim far-ends.	68
5.20	(4,4) flexing crossbar model 1.	69
5.21	Crosstalk noise analysis of flexing crossbar model 1.	70
5.22	Crosstalk noise waveforms at the far end of victim lines of model 1.	70
5.23	(4,4) flexing crossbar model 2.	71
5.24	Crosstalk noise analysis of flexing crossbar model 2.	71
5.25	Crosstalk noise waveforms at the far end of victim lines of model 2.	72
5.26	(4,4) flexing crossbar model 3.	72
5.27	Crosstalk noise analysis of flexing crossbar model 3.	73
5.28	Crosstalk noise waveforms at the far end of victim lines of model 3.	73
5.29	Crosstalk noise comparison at the far end of 1 st adjacent wires. The architecture with close crosspoints (Model 1) has the maximum crosstalk noise.	75
5.30	Crosstalk noise comparison at the far end of 2 nd adjacent wires.	75
5.31	Crosstalk noise comparison at the far end of 3 rd adjacent wires.	76
5.32	Crosstalk noise waveforms at the far end of victim lines of a conventional crossbar switch.	76
A.1	Simulink model for Figure 5.12.	98

List of Tables

4.1	Transmission line model parameters of an interconnect wire on FR4 material.	32
4.2	Electrical resistivity of commonly used conductors at 22°C. Retrieved from [7].	37
4.3	Relative permittivities of some typical dielectric materials where $\varepsilon_0 = 8.854 \times 10^{-12} F/m$ and $\varepsilon = \varepsilon_r \cdot \varepsilon_0$. Retrieved from [8].	38
4.4	Wire area capacitance values for typical 0.25 μm CMOS process. The values are given in $(aF/\mu m^2)$. Retrieved from [8].	39
4.5	Fringing capacitance values for typical 0.25 μm CMOS process. The values are given in $(aF/\mu m)$. Retrieved from [8].	39
4.6	Coupling capacitance values for typical 0.25 μm CMOS process with minimally spaced wires. The values are given in $(aF/\mu m)$. Retrieved from [8].	39
4.7	MOSFET model parameters used in this thesis. Retrieved from [9].	43
5.1	Bit energy values on different network architectures. Retrieved from [69]	52
5.2	Comparison of the crosstalk noise at the victim far-end of two capacitively coupled lines computed by MATLAB Simulink and the methods [3–6] in Volts. $R_{s1} = R_{s2} = 0$ and $V_{DD} = 2V$	67
5.3	Crosspoint distances on the compared flexing crossbars. d denotes the unit distance.	74

Chapter 1

Introduction

Interconnection networks may be characterized by a number of properties such as their topology, operational characteristics and functional capabilities. Crossbar is an interconnection network with multiple input-output terminals whose switches are arranged in a grid of interconnects. Connections are formed by closing switches located at the intersections of interconnecting lines that corresponds to the elements of matrix. Literally, crossbar networks consist of crossing metal bars that provide paths between inputs and outputs. Solid state semiconductor chip implementations realize the same switching topology in VLSI. However, rapid technology scaling and demand for higher operation frequencies make crosstalk noise a major source of performance degradation in crossbar switching networks.

Crosstalk noise refers to an undesired spurious signal caused by coupling of signal. It may occur as a result of inductively induced voltages or parasitic capacitances between interconnects inside VLSI chips. In general, chip designers ignore inductive effects on interconnects since extracting and modeling of these effects are extremely difficult due to their global nature. This is justifiable since inductive coupling and magnetic effects are negligible as compared to capacitive coupling effects [3, 5–7, 10, 11]. Moreover, increase in VLSI circuit density due to the scaling down of dimensions and lower spacing between interconnects makes capacitive coupling a more serious problem. Nonetheless, inductive effects should

be taken into account in high frequency applications, especially for wide clock and power wires [7] and long interconnects.

Demanding performance requirements lead to extensive use of dynamic circuit techniques that can considerably reduce area and delay, and increase speed for CMOS integrated circuits [12]. In very large integrated circuits, major challenges include layout delays, high power dissipation at high frequencies of operation, increased interconnect delays, and crosstalk noise. It has been shown that signal integrity problems in interconnects determine the performance of overall circuit. It is important to predict signal degradation like propagation delay, delay variation, voltage peaks, crosstalk noise, signal overshoot, ringing and attenuation in early design cycles as these can critically affect system response.

Analysis and reduction of noise becomes critical for high-speed VLSI circuits with the continuous increase in the operating frequencies and technology scaling [3, 4, 11]. In the presence of reduced power supply voltages to sustain drive strength in deep submicron circuits; threshold voltages are also reduced, resulting in lower noise margins. Among the various sources of noise, crosstalk due to the capacitive coupling effects is the dominant source of noise in current CMOS digital integrated circuits. Contemporary high-speed CMOS technologies accommodate much more metal layers with increased density and reduced spacing between interconnects lead to significant increase in capacitive coupling effects that deteriorates the signal integrity. The severe adverse effects of coupling noise impose timing problems that can bring delay and then circuit malfunctions.

A poor understanding of crosstalk can lead to overly conservative design rules, resulting in poor performance. It can also lead to logic errors which may only be triggered by certain logic combinations which are difficult to detect. Thus, to properly deal with this problem and to design noise-immune chips, a proper interconnect modeling is required.

Broadly speaking there are two main ways to model on-chip interconnects: simulation tools and closed form analytical expressions. HSPICE is the most common simulation program that uses numerical integration and convolution

techniques to produce accurate results. SPICE models include both lumped circuits and models based on delay extraction techniques such as the method of characteristics. Using simulation programs can be considered as a time-saving approach. However, interconnect simulations suffer from that myriad of issues require sophisticated settings. In order to avoid the computational complexity of SPICE simulations, analytical models can be used. Analytic models are usually effective for obtaining the far end solutions. Therefore, an accurate analytical model is essential for efficient and reliable noise analysis. Lumped RC modeling with coupling capacitance between neighboring wires can give accurate behavior of VLSI circuits with smaller feature sizes. Kuhlmann and Sapatnekar [13] make the following statement: coupling capacitances become substantial as their magnitude evolves comparable to the parasitic capacitance of a wire and area capacitances. This causes an increasing susceptibility to failure on account of the inadvertent noise and leads to a need for accurate noise estimation method. An incorrect estimation of the noise cause either functional failures in case of underestimation or wasted design resources because of overestimation. For dimensionally larger interconnects such as chip-to-chip wires, using transmission line models that consider inductive effects would be more suitable.

In light of the foregoing discussion, this thesis investigates the effects of crosstalk noise on flexing crossbar networks and what precautions can be taken or how flexing crossbars can be designed to alleviate the adverse effects of noise. This study proposes an efficient method for the estimation. The estimation method is also applicable to other submicron VLSI circuits. Lumped circuit theory is utilized to estimate crosstalk noise due to coupling effects and means of crosstalk reduction are investigated. Peak crosstalk noise amplitude, occurrence time, and time domain waveform are represented in closed form expressions. This research also introduces an empirical approach to compute the best case victim-aggressor alignment that minimizes the crosstalk noise on victim lines. In addition, it suggests a geometric approach for reducing the adverse effects of crosstalk noise on flexing crossbars. Delay and signal quality for varied lengths of interconnect wires on interconnection networks using lossy transmission line theory are analyzed and examined in detail. Furthermore, crossbar networks are compared with other

interconnection networks in terms of power consumptions.

The rest of the thesis is structured as follows. Chapter 2 presents the relevant literature and background for the presented research. Chapter 3 gives insight into interconnection circuits with current trends. Chapter 4 provides transmission line model for interconnect wires, explains interconnection modeling using a lumped circuit model, and presents simple equivalent circuit model for pass transistors and transmission gates. Chapter 5 describes a comparative power consumption analysis of interconnection networks and crosstalk analysis of flexing crossbars under victim aggressor model. Chapter 6 provides concluding remarks and suggests potential directions for future research.

Chapter 2

Literature Review

We are living in an era in which high performance computers, machines and systems are omnipotent for a variety of tasks. Such complex systems require sophisticated networks to reduce the communication overhead among their processors. It is a crucial research objective for us to address this issue. Accordingly, this chapter gives a detailed account of research findings, by providing a two faceted literature survey; first on interconnection networks and second on crosstalk noise. Among the numerous studies in these two fields of research, Thurber [14], Masson [15], Feng [16], Oruc [17,18] will be the basis of our survey on interconnection networks. We will also include more recent research results relating to on-chip networks. In particular, Kumar et al. [19], Kim et al. [20], Kao and Chao [21] and Oruc [18,22] will be part of our survey. Crosstalk noise has emerged as a consequence of the reduction in circuit dimensions as VLSI technologies improved. Hence it is a relatively new research issue and Vittal et al. [6], Kuhlmann and Sapatnekar [13], Elgamel and Bayoumi [23], Heydari and Pedram [11] will constitute the main articles in our survey. The research efforts in these references will be highlighted and evaluated in conjunction with our findings in this chapter.

2.1 Interconnection Networks

As feature sizes of VLSI circuits become smaller and processors become faster, more processors are being integrated into a single chip to obtain parallel processors for higher performance. Interconnection networks have emerged as an alternative to the buses to deal with the increasing bandwidth demands of such architectures. Early research results on interconnection networks were surveyed in [14], where it is emphasized that interconnecting subunits in a multiprocessor is a key research problem. As digital systems become complicated, the severity of this problem also increases. It is further pointed out that, in the limiting case, processor speeds cannot be increased further using faster system components only, it is stressed that any further speed-up will likely result from changes in the organization and construction of hardware, rather than by basic circuit enhancements [14].

Earlier studies on interconnection networks focused on the design of nonblocking networks. Reducing the number of crosspoints without compromising the connection power on a full crossbar was the principal objective. The seminal paper of Clos [24] on nonblocking networks was published in Bell Systems Technical Journal in 1953 at a time when there were no parallel computers, establishing the foundation of the field on interconnection networks. Clos aimed to sustain numerous telephone connections in a circuit switched telephone network without placing direct connection or crosspoint between every caller and receiver. Oruc [17] pointed out that a telephone network serving n customers would require $n(n - 1)/2$ crosspoints if every customer was directly connected to every other customer and assuming that each crosspoint could sustain a bidirectional communication. Crosspoints were implemented by bulky electromechanical devices and vacuum tubes in early fifties. Thus such networks were not feasible and a solution had to be found in the network design or architecture domain [17]. It was also stated in [17] that Clos designed his strictly nonblocking 3-stage networks utilizing orders of magnitude fewer crosspoints than an ordinary full crossbar would require a much of what followed since then were refinements of this construction with a few notable exceptions. It was further

mentioned that subsequent to findings of Clos, researchers in the field turned their attention to the reduction of the number of rearranged calls in a 3-stage network to accommodate and minimize number of crosspoints in nonblocking networks [17]. Beizer [25], Benes [26], and Paull [27] were credited much of this work. It was pointed out that Benes [26, 28, 29] focused on combinatorial and topological properties of rearrangeable networks. Other studies on rearrangeable networks were reported in Joel [30] and Opferman and Tsao-Wu [31].

Another problem associated with Clos networks was to minimize the number of switches. Extensions along this line include works of Bassalygo and Pinsker [32] and Cantor [33]. In his study, Cantor reduced the complexity of n input Clos network to $O(n \log^2 n)$ switches. In a subsequent study, Bassalygo and Pinsker further minimized the crosspoint complexity of strictly nonblocking networks to $O(n \log n)$. Further results on strictly nonblocking networks dealt with reducing the constants in the crosspoint complexity of such networks [34, 35]. Much of this work was based on Pinsker's seminal paper on concentrator switches [36]. Proving the existence of an extensive graph in the construction of strictly nonblocking networks was a major accomplishment in these studies.

The research on interconnection networks throughout the three decades including 1950s, 60s and 70s mostly dealt with interconnection issues in the telecommunications field. In 1971, Intel announced the first microprocessor chip ever, and efforts for constructing high performance computers intensified. Research on interconnection networks was coupled with parallel processing studies in the second half of 70s. Variants of Benes network such as shuffle-exchange, Omega, baseline, Indirect Cube, Generalized Cube were introduced in this period [17]. Following these studies, parallel computing researchers dominated the field of interconnection networks in a few years. Lawrie used his Omega network [37] to effectively interconnect processing elements in a parallel processor. He further examined data access and alignment problems in array processors [38]. He introduced a method to access rows, columns, diagonals and backward diagonals and perform other permutation and indexing of an array stored in a memory device without any contentions. Lang [39] extended Lawrie's network to realize any permutation in much less shuffle exchange steps.

In the parallel processing domain, research on interconnection networks further expanded with parallel processing studies during the eighties. The main goal was to obtain high performance parallel processors [40]. A number of blocking switches such as reverse exchange and baseline network were introduced by Feng and Wu [41, 42] for parallel processors. During 1990s a renewed research interest on 3-stage network designs produced a number of empirical findings. Some of the studies on nonblocking switching networks and routing algorithms were reported in [43–47]. Feldman et al. [43] present a new principle for establishing wide-sense nonblocking interconnection networks. In these networks, router tries to satisfy requests to build or demolish a connection. Wide sense nonblocking networks are capable of establishing new paths between unused input-output pairs by making sure that remains nonblocking as new paths are added [48].

In [15], Masson focuses on circuit switching in interconnection networks and points out that the subject fundamentally must deal with the design and analysis of crosspoint arrangements. He states that designing efficient interconnection networks is vital to designing high performance systems. He further points out that early research on interconnection networks was stimulated by the needs of the telecommunication industry. In [44], Yang and Masson introduced a new nonblocking circuit switching network. They named their network as nonblocking broadcast network as an input can be connected to multiple outputs. Yang [47] reported that there is a significant difference in crosspoint complexities of multicast networks and permutation networks. In her study, she presented a low cost interconnection network class supporting vast amount of multicast connections in a nonblocking fashion.

It is evident that research on interconnection networks will continue to evolve. The contemporary research trends imply a few tracks for the future. In the last few decades, interconnection network research has focused on photonic networks [21, 49], and network on-chip architectures [18, 19, 22, 50]. Kim et al. [20] introduced a new topology named flattened butterfly network that has half the cost of similar performance Clos network. Kao and Chao [21] proposed photonic on-chip waveguides as an alternative for long interconnection networks to overcome speed and power issues. Authors introduced a bufferless photonic

Clos network (BLOCON) to utilize silicon photonics. They also presented a link allocation scheme to ease the routing problem and two scheduling algorithms to resolve the contention problem of Clos network. The network on chip idea seeks to reduce the complexity of interchip connections by placing the entire interconnection network in a single chip [50]. Dally and Towles [50] introduced the idea of tiling to build a network on chip system. Kumar et al. [19] proposed a packet switching Network-on-Chip (NoC) structure for developing large and complex processor housing many resources on a single chip. Their proposed architecture integrates physical and architectural level designs. Further, they asserted that their architectural NoC template is capable of developing different applications which can be modeled as communication tasks. Significant problem in building a network on chip architecture is to identify the most effective switching topology to interconnect computational components. Others have proposed fat-trees and high radix Clos networks to build network on chip systems [21, 51, 52].

The one issue with tile based switching topologies is the locality of interconnections. As stated in [18] this results in uneven interprocessor distances increase communication overhead in processing elements and potential congestions. It is also claimed in [18] that fat-tree, Clos, and butterfly switching topologies require complex routing algorithms and this will likely add a significant communication overhead to computations within an on chip network that utilizes one of these switching networks. A new switching topology, called one sided binary tree-crossbar switch was introduced in [18] to mitigate with these problems. It was stated in [18] that switch is self routing and nonblocking. In this thesis, two sided variant of this binary tree crossbar switch will be further analyzed for power consumption and crosstalk issues.

2.2 Lumped Circuit Models of Interconnection Networks

In this section, we will survey relative literature on circuit models of interconnection networks. When signals are transmitted along interconnection networks, the proximity of wires causes crosstalk noise. Hence, the condition of a wire depends on the condition of its adjacent wires in a coupled system. Crosstalk noise may induce some adverse effects such as undershooting, overshooting, glitches, and increasing signal delay. Since integrated circuit densities incessantly grow, the crosstalk noise continues to pose a great problem for all high-performance VLSI circuits.

To analyze the crosstalk noise between neighboring interconnect wires, the circuit can be separated into two parts; victim and aggressor. The wire carrying the input signal is an *aggressor*, and the wire attacked by aggressor is a *victim*. Aggressor and victim nets are adjacent to each other, and the connection between them can be modeled by using coupling elements.

Many types of methods can be used to evaluate the crosstalk noise between the aggressor and the victim nets. Various transmission line equations were solved in [10, 53] and an analytical formula for peak crosstalk noise in capacitively coupled interconnect wires was obtained. This formula can be used for fully coupled structures, but it is not suitable for the general RC trees or the partially coupled wires [54].

Using computer aided simulation programs such as HSPICE is an accurate approach, but it is time consuming [11]. For chip designers, deriving analytical formulas that can determine noise waveform is more attractive than running a simulation program especially during early periods in the design process [11]. In addition, utilizing a simulator is computationally inefficient because of the complicated settings, and hence, is not as valuable for large topologies. Circuit models of interconnection networks can be represented as linear time-invariant systems. Therefore, model reduction methods [55–57] can be incorporated into

the analysis to reduce the computational complexity. In doing so, simulation programs can estimate the behavior of the noise more precisely.

Electrical problems due to crosstalk noise in interconnections were extensively investigated since the first appearance of large scale integration. To estimate the crosstalk noise many different methods [3–6, 11, 13, 58–60] have been proposed. The accuracy of these methods is verified by comparing their performance with that of HSPICE simulations. These methods are adopted by designers since their prediction accuracy is more acceptable than circuit simulators.

In the late 1990s and early 2000s, many researchers focused on the problem of deriving analytical formulas for crosstalk noise in integrated circuits [11]. During this period, new techniques were proposed to alleviate the problem. In [5], Vittal and Marek-Sadowska provided an upper bound for the peak crosstalk noise voltage in on-chip interconnects using RC circuit model. Their method utilizes dynamic noise margins rather than static ones. However, wire resistances were not taken into account in this work. In a consecutive study [6], some geometric considerations were utilized to obtain mathematical expressions for the noise properties such as peak amplitude voltage and the pulse width. Since the noise margins of switching elements are mainly dependent on both peak amplitude and width of the noise, estimating these properties is crucially important.

In order to determine peak crosstalk noise voltage in integrated circuits, a new methodology was offered by Devgan in [4]. This work can be considered as a milestone analytical study on crosstalk noise estimation performed up to now, and is similar in concept to Elmore delay in timing analysis. Besides, the proposed technique can be performed by inspection without using any matrix construction and factorization [4]. It is simple yet accurate in most cases, but exhibits increasing estimation error when the rise times of applied signals are short. It must also be noted that Devgan’s method cannot estimate the noise pulse width.

In [13], Kuhlmann and Sapatnekar proposed a time efficient crosstalk noise estimation method based on Devgan’s method. They used an RLC equivalent

model for interconnects in their method, and claimed that their metric estimates the crosstalk noise with a higher precision as compared to SPICE while the other fast noise computation methods overestimate it. They further asserted that Devgan’s metric has some limitations in that the victim net crosstalk noise is proportional to the slope of the input signal transient. This constitutes a major problem when the input signal is a step function. In such cases, crosstalk noise on the victim line goes to infinity. This is impossible in the sense that supply voltage restricts the maximum noise that can be produced [13]. They also pointed out that crosstalk noise has no dependence on the ground capacitances in Devgan’s model.

Cong et al. [54] proposed a lumped $2\pi RC$ model and apply it to noise constrained optimizations. Their model provides closed-form formulas for the waveform of crosstalk noise. They demonstrated their model’s capability in two applications; (i) noise reductive optimization rule generation, (ii) concurrent wire spacing to multiple nets for noise constrained interconnect minimization [54]. Their research findings show that the peak amplitude of the noise has more impact than the pulse width of the noise on functional failures.

Takahashi et al. in [61] also proposed a $2\pi RC$ model based crosstalk estimation method for generic RC circuits. Their methodology derives a closed-form waveform of crosstalk noise using an analytic expression. They also estimated the delay induced by the crosstalk from the noise waveform. The proposed model’s main shortcoming, however, is the increase of estimation error rate with the length of interconnects.

Other extensions to Devgan’s methodology were reported in [3, 11]. In their study, Heydari and Pedram modified Devgan’s method to introduce a new expression capable of predicting the peak amplitude, pulse width, and the time-domain crosstalk noise waveform on an RC interconnect [11]. Their approach estimates crosstalk noise waveforms with high accuracy. Nonetheless, the method requires complete coupling information of the whole network to obtain a valid result, while Devgan’s method can produce a result with partial coupling information.

In this thesis, we compare some crosstalk noise metrics proposed in [4–6, 11]. Based on the above review and initial evaluation, we provide a comparative analysis of these noise metrics, and describe our noise analysis method. By using flexing crossbars as benchmark circuits, we obtain noise tolerant flexing crossbar topologies.

Chapter 3

On-Chip Interconnection Networks

Along with the significantly increasing demand for higher processing speeds, communication units become main limiting factor in the performance of many digital systems. Buses cannot keep up with increasing bandwidth, delay, and power demands of such structures. It seems that dedicated wiring is not an effective approach for interconnecting components in digital systems especially those systems with high bandwidth requirements. Furthermore, dedicated wiring takes more area and it increases system complexity. Therefore, various interconnection network designs have been offered to alleviate this problem. Interconnection networks enable limited bandwidth to be shared such that it can be utilized efficiently [2]. Therefore, they constitute an economically feasible high-speed solution to communication problems which makes them the key factor in the success of future digital systems.

Interconnection networks can be used in many different applications. Figure 3.1 shows a basic real time processing system. Interconnection networks can be used to couple different processes in such a system, and many other systems that used a multitude of resources. Figure 3.2 shows a more general real time processing system that involves interconnections between a set of processors and

a set of memory modules. Interconnection networks can facilitate both on-chip and chip-to-chip communications.

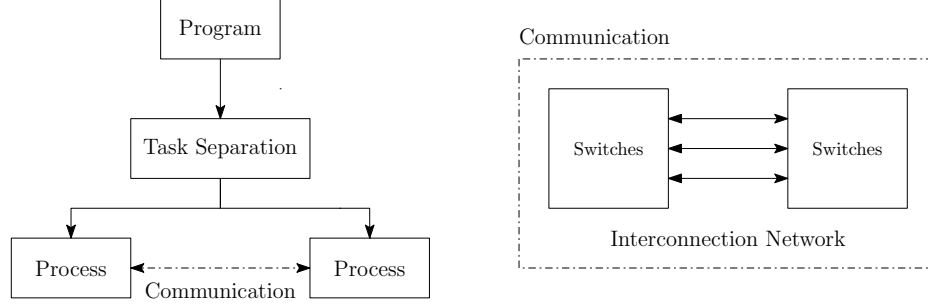


Figure 3.1: Overview of a real time processing system.

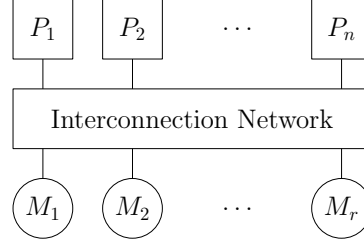


Figure 3.2: General real time processing system model.

In general, an interconnection network is a system that transmits data among *input* and *output* terminals which are connected together by set of switches and links. *Callers* and *receivers* use these terminals as entry and exit points [1]. Figure 3.3 shows block diagram of an $n \times r$ interconnection network that has n inputs (callers) and r outputs (receivers). Interconnection networks consist of permanent links and controllable switches such that different interconnection functions can be realized by properly configuring the switches. Capability of realizing switching functions determines the switching power of an interconnection network [1].

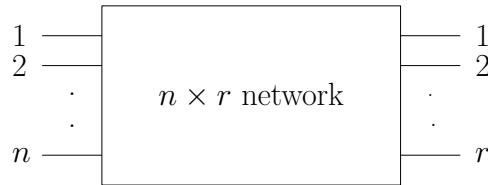


Figure 3.3: Block diagram of an interconnection network.

An interconnection network can be broadly categorized by a number of properties such as its control policy, switching policy, operational characteristics and network topology. The control functions of an interconnection network can be managed by either centralized or distributed controller. There are three switching policies: circuit switching, packet switching, and integrated switching. In circuit switching, physical paths are used to interconnect inputs and outputs. In packet switching, data are divided into packets and routed through network without setting up physical paths among inputs and outputs. Integrated switching combines powers of circuit and packet switching. There are three operation modes for interconnection networks: synchronous, asynchronous and combined mode. Synchronous communications are synchronized by an external clock and asynchronous communication is synchronized by special signals. In combined communication, both synchronous and asynchronous communications are used. Topology of a network refers to the physical arrangements of links and switches that set up connections. The links are actually physical wires, switching elements are devices connecting set of input and output links together. Figure 3.4 shows a topological taxonomy of interconnection networks in which they can be classified as static or dynamic.

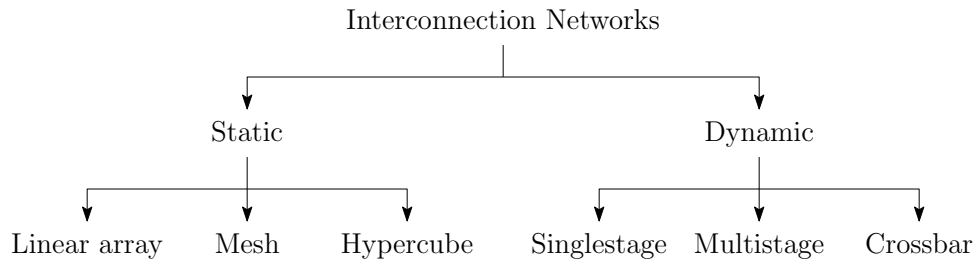


Figure 3.4: A topological classification of interconnection networks.

Static networks provide fixed connections between terminals. In static networks, connections cannot be changed and messages must be routed along established links. Static networks can be categorized further in regard to their topological patterns as linear array, mesh or hypercube topology. Linear arrays, rings, n-dimensional meshes, n-cubes are well known examples of static networks [62].

Dynamic networks provide reconfigurable connections between terminals.

Switches are fundamental components of dynamic networks. Dynamic networks can change their interconnectivity dynamically by setting their switches [62]. Dynamic networks can be divided into three topological classes: singlestage, multistage, and crossbar. Singlestage networks, also called recirculating networks, consist of a single switching stage cascaded to the links. Various connections and permutations are constructed by recirculating the data flow several times through the network. Multistage networks are more complicated structures that comprise multiple switching stages cascaded to the links. These types of networks are capable of interconnecting any one of input and output terminals together due to the simplicity of creating connections with the help of multiple stages. They can be further classified as blocking, nonblocking and rearrangeable nonblocking. Concurrent connections between more than one pair of input-output terminals may cause contentions in blocking networks. Banyan, omega, flip, indirect binary n-cube, and delta networks are examples of blocking networks. Nonblocking networks originated from Clos network [24]. Rearrangeable nonblocking networks can create all possible connections between multiple input-output terminals by rearranging their connections. They can establish new connections or destroy existing connections by requests. The Benes network [26,28,29] is an example of a rearrangeable nonblocking network. Crossbar switches are nonblocking networks in which every input and every free output terminals can be connected together.

In this thesis, we will mainly be concerned with crossbar networks. Further explanations about crossbar networks will be provided later of this chapter.

3.1 Elementary Switching Structures

An elementary switch is a device used to interrupt the data flow or diverting it from one terminal to another. Data flow between terminals may be unidirectional or bidirectional. Elementary switches have one or more set of terminals, which are connected to the links. Multiple-input, multiple-output switches can be realized using simple on-off switches as shown in Figure 3.5. Oruc [1] states that an $n \times r$ elementary switch can be constructed by fanning out each of the n inputs to all

r outputs using r on-off switches. In other words, an $n \times r$ elementary switch requires nr on-and-off switches.

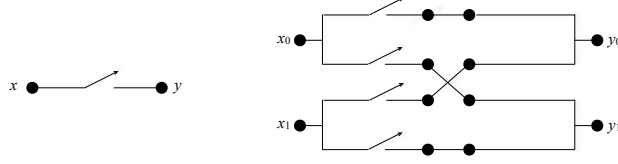


Figure 3.5: On-off and 2×2 elementary switches.

As long as the capacity of an elementary switch is sufficient, a terminal may communicate with more than one terminal. In elementary switches, congestions may occur only on the terminals. Barring a capacity constraint, an arbitrary input-output pair can communicate with each other. Therefore, elementary switches are nonblocking networks.

Elementary switches provide nonblocking switching but they have a critical disadvantage in that their complexities increase linearly with both input and output numbers. Moreover, fan-in and fan-out (in and out degrees of vertices) grow linearly with input and output numbers. Consequently, elementary switches are not utilized in physical layers of interconnection networks as n and r become large [1].

3.2 Binary Tree Switching Structures

Binary tree switches can be utilized in order not to encounter fan-in and fan-out problems of elementary switches [1]. An $n \times r$ binary tree switch is obtained by replacing the on and off switches in an $n \times r$ elementary switch by a cascade of $\log_2(r)$ stages of $2n(r-1)$ on-off switches with $\log_2(n)$ stages of $2r(n-1)$ on-off switches [1]. Figure 3.6 shows a 4×4 binary tree-switch. The full circles located in the middle show permanent links.

All the paths between any input-output pairs are unique in these structures. In order to create connection between an input and an output, it is sufficient to

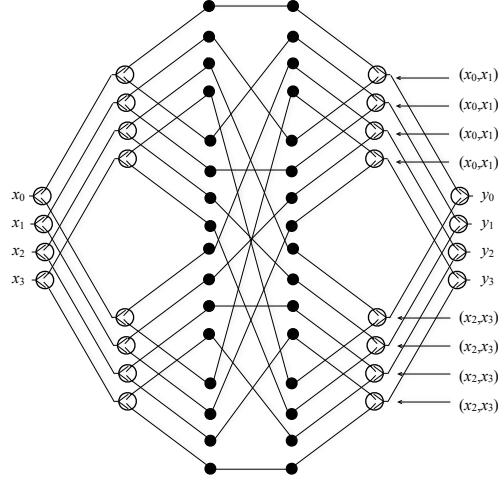


Figure 3.6: A 4×4 binary tree switch. Retrieved from [1].

turn on the switches along the direction of the relevant output.

Constructing a path requires setting some of the $n(r - 1) + r(n - 1)$ on-off switches in such structures. Simpler and more powerful constructions can be designed by replacing the switches in either left or right binary trees by permanent links.

3.3 Crossbar Switches

Crossbar switches directly connect input-output terminals together without using any intermediate stages. They can be viewed as a grid, i.e., number of vertical and horizontal links connected by a switch at each intersection [62]. In crossbars, it is possible to establish a connection between any input terminal and any output terminal just by setting the crosspoint switches located at the intersections. Crosspoints can be turned on or off with regards to the requests. Therefore, crossbars allow to utilize all possible permutations.

Formally, an $n \times r$ crossbar switch is an $n \times r$ array of crosspoints each of which may be turned on or off to connect a set of n inputs with a set of r outputs [1]. Figure 3.7 shows 4×4 crossbar switch. The full circles inside the

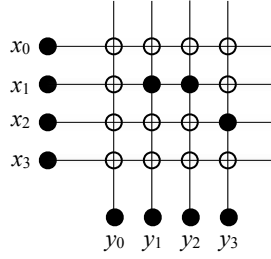


Figure 3.7: A 4×4 crossbar switch. Retrieved from [1].

grid are crosspoints that are closed to create the requested connections between input x_1 and outputs y_1, y_2 , and input x_2 and output y_3 .

3.4 Flexing Crossbar Switches

Conventional crossbar switches do not restrict fan-out of inputs or fan-in of outputs. In $n \times r$ crossbar switch, each input is connected to all r outputs and each output is connected to all n inputs. As in the elementary switch model, this makes an $n \times r$ crossbar infeasible as n and r become large. To alleviate this problem, Oruc [1, 18, 22] offers to combine the binary tree and crossbar models together. Resulting network is called flexing crossbar or binary tree crossbar.

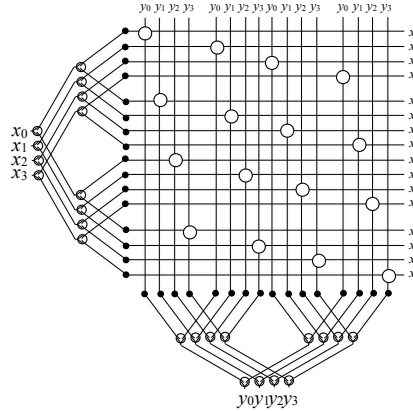


Figure 3.8: A 4×4 flexing crossbar. Retrieved from [1].

Figure 3.8 shows a 4×4 flexing crossbar. Empty circles indicate the crosspoints. The binary tree on the left distribute the inputs to the terminals of the crossbar

switch located in the middle. In a like manner, the binary tree at the bottom of the structure brings together crossbar switch terminals at the outputs.

Flexing crossbars are rearrangeable nonblocking networks which are superior than crossbars in point of interconnection capabilities. The crossbar array in the middle of flexing crossbar allows any input-output connection without blocking other inputs and outputs. Nevertheless, inefficient use of crossbar array creates an area problem, i.e., nr intersections serve as crosspoints out of n^2r^2 .

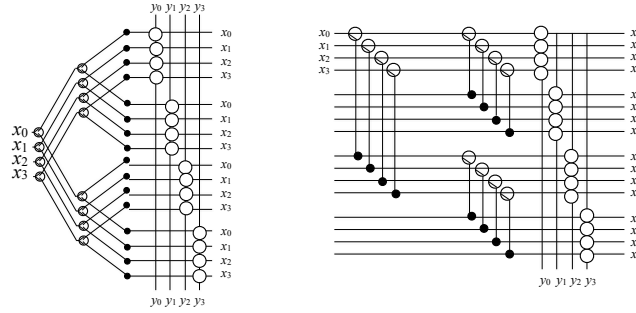


Figure 3.9: A 2-level binary tree crossbar switch with direct outputs and its crossbar realization. Retrieved from [1]

Oruc [1] further improves the model by changing the number of duplications of inputs and outputs produced by the binary trees as in Figure 3.9. It should be noted that, in the new configurations only 1×2 elementary switches are used. In the structure at the left side of the figure, the number of vertical lines are reduced from 16 to 4 by removing the binary tree at the bottom. Consecutively, the number of intersections reduced from 256 to 64 and one-fourth of these intersections is populated by crosspoints.

3.5 Physical Realizations of Crossbar Switches

In order to construct actual crossbar switches, theoretical models should be converted to physical models. There are different technologies to this purpose. Early implementations were established on electromechanical principles [1]. Primitive telephone networks were established with crossbars which contain

mechanical relays as represented in Figure 3.10. In such crossbars, connections between inputs and outputs are bidirectional.

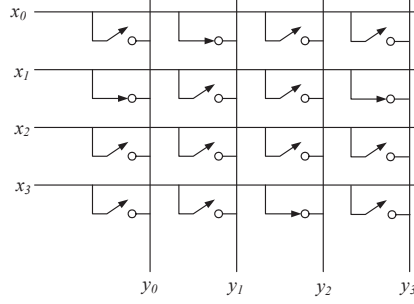


Figure 3.10: A 4×4 crossbar network made by mechanical switches. Retrieved from [2].

Today, however, crossbar switches are implemented using digital and optical technologies. One of the recent crossbar switch technology is pass transistor realization. In such realizations, MOSFET¹ solid state devices are employed as shown in Figure 3.11.

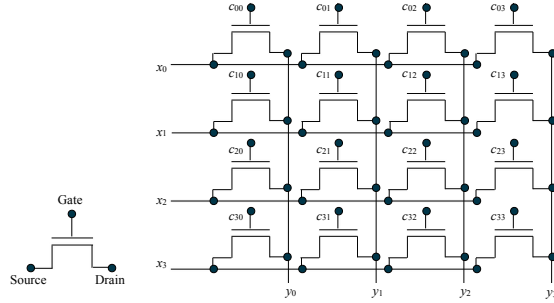


Figure 3.11: A 4×4 crossbar switch with direct links realized by N-type MOSFETs. Retrieved from [1].

In pass transistor circuits, we can think of the MOSFETs as simple switches such that they serve as on-off switches between *source* and *drain* terminals by way of *gate* terminal. Gate voltage controls current between source and drain terminals in MOSFETs employed as pass transistors.

There are two types of MOSFET devices, NMOS and PMOS. The polarity of their gate voltage is the main difference between them. When the gate voltage of

¹Metal Oxide Semiconductor Field Effect Transistor

an NMOS device is positive (V_{DD}), it is *on* or in transmission state. The PMOS device operates in a complementary way to the NMOS device. It is *on* when its gate voltage is negative ($-V_{DD}$). Both of them can be used in crossbar switches with proper gate voltages.

Figure 3.11 shows a pass transistor implementation of a 4×4 crossbar switch. In order to establish connections between input-output terminals, MOSFETs are individually set on or off. Both Figure 3.10 and Figure 3.11 are functionally equivalent models. Furthermore, as long as outputs are requested by no more than one input, these networks last nonblocking.

Chapter 4

Interconnection Network Modeling

In order to analyze interconnection networks, there are two general analytical tools available in literature: lumped element and distributed element methods. The lumped element or lumped circuit model represents the electrical properties of the structure by a circuit, consisting of ideal electrical components such as resistors, inductors, and capacitors connected to one another using lossless wires. The distributed element or transmission line model considers that the circuit attributes are distributed continuously all over the circuit.

An interconnection network comprises collections of switches, buffers and transistors. At each level of hierarchy, signals or packets are transported on interconnect wires. An interconnect wire can be considered as a distributed element model with a resistance and capacitance per unit length. Electrical characteristics of an interconnect wire can be estimated with lumped circuit elements [7]. Other interconnection network components such as switches and transistors can also be modeled using lumped element method to simplify the network. Modeling with lumped element method enables networks to be analyzed by using ordinary circuit theory. On the other hand, transmission line theory bridges the gap between complete field analysis, which is a numerical method

for designing and developing electromagnetic application products, and circuit theory [63]. We can approach the signal transmission phenomena from two different angles, i.e., the extension of circuit theory or the specialization of Maxwell's equations [64].

Electrical size is the main difference between circuit and transmission line theory. In transmission lines, physical dimensions of a network are considered as a sizable fraction of the wavelength, while electrical component sizes in circuit analysis are much smaller than the wavelength. Hence, transmission lines are actually distributed parameter systems where signals can change in magnitude and phase over distance, while circuit theory concerns with lumped elements, where signals do not change over the length of wires [64].

4.1 Transmission Line Model for Interconnect Wires

Transmission lines contain at least two conductors. Figure 4.1 illustrates such a representation of an interconnect wire.

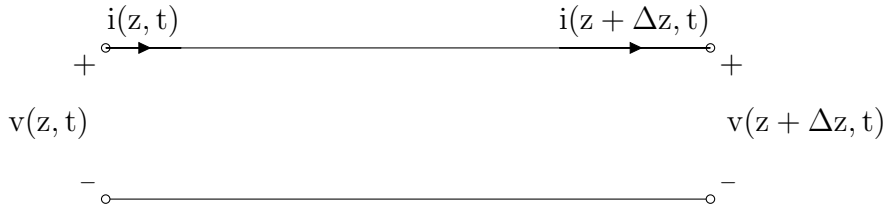


Figure 4.1: Schematic representation of a transmission line as two parallel lines.

The infinitesimal length, Δz , of wire can be modeled with lumped circuit elements, i.e., R , L , G , and C that are defined as follows:

- R , series resistance per unit length, in Ω/m .
- L , series inductance per unit length, in H/m .
- G , parallel conductance per unit length, in S/m .
- C , parallel capacitance per unit length, in F/m .

R and G represent loss, and they are captured from the finite conductivity of the conductors, and the dielectric loss of the material between the conductors, respectively [64].

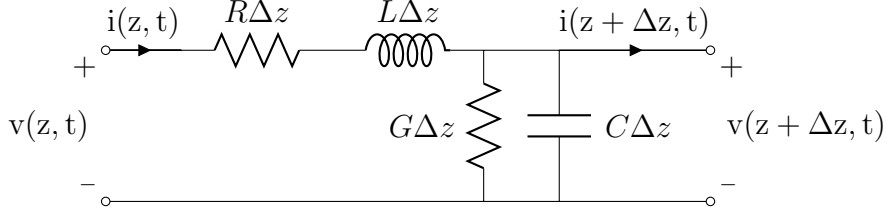


Figure 4.2: General transmission line model.

Figure 4.2 represent a general transmission line model. Cascade connections of this infinitesimal length circuit converge to a finite length transmission line. Kirchhoff's circuit laws lead to

$$v(z, t) - R\Delta z i(z, t) - L\Delta z \frac{\partial i(z, t)}{\partial t} - v(z + \Delta z, t) = 0 \quad (4.1)$$

$$i(z, t) - G\Delta z v(z + \Delta z, t) - C\Delta z \frac{\partial v(z + \Delta z, t)}{\partial t} - i(z + \Delta z, t) = 0 \quad (4.2)$$

Dividing Eq. 4.1 and 4.2 by Δz and letting $\Delta z \rightarrow 0$ gives the following equations which are known as telegrapher equations [64].

$$\frac{\partial v(z, t)}{\partial z} = -Ri(z, t) - L \frac{\partial i(z, t)}{\partial t} \quad (4.3)$$

$$\frac{\partial i(z, t)}{\partial z} = -Gv(z, t) - C \frac{\partial v(z, t)}{\partial t} \quad (4.4)$$

In terms of phasors, the coupled equations can be written as

$$\frac{dV(z)}{dz} = -(R + j\omega L)I(z) \quad (4.5)$$

$$\frac{dI(z)}{dz} = -(G + j\omega C)V(z) \quad (4.6)$$

where $\omega = 2\pi f$ is the angular frequency. To obtain wave equations for $V(z)$ and $I(z)$, Eq. 4.5 and Eq. 4.6 can be solved simultaneously

$$\frac{d^2V(z)}{dz^2} - \gamma^2V(z) = 0 \quad (4.7)$$

$$\frac{d^2I(z)}{dz^2} - \gamma^2I(z) = 0 \quad (4.8)$$

where $\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)}$ is the complex propagation coefficient whose real part α is the attenuation constant, with units m^{-1} , and whose imaginary part is the phase constant β , with units rad/m . These quantities are functions of frequency. Solutions of the transmission line equations can be found as

$$V(z) = V_0^+ e^{-\gamma z} + V_0^- e^{\gamma z} \quad (4.9)$$

$$I(z) = I_0^+ e^{-\gamma z} + I_0^- e^{\gamma z} \quad (4.10)$$

where the $e^{-\gamma z}$ indicates wave propagation in the $+z$ direction, and the $e^{\gamma z}$ indicates wave propagation in the $-z$ direction. $V_0^\pm$ and $I_0^\pm$ are constants defined by boundary conditions. Using the coupled equations, the following transmission line parameters can be found from the solutions of transmission line equations

$$Z_0 = \frac{V_0^+}{I_0^+} = -\frac{V_0^-}{I_0^-} = \frac{R + j\omega L}{\gamma} \quad (4.11)$$

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)} \quad (4.12)$$

Characteristic impedance Z_0 and complex propagation constant γ are the most important parameters of a transmission line. They depend on the distributed circuit parameters R, L, G, C of the line and frequency ω but not the length of the line.

4.1.1 The Transmission Matrix

A transmission line is a two-port network and in practice they are usually analyzed by approximating them by a cascade connection of two-port devices as illustrated in Figure 4.3.

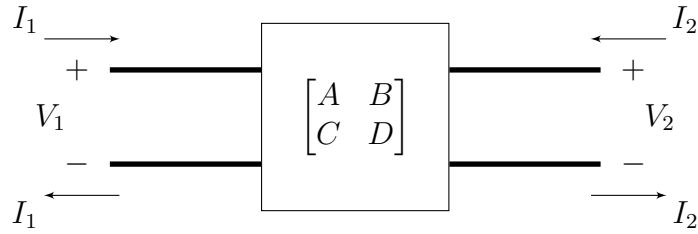


Figure 4.3: A two-port network and transmission matrix of it.

Linear two-port devices can be defined using number of equivalent circuit parameters, i.e., their transmission (ABCD), impedance (Z), admittance (Y), or scattering (S) matrices. These representations can be converted to each other, and they establish relations between the following variables

- V_1 , voltage across 1st port
- I_1 , current into 1st port
- V_2 , voltage across 2nd port
- I_2 , current into 2nd port

where V_i and I_i represent the Fourier (Laplace) transforms or the phasors of the voltages and currents ($i = 1, 2$).

In this study, we use transmission matrix representation whose entries satisfy the following linear relationship

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \quad (4.13)$$

In order to use transmission matrices, we need to determine A, B, C, D values. Assume that Z_{sc} , and Z_{oc} are the impedances reflected to the input ports when the output ports are short-circuited and open-circuited, respectively. According to Eq. 4.13, these impedances are given by

$$Z_{sc} = \frac{B}{D} \quad Z_{oc} = \frac{A}{C} \quad (4.14)$$

$A = D$ for symmetric two-port networks, and determinant of an $ABCD$ matrix satisfies $AD - BC = 1$ for linear two-port networks. Z_0 and θ , the symmetric reciprocal two-port networks can be characterized, where Z_0 is the characteristic impedance at the input ports of the network when the output ports are matched, i.e., terminated by a load impedance Z_0 and $\theta = \ln(I_1/I_2)$ is the propagation constant where I_1 and I_2 are port currents at the matched condition. A cascaded two-port network consists of n equivalent symmetric reciprocal two-port networks with characteristic impedances Z_0 cascaded together and has an equivalent characteristic impedance Z_0 [65]. The propagation coefficient θ of this cascaded two-port network is

$$\theta = \sum_{k=1}^n \theta_k \quad (4.15)$$

Transmission matrix parameters can be expressed using Z_0 and θ .

$$A = D = \cosh \theta, \quad B = Z_0 \sinh \theta, \quad C = \frac{1}{Z_0} \sinh \theta \quad (4.16)$$

As the length of the transmission line Δz approaches zero, θ_k also approaches zero. Yet, $n\Delta z$ remains unchanged since the number of sections n goes to infinity. Using the first-order approximation, $\theta_k = \gamma\Delta z$, Eq. 4.15 leads to $\theta = \gamma d$ for the

line propagation constant where $n\Delta z$ is the line length. Therefore, $ABCD$ matrix modeling of the transmission line of length $n\Delta z$ yields

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \begin{bmatrix} \cosh(\gamma n\Delta z) & Z_0 \sinh(\gamma n\Delta z) \\ \frac{1}{Z_0} \sinh(\gamma n\Delta z) & \cosh(\gamma n\Delta z) \end{bmatrix} \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \quad (4.17)$$

4.1.2 Delay and Signal Quality Analysis of Interconnect Wires

Rapid technology scaling and demand for higher operation frequencies make difficult to provide input-output interfaces that can sustain communication over the chip. Information rates are increased substantially in order to prevent bottlenecks. Traditionally, the data are carried over circuit traces in chips. Nonetheless, links transmitting information at higher frequencies may encounter the inherent interconnect bandwidth limitations [66].

In this section, we investigate the links operating at frequencies up to 50 GHz by computing magnitude, step, and impulse responses. The analysis provides important information about signal gain and propagation delay over the links. For different lengths of interconnects, identifying the propagation delays is of vital importance. Propagation delays may become significant compared to bit periods at higher frequencies even for interconnects a few millimeters in length.

Packets over links are routed through chip areas containing switches, connectors and sockets on interconnection networks. Changes in the transmission geometries lead to different types of discontinuities such as bends, vias, and crossings. Uniformity of the electromagnetic field existing at the transmission line can be distorted due to these inevitable discontinuities. Moreover, frequency responses are sensitive to them. Therefore, link models should contain discontinuities. Links and these types of discontinuities can be viewed as linear two-port networks. Cascading these two-port descriptions produce the overall link model. In our analysis, all the two-port networks are described by transmission

matrices as a common form since the matrix entries can be easily obtained from the distributed parameters per unit length. Transmission matrix of an entire link is calculated by multiplying the constituent transmission matrices.

There are precise and scalable transmission line models for interconnect wires available in the literature [64, 66]. Link discontinuities are not taken into account in these models due to their complicated structures. Furthermore, specific simulation tools are needed to model these effects properly. In this study, we carry out the analysis for the links with and without discontinuities. Parameters such as discontinuity locations, wire lengths or loss tangent values are varied to evaluate the performance of various kinds of interconnect wires.

Figure 4.4 depicts a simplified lossy differential microstrip model for an interconnect link without discontinuities. A typical channel is shown in Figure 4.5 with packaging, mismatched terminations, and lossy differential microstrip interconnect. Packagings, LC circuits, are placed at both ends of the link to model discontinuities.

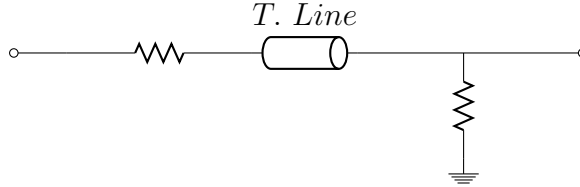


Figure 4.4: Transmission line model of an interconnect without IC packaging.

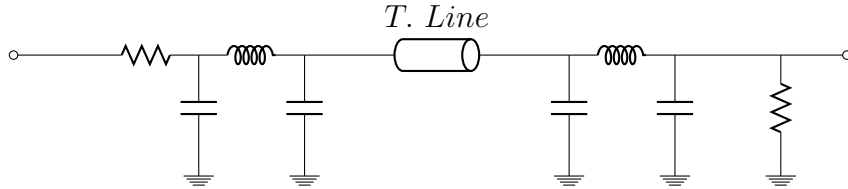


Figure 4.5: Transmission line model of an interconnect with IC packaging.

Alternating electric current density tends to be largest near the surface of the conductor, and it is decayed rapidly with depth inside the conductor. Most of the electric current flows through the *skin* of the conductor that is lying between surface and skin depth level of the conductor. This phenomenon is called skin effect. Effective resistance of the conductor increases at higher frequencies where

the skin depth is shorter. Skin effects are represented by R in Figure 4.2 by using a complex frequency dependent function which contains skin effect constant R_s . Skin effect is not an important issue for narrower wires.

Conduction in the dielectric material is generally negligible, $G_0 = 0$. The time varying electromagnetic field in the dielectric material produced by alternating electric current increases with frequency, and causes heating and loss. This is modeled using frequency dependent capacitance C . Dielectric losses place bandwidth limitations on chip communications. Therefore, low loss materials are used as dielectrics to overcome this problem [66].

Table 4.1: Transmission line model parameters of an interconnect wire on FR4 material.

Parameter	Value
R_0 (Ω/m)	0.0001
R_s ($\Omega/m\sqrt{Hz}$)	$8.7 \cdot 10^{-9}$
L_0 (nH/m)	370
G_0 (pS/m)	1
C_0 (pF/m)	148
$Z_0(\Omega)$	100
f_0 (GHz)	10
c (m/s)	$2.998 \cdot 10^8$
ε_0 (pF/m)	8.85
ε_r	4.9
θ_0	0.021

In the analysis, R , L , G , and C parameters are converted into transmission matrices by using Eq. 4.17. The interconnect wire model is considered as a differential microstrip line with 100Ω matched terminations on a typical FR4 dielectric material. The values used in the analysis are given in Table 4.1. In the table, R_0 is the DC resistance of interconnect per unit length, R_s is the resistivity coefficient of skin effect impedance, ε_r is relative permittivity, ε_0 is free space permittivity, θ_0 is the loss tangent, f_0 is the frequency in which AC parameters are specified, Z_0 is characteristic impedance, and $\nu_0 = \frac{c}{\varepsilon_r}$ is the propagation velocity. Transmission line quantities L and G values are frequency independent quantities, and they are equal to $L_0 = \frac{Z_0}{\nu_0}$ and G_0 , respectively. The values of R

and C are frequency dependent and given in Eq. 4.18.

$$R = R_0 + R_s(1 + j)\sqrt{f}, \quad C = C_0 \left(j \frac{f}{f_0} \right)^{-2\theta_0/\pi} \quad (4.18)$$

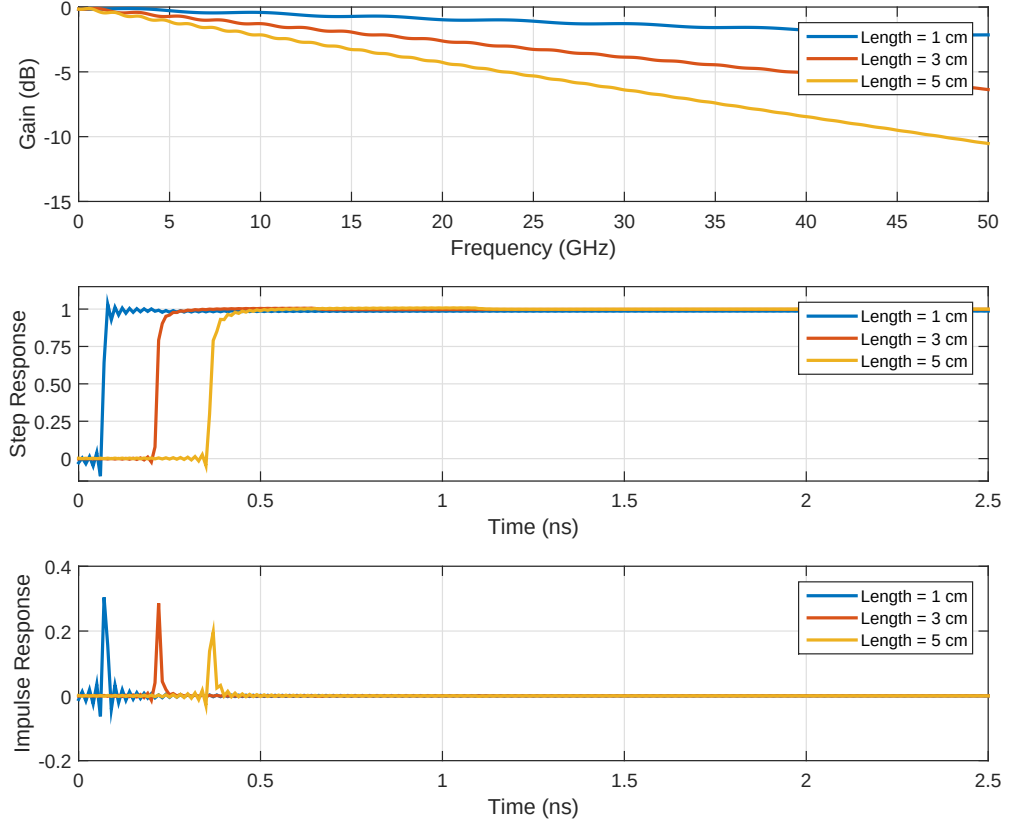


Figure 4.6: Frequency, step and impulse responses of 1, 3, and 5 cm interconnect links without IC packaging.

The simulation starts from a lossy transmission line description including skin effect and dielectric loss, calculates frequency-dependent $RLGC$ parameters, creates transmission matrices for the transmission line and with and without a simple package model to describe the behavior of two-port network. It then combines them and plots the resulting channel response in the time and frequency

domains. The simulation is valid for both interconnect and chip to chip wires.

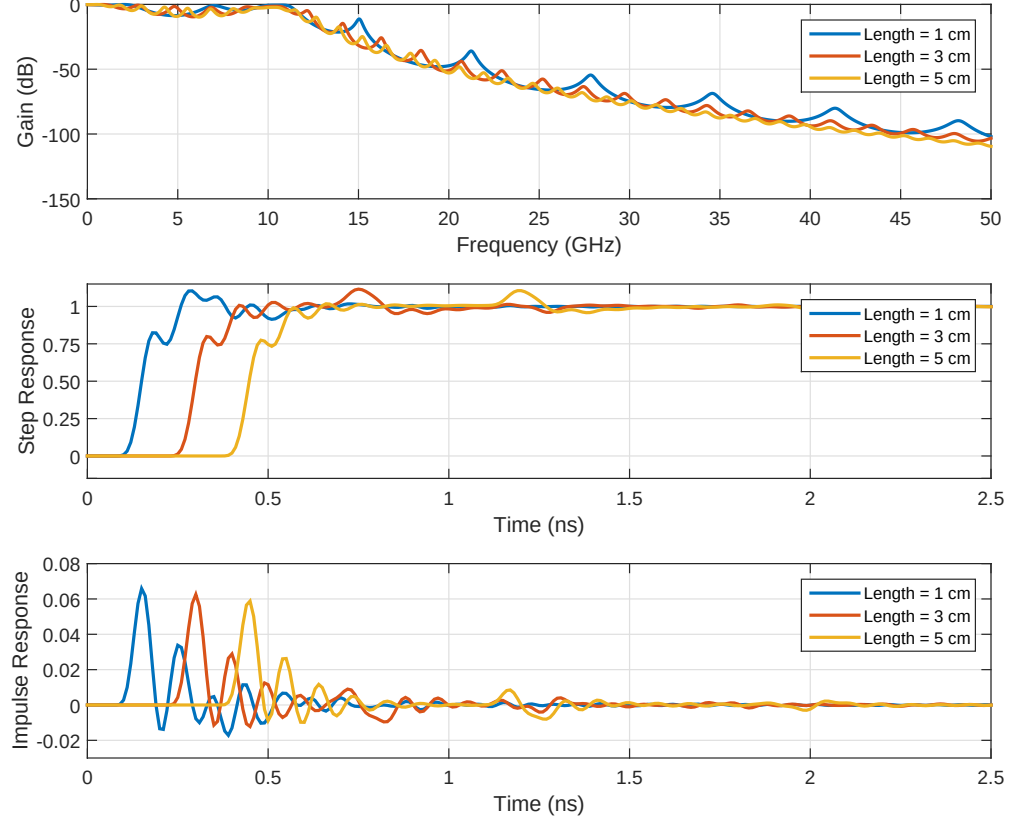


Figure 4.7: Frequency, step and impulse responses of 1, 3, and 5 cm interconnect links with IC package models at either end.

Figure 4.6 presents frequency, step and impulse responses of 1, 3, and 5 cm interconnect links without IC packaging. It is seen that in Figure 4.6 (a) there is a three-fold decrease in signal level over a 5 cm transmission line. In Figure 4.6 (b,c) step and impulse responses suggest that the transmission line delay is less than 0.5 ns.

The effects of discontinuities on frequency, step and impulse responses are given in Figure 4.7. It can be seen that 3 dB-bandwidth drops from 13 GHz to 2.15 GHz by including package models into a 5 cm link. Including discontinuities to the link

models causes large ripples in the frequency responses. These effects are called reflections and they can also be seen in the step and impulse responses. Increasing interconnect lengths cause higher losses at higher frequencies. It should also be noted that, as the interconnect length increases, the amount of propagation delay also increases in both cases.

These results indicate that switch designs with wire lengths up to 5 cm would cause a three-fold drop in signal gain and propagation delay that is less than 0.5 ns when operated at frequencies up to 50 GHz.

4.2 Lumped Circuit Models

Switches and wires in interconnection networks can be modeled using lumped circuits. In particular, a switch can be modeled as a serial resistor as shown in Figure 4.8. When the switch is open, there is no current between the nodes where it is connected. However, when the switch is closed current encounters a serial resistance which is set to $20\ \Omega$ in the simulations.



Figure 4.8: An elementary switch can be modeled as a serial resistor.

Wires connect transistors and switches together and play an important role in the performance of interconnection networks. Correct modeling of interconnect wires is essential for making accurate analyses of interconnection networks. In traditional VLSI circuits, interconnect wires had low resistances since they were wide and thick, and they have lumped capacitances. They could be considered as having equal electric potentials. With the advances in VLSI technologies, wires have become narrower, their resistances are increased, and may delays on wires exceed the gate delays. Besides, when wires are closed together they get capacitively coupled together, and this induces transient undesirable signals on neighboring wires, leading to crosstalk noise.

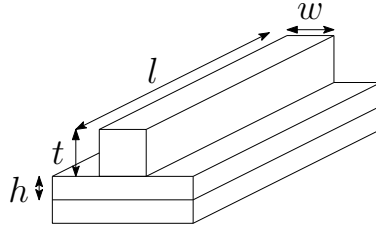


Figure 4.9: Interconnect wire geometry.

Figure 4.9 shows a wire that has length l , width w , thickness t , and dielectric height h . The wire resistance is a function of the wire's cross sectional area. Narrow wires have larger resistances since they constrict the current flow. The wire capacitance is a function of the wire's height and area. Inductive coupling and magnetic effects can also be included in the analysis. However, this adds more complexity to the modeling of interconnect wires.

In general, wires are distributed circuits with resistance, capacitance, conductance, and inductance per unit length. Their behavior can be approximated using lumped circuit models. $L-model$, $\pi-model$, and $T-model$ are three standard lumped circuit approximations. In interconnection networks, each wire section in between switching elements or crosspoints can be considered as wire segments. Figure 4.10 shows how the wire segments can be modeled using lumped circuit elements. The lumped circuit approximation converges to the true distributed circuit, as the number of segments goes to infinity. In this thesis, we use the $L-model$ to describe interconnect wires.

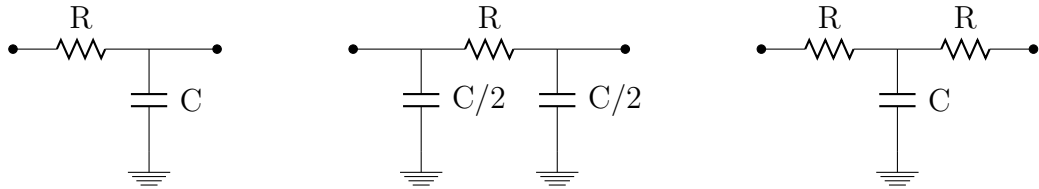


Figure 4.10: Depiction of $L-model$, $\pi-model$, and $T-model$ approximations to distributed RC circuit.

The wire resistance is proportional to the length l and inversely proportional to the cross sectional area $t \cdot w$. Figure 4.9 shows a rectangular wire whose resistance

can be expressed as

$$R = \frac{\rho}{t} \frac{l}{w} \quad (4.19)$$

where ρ is the resistivity of the material ($\mu\Omega \cdot cm$). Since the thickness value t is a coefficient for a given technology, Eq. 4.19 can be rewritten as the following

$$R = R_{\square} \frac{l}{w} \quad (4.20)$$

where $R_{\square}$ is the sheet resistance ($\Omega/square$).

Some frequently used materials are given with their electrical resistivities in Table 4.2. Aluminum and copper are the most preferred interconnect materials due to their low cost and their compatibility with the IC production processes.

Table 4.2: Electrical resistivity of commonly used conductors at 22°C. Retrieved from [7].

Material	ρ ($\mu\Omega \cdot cm$)
Silver (Ag)	1.6
Copper (Cu)	1.7
Gold (Au)	2.2
Aluminum (Al)	2.8
Tungsten (W)	5.3
Titanium (Ti)	43

Extracting and modeling of the wire capacitance in an integrated circuit is not an easy task due to the three dimensional structure of interconnects. An interconnect wire can be modeled as a conductor over the ground plane. Parallel plate capacitances to the ground plane, fringing capacitances along the wire, and the coupling capacitances to the neighboring interconnects are the main constituents of the wire capacitance as illustrated in Figure 4.11. However, there is a simple first order model available in the literature [7, 8]. In this model, the wire capacitance is expressed in terms of the wire's height t and area $w \cdot l$. The approximate formula of the total capacitance is given in Eq. 4.21.

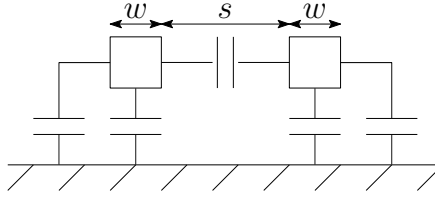


Figure 4.11: Capacitive effects on interconnect wires that have length l , width w , thickness t , and dielectric height h .

$$C = \frac{\varepsilon}{h}wl \quad (4.21)$$

where ε represents the permittivity of the dielectric layer. Relative permittivity of some typical dielectric materials are given in Table 4.3.

Table 4.3: Relative permittivities of some typical dielectric materials where $\varepsilon_0 = 8.854 \times 10^{-12} F/m$ and $\varepsilon = \varepsilon_r \cdot \varepsilon_0$. Retrieved from [8].

Material	ε_r
Free Space	1
Aerogels	1.5
Polyimides	3-4
Silicon Dioxide	3.9
Glass Epoxy	5
Silicon Nitride	7.5
Alumina	9.5
Silicon	11.7

The fringing and coupling (or interwire) capacitances are more complex to calculate and require an electromagnetic field solver for exact results. However, for standard CMOS processes, typical capacitances are known. Area, fringe and coupling capacitances for typical $0.25 \mu m$ CMOS process are presented in Table 4.4, Table 4.5, and Table 4.6, respectively. These tables show the capacitance values for $0.25 \mu m$ CMOS process with 1 layer of polysilicon substrate and 4 layers of aluminum dielectrics. If the wires are placed isolated from the active devices, field columns are used. Accordingly, when the wires routed through the area contains active devices, active columns are used. In this study, we use a two layered structure where interconnect wires, switches, and transistors

located at the same layer.

Table 4.4: Wire area capacitance values for typical $0.25\ \mu m$ CMOS process. The values are given in $(aF/\mu m^2)$. Retrieved from [8].

Layer	Field	Active	Substrate	L1	L2	L3	L4
Substrate	88						
L1	30	41	57				
L2	13	15	17	36			
L3	8.9	9.4	10	15	41		
L4	6.5	6.8	7	8.9	15	35	
L5	5.2	5.4	5.4	6.6	9.1	14	38

Table 4.5: Fringing capacitance values for typical $0.25\ \mu m$ CMOS process. The values are given in $(aF/\mu m)$. Retrieved from [8].

Layer	Field	Active	Substrate	L1	L2	L3	L4
Substrate	54						
L1	40	47	54				
L2	25	27	29	45			
L3	18	19	20	27	49		
L4	14	15	15	18	27	45	
L5	12	12	12	14	19	27	52

Table 4.6: Coupling capacitance values for typical $0.25\ \mu m$ CMOS process with minimally spaced wires. The values are given in $(aF/\mu m)$. Retrieved from [8].

Layer	Substrate	L1	L2	L3	L4	L5
Capacitance	40	95	85	85	85	115

4.3 Pass Transistors and Transmission Gates

MOSFETs can be considered as simple switches, serving as on-off switches between *source* and *drain* terminals by way of a *gate* terminal that controls the current between the source and drain terminals as shown in Figure 4.12. MOSFETs employed in this manner are called pass transistors.

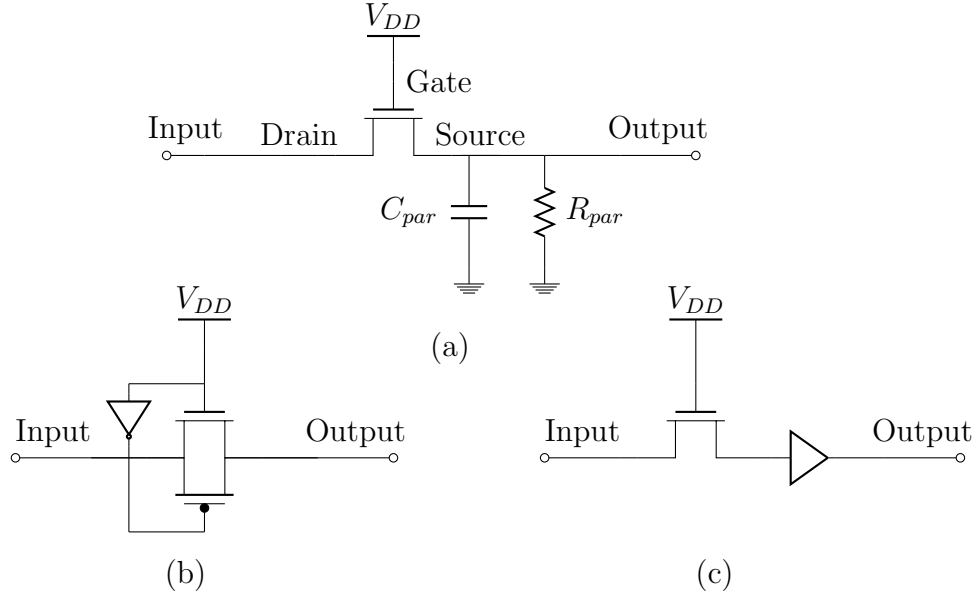


Figure 4.12: Methodologies to avoid the low output voltage problem of pass transistor circuits.

A single pass transistor suffers from threshold drops. In order to pull the output voltage to the rail, additional circuitry is necessary. In the realizations of interconnection networks, pass transistors can be replaced by more robust circuits such as complementary MOS (CMOS) circuits or transmission gates to improve the operational performance of the switches as shown in Figure 4.12 (b, c).

Assume that an N-type MOSFET is used as a pass transistor as shown in Figure 4.12 (a). When the gate terminal voltage is high (V_{DD}), the input signal is transmitted to the output terminal regardless of its voltage level. However, electrical current, which flows through MOSFET, charges up the parasitic capacitance C_{par} until the output-gate voltage difference reaches to the threshold voltage V_T , making the output voltage value lower than the input voltage. When the gate terminal voltage is low, MOSFET becomes non-conductive, and the parasitic capacitance discharges through the parasitic resistor R_{par} to the ground. If the gate terminal voltage is high and the input voltage is low, the parasitic capacitance discharges through the pass transistor and input terminal [67]. This phenomenon is illustrated in Figure 4.13.

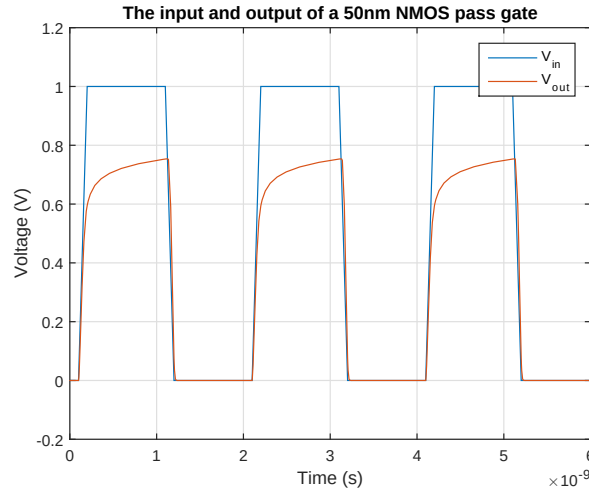


Figure 4.13: Simulation of a pass transistor.

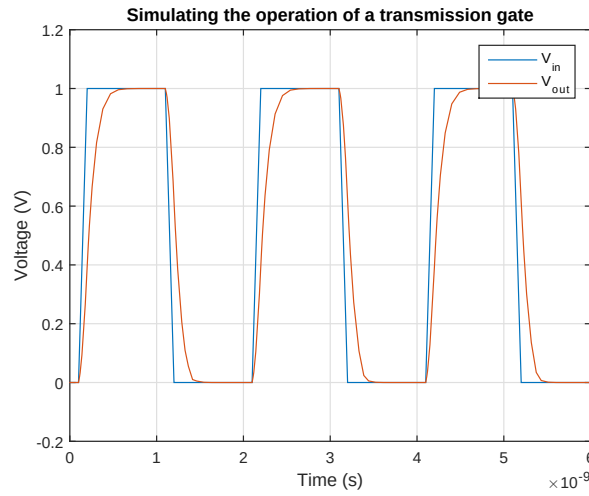


Figure 4.14: Simulation of a transmission gate.

Transmission gates solve the threshold drop problem by their rail-to-rail output voltage swing. Transmission gates consist of parallel pairs of N-type and P-type pass transistors. When the transmission gate is conductive, the output voltage reaches the same voltage level with the input terminal as can be seen in Figure 4.14. As an alternative method, the output voltage can be boosted to the input voltage level by using additional circuits.

In the rest of the thesis, we will use transmission gates at the crosspoints of

interconnection networks, employ the RC equivalent model of transmission gates given in Figure 4.15. The input-output relation of RC equivalent circuit is shown in Figure 4.16. Typical values of the components in this circuit are shown in Table 4.7 where R and C are the resistance and capacitance values for the RC equivalent of corresponding MOSFET in the transmission gate.

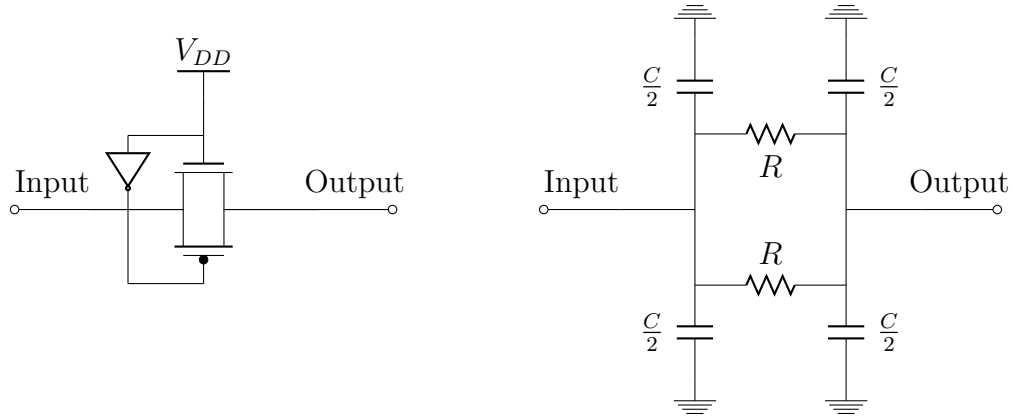


Figure 4.15: A transmission gate circuit and its RC equivalent.

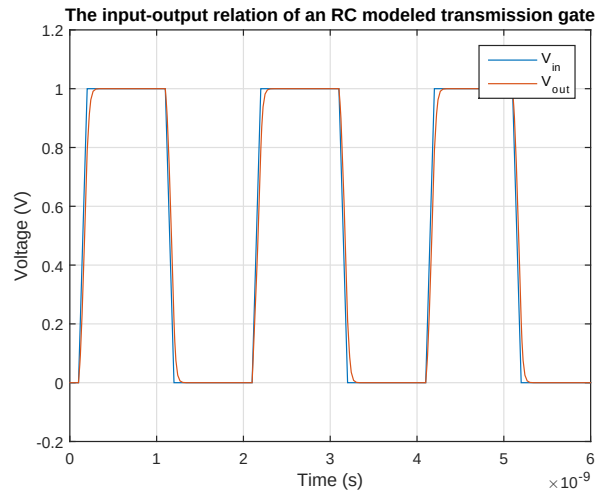


Figure 4.16: RC model simulation of a transmission gate.

Table 4.7: MOSFET model parameters used in this thesis. Retrieved from [9].

Technology	Size	R	C
NMOS (long-channel)	10 μm by 1 μm	1500 Ω	17.5 fF
PMOS (long-channel)	30 μm by 1 μm	1500 Ω	52.5 fF
NMOS (short-channel)	0.5 μm by 50 nm	3400 Ω	0.625 fF
PMOS (short-channel)	1 μm by 50 nm	3400 Ω	1.25 fF

Chapter 5

Power Consumption and Crosstalk Analysis on Interconnection Networks

Along with the continuous increase in the operating frequencies and technology scaling, power and noise optimization have become critical properties for interconnection networks. Accordingly, this chapter gives a detailed account of comparative power consumption analysis of interconnection networks and crosstalk noise analysis of flexing crossbars under the victim aggressor model.

5.1 Power Consumption in Interconnection Networks

Different power dissipation measures have to be considered depending upon the design problem. For example, average power dissipation P_{avg} is important for cooling or battery requirements whereas the peak power value P_{peak} is important for supply line sizing [8]. Before proceeding any further, it would be appropriate to give informative definitions about terms for power and energy.

The instantaneous power $P(t)$, which is measured in Watt (W) or equivalently Joule/s (J/s), consumed by a circuit element is the product of the voltage across its terminals and the current through it.

$$P(t) = I(t)V(t) \quad (5.1)$$

The energy, which is usually expressed in Joule (J) or equivalently Watt · s (W · s), consumed or supplied over time interval $t \in [0, T]$ is equal to the integral of the instantaneous power

$$E = \int_0^T P(t) dt \quad (5.2)$$

Average power is calculated by dividing the energy by time interval

$$P_{avg} = \frac{E}{T} = \frac{1}{T} \int_0^T P(t) dt \quad (5.3)$$

Peak power is the maximum value of the instantaneous power over time interval $t \in [0, T]$

$$P_{peak} = I_{peak}V_{peak} = \max[P(t)] \quad , t \in [0, T] \quad (5.4)$$

The voltage and current are related by the expression $V = IR$ (Ohm's law). Accordingly, the instantaneous power converted from electricity to heat in a resistor is equal to $P_R(t) = \frac{V_R^2(t)}{R} = I_R^2(t)R$. Unlike a resistor, an ideal capacitor does not dissipate any power. When it is charged from 0 to V_C (voltage at the terminals of the capacitor), it stores an energy given by $E_C = C \int_0^{V_C} V(t) dV = \frac{1}{2}CV_C^2$. This energy is released when the capacitor discharges back to 0.

Figure 5.2 shows a pass transistor and a transmission gate. They become active when the input switches from 0 to 1, and the load capacitor is charged to

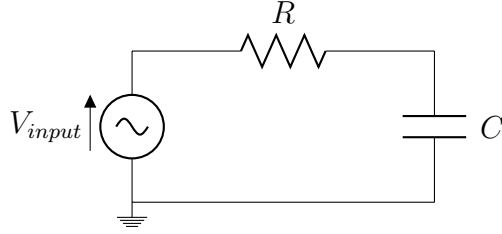


Figure 5.1: First order RC Network

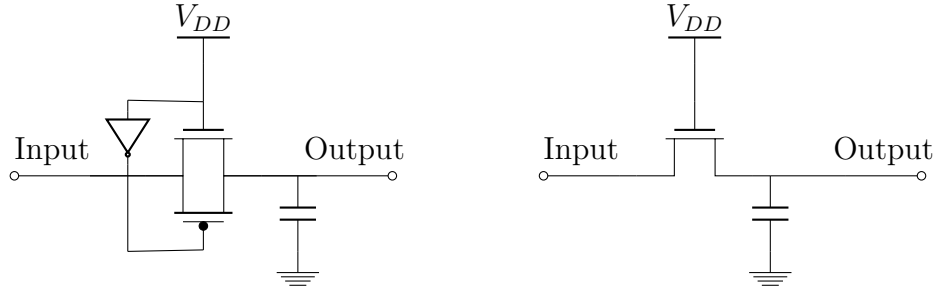


Figure 5.2: Transmission gate and pass transistor circuits.

V_{DD} . When the input switches back from 1 to 0, the capacitor discharges. The stored energy is dissipated in the MOSFETs.

From the above analysis, we can conclude that power dissipation is function of switching frequency in CMOS circuits. If a CMOS circuit changes its state at some frequency, f_{sw} , over time interval T , the capacitive load will be charged and discharged $T \cdot f_{sw}$ times. The average power dissipation can be computed as follows:

$$P = \frac{E}{T} = \frac{T f_{sw} C V_{DD}^2}{T} = f_{sw} C V_{DD}^2 \quad (5.5)$$

Resulting value is also called dynamic power. In general, f_{sw} is given by the product of the activity factor α and the clock frequency f [7]. Correspondingly, the dynamic power dissipation is expressed as

$$P = \alpha C V_{DD}^2 f \quad (5.6)$$

A chip's size and speed is closely related with its organizational circuitry. Therefore, counting only the transistors in a chip is a deceptive approach to calculate the power dissipation [68]. In interconnection networks, dynamic power is the dominant component of the total power consumed. Therefore, this work considers dynamic power dissipation to evaluate interconnection networks. Inside an interconnection network, different signals travel along different paths, and traffic load along path changes from time to time [69]. Therefore, we need to make some assumptions about power dissipation in switching nodes. In our analysis, we assume that all interconnection networks are produced with the same technology. In addition, the area calculations are done assuming that there is only one metal layer despite the fact that VLSI chips consist of many overlapping layers. This is a reasonable assumption given that layers are generally insulated from one another and noise issues are limited to a single layer. In particular, we shall adopt the bit energy model [69]. However, our analysis is different from [69] as we take into account the cell sizes and possible scaling in this bit energy model, and examine power consumption of different topologies under full network traffic. According to bit energy model, total energy consumption is calculated by summing up the energy consumed on node switches, E_S , on interconnect wires, E_W , on multiplexers, E_M , and on internal buffers E_B . Definitions of these energies are given in the following paragraph.

Crosspoints and multiplexers are located in between input-output terminals inside the networks, and they are used to route signals from one stage to another. When a bit passes over a crosspoint (or passes over a multiplexer), it consumes energy in the amount of E_S (or E_M), at a rate of the switching frequency of its logic gates. The signal on interconnect wires are toggled during switchings. During this charging or discharging process, energy in the amount of E_W is consumed. E_W is a function of the wire length, which is estimated using Thompson's grid model [68] by mapping an interconnection network into a grid and then counting the cells in the grid. In addition to these, data bits are temporarily stored in buffers in case of contention. Energy consumption on internal buffers, E_B , emanates from access requests and memory cleaning operations.

5.1.1 The Grid Model for Crossbar Switches

Recall that an $n \times n$ crossbar network is an array of crosspoints to connect a set of n inputs with a set of n outputs. Crosspoints on the crossbar network can be realized by pass transistors or transmission gates. Since every input-output connection has its dedicated path, crossbar networks are nonblocking and no internal buffering is needed. In addition to that, in our analysis, we assume that there is no *destination contention* problem in the crossbar networks. Accordingly, we also assume that only n out of $n \times n$ crosspoints can operate at the same time. Figure 5.3 shows Thompson's grid model of a 2×2 crossbar network, where each crosspoint occupies only 1 cell. However, both vertical and horizontal wires require one additional cell so that each interconnect wires have $2n$ length. Therefore, the worst case energy requirement of a crossbar network fabrics is given in Eq. 5.7.

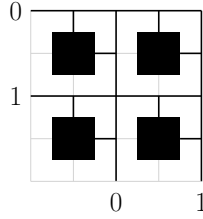


Figure 5.3: Thompson grid model of a 2×2 crossbar network.

$$E_{crossbar} = n \cdot E_S + 2 \cdot (2n) \cdot E_W \quad (5.7)$$

where E_S is the bit energy for the crosspoint and E_W is the bit energy of a Thompson grid wire. Thus, energy consumption of the crossbar network linearly increases with the number of input and output terminals n .

5.1.2 Flexing Crossbar Interconnection Network

An $n \times n$ flexing crossbar network connects a set of n inputs with a set of n outputs. Only nr intersections serve as crosspoints out of n^2r^2 due to the inefficient use

of crossbar array in the middle of flexing crossbar. However, the crossbar array allows any input-output connection without blocking other inputs and outputs, i.e., all of the crosspoints can operate at the same time. Nevertheless, we assume that an input is not communicated more than one output at the same time for concreteness. In a flexing crossbar, both vertical and horizontal wires have $2n^2$ length, and each crosspoint occupies only 1 cell as in a crossbar network.

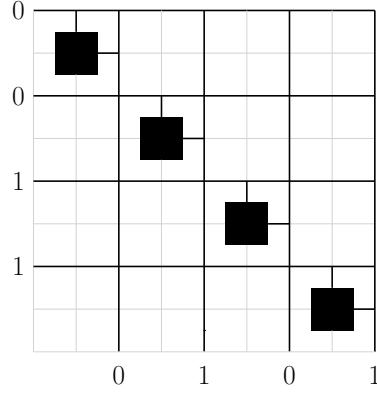


Figure 5.4: Thompson grid model of a 2×2 flexing crossbar network.

Figure 5.3 shows Thompson grid model of a 2×2 flexing crossbar network. Eq. 5.8 gives the worst case energy requirement of a flexing crossbar network.

$$E_{flexing\ crossbar} = n \cdot E_S + (2n^2 + 2) \cdot E_W \quad (5.8)$$

It is possible to reduce the energy requirement of the $n \times n$ flexing crossbar by scaling the cell size by a factor of n . In this case, the total bit energy can be shown to be equal to that of $n \times n$ crossbar.

5.1.3 Baseline Interconnection Network

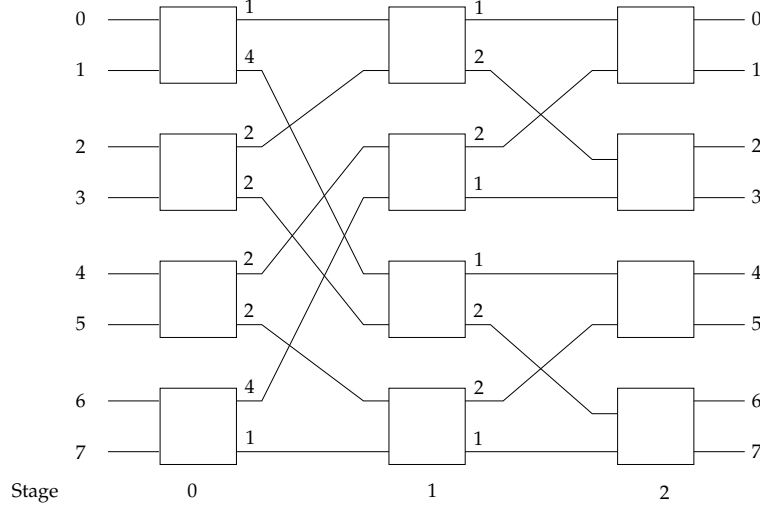


Figure 5.5: An 8×8 Baseline network

An $n \times n$ baseline network has $n = 2^s$ inputs and $n = 2^s$ outputs, where $s = \log_2(n)$ is the number of stages. It has total of $\frac{1}{2}n\log_2(n)$ switches in s stages. Figure 5.5 shows a 8×8 baseline network with Thompson grid lengths of interconnect wires. Thompson grid model for the first stage of this network is represented in Figure 5.6. It can be seen that the longest interconnect wire has a length of 4 cells.

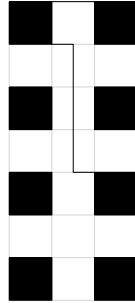


Figure 5.6: Thompson grids on a Baseline network

Baseline networks suffer from *interconnect contention* problems such that different data paths can use the same interconnect [69]. Therefore, internal buffers are needed in the switches. The worst case bit energy formula of the

baseline network is given in Eq. 5.9.

$$E_{baseline} = E_B + \sum_{k=0}^s \sum_{i=0}^{n-1} E_{W(i,k)} + \left(\frac{1}{2} n \log_2(n) \right) E_S \quad (5.9)$$

where s denotes the number of stages and n denotes the number of inputs.

5.1.4 Fully-Connected Interconnection Network

An $n \times n$ fully-connected interconnection network uses multiplexers to switch one of the n inputs through to a common output as shown in Figure 5.7. Each multiplexer is controlled by a select input that determines which input should be transmitted to the output.

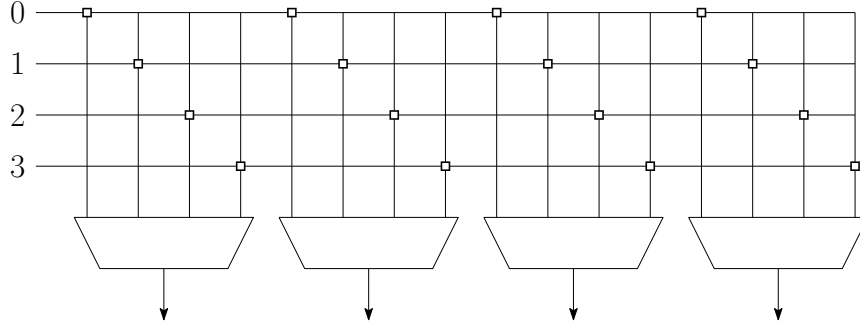


Figure 5.7: Fully connected network

In fully connected networks each input-output connection has its dedicated data path as in crossbar networks. Therefore, fully-connected networks are also nonblocking such that no internal buffers are needed in the power modeling. Energy is consumed on interconnect wires and multiplexers. Eq. 5.10 gives the worst case energy consumption.

$$E_{fully-connected} = n \cdot E_M + (n^2 + n) E_W \quad (5.10)$$

It should be noted that in fully connected networks, each bit consumes energy

only at the multiplexer that it passes. Wire lengths can be directly calculated from Figure 5.7 such that horizontal wires have length of n^2 grids whereas vertical wires have length of n grids.

5.1.5 Power Consumption Analysis Results

The bit energy value represents the energy consumption for one bit. We assume that all data paths require same amounts of energy to be able to easily calculate the total power consumption of the entire fabric. In our analysis, we compare power consumption of flexing crossbars with other interconnection network architectures for different number of input-output terminals, namely, 4×4 , 8×8 , 16×16 and 32×32 . We use the bit energy values provided in [69]. Table 5.1 lists these values.

Table 5.1: Bit energy values on different network architectures. Retrieved from [69]

Architecture	$E_S(\times 10^{-15} J)$	$E_W(\times 10^{-15} J)$	$E_B(\times 10^{-15} J)$				$E_M(\times 10^{-15} J)$			
			N = 4	N = 8	N = 16	N = 32	N = 4	N = 8	N = 16	N = 32
Crossbar	220	87	-	-	-	-	-	-	-	-
Flexing Crossbar	220	87	-	-	-	-	-	-	-	-
Baseline	1080	87	140	140	154	222	-	-	-	-
Fully Connected	-	87	-	-	-	-	431	782	1350	2515

Figure 5.8 presents the estimated power consumption values of 4 different network topologies under full traffic throughput and 480 Mbps data rate. It illustrates the impact of number of terminals on power. According to the analysis results, the following conclusions can be drawn:

As the number of terminals increases or equally as the interconnection networks become larger, interconnect wires, instead of switching nodes, increasingly dominate the power consumption. Besides, increasing the number of terminals causes an exponential increase in the area occupied by a flexing crossbar on a chip. Hence, flexing crossbar has the highest power consumption among all architectures with large number of terminals. This is an expected result, as larger chips obviously dissipate more power than the smaller ones. However, flexing crossbars do not waste so much power even for 32×32 configuration. Moreover,

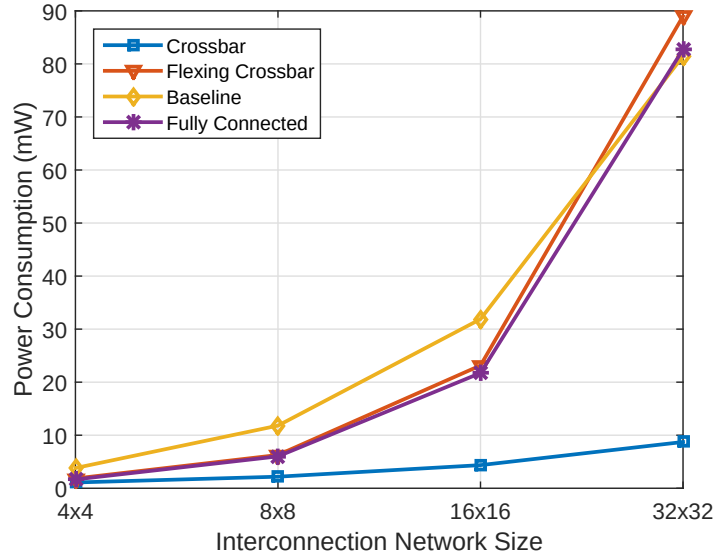


Figure 5.8: Power consumption under full traffic throughput

as it will be shown in the following they exhibit improved noise immunity to both crosstalk and discontinuity dependent attenuation. Therefore, flexing crossbars ensure optimal transmission over the VLSI chips.

5.2 Noise in Interconnection Networks

Electrical noise refers to unwanted random fluctuations in electrical signals. In the presence of reduced power supply voltages to sustain drive strength in deep submicron circuits; threshold voltages are also reduced, resulting in lower noise margins. Therefore, analysis and reduction of noise becomes critical for high-speed VLSI circuits. Noise appears in a variety of forms, and affects circuits in different ways. It can be broadly classified under two types:

- Inherent circuit noise arises within the circuit. It results from the discrete and random movement of charges in a wire or device. Thermal noise, shot noise, phase noise, avalanche noise, burst noise and flicker noise can be given as examples of inherent noise.

- Interfering noise, which is picked up from the outside, is due to coupling from the signals in adjacent links, circuits or devices. This type of noise is also referred to as crosstalk.

Among the various sources of noise, crosstalk due to the capacitive coupling effects is the dominant source of noise in current CMOS digital integrated circuits. Contemporary high-speed CMOS technologies accommodate much more metal layers with increased density and reduced spacing between interconnects leads to significant increase in capacitive coupling effects that deteriorates the signal integrity. The severe adverse effects of coupling noise impose timing problems that can bring delay and then circuit malfunctions. Therefore, demanding performance requirements make crosstalk noise a major source of performance degradation in crossbar switching networks.

A poor understanding of crosstalk can lead to overly conservative design rules resulting in poor performance. It can also lead to irregular logic errors which may only be triggered by certain logic combinations which are difficult to detect. Thus, to properly deal with this problem and to design noise immune chips a suitable crosstalk noise estimation method is required. In this part of the thesis, we make comparative analysis of some crosstalk noise metrics proposed in [4–6, 11], and we suggest a geometric approach reducing the adverse effects of crosstalk noise. We use interconnect modeling method presented in Chapter 4. By using flexing crossbars as benchmark circuits, we propose noise tolerant topologies.

5.3 Crosstalk Noise in Flexing Crossbars

Crosstalk may occur as a result of inductively induced voltages or parasitic capacitances between interconnects inside VLSI chips. In general, designers ignore inductive effects on interconnects since extracting and modeling of these effects are extremely difficult due to their global nature. This is a justifiable idea since inductive coupling and magnetic effects are so small compared to capacitive coupling effects that they can be ignored [3, 5–8, 10, 11]. The interconnect wire

modeling method presented in Chapter 4 allows us to properly extract the parasitic capacitances associated with interconnects, which is an essential step to characterize and reduce crosstalk noise.

Figure 5.9 indicates N neighboring N segmented microstrip lines with their coupled distributed RC circuit model. In such structures, it is difficult to derive an expression to estimate crosstalk noise due to the complexity of the circuit. Besides, crosstalk mainly affects the lines which are close to the aggressor lines. Therefore, researchers consider only two parallel on-chip lines running parallel on the same metal layer as a circuit model to estimate the coupling for on-chip interconnects and derive an analytical expression for the crosstalk noise [4–6, 11, 70].

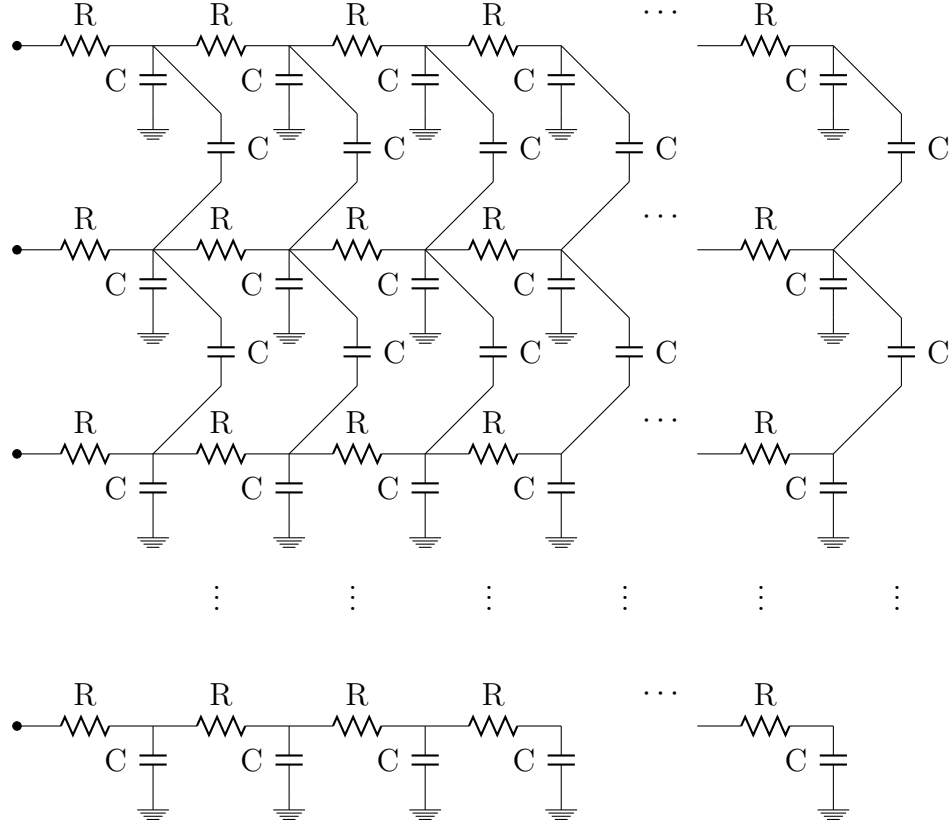


Figure 5.9: Schematic representation of N capacitively coupled interconnect wires.

We start the analysis by reviewing a simplified model to give the basic concepts of the crosstalk noise and to clarify its importance. Figure 5.10 shows a section

of two closely placed on-chip interconnect wires. Assume that the wire lengths are less than $\lambda/4$. A part of the signal transmitted by an *aggressor* couples into a *victim* via capacitance C_c , and goes to ends of the victim line R_{21} and R_{22} . Now, let us make a comparative noise analysis of the circuit given in Figure 5.10 for two different cases. Similar analysis can be found in [70]. In our analysis, we suppose that coupling capacitance between aggressor and victim lines is $C_c = 20 \text{ fF}$, capacitance of the victim line to ground is $C_2 = 60 \text{ fF}$, and aggressor lines carries a 1 GHz sinusoidal signal V_1 of 1 V peak value.

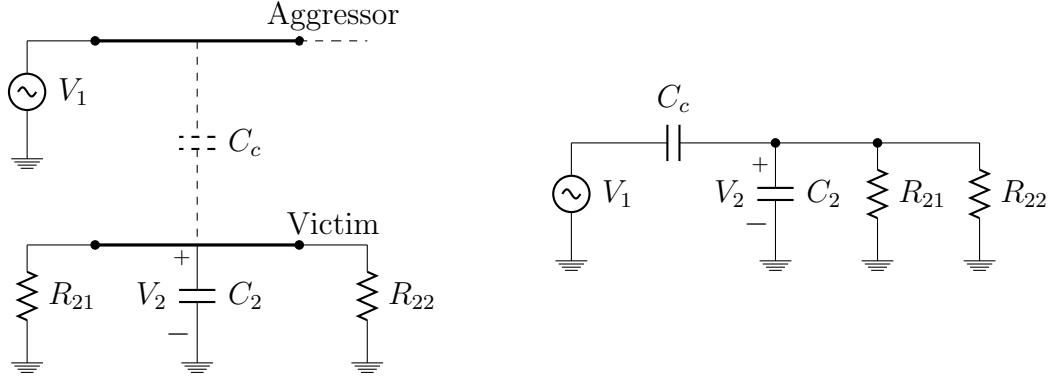


Figure 5.10: Capacitive coupling between parallel wires and the equivalent circuit.

Case 1: Assume that resistances R_{21} and R_{22} are very high, i.e., either ends of the victim line is open circuit. Noise voltage V_2 is a fraction of the input signal V_1 that contaminates the victim line, and can be calculated as

$$V_2 = V_1 \frac{Z_2}{Z_2 + Z_c} = V_1 \frac{\frac{1}{j\omega C_2}}{\frac{1}{j\omega C_2} + \frac{1}{j\omega C_c}} = V_1 \frac{C_c}{C_2 + C_c} \quad (5.11)$$

Eq. 5.11 is not frequency dependent in which case the victim line has infinite loads. The ratio of C_2 and C_c has a direct effect on the amount of the voltage picked up from capacitive coupling. C_2 can be increased or C_c can be decreased to reduce the coupling. This can be physically accomplished by separating the aggressor and victim lines from each other.

Substituting numerical values into Eq. 5.11 gives

$$V_{2 \text{ peak}} = 1 \frac{20 \text{ fF}}{20 \text{ fF} + 60 \text{ fF}} = 0.25 \text{ V} \quad (5.12)$$

It can be stated that the resulting crosstalk value is surprisingly strong.

Case 2: Let the parallel combination of R_{21} and R_{22} be $R_2 = 100 \Omega$ and let Z_2 be the equivalent impedance of R_2 and C_2 in parallel

$$Z_2 = \frac{R_2 \frac{1}{j\omega C_2}}{R_2 + \frac{1}{j\omega C_2}} = \frac{R_2}{1 + j\omega C_2 R_2} \quad (5.13)$$

The peak value of the crosstalk noise on the victim line is

$$V_2 = V_1 \frac{Z_2}{Z_2 + Z_c} = V_1 \frac{\frac{R_2}{1 + j\omega C_2 R_2}}{\frac{R_2}{1 + j\omega C_2 R_2} + \frac{1}{j\omega C_c}} = V_1 \frac{j\omega C_c R_2}{1 + j\omega R_2 (C_2 + C_c)} \quad (5.14)$$

Whenever $\omega R_2 (C_2 + C_c) \gg 1$ or equivalently $f \gg 1 / (2\pi R_2 (C_2 + C_c))$, Eq. 5.14 reduces to Eq. 5.11. We can conclude that the high frequency behavior converges to the first case where the victim line has infinite loads at its ends (R_{21} and R_{22} are very high, i.e., open circuits). With the given numerical values, frequency bound is found as $f \gg 19.89 \text{ GHz}$.

When $f \ll 19.89 \text{ GHz}$ Eq. 5.14 converges to

$$V_2 \approx V_1 (j\omega C_c R_2) \quad (5.15)$$

Substituting numerical values, we obtain

$$|V_{2 \text{ peak}}| \approx 1 \cdot (2\pi 10^9 \cdot 20 \cdot 10^{-15} \cdot 100) \approx 12.6 \text{ mV} \quad (5.16)$$

Resulting crosstalk value is 1/20 of the value calculated in the first case. Reducing the victim line impedance reduces also the peak value of crosstalk noise. Therefore in order to reduce the capacitive crosstalk, providing a low impedance victim line is sufficient.

We will now consider the circuit given in Figure 5.9 for two parallel lines which is illustrated in Figure 5.11.

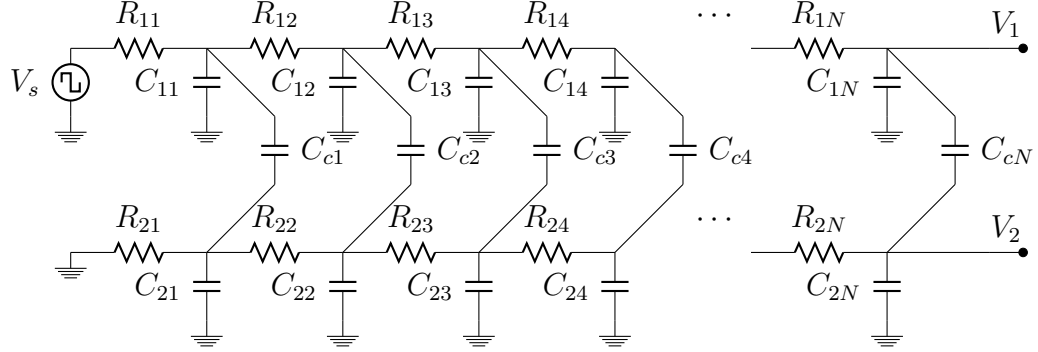


Figure 5.11: Schematic representation of capacitively coupled aggressor and victim lines.

According to Devgan [4], victim line, aggressor line, and coupling between them can be represented by a linear circuit description. The state space representation of the system is presented in Eq. 5.17.

$$\begin{bmatrix} C_1 & C_c \\ C_c & C_2 \end{bmatrix} \begin{bmatrix} sV_1 \\ sV_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} V_s \quad (5.17)$$

where C_1 , C_2 , and C_c are capacitance arrays, V_s is the input signal, V_1 is an array of node voltages on the aggressor line, V_2 is an array of node voltages on the victim line, A_{11} and A_{22} are the equivalent resistance arrays of the aggressor and victim lines, respectively. Expanding the state space expression gives the following equations

$$sC_1V_1 + sC_cV_2 = A_{11}V_1 + A_{12}V_2 + B_1V_s \quad (5.18)$$

$$sC_cV_1 + sC_2V_2 = A_{21}V_1 + A_{22}V_2 + B_2V_s \quad (5.19)$$

Or equivalently,

$$V_1 = (sC_1 - A_{11})^{-1} \left[(A_{12} - sC_c)V_2 + B_1V_s \right] \quad (5.20)$$

$$V_2 = (sC_2 - A_{22})^{-1} \left[(A_{21} - sC_c)V_1 + B_2V_s \right] \quad (5.21)$$

Substituting Eq. 5.20 into Eq. 5.21 gives

$$V_2 = (sC_2 - A_{22})^{-1} \left[(A_{21} - sC_c)(sC_1 - A_{11})^{-1} \left[(A_{12} - sC_c)V_2 + B_1V_s \right] + B_2V_s \right] \quad (5.22)$$

Eq. 5.22 can be rewritten as

$$\left[(sC_2 - A_{22}) - (A_{21} - sC_c)(sC_1 - A_{11})^{-1}(A_{12} - sC_c) \right] V_2 = (A_{21} - sC_c)(sC_1 - A_{11})^{-1}B_1V_s + B_2V_s \quad (5.23)$$

A_{12} indicates the resistive path from aggressor to victim line, and A_{21} indicates a resistive path from victim to aggressor line. However, the aggressor and victim lines are not on the same line and there is no physical connection between them. Therefore, A_{12} and A_{21} are equal to zero for all coupling noise issues. In a like manner, B_2 indicates a resistive path from source to the victim line which means that the victim line is directly excited by the source. Similarly, B_2 can be taken as zero. Therefore, Eq. 5.17 becomes

$$\begin{bmatrix} C_1 & C_c \\ C_c & C_2 \end{bmatrix} \begin{bmatrix} sV_1 \\ sV_2 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} V_s \quad (5.24)$$

Eq. 5.22 is simplified to

$$\left[(sC_2 - A_{22}) - sC_c(sC_1 - A_{11})^{-1}sC_c \right] V_2 = -sC_c(sC_1 - A_{11})^{-1}B_1V_s \quad (5.25)$$

$$H(s) = \frac{V_2}{V_s} = \frac{-sC_c(sC_1 - A_{11})^{-1}B_1}{(sC_2 - A_{22}) - sC_c(sC_1 - A_{11})^{-1}sC_c} \quad (5.26)$$

It can be seen from Eq. 5.26 that $H(s)$ has a zero at $s = 0$. The peak value of crosstalk noise that can be induced at the far end of the victim line (V_2) can be computed by using this fact. Applying an infinite ramp signal to the aggressor input (V_s) produces a delayed infinite ramp at the output of the aggressor line (V_1). Nevertheless, all the capacitances on the victim line are charged up to their maximum value. Whether or not the input signal is infinite, derivative of

the coupled noise is zero at $t = \infty$. At that time the coupled noise V_2 is at its maximum value.

The maximum value of V_2 can be computed from Eq.5.26 using the final value theorem. When $t \rightarrow \infty$ or equally $s \rightarrow 0$, the final value of V_2 is computed as follows:

$$V_{2,max} = \lim_{s \rightarrow 0} sV_2 = \lim_{s \rightarrow 0} sH(s)u(s) = \lim_{s \rightarrow 0} sH(s) \frac{\dot{u}}{s^2} = \lim_{s \rightarrow 0} H(s) \frac{\dot{u}}{s} \quad (5.27)$$

$$V_{2,max} = -A_{22}^{-1}C_cA_{11}^{-1}B_1\dot{u} = -A_{22}^{-1}C_cA_{11}^{-1}B_1\frac{V_{DD}}{t_r} \quad (5.28)$$

where t_r is the rise time of the aggressor signal. For the sake of simplicity, Devgan assumes that the rise and fall times are equal. According to Eq. 5.28, noise voltage reaches its maximum value at $t = t_r$. This is a justifiable idea. However, if the rise time is fast, then maximum noise value goes to infinity. Therefore, we can assert that Devgan's method cannot accurately estimate the peak crosstalk values when the applied signals have shorter rise-times.

To compare this approach with others, let us consider a two-segmented circuit that contains an aggressor and a victim lines. The circuit structure is given in Figure 5.12.

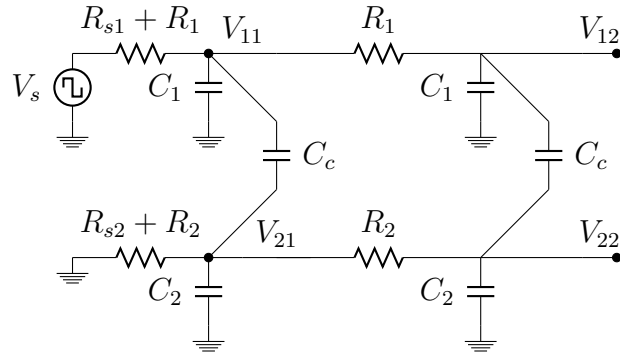


Figure 5.12: Schematic representation of capacitively coupled two segmented aggressor and victim lines.

In Figure 5.12, V_{21} and V_{22} represent the voltages at near-end and far-end of the victim line, respectively. For this circuit, Devgan's metric yields the following

equations.

$$V_{21,Devgan} = (2R_2 + 2R_{s2})C_c \frac{V_{DD}}{t_r} \quad (5.29)$$

$$V_{22,Devgan} = (2R_2 + 3R_{s2})C_c \frac{V_{DD}}{t_r} \quad (5.30)$$

Crosstalk noise at the far end of the victim line is of great importance in our analysis since it directly affects the outputs. Let the parasitics values are $C_1 = 60fF$, $C_2 = 120fF$, $C_c = 100fF$, $R_{s1} = 20\Omega$, $R_{s2} = 20\Omega$, $R_1 = 100\Omega$, $R_2 = 100\Omega$, $t_r = 0.08ns$ and $V_{DD} = 2V$.

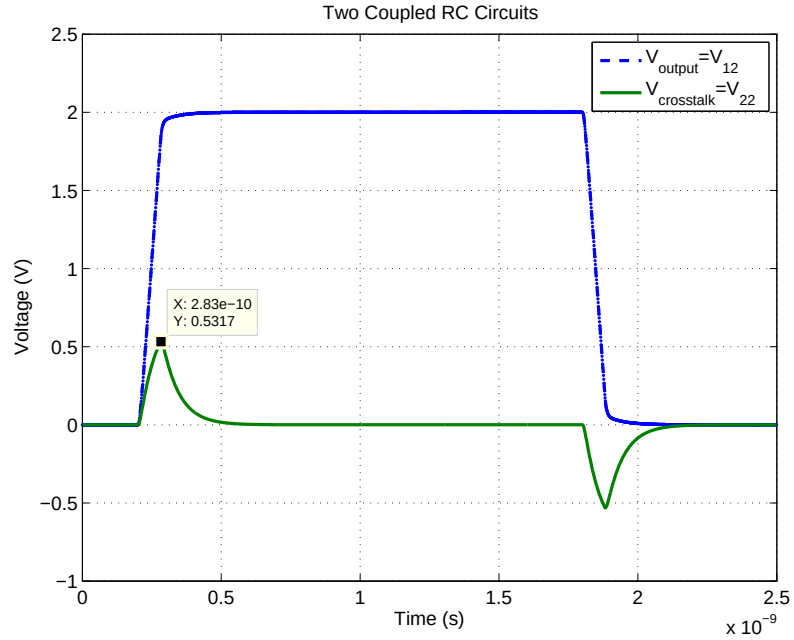


Figure 5.13: Output voltages at the far end of aggressor ($V_{out} = V_{12}$) and victim ($V_{crosstalk} = V_{22}$) lines .

By using Eq. 5.30, maximum crosstalk voltage at the far end of the circuit (V_{22}) is found as $0.65V$. According to the HSPICE and Simulink simulations given in Figure 5.13, the peak value of V_{22} is $0.53V$. The estimation error is 23%.

Let us repeat the analysis for another set of parasitics. Assume that $C_1 = 235fF$, $C_2 = 220fF$, $C_c = 200fF$, $R_{s1} = 20\Omega$, $R_{s2} = 30\Omega$, $R_1 = 100\Omega$, $R_2 = 100\Omega$, $t_r = 0.08ns$ and $V_{DD} = 2V$. Using Devgan's method, the peak crosstalk noise is again found as $0.65V$. However, according to the simulations, the peak

noise value at the far end of the victim line is $0.36V$. Figure 5.14 shows the simulation results. The estimation error is 80%.

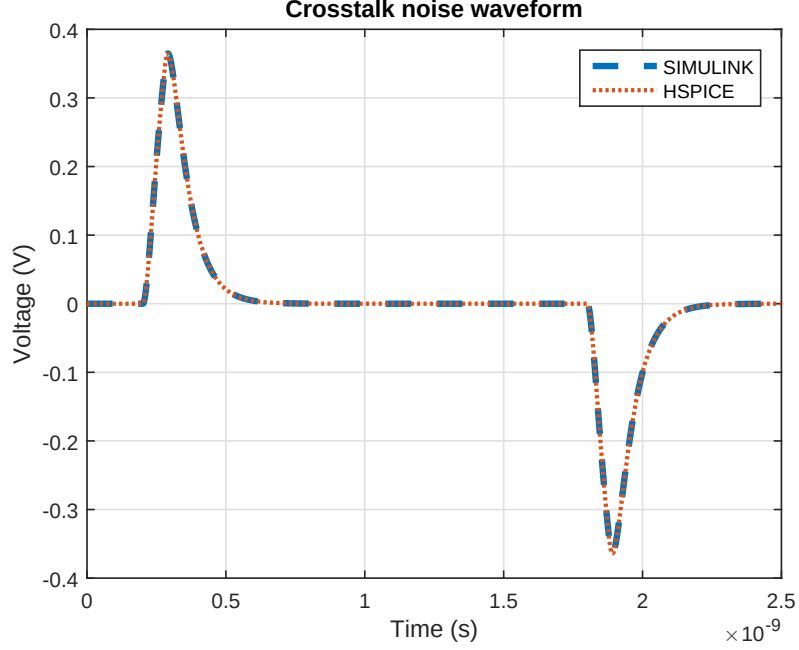


Figure 5.14: HSPICE and Simulink simulations give the same crosstalk noise waveforms.

Heydari and Pedram alter Devgan's method, and present a new expression capable of predicting the crosstalk waveforms [3, 11]. The authors agree with Devgan's result that crosstalk noise reaches the maximum value at $t = t_r$. According to their observation, capacitive crosstalk at each node on the victim line is a rising exponential function during the time interval that the applied signal to the aggressor line is rising. Maximum value of the crosstalk noise at each node is actually equal to the corresponding rising exponential function at $t = t_r$. Authors propose the following metric for crosstalk noise estimation based on Devgan's method

$$V_{2,max} = V_2 \left(I - \exp \left(\text{diag} \left(-\frac{t_r}{\tau_i} \right) \right) \right) \quad \text{for } i = 1, 2, \dots, N \quad (5.31)$$

where $\text{diag}(x)$ is a diagonal matrix with all its main diagonal entries are x . τ_i

represents the time constant of i^{th} node on the victim line, and V_2 is a vector and computed by Devgans metric. It contains steady state voltage values of the noise at the victim nodes. On the victim line, each node sees two capacitances: a coupling capacitance, C_{ci} , and a ground capacitance, C_{2i} . Time constant τ_i at each victim line node is obtained by summing up the individual time constants of these two capacitances, C_{2i} and C_{ci} . An individual time constant value is equal to the product of the corresponding capacitance and the equivalent circuit resistance while all the other capacitances are open circuited [3, 11]. In order to calculate time constants, authors construct an equivalent circuit containing ground (C_{2i}), and coupling (C_{ci}) capacitances, and equivalent resistances, ($R_{1i,eq.}$, $R_{2i,eq.}$) as shown in Figure 5.15. In the equivalent circuit, all the other capacitances are replaced with open circuit connections. Therefore, equivalent node resistances are given as $R_{1i,eq.} = \sum_{j=1}^i R_{1i}$ and $R_{2i,eq.} = \sum_{j=1}^i R_{2i}$. Characteristic polynomial of the second order transfer function of this circuit is

$$\begin{aligned}\Lambda_i(s) &= R_{1i,eq.} R_{2i,eq.} C_{2i} C_{ci} s^2 + \left([(R_{1i,eq.} + R_{2i,eq.}) C_{ci} + R_{2i,eq.} C_{2i}] s + 1 \right) \\ &= R_{1i,eq.} R_{2i,eq.} C_{2i} C_{ci} s^2 + \tau_{v,i}\end{aligned}\quad (5.32)$$

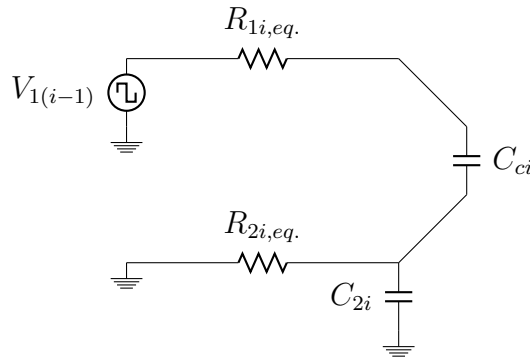


Figure 5.15: Equivalent circuit constructed to calculate the time constant of the i^{th} node of the victim line. Retrieved from [3]

The time constant of this circuit, $\tau_{v,i}$, is equal to the first order term in the characteristic polynomial. However, input voltage source should be a unit step

function with zero rise time for $\tau_{v,i}$ to be equal to the time constant of i^{th} node in the victim line. This is not the case in real systems. Based on this fact, Heydari and Pedram state that the time constant of i^{th} node on the victim line, τ_i , is also depended on the propagation delay of the signal coming from other paths, $\tau_{a,i}$ established by the coupling capacitances $C_{ck}(k = 1, 2, \dots, i - 1)$ [11]. Therefore, the overall delay from the input signal node to the i^{th} node of the victim line is

$$\tau_i = \zeta \cdot [R_{1i,eq}.R_{2i,eq}.C_{2i}C_{ci}s^2 + \tau_{a,i}] \quad \text{for } i = 1, 2, \dots, N \quad (5.33)$$

where $\tau_{a,i}$ is

$$\begin{aligned} \tau_{a,i} = & R_{1i,eq}.(C_{ci} + C_{1i}) \\ & + \sum_{k=1}^{i-1} [R_{1k,eq}.(C_{ck} + C_{1k}) + R_{2k,eq}.(C_{ck} + C_{2k})] \quad \text{for } i = 1, 2, \dots, N \end{aligned} \quad (5.34)$$

and ζ is a constant factor for the delay increase due to the finite input signal slope. In their analysis [3], Heydari and Pedram use $\zeta = 1.07$. Combining Eq. 5.33 and Eq. 5.34 yields the following expression for τ_i where $i = 1, 2, \dots, N$:

$$\tau_i = \zeta \cdot \left[R_{1i,eq}.C_{ci} + \sum_{k=1}^{i-1} [R_{1k,eq}.(C_{ck} + C_{1k}) + R_{2k,eq}.(C_{ck} + C_{2k})] \right] \quad (5.35)$$

The peak value of the crosstalk noise can be computed using Eq. 5.31 with the expression of τ_i . Heydari and Pedram's metric yields the analytical expressions for the maximum values of the near-end and far-end crosstalk noise on Figure 5.12:

$$\begin{aligned} V_{21,Heydari-Pedram} &= V_{21,Devgan} \left(1 - \exp \left(-\frac{t_r}{\tau_1} \right) \right) \\ &= (2R_2 + 2R_{s2})C_c \frac{V_{DD}}{t_r} \left(1 - \exp \left(-\frac{t_r}{\tau_1} \right) \right) \end{aligned} \quad (5.36)$$

where $\tau_1 = 1.07[(R_1 + R_{s1})(2C_c + C_1) + (R_2 + R_{s2})(C_c + C_2)]$

$$\begin{aligned}
V_{22,Heydari-Pedram} &= V_{21,Devgan} \left(1 - \exp \left(-\frac{t_r}{\tau_2} \right) \right) \\
&= (2R_2 + 2R_{s2})C_c \frac{V_{DD}}{t_r} \left(1 - \exp \left(-\frac{t_r}{\tau_2} \right) \right)
\end{aligned} \tag{5.37}$$

where $\tau_2 = 1.07[(2R_1 + R_{s1})(C_c + C_1) + (2R_2 + R_{s2})C_2 + (R_1 + R_2)C_c]$.

Given the equivalent time constant of noise (τ_2), and the values of the parasitics, the crosstalk noise waveform at the far-end of the victim line can be calculated as follows

$$V_{22}(t) = \begin{cases} (2R_2 + 2R_{s2})C_c \frac{V_{DD}}{t_r} \left(1 - \exp \left(-\frac{t}{\tau_2} \right) \right), & 0 \leq t \leq t_r \\ (2R_2 + 2R_{s2})C_c \frac{V_{DD}}{t_r} \left(1 - \exp \left(-\frac{t_r}{\tau_2} \right) \right) \exp \left(-\frac{t-t_r}{\tau_2} \right), & t \geq t_r \end{cases} \tag{5.38}$$

For the cases, where the input signal has smaller rise time, maximum value of the noise becomes large and accordingly noise waveform rolls down quickly. Maximum crosstalk value is inversely proportional to the input signal rise time. When the rise time is large in proportion to the delays, Devgan's metric can estimate the maximum crosstalk value as accurate as Heydari and Pedram's metric. Figure 5.16 shows the relation between maximum crosstalk noise on the far end of victim line and input rise time in Devgan's method, Heydari and Pedram's method, and in MATLAB Simulink simulations. It can be seen that, Heydari and Pedram's method performs nearly well as MATLAB Simulink and HSPICE.

Apart from Devgan [4], and Heydari and Pedram [3,11], Vittal et al. presented their exclusive crosstalk estimation metrics in [5,6]. In their studies, Vittal et al. consider the circuit given in Figure 5.17 which is a variant of Figure 5.12. According to their method presented in [5] and [6], resulting crosstalk peak value can be computed by the following equations

$$V_{22,Vittal'97} = \frac{2R_2C_c}{R_1(C_1 + C_3 + 2C_c) + R_2(C_2 + C_4 + 2C_c)} \tag{5.39}$$

$$V_{22,Vittal'99} = \frac{(2R_2 + R)C_c}{R_{den}} \tag{5.40}$$

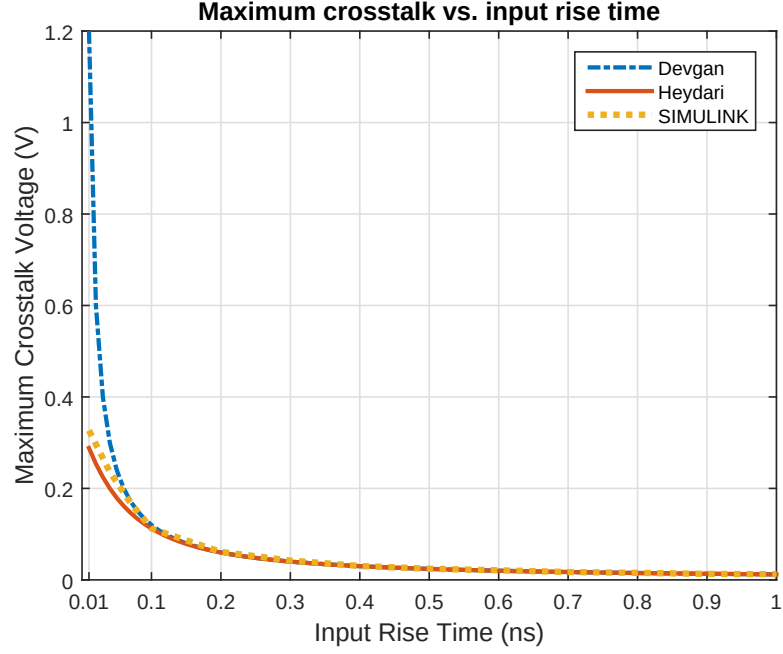


Figure 5.16: Relation between maximum crosstalk noise on the far end of victim line and input signal rise time. $R_1 = R_2 = 100 \Omega$, $C_1 = 50 \text{ fF}$, $C_2 = 60 \text{ fF}$, $C_c = 30 \text{ fF}$, $V_{DD} = 2 \text{ V}$

where

$$R_{den} = 2R_1(C_1 + C_c - 3 + 2C_c) + R_2(C_2 + C_4 + 2C_c) + R(C_3 + C_4 + 2C_c) - \left(\frac{R_2 \times R/2}{R_2 + R/2} \right) (C_2 + C_4 + 2C_c) \quad (5.41)$$

To be able to compare all the methodologies, we make the following assumptions: $C_3 = C_1$, $C_4 = C_2$, $R = R_1$. Therefore, Eq. 5.39 and Eq. 5.40 reduce to

$$V_{22,Vittal'97} = \frac{R_2 C_c}{R_1(C_1 + C_c) + R_2(C_2 + C_c)} \quad (5.42)$$

$$V_{22,Vittal'99} = \frac{(2R_2 + R_1)C_c}{2R_1(C_1 + C_c) + 2R_2(C_2 + C_c) + R_1(C_1 + C_2) - 2\left(\frac{R_2 \times R_1/2}{R_2 + R_1/2}\right)(C_2 + C_c)} \quad (5.43)$$

In order to measure the accuracy levels of the methods proposed in [3–6], number of experiments are performed. Simulation results on the two-line structure in 130-nm CMOS technology using these methods and MATLAB

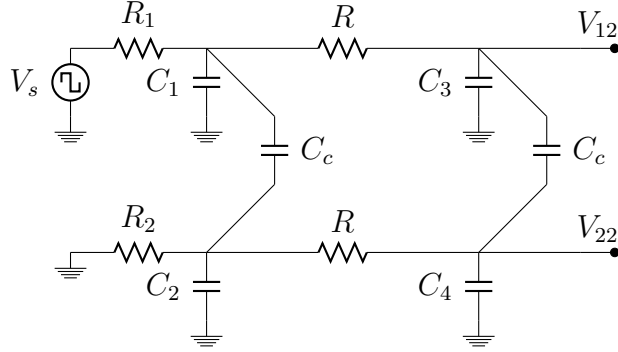


Figure 5.17: Schematic representation of capacitively coupled two segmented aggressor and victim lines.

Simulink are reported in Table 5.2, and Figure 5.18. Aggressor and victim line driver strengths are greatly different in these experiments. The results testify that Heydari and Pedram's method [3] shows higher accuracies compared to other methods [4–6].

Table 5.2: Comparison of the crosstalk noise at the victim far-end of two capacitively coupled lines computed by MATLAB Simulink and the methods [3–6] in Volts. $R_{s1} = R_{s2} = 0$ and $V_{DD} = 2V$.

R_1	R_2	C_1	C_2	C_c	t_r	Simulink	Devgan	Vittal 1997	Vittal 1999	Heydari
100	100	5.00E-14	6.00E-14	3.00E-14	5.00E-11	0.2491	0.24	0.1765	0.3176	0.1793
100	100	6.00E-14	6.00E-14	3.00E-14	4.00E-10	0.045	0.03	0.1667	0.3	0.03
50	300	7.00E-14	7.00E-14	5.00E-14	1.00E-10	0.5394	0.6	0.3571	0.4142	0.4376
70	80	7.00E-14	6.00E-14	5.00E-14	3.00E-10	0.08	0.0533	0.2326	0.3993	0.0533
70	80	1.00E-13	1.20E-13	9.00E-14	3.00E-10	0.1427	0.096	0.2392	0.4102	0.0952
40	80	1.00E-13	1.20E-13	6.00E-14	1.00E-10	0.2517	0.192	0.2308	0.3319	0.1743
40	100	1.00E-13	1.20E-13	6.00E-14	9.00E-11	0.3121	0.2667	0.2459	0.3338	0.2252
30	70	1.00E-13	1.20E-13	1.00E-13	8.00E-11	0.4111	0.35	0.3271	0.452	0.299
100	30	1.20E-13	7.00E-14	1.00E-13	8.00E-11	0.1543	0.15	0.1107	0.3412	0.1058
20	100	6.00E-14	1.20E-13	1.00E-13	8.00E-11	0.5317	0.5	0.3968	0.4686	0.4143
90	200	8.00E-14	2.20E-13	1.60E-13	8.00E-11	0.5935	1.6	0.3279	0.4547	0.5498
40	60	7.00E-14	1.00E-13	1.00E-13	3.00E-11	0.53	0.8	0.3191	0.4992	0.436
20	20	2.35E-13	2.20E-13	2.00E-13	8.00E-11	0.4187	0.65	0.2604	0.4335	0.3896
20	20.4	2.35E-13	2.20E-13	1.40E-13	8.00E-11	0.3742	0.4536	0.2121	0.3532	0.2869

Now, we will scrutinize the crosstalk noise in flexing crossbars. Figure 5.19 shows the effects of the locations of switching elements in capacitively coupled two line structures. If the switches at the both end of the lines placed closer to each other, resulting crosstalk noise will be much more than the case in which the switches placed far from each other.

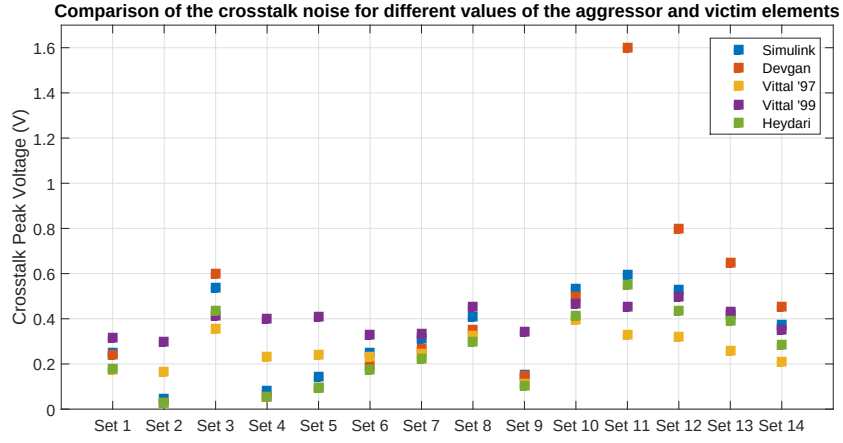


Figure 5.18: Comparison of the crosstalk noise computed by MATLAB Simulink and the methods proposed in [3–6].

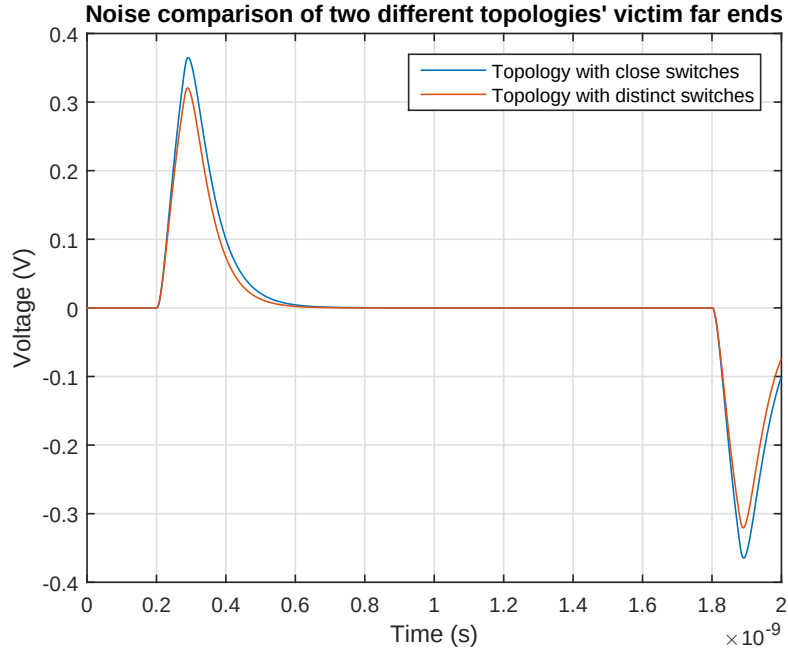


Figure 5.19: Crosstalk noise comparison of two different topologies' victim far-ends.

Based on the above review and analysis, we suggest a geometric approach in which the crosspoints are placed distantly from each other. This approach alleviates the crosstalk noise problem. As the distance between the crosspoints

increases, the crosstalk noise on victim lines decreases. To verify this conclusion, crosstalk noise analyses for three 4×4 flexing crossbars with different locations of crosspoints are performed separately, and the obtained results are compared with each other. Figure 5.20, Figure 5.23, and Figure 5.26 depict these flexing crossbar models. Figure 5.21, Figure 5.24, and Figure 5.27 show the input (V_{input}) and the output signal waveforms. In all of the experiments, 1 V square wave signal of 1 GHz frequency is applied to the input terminal x_0 in which the line is called aggressor. 1st, 2nd, and 3rd adjacent lines represent the victims. Crosstalk noise waveforms at the far ends of the victim lines are shown in more detail in Figure 5.22, Figure 5.25, and Figure 5.28.

Flexing crossbars can be described by a binary matrix called incidence matrix ($\mathcal{I}$) of a graph. Incidence matrices of graphs consist of both 0 and 1 entries where 1 indicates that there is a crosspoint between corresponding input-output pairs [1]. Incidence matrices provide precise indexing of the crosspoint positions. The incidence matrices of the compared $n \times r$ flexing crossbar models are given in Eq. 5.44, Eq. 5.46, and Eq. 5.48. In the representations, rows $kn+i$, $0 \leq k \leq u-1$ represent the copies of inputs i , $0 \leq i \leq n-1$. In addition, non-indexed entries are all set to 0, each input is replicated u times, and outputs are not replicated.

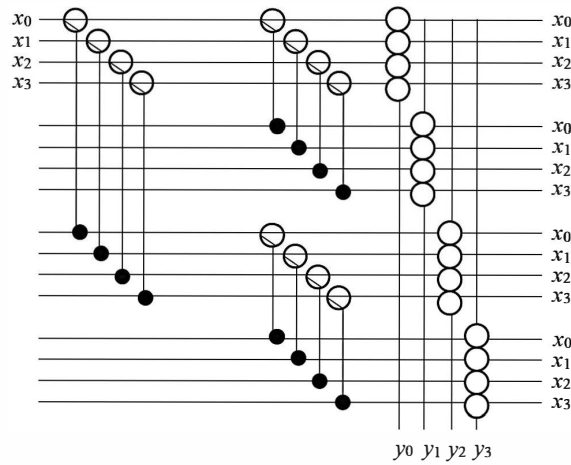


Figure 5.20: (4,4) flexing crossbar model 1.

$$\mathcal{I}_{Model\ 1} \left[kn + i, \frac{kr}{u} + j \right]_{nu,r} = 1 \quad (5.44)$$

where $0 \leq i \leq (n - 1)$, $0 \leq j \leq (\frac{r}{u} - 1)$ and $0 \leq k \leq (u - 1)$. If we consider a 4×4 flexing crossbar, the above expression is simplified to

$$\mathcal{I}_{Model\ 1}[4k + i, k]_{4 \times 4, 4} = 1 \quad (5.45)$$

where $0 \leq k \leq 3$ and $0 \leq i \leq 3$.

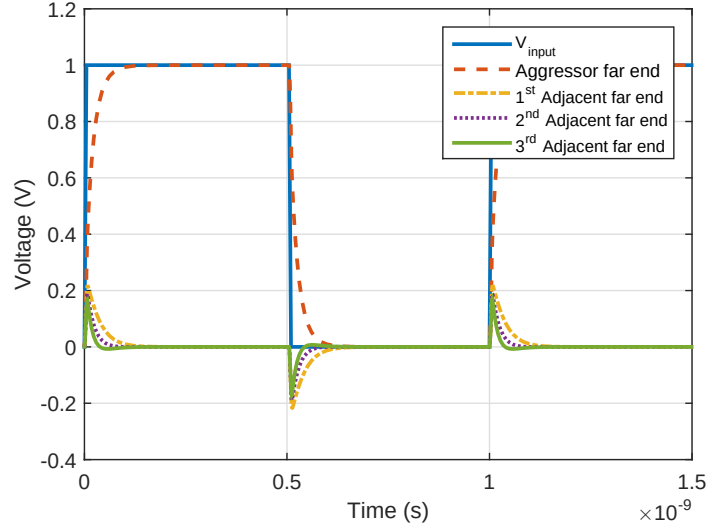


Figure 5.21: Crosstalk noise analysis of flexing crossbar model 1.

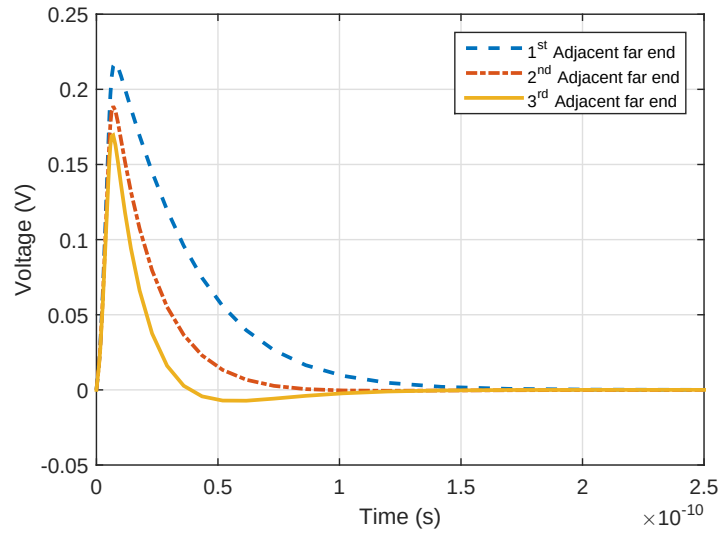


Figure 5.22: Crosstalk noise waveforms at the far end of victim lines of model 1.

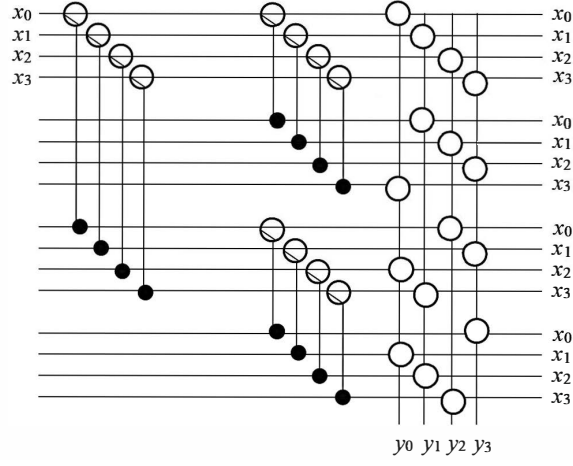


Figure 5.23: (4,4) flexing crossbar model 2.

$$\mathcal{I}_{Model\ 2} \left[kn + i, \ k + i - \left\lfloor \frac{k+i}{u} \right\rfloor \times r \right]_{nu,r} = 1 \quad (5.46)$$

where $0 \leq i \leq (n - 1)$, $0 \leq j \leq \left(\frac{r}{u} - 1\right)$ and $0 \leq k \leq (u - 1)$. If we consider a 4×4 flexing crossbar, the above expression is simplified to

$$\mathcal{I}_{Model\ 2} \left[4k + i, \ k + i - \left\lfloor \frac{k+i}{4} \right\rfloor \times 4 \right]_{4 \times 4,4} = 1 \quad (5.47)$$

where $0 \leq k \leq 3$ and $0 \leq i \leq 3$.

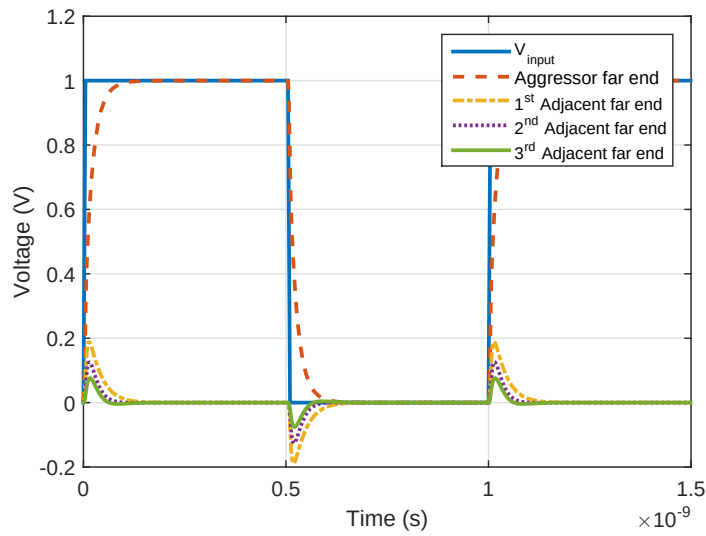


Figure 5.24: Crosstalk noise analysis of flexing crossbar model 2.

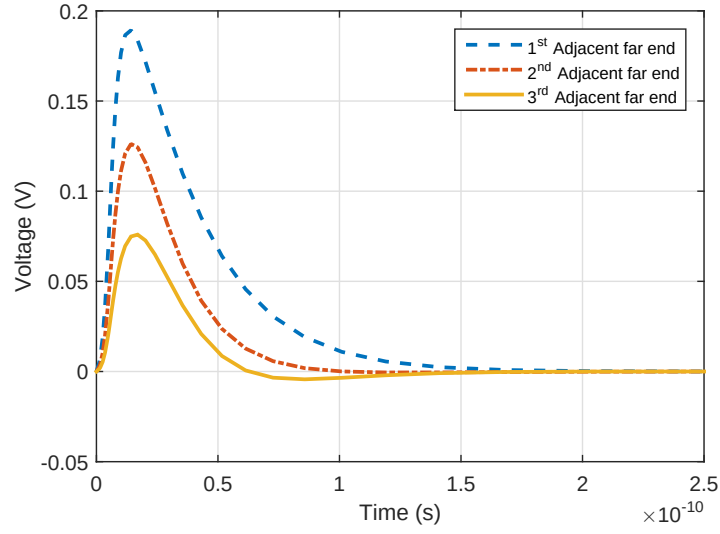


Figure 5.25: Crosstalk noise waveforms at the far end of victim lines of model 2.

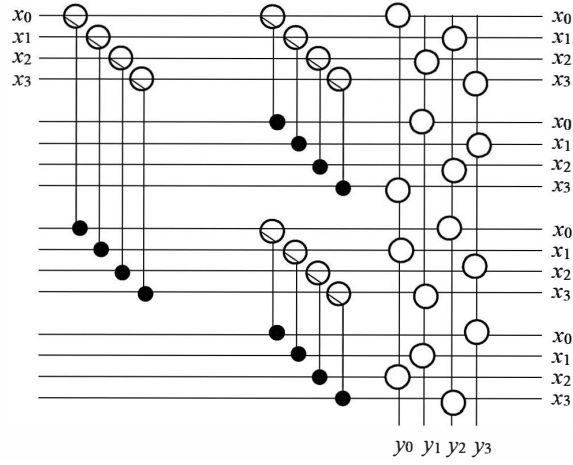


Figure 5.26: (4,4) flexing crossbar model 3.

$$\mathcal{I}_{Model\ 3} \left[kn + i, \ k + 2i + \lfloor \frac{i}{2} \rfloor - \left\lfloor \frac{k + 2i + \lfloor \frac{i}{2} \rfloor}{u} \right\rfloor \times r \right]_{nu,r} = 1 \quad (5.48)$$

where $0 \leq i \leq (n - 1)$, $0 \leq j \leq (\frac{r}{u} - 1)$ and $0 \leq k \leq (u - 1)$. If we consider a

4×4 flexing crossbar, the above expression is simplified to

$$\mathcal{I}_{Model\ 3} \left[4k + i, \ k + 2i + \lfloor \frac{i}{2} \rfloor - \left\lfloor \frac{k + 2i + \lfloor \frac{i}{2} \rfloor}{4} \right\rfloor \times 4 \right]_{4 \times 4, 4} = 1 \quad (5.49)$$

where $0 \leq k \leq 3$ and $0 \leq i \leq 3$.

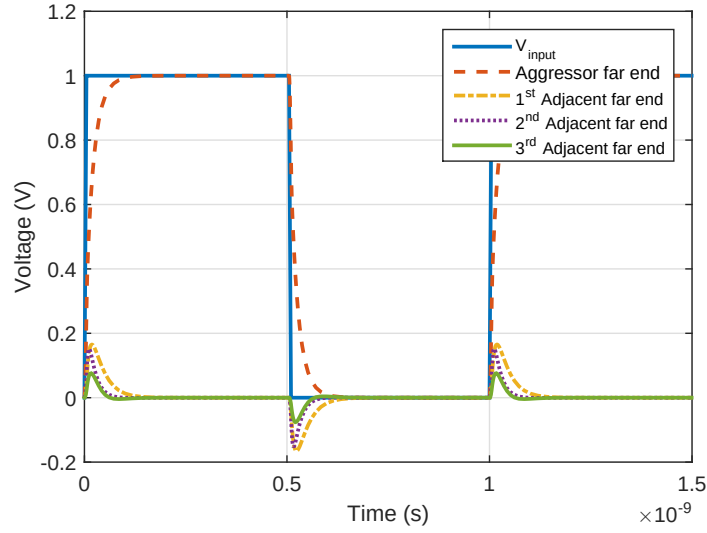


Figure 5.27: Crosstalk noise analysis of flexing crossbar model 3.

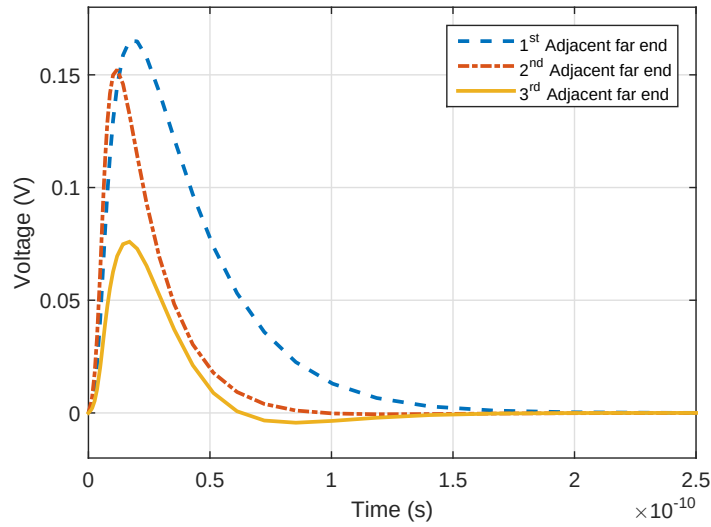


Figure 5.28: Crosstalk noise waveforms at the far end of victim lines of model 3.

Representations of the incidence matrices of the generated flexing crossbars are constructed as follows

$$\mathcal{I}_{Model\ 1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathcal{I}_{Model\ 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathcal{I}_{Model\ 3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Table 5.3: Crosspoint distances on the compared flexing crossbars. d denotes the unit distance.

Distance from the aggressor far end to	Model 1	Model 2	Model 3
1 st adjacent far end	d	$\sqrt{2}d$	$\sqrt{5}d$
2 nd adjacent far end	$2d$	$\sqrt{8}d$	$\sqrt{5}d$
3 rd adjacent far end	$3d$	$\sqrt{18}d$	$\sqrt{18}d$

It is shown that in all experiments, crosstalk noise levels are obtained within the permissible limits. According to the simulation results the maximum crosstalk noise value is $0.22\ V$. The experiments are repeated by using analytical methods. Using Vittal and Marek-Sadowska's method [5] the peak crosstalk noise value is computed as $0.3326\ V$, and using Heydari and Pedram's method [3] the peak

crosstalk value is found as 0.2336 V. Analytical expressions estimate the peak crosstalk noise value successfully.

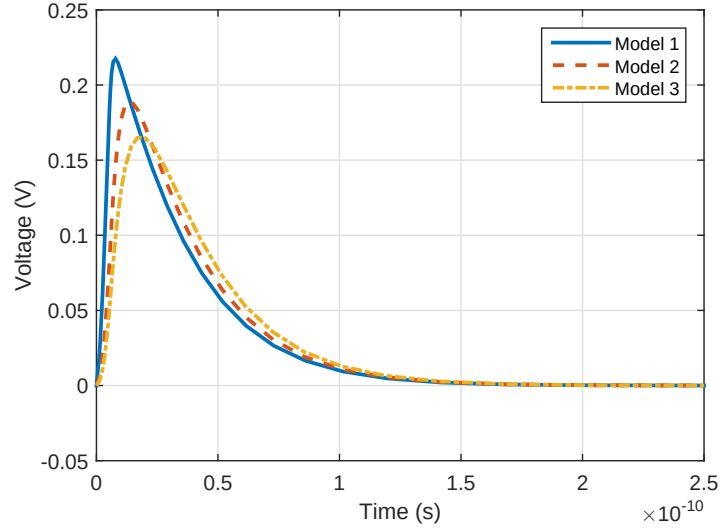


Figure 5.29: Crosstalk noise comparison at the far end of 1st adjacent wires. The architecture with close crosspoints (Model 1) has the maximum crosstalk noise.

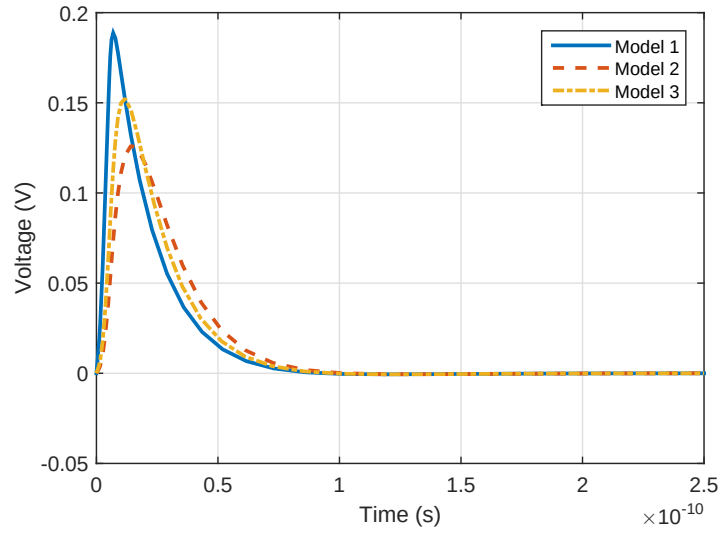


Figure 5.30: Crosstalk noise comparison at the far end of 2nd adjacent wires.

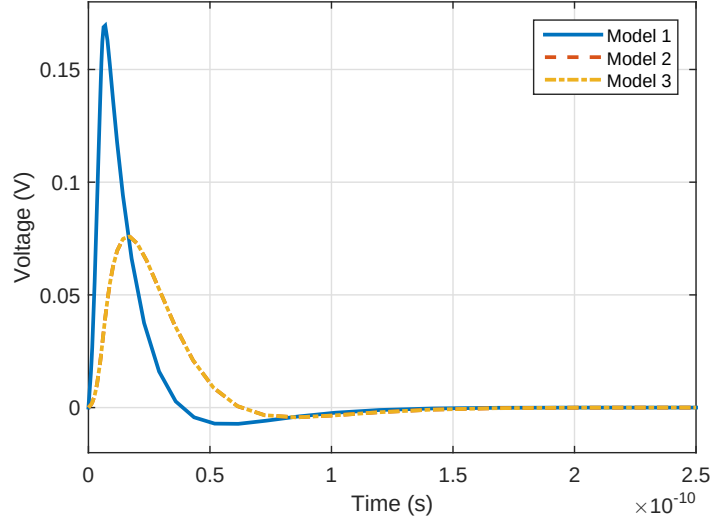


Figure 5.31: Crosstalk noise comparison at the far end of 3rd adjacent wires.

A similar analysis for crossbar switches given in Figure 3.7 has also been carried out and the results are shown in Figure 5.32. The crosstalk noise at the far end of the 1st adjacent wire is computed as 0.3078 V whereas the maximum crosstalk noise in flexing crossbars is 0.22 V. These results indicate that flexing crossbars exhibit smaller crosstalk noise as compared to crossbar switches.

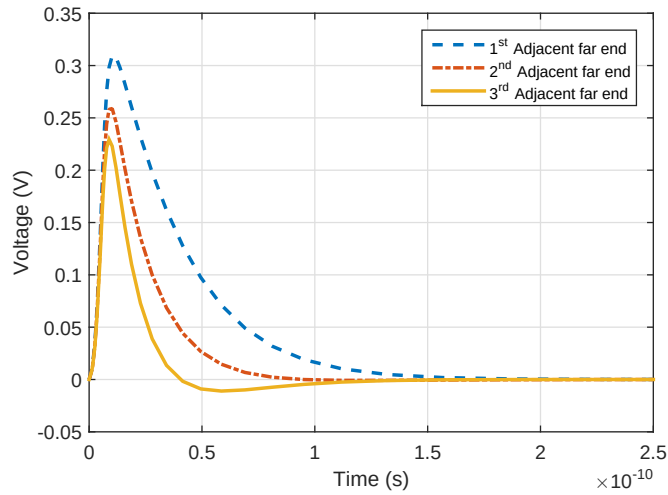


Figure 5.32: Crosstalk noise waveforms at the far end of victim lines of a conventional crossbar switch.

Increasing the coupled length and aggressor line length increases the crosstalk voltage. Similarly, increasing the victim length, or the load capacitance decrease the crosstalk peak voltage. In our analysis, we mainly investigated how the crosspoints should be placed to minimize power consumption and cross-coupling. Crosspoint distances of the compared models are given in Table 5.3. Figure 5.29, Figure 5.30, and Figure 5.31 verify our conclusion, i.e., flexing crossbar architectures with close crosspoints are more affected by crosstalk noise than the architectures with distinct crosspoints. Therefore, the crosspoints should be placed distinctly to minimize cross-coupling, and we should consider this fact as a design rule.

Chapter 6

Conclusion

Conventional crossbar switches do not restrict fan-out of inputs or fan-in of outputs. This makes an $n \times r$ crossbar infeasible as n and r become large. To alleviate this problem the binary tree and crossbar models are combined together. Resulting network is called flexing crossbar. The only drawback of an $n \times r$ flexing crossbar is the fact that the area of the crossbar switch located in the middle grows proportionally to n^2 and r^2 . In this study, we analyze delay and signal quality for varied lengths of interconnect wires on interconnection networks using lossy transmission line theory. This analysis shows that signal quality does not diminish despite longer wire lengths of flexing crossbars, i.e., up to 5 GHz operating frequency even long interconnect wires over 5 cm length can operate without facing any signal integrity issue. Hence, flexing crossbars can perform as well as the traditional ones without facing any deterioration.

Demanding performance requirements leads to extensive use of dynamic circuit techniques that can considerably reduce area and delay, and increase speed for CMOS integrated circuits [12]. In very large integrated circuits, major challenges include layout delays, high power dissipation at high frequencies of operation, increased interconnect delays, and crosstalk noise. It has been shown that signal integrity in interconnects determine the performance of overall circuit. It is important to predict signal degradation like propagation delay, delay variation,

voltage peaks, crosstalk noise, signal overshoot, ringing and attenuation in early design cycles as these can critically affect system response.

Analysis and reduction of noise becomes critical for high-speed VLSI circuits with the continuous increase in the operating frequencies and technology scaling [3, 4, 11]. In the presence of reduced power supply voltages to sustain drive strength in deep submicron circuits; threshold voltages are also reduced, resulting in lower noise margins. Among the various sources of noise, crosstalk due to the capacitive coupling effects is the dominant source of noise in current CMOS digital integrated circuits.

Contemporary high-speed CMOS technologies accommodate much more metal layers with increased density and reduced spacing between interconnects lead to significant increase in capacitive coupling effects that deteriorates the signal integrity. The severe adverse effects of coupling noise impose timing problems that can bring delay and can cause a circuit malfunction.

In this thesis, we investigated the effects of crosstalk noise on flexing crossbars and what precautions can be taken or how flexing crossbars can be designed to alleviate the adverse effects of noise. We introduce an efficient method for the estimation. The estimation method is also applicable to other submicron VLSI circuits. Lumped circuit theory is utilized to estimate crosstalk noise due to coupling effects and means of crosstalk reduction are investigated. Peak crosstalk noise amplitude, occurrence time, and time domain waveform have been represented in closed form expressions. We also provide an empirical approach to compute the best case victim-aggressor alignment that minimizes the crosstalk noise on victim lines.

Crosstalk reduction is a critical step in modern day chip design, and structure optimization is a key component in this process. According to empirical results of this study, the following facts should be taken into consideration to reduce the crosstalk noise from a design methodology viewpoint.

Increasing the lateral spacing between the adjacent lines reduces the crosstalk

noise by decreasing the cross coupling. However, it needs larger layout area in chip. Similarly, placing aggressor and victim lines perpendicular to each other decreases the cross-coupling effects. As the wire resistances reduce, victim line capacitances increase. Inserting buffers or repeaters into wires divide them into smaller portions, and reduces the wire resistances. This can also be accomplished by using the wider interconnect wires. Peak crosstalk value can be reduced by ensuring proper rise times for the signals. Slowing down the signal transitions on aggressor lines gives more time for victim lines to flow current. Finally, placing the crosspoints away from each other reduces the crosstalk noise by minimizing the cross-coupling.

Bibliography

- [1] A. Y. Oruc, *Foundations of Interconnection Networks*. CRC Press, 2015.
- [2] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, 2003.
- [3] P. Heydari and M. Pedram, “Analysis and Reduction of Capacitive Coupling Noise in High-Speed VLSI Circuits,” in *Computer Design, 2001. ICCD 2001. Proceedings. 2001 International Conference on*, pp. 104–109, IEEE, 2001.
- [4] A. Devgan, “Efficient coupled noise estimation for on-chip interconnects,” in *Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, pp. 147–151, IEEE Computer Society, 1997.
- [5] A. Vittal and M. Marek-Sadowska, “Crosstalk reduction for VLSI,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 290–298, 1997.
- [6] A. Vittal, L. H. Chen, M. Marek-Sadowska, K.-P. Wang, and S. Yang, “Crosstalk in VLSI interconnections,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 18, no. 12, pp. 1817–1824, 1999.
- [7] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, 4th ed., 2010.

- [8] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, “Digital integrated circuits—a design perspective,” 2003.
- [9] R. J. Baker, *CMOS Circuit Design, Layout, and Simulation*. 2010.
- [10] T. Sakurai, “Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSIs,” *Electron Devices, IEEE Transactions on*, vol. 40, no. 1, pp. 118–124, 1993.
- [11] P. Heydari and M. Pedram, “Capacitive coupling noise in high-speed VLSI circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 478–488, 2005.
- [12] P. Larsson and C. Svensson, “Noise in digital dynamic cmos circuits,” *Solid-State Circuits, IEEE Journal of*, vol. 29, no. 6, pp. 655–662, 1994.
- [13] M. Kuhlmann and S. S. Sapatnekar, “Exact and efficient crosstalk estimation,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 7, pp. 858–866, 2001.
- [14] K. J. Thurber, “Interconnection networks: a survey and assessment,” in *Proceedings of the May 6-10, 1974, national computer conference and exposition*, pp. 909–919, ACM, 1974.
- [15] G. M. Masson, G. C. Gingher, and S. Nakamura, “A sampler of circuit switching networks,” *Computer*, vol. 12, no. 6, pp. 32–48, 1979.
- [16] T. Y. Feng, “A survey of interconnection networks,” *Computer*, vol. 14, pp. 12–27, Dec. 1981.
- [17] A. Y. Oruc, “Multiple tracks of research on interconnection networks,” in *Proceedings of the 1995 ICPP Workshop on Challenges for Parallel Processing*, pp. 16–23, 1995.
- [18] A. Y. Oruc, “One-sided binary tree crossbar switching for on-chip networks,” in *Conference on Information Sciences and Systems , 49th Annual*

Conference on, Johns Hopkins University Whiting School of Engineering
Department of Electrical and Computer Engineering, 2015.

- [19] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, “A network on chip architecture and design methodology,” in *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*, pp. 105–112, IEEE, 2002.
- [20] J. Kim, W. J. Dally, and D. Abts, “Flattened butterfly: a cost-efficient topology for high-radix networks,” *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 126–137, 2007.
- [21] Y.-H. Kao and H. J. Chao, “Design of a bufferless photonic clos network-on-chip architecture,” *Computers, IEEE Transactions on*, vol. 63, no. 3, pp. 764–776, 2014.
- [22] A. Y. Oruc, “Flexing crossbars: Next generation packet switches,” April 2014.
- [23] M. a. Elgamel and M. a. Bayoumi, “Analysis and Optimization in Deep Submicron Technology,” *Ieee Circuits and Systems Magazine*, 2003.
- [24] C. Clos, “A study of nonblocking switching networks,” *Bell System Technical Journal*, vol. 32, pp. 406–424, 1953.
- [25] B. Beizer, “The analysis and synthesis of signal switching networks,” in *Proceedings of the Symposium on Mathematical Theory of Automata*, pp. 563–576, 1962.
- [26] V. E. Beneš, “Heuristic Remarks and Mathematical Problems Regarding the Theory of Connecting Systems,” *Bell System Technical Journal*, vol. 41, no. 4, pp. 1201–1247, 1962.
- [27] M. Paull, “Reswitching of connection networks,” *Bell System Technical Journal*, vol. 41, no. 3, pp. 833–855, 1962.

- [28] V. E. Beneš, “Optimal rearrangeable multistage connecting networks,” *Bell System Technical Journal*, vol. 43, no. 4, pp. 1641–1656, 1964.
- [29] V. E. Beneš, *Mathematical theory of connecting networks and telephone traffic*, vol. 17. Academic press New York, 1965.
- [30] A. E. Joel, “On permutation switching networks,” *Bell System Technical Journal*, vol. 47, no. 5, pp. 813–822, 1968.
- [31] D. C. Opferman and N. T. Tsao-Wu, “On a class of rearrangeable switching networks part i: Control algorithm,” *Bell System Technical Journal*, vol. 50, no. 5, pp. 1579–1600, 1971.
- [32] L. A. Bassalygo and M. S. Pinsker, “Complexity of an optimum nonblocking switching network without reconnections,” *Problemy Peredachi Informatsii*, vol. 9, no. 1, pp. 84–87, 1973.
- [33] D. G. Cantor, “On non-blocking switching networks,” *Networks*, vol. 1, no. 4, pp. 367–377, 1971.
- [34] F. Chung, “On Concentrators, Superconcentrators, Generalizers, and Nonblocking Networks,” *Bell System Technical Journal*, vol. 58, no. 8, pp. 1765–1777, 1978.
- [35] L. A. Bassalygo, “Asymptotically optimal switching circuits,” *Problemy Peredachi Informatsii*, vol. 17, no. 3, pp. 81–88, 1981.
- [36] M. S. Pinsker, “On the complexity of a concentrator,” in *7th International Teletraffic Conference*, vol. 4, pp. 1–318, 1973.
- [37] D. H. Lawrie, “Memory-processor connection networks,” 1973.
- [38] D. H. Lawrie, “Access and alignment of data in an array processor,” *Computers, IEEE Transactions on*, vol. 100, no. 12, pp. 1145–1155, 1975.
- [39] T. Lang, “Interconnections between processors and memory modules using

- the shuffle-exchange network,” *Computers, IEEE Transactions on*, vol. 100, no. 5, pp. 496–503, 1976.
- [40] A. Bouhraoua and M. Elrabaa, “A high-throughput network-on-chip architecture for systems-on-chip interconnect,” in *System-on-Chip, 2006. International Symposium on*, pp. 1–4, IEEE, 2006.
 - [41] C.-L. Wu and T.-Y. Feng, “The reverse-exchange interconnection network,” *Computers, IEEE Transactions on*, vol. 100, no. 9, pp. 801–811, 1980.
 - [42] C.-L. Wu and T.-Y. Feng, “The universality of the shuffle-exchange network,” *Computers, IEEE Transactions on*, vol. 100, no. 5, pp. 324–332, 1981.
 - [43] P. Feldman, J. Friedman, and N. Pippenger, “Wide-sense nonblocking networks,” *SIAM Journal on Discrete Mathematics*, vol. 1, no. 2, pp. 158–173, 1988.
 - [44] Y. Yang and G. M. Masson, “Nonblocking broadcast switching networks,” *Computers, IEEE Transactions on*, vol. 40, no. 9, pp. 1005–1015, 1991.
 - [45] G. Lin and N. Pippenger, “Parallel algorithms for routing in nonblocking networks,” *Mathematical systems theory*, vol. 27, no. 1, pp. 29–40, 1994.
 - [46] P. M. Y.-J. Tsai and D. Robinson, “A survey of collective communication in wormhole-routed massively parallel computers,” tech. rep., Tech. Report MSU-CPS-94-35, Dept. of Computer Science, Michigan State Univ., East Lansing, Mich, 1994.
 - [47] Y. Yang, “A class of interconnection networks for multicasting,” *Computers, IEEE Transactions on*, vol. 47, no. 8, pp. 899–906, 1998.
 - [48] S. Arora, T. Leighton, and B. Maggs, “On-line algorithms for path selection in a nonblocking network,” in *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pp. 149–158, ACM, 1990.

- [49] A. Shacham, K. Bergman, and L. P. Carloni, “Photonic networks-on-chip for future generations of chip multiprocessors,” *Computers, IEEE Transactions on*, vol. 57, no. 9, pp. 1246–1260, 2008.
- [50] W. J. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” in *Design Automation Conference, 2001. Proceedings*, pp. 684–689, IEEE, 2001.
- [51] Y.-H. Kao, N. Alfaraj, M. Yang, and H. J. Chao, “Design of high-radix clos network-on-chip,” in *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pp. 181–188, IEEE, 2010.
- [52] Z. Wang, J. Xu, X. Wu, Y. Ye, W. Zhang, M. Nikdast, and X. Wang, “Floorplan optimization of fat-tree based networks-on-chip for chip multiprocessors,” 2012.
- [53] H. Kawaguchi and T. Sakurai, “Delay and noise formulas for capacitively coupled distributed rc lines,” in *Design Automation Conference 1998. Proceedings of the ASP-DAC’98. Asia and South Pacific*, pp. 35–43, IEEE, 1998.
- [54] J. Cong, D. Z. Pan, and P. V. Srinivas, “Improved crosstalk modeling for noise constrained interconnect optimization,” in *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*, pp. 373–378, ACM, 2001.
- [55] L. Pillage and R. Rohrer, “Asymptotic waveform evaluation for timing analysis,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 9, pp. 352–366, Apr 1990.
- [56] P. Feldmann and R. Freund, “Efficient linear circuit analysis by pade approximation via the lanczos process,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 14, pp. 639–649, May 1995.
- [57] K. J. Kerns and A. T. Yang, “Stable and efficient reduction of large,

- multiport rc networks by pole analysis via congruence transformations,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 16, no. 7, pp. 734–744, 1997.
- [58] K. Aingaran, F. Klass, C.-M. K. C.-M. Kim, C. Amir, J. Mitral, E. You, J. Mohd, and S.-K. D. S.-K. Dong, “Coupling noise analysis for VLSI and ULSI circuits,” *Proceedings IEEE 2000 First International Symposium on Quality Electronic Design (Cat. No. PR00525)*, 2000.
- [59] J. Z. J. Zhang and E. Friedman, “Crosstalk noise model for shielded interconnects in VLSI-based circuits,” *IEEE International [Systems-on-Chip] SOC Conference, 2003. Proceedings.*, pp. 243–244, 2003.
- [60] W. Gong, W. Yu, Y. Lu, Q. Tang, Q. Zhou, and Y. Cai, “A parasitic extraction method of VLSI interconnects for pre-route timing analysis,” in *Communications, Circuits and Systems (ICCCAS), 2010 International Conference on*, pp. 871–875, IEEE, 2010.
- [61] M. Takahashi, M. Hashimoto, and H. Onodera, “Crosstalk noise estimation for generic RC trees,” in *Computer Design, 2001. ICCD 2001. Proceedings. 2001 International Conference on*, pp. 110–116, IEEE, 2001.
- [62] M. R. Zargham, *Computer Architecture: Single and Parallel Systems*. Prentice-Hall, Inc., 1996.
- [63] S. B. Dhia, M. Ramdani, and E. Sicard, *Electromagnetic Compatibility of Integrated Circuits: Techniques for low emission and susceptibility*. Springer Science & Business Media, 2006.
- [64] D. M. Pozar, *Microwave engineering*. John Wiley & Sons, 4th ed., 2012.
- [65] P. L. Peres, C. R. De Souza, and I. S. Bonatti, “ABCD matrix: a unique tool for linear two-wire transmission line modelling,” *International Journal of Electrical Engineering Education*, vol. 40, no. 3, pp. 220–229, 2003.

- [66] A. C. Carusone, “Front-end circuits for multi-gb/s chip-to-chip links,” in *Circuits at the nanoscale: communications, imaging, and sensing* (K. Iniewski, ed.), CRC press, 2008.
- [67] K. Yano and S. Muroga, “Pass transistors,” in *The VLSI handbook* (W. K. Chen, ed.), CRC press, 2006.
- [68] C. D. Thompson, *A Complexity Theory for VLSI*. PhD thesis, Carnegie-Mellon University, 1980.
- [69] T. T. Ye, G. D. Micheli, and L. Benini, “Analysis of power consumption on switch fabrics in network routers,” in *Proceedings of the 39th annual Design Automation Conference*, pp. 524–529, ACM, 2002.
- [70] G. Vasilescu, *Electronic Noise and Interfering Signals: Principles and Applications*. Springer, 2005.

Appendix A

Code

MATLAB[®] code: `transmission_line_analysis.m`

```
1  % Transmission Line Analysis
2  clc; clear all; close all;
3  % Parameters
4  w_0 = 2*pi*10e9;
5  theta_0 = 0.021;          k_r = 87;
6  f = linspace(0,50e9,1000); w = 2*pi*f;
7  c = 2.998e8;              eps_r = 4.9;
8  v0 = sqrt(1/eps_r)*c;
9  Z0 = 100;                  L0 = Z0/v0;
10 G0 = 1e-12;                C0 = 1/(Z0*v0);
11 RDC = 0.0001;              RAC = (k_r*(1+j)*sqrt(w/w_0));
12 % RLGC values
13 R = sqrt(RDC^2 + RAC.^2);
14 L = L0*ones(1,length(f));
15 G = G0*ones(1,length(f));
16 C = C0*(j*w./w_0).^(-2*theta_0/pi);
17
18 if f(1)==0
19     C(1) = C(2);
```

```

20 end
21
22 wire_length = [0.01 0.03 0.05];
23 for i=1:1:3
24     d = wire_length(i);
25     w = 2*pi*f;
26     gammad = d*sqrt((R+j*w.*L).*(G+j*w.*C));
27     z0 = sqrt((R+j*w.*L)./(G+j*w.*C));
28     z0(1) = sqrt((R(1)+j*w(1).*L(1))./(G(1)+j*w(1).*C(1)));
29     % Transmission line ABCD matrix
30     transmission_line.A = cosh(gammad);
31     transmission_line.B = z0.*sinh(gammad);
32     transmission_line.C = sinh(gammad)./z0;
33     transmission_line.D = transmission_line.A;
34
35     % Bondwire package ABCD matrix
36     % pad admittance
37     l = length(j*2*pi*f*0.5e-12);
38     pad.A = ones(1,l);
39     pad.B = zeros(1,l);
40     pad.C = j*2*pi*f*0.5e-12;
41     pad.D = ones(1,l);
42     % bondwire impedance
43     l = length(j*2*pi*f*10e-10);
44     bondwire.A = ones(1,l);
45     bondwire.B = j*2*pi*f*10e-10;
46     bondwire.C = zeros(1,l);
47     bondwire.D = ones(1,l);
48
49     package = series_connection(pad,series_connection(bondwire,pad));
50
51     % Source impedance ABCD matrix
52     l = length(120*ones(1,length(f)));

```

```

53     source.A = ones(1,1);
54     source.B = 120*ones(1,length(f));
55     source.C = zeros(1,1);
56     source.D = ones(1,1);
57
58     % Termination admittance ABCD matrix
59     l = length(ones(1,length(f))./80);
60     termination.A = ones(1,1);
61     termination.B = zeros(1,1);
62     termination.C = ones(1,length(f))./80;
63     termination.D = ones(1,1);
64
65     % ABCD matrix of the whole wire
66     % -with discontinuities
67     wire = series_connection(source,series_connection(series_connection...
68     (package,series_connection(transmission_line,package)),termination));
69     % -without discontinuities
70     wire = series_connection(source,series_connection(transmission_line,...
71     termination));
72
73     % Frequency response
74     H = 1./(wire.A);
75
76     % Step and impulse responses
77     Hd = [H conj(H(end-1:-1:2))];
78     h = real(ifft(Hd)); % impulse response
79     hstep = conv(h,ones(1,length(h)));
80     hstep = hstep(1:length(h)); % step response
81     t = linspace(0,1/f(2),length(h)+1);
82     t = t(1:end-1);
83
84     subplot(3,1,1);
85     plot(1e-9*f,20*log10(abs(H)),'linewidth',2) %%

```

```

86     set(gca,'FontSize',12);
87     hold all
88     grid on
89     xlabel('Frequency (GHz)');
90     ylabel('Gain (dB)');
91     legend('Length = 1 cm','Length = 3 cm','Length = 5 cm');
92
93     subplot(3,1,2);
94     plot(t*1e9,hstep,'linewidth',2);
95     set(gca,'xlim',[0 4],'ylim',[-.02 .42],'FontSize',12);
96     hold all
97     grid on
98     xlabel('Time (ns)');
99     ylabel('Step Response')
100    legend('Length = 1 cm','Length = 3 cm','Length = 5 cm');
101
102    subplot(3,1,3);
103    plot(t*1e9,h,'linewidth',2);
104    set(gca,'xlim',[0 4],'ylim',[-.02 .42],'FontSize',12);
105    hold all
106    grid on
107    xlabel('Time (ns)');
108    ylabel('Step Response')
109    legend('Length = 1 cm','Length = 3 cm','Length = 5 cm');
110 end

```

MATLAB[®] code: series_connection.m

```

1 function system = series_connection(system_1,system_2);
2     % Series connection of two 2-port networks
3     system.A = system_1.A.*system_2.A + system_1.B.*system_2.C;
4     system.B = system_1.A.*system_2.B + system_1.B.*system_2.D;
5     system.C = system_1.C.*system_2.A + system_1.D.*system_2.C;
6     system.D = system_1.C.*system_2.B + system_1.D.*system_2.D;

```

7 end

MATLAB[®] code: power_consumption_comparison.m

```
1 % Power Consumption Analysis
2 for k = 1:1:4;
3     Es_crossbar = 220e-15;
4     Es_flexing = 220e-15;
5     Es_baseline = 1821e-15;
6     Ew = 87e-15;
7     switch k
8         case 1
9             N = 4;
10            Eb_baseline = 140e-15;
11            baseline_w_length = 6;
12            Em_fully = 431e-15;
13        case 2
14            N = 8;
15            Eb_baseline = 140e-15;
16            baseline_w_length = 30;
17            Em_fully = 782e-15;
18        case 3
19            N = 16;
20            Eb_baseline = 154e-15;
21            baseline_w_length = 92;
22            Em_fully = 1350e-15;
23        case 4
24            N = 32;
25            Eb_baseline = 222e-15;
26            baseline_w_length = 274;
27            Em_fully = 2515e-15;
28    end
29    E_Crossbar(k) = N*Es_crossbar + 2*(2*N)*Ew;
30    E_Flexing(k) = N*Es_flexing + (2*(N^2)+2)*Ew;
```

```

31         E_Baseline(k) = Eb_baseline + (0.5*N*log2(N))*...
32             Es_baseline + (baseline_w_length)*Ew ;
33         E_Fully(k)    = N*Em_fully + (N^2 + N)*Ew;
34     end
35
36     plot(E_Crossbar,'-s','linewidth',2)
37     xlim([0 5])
38     set(gca,'XTick',1:4)
39     set(gca,'XTickLabel',{'4x4', '8x8', '16x16', '32x32'},'fontsize',12)
40     grid on
41     hold all
42     plot(E_Flexing,'-o','linewidth',2)
43     plot(E_Baseline,'-v','linewidth',2)
44     plot(E_Fully,'-d','linewidth',2)
45     xlabel('Interconnection Network Size')
46     legend('Crossbar','Flexing Crossbar','Baseline',...
47         'Fully Connected','Location','NW')
48     close all
49
50     % data rate of the networks = 480Mbit/s
51     P_Crossbar      = 480e6*E_Crossbar;
52     P_Flexing       = 480e6*E_Flexing;
53     P_Baseline      = 480e6*E_Baseline;
54     P_Fully         = 480e6*E_Fully;
55     figure();
56     plot(1e3*P_Crossbar,'-s','linewidth',2)
57     % xlim([0 5])
58     set(gca,'XTick',1:4)
59     set(gca,'XTickLabel',{'4x4', '8x8', '16x16', '32x32'},'fontsize',12)
60     grid on
61     hold all
62     plot(1e3*P_Flexing,'-v','linewidth',2)
63     plot(1e3*P_Baseline,'-d','linewidth',2)

```

```

64 plot(1e3*P_Fully,'-*','linewidth',2)
65 xlabel('Interconnection Network Size')
66 ylabel('Power Consumption (mW)')
67 legend('Crossbar','Flexing Crossbar','Baseline',...
68        'Fully Connected','Location','NW')

```

MATLAB[®] code: crosstalk_comparison.m

```

1  %% Comparison of crosstalk noise - Table 5.2 - Figure 5.18
2  clc; clear all; close all;
3  Rs1_array =[0 0 0 0 0 0 0 0 0 0 0 0 20 20];
4  Rs2_array =[0 0 0 0 0 0 0 0 0 0 0 0 30 29.6];
5  R1_array = [100 100 50 70 70 40 40 30 100 20 90 40 20 20];
6  R2_array = [100 100 300 80 80 80 100 70 30 100 200 60 20 20.4];
7  C1_array = [50 60 70 70 100 100 100 100 120 60 80 70 235 235].*10^(-15);
8  C2_array = [60 60 70 60 120 120 120 120 70 120 220 100 220 220].*10^(-15);
9  Cc_array = [30 30 50 50 90 60 60 100 100 100 160 100 200 140].*10^(-15);
10 tr_array = [0.05 0.4 0.1 0.3 0.3 0.1 0.09 0.08 0.08 0.08 0.08 0.08 0.03 0.08...
11             0.08].*10^(-9);
12 VDD=2;
13
14 for i=1:length(Cc_array)
15     Rs1= Rs1_array(i);
16     Rs2= Rs2_array(i);
17     R1 = R1_array(i);
18     R2 = R2_array(i);
19     C1 = C1_array(i);
20     C2 = C2_array(i);
21     Cc = Cc_array(i);
22     tr = tr_array(i);
23
24     % Devgan 1997
25     V21ss(i) = 2*(R2+Rs2)*Cc*(VDD/tr); % tr is rise time of input signal
26     V22ss(i) = (2*R2+3*Rs2)*Cc*(VDD/tr);

```

```

27
28     % Heydari 2005
29     % function of Devgan's metric
30     Tau_d1 = 1.07*((R1 + Rs1)*(2*Cc + C1) + (R2 + Rs2)*(Cc + C2));
31     V21max(i) = V21ss(i)*(1-exp(-tr/Tau_d1));
32     Tau_d2 = 1.07*((2*R1 + Rs1)*(Cc + C1) + (2*R2 + Rs2)*C2 + ...
33         (R1 + R2)*Cc);
34     V22max(i) = V21ss(i)*(1-exp(-tr/Tau_d2));
35
36     % Vittal 1997 & 1999
37     % There are two results that belong to Vittal
38     % Vr0 means interconnect resistance is 0 - Vittal 1997
39     % Vr, iterconnect resistance is not 0 - Vittal 1999
40     R1=Rs1+R1;
41     R2=Rs2+R2;
42     C3=C1;
43     C4=C2;
44     R=R1;
45     % R=R2;
46     X=Cc;
47     % when interconnect resistance is zero - Vittal 1997
48     Vr0(i) = (2*R2*X)/(R1*(C1 + C3 + 2*X) + R2*(C2 + C4 + 2*X));
49     % when interconnect resistance is not zero - Vittal 1999
50     Vr(i) = ((2*R2+R)*X)/(R1*(C1+C3+2*X)+R2*(C2+C4+2*X)-((R2*(R/2))/...
51         (R2+(R/2)))*(C2+C3+2*X));
52 end
53 % Comparison of crosstalk noise
54 % Devgan - Vittal - Heydari
55 simulink_res=[0.249077075338473 0.044999953320821 0.539421103886877...
56     0.079987848756963 0.142739988009513 0.251670465250145...
57     0.312079538543642 0.411051692018822 0.154254820559522...
58     0.531691894361946 0.593522109149635 0.529989396343630...
59     0.418684693503760 0.374170800506030];

```

```

60
61 result = [R1_array' R2_array' C1_array' C2_array' Cc_array' tr_array'...
62     simulink_res' V22ss' Vr0' Vr' V22max']
63 parameters = {'R1','R2','C1','C2','Cc','tr','Simulink','Devgan',...
64     'Vittal 1997','Vittal 1999','Heydari'};
65 xlswrite('+ crosstalk noise vittal-devgan-heydari',result,'Sheet1','A2')
66 xlswrite('+ crosstalk noise vittal-devgan-heydari',...
67     parameters,'Sheet1','A1')
68
69 plot(result(:,7:11),'s','linewidth',5,'MarkerSize',3)
70 grid on
71 xlim([0 15])
72 ylim([0 1.7])
73 set(gca,'XTick',1:14)
74 set(gca,'XTickLabel',{'Set 1', 'Set 2', 'Set 3', 'Set 4', 'Set 5',...
75     'Set 6', 'Set 7', 'Set 8', 'Set 9'...
76     , 'Set 10', 'Set 11', 'Set 12', 'Set 13', 'Set 14'},'fontsize',12)
77 legend('Simulink','Devgan','Vittal ''97','Vittal ''99','Heydari')
78 % xticklabel_rotate
79 ylabel('Crosstalk Peak Voltage (V)')

```

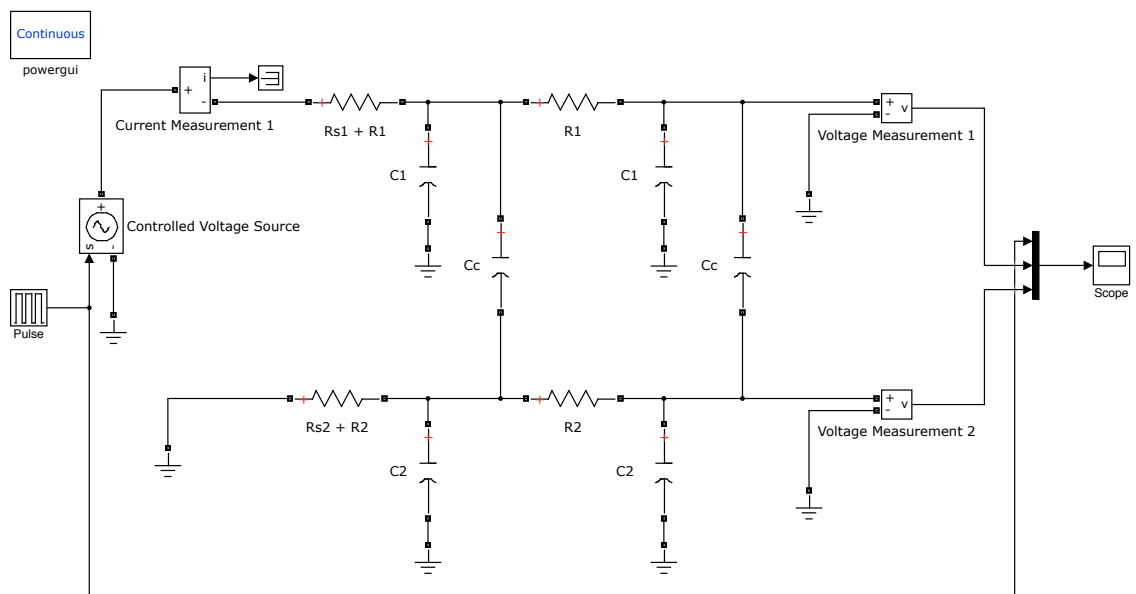


Figure A.1: Simulink model for Figure 5.12.