Delay Uncertainty in High Performance Clock Distribution Networks

by

Dimitrios Velenis

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

Supervised by
Professor Eby G. Friedman

Department of Electrical and Computer Engineering
School of Engineering and Applied Sciences
The College

University of Rochester
Rochester, New York

2003

UMI Number: 3102306

UMI®

UMI Microform 3102306

Copyright 2003 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against

unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Dedication

To my family, my parents Stamatios and Penelopi, my brother Patroklos, and my wife Andreia Marina who brings all the happiness in my life.

Curriculum vitae

The author was born in Kavala, Macedonia, Greece on July 22, 1974. He attended the Technical University of Crete, in Chania Greece, from 1993 to 1998 and graduated in July of 1998 with a Bachelor of Science degree in Electronics and Computer Engineering. He came to the University of Rochester in August of 1998, and began graduate studies at the Department of Electrical and Computer Engineering. He received his Masters of Science degree in May 2000. He performed research with Intel Corporation, Santa Clara, CA in 1999 and with IBM Corporation at Austin Research Lab, Austin, TX in 2001. From 1998 to 2003, he pursued research in very large scale integrated circuits under the direction of Professor Eby G. Friedman, in the areas of timing optimization and noise effects in high performance synchronous integrated circuits with emphasis in the design of multi-gigahertz clock distribution networks.

Acknowledgements

I would like to express my most sincere feelings of gratitude to my academic advisor, Professor Eby G. Friedman, for his enthusiastic encouragement and enlightment during all my years as a doctoral student. His inexhaustible dedication and enthusiasm have become the source of my inspiration, while his brilliant research ideas earned my deepest respect. In addition, he has been a great mentor for me, having a human interest for many aspects of my life. I wish I will be able to give to the people around me just a fraction of the motivation and support that Professor Friedman has offered me. I would like to wish to Professor Friedman to offer this opportunity to many more students in the future.

I am also deeply indebted to Professor Marios C. Papaefthymiou for his sound contribution in my research effort. His determination, clear focus, and capability to formulate complicated problems were great assets during the development of the work presented in this dissertation. My collaboration with him was a very productive, and at the same time, enjoyable experience.

I would also like to thank Professors David Albonesi, John C. Lambropoulos, and Martin Margala for serving in my committee and for their influence and suggestions regarding my work. I am also grateful to the entire faculty, staff, and students of the Department of Electrical and Computer Engineering for putting their hearts into filling everyday life with excitement and joy. Special thanks to Profes-

sor Alexander Albicki, Amy Freitas, Maureen Muar, Debra Neiner, Jim Prescott, Yana Shinkman, John Simonson, Melissa Singkhamsack, and John Strong.

In particular, I would like to recognize the contribution of the people that we shared together the whole experience. I would like to thank all the members of the High Performance VLSI/IC Design and Analysis Laboratory for the unforgettable moments we had together. It was an honor for me to work in the same team with Volkan Kursun, Andrey Mezhiba, Boris Andreev, Magdy El-Moursy, Dr. Ivan Kourtev, Dr. Tianwen (Kevin) Tang, Dr. Yehea Ismail, Dr. Radu Secareanu, Dr. Victor Adler, Weize Xu, Junmou Zhang, Mikhail Popovich, Guoqing Chen, Jonathan Rosenfeld, and Vasilis Pavlidis. Special thanks to Mrs. RuthAnn Williams for the true love and the great joy she offered to us in every single day.

I would also like to thank all my friends for all the great time we had together that became wonderful memories. It was a great pleasure to meet with Dimitrios Katsis, Christos Koulouvatianos, Grigoris Maglis, Thanos Papathanasiou, Stephanos Nitodas, Panagiotis Kambylafkas, and Manos Fitrakis.

Finally, I would like to thank my family for their unconditional love, support, and understanding. They have always been on my side, teaching me to challenge life and believe in myself. To my loving wife Andreia Marina—I am really blessed to share my life with you. Thank you for your ocean of love and support. You are my inspiration and my source of happiness, I wouldn't have made it without you.

This work was supported in part by the Semiconductor Research Corporation under Contract No. 99-TJ-687, and No. 2003-TJ-1068, the DARPA/ITO under AFRL Contract F29601-00-K-0182, grants from the New York State Office of Science, Technology & Academic Research to the Center for Advanced Technology - Electronic Imaging Systems and to the Microelectronics Design Center, and by grants from Xerox Corporation, IBM Corporation, Intel Corporation, Lucent Technologies Corporation, Eastman Kodak Company and Photon Vision Systems, Inc.

Abstract

The effect of noise on circuit operation and reliability has become an important issue in the design of high performance integrated circuits. The main effect of noise is the degradation of signal integrity causing uncertainty in the signal delay. The uncertainty of the propagation delay of a signal can cause a catastrophic violation of the timing constraints within a system.

One way to improve the tolerance of a system to delay uncertainty is by relaxing the timing constraints of the critical paths. A methodology that implements this concept by applying non-zero clock skew scheduling is described. Furthermore, a variation of this methodology is presented that reduces the power dissipated in the fast data paths of a system.

One of the most critical signals in a synchronous digital circuit is the clock signal. It is important to reduce the uncertainty of the clock signal delay, particularly of the clock signals driving the registers belonging to the most critical data paths. A methodology that controls the topology of the clock tree so as to improve the tolerance of the clock signal to delay uncertainty is described. The primary target of that methodology is to reduce the non-common portion of the clock tree among the clock paths that drive the registers of the most critical data paths. An algorithm that implements this methodology and extracts the clock tree topology is presented. The application of the algorithm to a set of benchmark circuits demonstrates a significant reduction in the delay uncertainty of the clock signal in the most critical data paths.

Furthermore the effect of layout design elements on the delay uncertainty of the clock signal is considered. The delay uncertainty introduced by device parameter variations and interconnect crosstalk is investigated. It is demonstrated that increasing the size of the clock buffers reduces the effects that introduce delay uncertainty, albeit with increasing the power dissipation on the clock tree. The dependence of delay uncertainty upon physical characteristics is leveraged in the development of a methodology for clock tree physical layout design. The developed methodology utilizes buffer insertion and layout enhancement techniques to reduce the delay uncertainty of the clock signal to the registers of the most critical data paths. The primary tradeoff in the application of the proposed techniques is between the power dissipation and total area of a clock distribution network.

Contents

Dedication **ii**

Curriculum Vitae **iii**

Acknowledgements **iv**

Abstract **vii**

List of Tables **xii**

List of Figures **xiv**

1 Introduction **1**

2 Delay Uncertainty in High Speed CMOS Circuits **8**

- 2.1 Variations of CMOS device parameters 10
  - 2.1.1 Effective channel length variation 10
  - 2.1.2 Variations in carrier mobility 13
  - 2.1.3 Variations in the gate oxide thickness 15
  - 2.1.4 Threshold voltage variation 16
- 2.2 Variations in interconnect parameters 22

2.3 Interconnect noise 25
  2.3.1 Capacitive interconnect coupling 25
  2.3.2 Inductance effects 27
2.4 Variations in system parameters 30
  2.4.1 Power supply fluctuations - IR drops 30
  2.4.2 Electromagnetic Interference (EMI) effects 32
  2.4.3 Temperature variations 34
2.5 Conclusions 34

3 Performance Enhancements Through Clock Skew Scheduling 36
3.1 Improving the Timing Constraints and Speed 37
  3.1.1 Background on clock skew scheduling 37
  3.1.2 Optimal clock skew schedule algorithm implementation . . 42
  3.1.3 Experimental results from application of optimum clock skew scheduling 45
3.2 Reducing Power in Non-Critical Data Paths 49
  3.2.1 The general technique and related delay constraints 50
  3.2.2 Application to critical data paths 53
  3.2.3 Results on a demonstration circuit 55
3.3 Conclusions 57

4 Clock Tree Topological Enhancements to Reduce Delay Uncertainty 59
4.1 Concept of the Algorithm 60
4.2 Description of the Algorithm 64
4.3 Incorporating non-zero clock skew scheduling 66

4.4 Experimental results 69
4.5 Conclusions 72

5 Buffer Sizing for Reduced Delay Uncertainty 73
5.1 Delay uncertainty due to device parameter variations 75
5.2 Delay uncertainty due to interconnect crosstalk 82
5.3 Power dissipation tradeoffs 89
5.4 Conclusions 92

6 Clock Tree Buffer Insertion and Layout Enhancements 93
6.1 Buffer insertion and sizing 94
6.2 Dedicated minimal clock tree driving the critical path registers . . 100
6.3 Conclusions 104

7 Conclusions 105

8 Future Research 110
8.1 Incorporating delay uncertainty into the design flow 111
8.2 Design techniques for reducing delay uncertainty 114
  8.2.1 Tapered buffer insertion for delay balancing 114
  8.2.2 Shielding of clock lines 116
8.3 New design trends and challenges 118

References 119

A Clock Tree Layouts 127

B Design Example 135

C Publications 138

List of Tables

2.1 Parameter values for electron and hole carriers at 300° K 14

3.1 Comparison between the original and the increased delay of data paths B, C, and D within the FUB illustrated in Fig. 3.6 56

3.2 Normalized power dissipation within the circuit block containing the latches 56

4.1 Reduction in delay uncertainty of the most critical data paths. BF describes the branching factor of the original binary tree. 71

5.1 Transistor size and power dissipation components for different buffer sizes. 89

6.1 Increase in power dissipation with increasing buffer size. 100

6.2 Tradeoff between the reduction in power dissipation and the increase in clock tree area. 103

6.3 Comparison between the reduction in power dissipation and the reduction in the aggregate buffer size. 104

B.1 Application of different design techniques to manage delay uncertainty. 136

List of Figures

1.1 Evolution of the microprocessor since the 1970's 3

2.1 Basic steps of the transistor source and drain creation process . . 11

2.2 V_{SB} variations due to floating sources along the N-channel tree portion of a three input NAND gate 19

2.3 Standard deviation of V_{th} as a function of t_{ox} 20

2.4 Standard deviation of V_{th} as a function of doping concentration . 21

2.5 Geometric parameters of the interconnect lines 22

2.6 Copper line thickness losses due to dishing and erosion effects . . 24

2.7 Variation in ILD thickness due to different pattern density of interconnect lines 24

2.8 Capacitive coupling among different interconnect layers 26

2.9 Capacitively coupled net 27

2.10 Transition time (t_r) versus the length of the interconnect line (l). The shaded area denotes the region where inductance is important 29

2.11 Voltage drop due to the resistance within the power distribution network 31

2.12 EMI-induced delay versus EMI phase differences for a logic transition 33

3.1 A local data path. 38

3.2 Examples of positive and negative clock skew. 39

3.3 Preventing timing hazards in synchronous digital systems. 41

3.4 Clock skew permissible range. 42

3.5 The minimum clock period extraction algorithm 43

3.6 Circuit graph of Itanium^{TM} FUB with normalized data path delays. 46

3.7 Variation of clock signal delay to different clock buffer sizes. . . . 48

3.8 Path delay distribution with zero clock skew scheduling 50

3.9 Increasing the delay of the fast data paths by downsizing the local latches that drive these paths. 51

3.10 The added delay of the fast data path should not violate the long path timing constraint. 52

3.11 Application of local clock skew to equalize the available idle time between the long and short delay data paths. (a) Initial timing of the data paths. (b) Timing of the data paths after the applicati on of local clock skew 54

4.1 Introduction of different clock signal delays to the non-common portions of the clock tree. 61

4.2 Uncertainty graph representation of a circuit. 62

4.3 Clock tree topology for the circuit shown in Fig. 4.2. 63

4.4 Iterations of the algorithm to reduce the input graph to a single node. 65

4.5 Incorporating non-zero clock skew scheduling within the CTT algorithm 68

4.6 Comparison between the algorithmically extracted CTT and a binary tree. 70

5.1 Dependence of device parameters on process, environmental, and system parameters 78
5.2 Reduction in delay with increasing inverter size 79
5.3 Uncertainty in inverter delay due to process, environmental, and system parameter variations 80
5.4 Delay uncertainty due to 10% V_{dd} variation for different input transition times 81
5.5 Effect of input signal transition time t_T on delay uncertainty . . . 82
5.6 Capacitive coupling between two interconnect lines 84
5.7 Simulation of capacitively coupled interconnect 85
5.8 Uncertainty of the signal delay of the buffer driving the victim line due to different switching activities of the aggressor line. Low coupling between the interconnect lines is considered 86
5.9 Reduction in delay uncertainty along the victim line with increasing buffer size 87
5.10 Delay uncertainty increases proportionally with capacitive coupling among the lines 88
5.11 Increase in power dissipation with buffer size 90
5.12 Power-Delay uncertainty Product (PD_UP) for device parameter variations, interconnect coupling, and the sum of both effects . . . 92

6.1 An example of a minimal rectilinear Steiner clock tree 96
6.2 Buffer insertion in the clock tree shown in Fig. 6.1 97

6.3 Increasing buffer size to reduce delay uncertainty 99
6.4 Dedicated clock tree and buffers to drive the critical registers in the circuit shown in Fig. 6.1 102

8.1 Incorporating delay uncertainty models into the design flow to estimate delay uncertainty earlier in the circuit design process . . . 113
8.2 Sizing of tapered buffers per stage to match upstream and downstream delays 115
8.3 Shielding a victim line with power supply lines 117
8.4 Peak noise increases as coupling occurs farther from the driver . . 117

A.1 Circuit # 1 - buffer insertion and sizing 128
A.2 Circuit # 1 - dedicated minimal clock tree 128
A.3 Circuit # 2 - buffer insertion and sizing 129
A.4 Circuit # 2 - dedicated minimal clock tree 129
A.5 Circuit # 3 - buffer insertion and sizing 130
A.6 Circuit # 3 - dedicated minimal clock tree 130
A.7 Circuit # 4 - buffer insertion and sizing 131
A.8 Circuit # 4 - dedicated minimal clock tree 131
A.9 Circuit # 5 - buffer insertion and sizing 132
A.10 Circuit # 5 - dedicated minimal clock tree 132
A.11 Circuit # 6 - buffer insertion and sizing 133
A.12 Circuit # 6 - dedicated minimal clock tree 133
A.13 Circuit # 7 - buffer insertion and sizing 134
A.14 Circuit # 7 - dedicated minimal clock tree 134

B.1 Example of circuit data paths 135

Chapter 1

Introduction

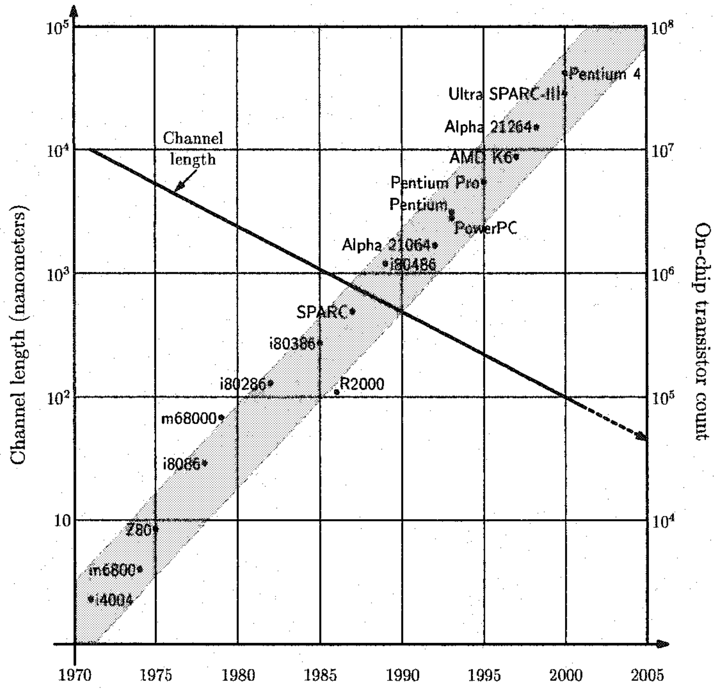
The microprocessor is commonly used in a broad spectrum of human activities varying from the sciences and engineering to arts and entertainment. The capability of the microprocessor to process and transfer large amounts of data and information that were inconceivable in the past has made possible a number of important breakthroughs. Furthermore, everyday life has radically been changed, affecting the way people access news, information, and knowledge, as well as the way people communicate with each other. All of these achievements have changed our lives forever and are based on the microprocessor. Within thirty years, the microprocessor has become the life support system of the modern world [1].

The first commercial integrated microprocessor was the 4004, launched by Intel Corporation in 1971. The 4004 offered approximately the same performance as the *ENIAC* with 18000 vacuum tubes did in 1946. The low cost (approximately \$200) and tiny size (12 mm^2) of the 4004 enabled engineers to create new categories of society changing products. At the same time, however, skeptics predicted that the market for a single integrated circuit computer would be tiny. Soon afterwards, these predictions were proved wrong. The computer-on-a-chip has become one of the largest markets in the world and a fundamental factor in the world economy.

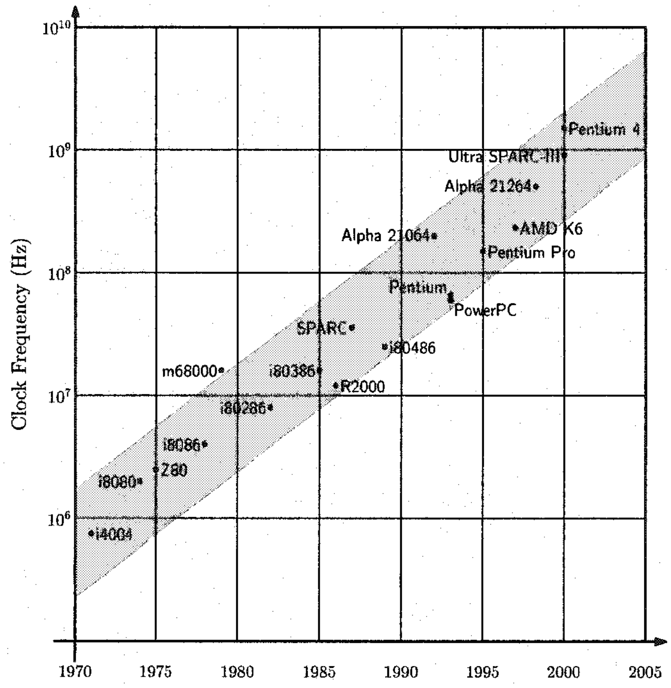
Since 1971, the development of the microprocessor has continued, steadily following Gordon Moore's prediction in the 1960's. Moore predicted that the density of integrated circuits (IC) would double every eighteen months. This statement is widely known as *Moore's law* [2]. The increase in circuit density is accompanied by a decrease in the on-chip feature size [3], enabling system designers to enhance the functionality of an IC by adding more components. The trend characterizing the reduction in the on-chip feature size (*i.e.*, the channel length) together with a significant increase in circuit density since 1971 are shown in Fig. 1.1(a). In addition, circuits with smaller device sizes can operate faster, permitting the operating clock frequency to be increased. This increasing trend in operating clock frequencies is shown in Fig. 1.1(b)

During the evolution of the microprocessor, the circuit design process has faced several serious challenges and obstacles. Providing solutions to overcome these challenges has permitted IC design and manufacturing technologies to mature and move forward into new eras where more challenging problems would be faced. In the 1970's, the primary design constraint was die area. Small wafer area and the high density of defects per die area degraded manufacturing yield and made the fabrication of large ICs extremely costly and non-profitable. Solutions were provided by the development of scaling techniques [3] which reduced the minimum feature size and die area. In addition, manufacturing process technologies have improved, dramatically reducing defect densities while improving manufacturing yield.

Scaling the feature size, in addition to relaxing area constraints, has also increased the functionality of the ICs, since more system components could be integrated onto a single die. In addition, circuits with smaller channel lengths are



(a) Reduction of channel length and increase of circuit density



(b) Increase in operating speed

Fig. 1.1: Evolution of the microprocessor since the 1970's

able to operate faster, boosting the operating frequency of the microprocessor, as shown in Fig. 1.1(b). The quest for improved functionality, faster circuit speed, and lower power dissipation has shifted the manufacturing process from PMOS to NMOS and finally to CMOS, introducing the era of VLSI circuits, where the term *VLSI* stands for Very Large Scale Integration.

As the number of transistor devices on an IC and the operating frequency have both been increasing, on-chip power dissipation has become a challenging issue. Several power reduction techniques and strategies have been developed to reduce the on-chip power dissipation. One of the most effective techniques is scaling the power supply voltage [4]. This approach exploits the quadratic dependence of power dissipation on supply voltage. The scaling of the supply voltage, however, reduces the noise margins of a circuit [5], thereby increasing the sensitivity of a system to noise.

The effect of noise on circuit operation and reliability has recently become an extremely important issue in the design of high performance integrated circuits. Presently, the leading integrated circuit manufacturers have the technological capability to mass produce VLSI circuits with a feature size of less than a hundred nanometers [6]. These technologies are identified with the term *very deep submicrometer* (VDSM), since the minimum feature size is well below the one micrometer mark. At these very small geometries, the deleterious effects caused by noise are aggravated. The challenges faced today in the design of VDSM circuits are greater than ever.

The main effect of noise is the degradation of signal integrity and an increased uncertainty in the signal delay. The research presented in this dissertation is specifically concerned with the effects of delay uncertainty in high performance

CMOS integrated circuits. The results of this research focus on improving the tolerance of a system to delay uncertainty.

The variation of process and environmental parameters is the primary source of noise that introduces uncertainty in the delay of the signals propagating within a circuit. At the transistor device level, the variation of geometric and electrical parameters such as the effective channel length and threshold voltage are some of the important effects that introduce delay uncertainty. In addition, a significant effect on the signal delay is caused by the coupling of signals [7] due to capacitive [8] and inductive [9] coupling among the wires. The variations of the geometric parameters of the interconnects due to limitations of the manufacturing process [10–12] also cause uncertainty in the delay of a signal propagating through an interconnect line. At the system level, variations in the power supply voltage [13], temperature variations [14], and electromagnetic effects [15] can affect the signal delay. These effects that introduce delay uncertainty are summarized in Chapter 2.

The uncertainty of the propagation delay of a signal can cause a catastrophic violation of the timing constraints within a system. With increasing clock frequencies, these constraints have become tighter and the sensitivity of a system to delay uncertainty has increased. One way to improve the tolerance of a system to delay uncertainty is by relaxing the timing constraints of the critical paths. In Chapter 3, a methodology that implements this concept through the application of non-zero clock skew scheduling is described. A variation of this methodology is applied to reduce the power dissipated in the fast data paths of a system. This problem is discussed and the application of the proposed solution to an industrial circuit is described in Chapter 3.

One of the most critical signals in a synchronous digital circuit is the clock signal. It is important to reduce the uncertainty of the clock signal delay, particularly the clock signals driving the registers belonging to the most critical data paths. In Chapter 4, a methodology is described that controls the topology of the clock tree so as to improve the tolerance of the clock signal to delay uncertainty. An algorithm that implements this methodology and extracts the clock tree topology is also presented in Chapter 4. The application of the algorithm to a set of benchmark circuits demonstrates a significant reduction in the delay uncertainty of the clock signal in the most critical data paths.

The delay of the clock signal propagating along a clock distribution network can be controlled by inserting clock buffers within the signal path. The variation of the device parameters, however, can change the current flow through that buffer, thereby introducing uncertainty in the buffer delay. The dependence of the uncertainty in the buffer delay upon the size of a buffer is investigated in Chapter 5. It is shown that the uncertainty in the buffer delay can be reduced by increasing the buffer size. Furthermore, it is demonstrated that increasing the buffer size reduces the delay uncertainty caused by crosstalk coupling among interconnect lines. The primary drawback of increasing the clock buffer size is the increase in power dissipation. The tradeoff between reducing the delay uncertainty and increasing the power dissipation is considered by introducing the Power-Delay-uncertainty-Product (PD_UP).

The delay uncertainty of a clock signal is strongly dependent upon the geometric characteristics and the spatial location of the clock lines and registers. Therefore, in order to estimate and reduce delay uncertainty, physical layout information should be incorporated into the clock distribution network design process. Two different strategies are described in Chapter 6 for synthesizing the clock tree layout. Both strategies reduce the delay uncertainty of the clock signals arriving at the registers of the most critical data paths. Clock buffer insertion and sizing is utilized in one approach, exploiting the reduction in delay uncertainty with increasing buffer size. In the second approach, the clock signal is distributed to the registers of the critical paths by a dedicated portion of the clock tree, which increases the common path among the clock signals. A tradeoff between power dissipation and wire length is demonstrated by the application of these two strategies to the synthesis of a clock layout for a set of benchmark circuits.

Conclusions of this dissertation are offered in Chapter 7. Finally, ideas for future research that will enhance and incorporate the presented methodologies into a unified system level design methodology to control the effects that introduce delay uncertainty are presented in Chapter 8.

Chapter 2

Delay Uncertainty in High Speed CMOS Circuits

The primary characteristic of the microelectronics revolution, as discussed in Chapter 1, is the rapid decrease in device size, producing phenomenal increases in circuit density, functionality, and operational clock frequencies [6,16]. Scaling of the device geometries supports the system-on-a-chip integration of multiple subsystems [17,18], greatly increasing the number of on-chip clocked elements. These effects have resulted in hundreds of thousands of elementary operations being executed in sequences specified by application-specific algorithms and controlled by a clock signal, operating within time periods much less than a nanosecond [19]. These constraints demonstrate the tight timing control of the arrival times of the clock signal at the many registers distributed throughout an integrated circuit. Deviations of the clock signal from the target delay can cause incorrect data to be latched within a register, resulting in a system malfunctioning.

The variation of the signal propagation delay is described by the term *delay uncertainty.* The effects that introduce uncertainty in the signal delay can generally be referred to as *noise.* Examples of effects that produce noise are the variations of process, environmental, and system parameters and interconnect signal coupling.

One common characteristic of these phenomena is that these effects are highly unpredictable. It is therefore extremely difficult to estimate and adequately model these effects. In addition, most of those parameters whose variation introduces delay uncertainty cannot be controlled within the circuit design process. For example, the difference in the arrival time of the clock signal between two paths due to different wire lengths in a clock distribution network can be estimated based on the impedance characteristics of a line. That difference in delay can also be compensated by certain design techniques, such as decreasing the length of the longer path, or inserting buffers of an appropriate size [20]. However, there is variation in the signal delay of these paths due to the non-uniformity of the interconnect lines caused by imperfections during the etching and metal deposition processes. Additional variations in delay are caused by the interaction between the clock signal and data signals on neighboring lines. These variations in delay are extremely difficult to predict and to compensate, creating delay uncertainty.

An approach to manage these delay uncertainties is to develop design methodologies that tolerate the parameter variations rather than compensate for these variations. A statistical analysis can be used to model those parameter variations, providing quantitative estimates of the delay uncertainty. These models are used to develop design methodologies that account for the effects that introduce delay uncertainty and produce systems capable of tolerating delay variations. The issue of delay uncertainty, however, remains a crucial one since the race for smaller devices and faster operational frequencies aggravates the effects that cause delay variations. Designing a system tolerant to delay uncertainty is a highly challenging task that involves precise balancing between design tradeoffs and operational specifications.

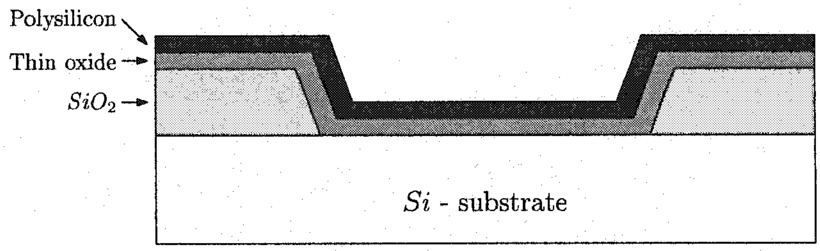
The effects that introduce delay uncertainty in different circuit elements are discussed in this chapter. Design methodologies and techniques that tolerate delay uncertainty are presented in the following chapters. The effects that cause uncertainty in the delay of a signal propagating through a CMOS device are presented in section 2.1. Geometric parameter variations of interconnect lines and the effect of these variations on delay are discussed in section 2.2. The effect of signal transition noise on interconnects is described in section 2.3. In section 2.4, the effects of system level parameter variations are discussed. Finally, some conclusions are presented in section 2.5.

2.1 Variations of CMOS device parameters

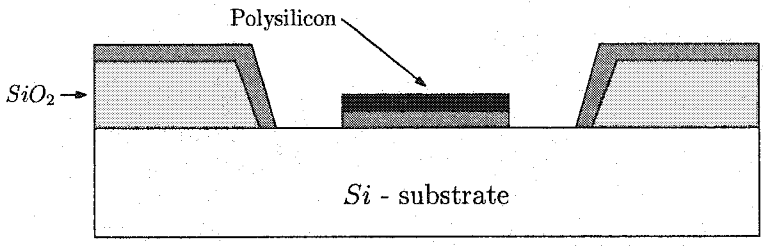
As device sizes are decreased, the effect of device parameter variations is aggravated. Assuming that the magnitude of the variation of an MOS device parameter remains constant, the per cent variation of this parameter increases inversely proportional with the scaling of the device. In this section, the variation of several device parameters and the effects on the signal propagation delay through a CMOS circuit are discussed. The variation of the effective channel length is discussed in subsection 2.1.1. Variations in carrier mobility are presented in subsection 2.1.2. Gate oxide thickness variation is described in subsection 2.1.3, while variations in threshold voltage are presented in subsection 2.1.4.

2.1.1 Effective channel length variation

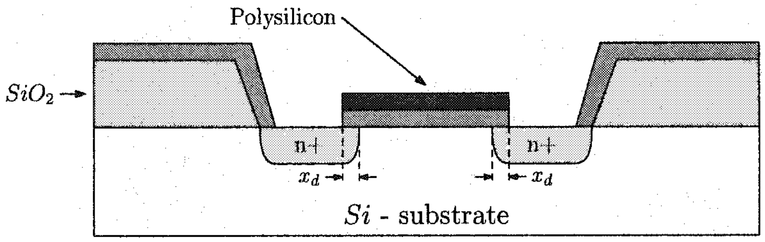
The basic steps of the semiconductor manufacturing process that creates the source and drain of an NMOS transistor are shown in Fig. 2.1 (the process that



(a) Deposition of thin oxide and polysilicon layers



(b) Etching the poly and oxide layer to expose the source and drain regions



(c) Doping of the source and drain regions

Fig. 2.1: Basic steps of the transistor source and drain creation process

creates a PMOS device is similar with only minor modifications). Initially, the silicon substrate is covered with a thick layer of SiO_2. Afterwards, the SiO_2 layer is etched in the area where a transistor will be created. This area is covered with a layer of thin oxide, followed by a layer of polysilicon, as shown in Fig 2.1(a). After deposition, the polysilicon layer is patterned and etched to form the interconnects and the MOS transistor gate. The thin oxide layer not covered by polysilicon is also etched away, exposing the bare silicon substrate on which the source and drain regions are formed [see Fig. 2.1(b)]. The length of the polysilicon and the thin oxide layers that remain on the silicon substrate define the *gate* and the *channel length* of the transistor. The exposed silicon surface is then doped with a high concentration of donors, either through diffusion or ion implantation. As shown in Fig 2.1(c), the donors penetrate the exposed areas of the silicon substrate, creating two n-type regions. Note that the polysilicon gate, which is patterned before doping, actually defines the location of the channel region as well as the locations of the source and drain regions. Since this process allows the direct positioning of the two regions relative to the gate, this process is also called a *self-aligned* process [21].

Ideally, the edge of the source and drain regions should be aligned with the edge of the gate oxide layer. Practically, however, both the source and drain regions tend to extend below the oxide by an amount x_d, the *lateral diffusion*, as shown in Fig. 2.1(c). The effective channel L_{eff} of a transistor therefore becomes shorter than the drawn gate length L by a factor of $\Delta_L = 2x_d$. As shown in [22], the lateral diffusion factor for a 0.25 μm CMOS technology is approximately $\Delta_L = 0.08$ μm. As a rule of thumb, the lateral diffusion factor is about 30% of the drawn transistor gate [6].

There are several effects that can cause variations in the effective channel length of a transistor. These variations can be either global (*i.e.*, variations of the effective channel length among transistors at different areas within a die), or local (*i.e.*, variation of the effective channel length within a single transistor).

One of the effects that can cause variations of the effective channel length is a change in the lateral diffusion of the source and drain areas below the transistor gate. The lateral diffusion x_d is proportional to the donor doping density at the source and drain areas. Therefore, a non-uniform doping density is possible which will create non-uniform lateral diffusion areas and variations in the effective channel length. Usually, a non-uniform doping density has a global profile across a die, therefore, variations in the effective channel length due to non-uniform doping densities occur between transistors at distant areas on an IC. Other effects such as misalignment or misplacement of the photoresist masks and resolution limitations of the photolithography process also contribute to global variations in the effective channel length.

Alternatively, etching imperfections of the gate oxide and polysilicon layers that define the transistor gate may create local variations in the effective channel length within a single transistor. These imperfections may occur randomly along a transistor. However, the wider a transistor gate, the higher the probability that these imperfections will occur. Therefore, this effect is greater in the case of wide transistors.

2.1.2 Variations in carrier mobility

When an electric field is applied across a semiconductor, the charge carriers within the semiconductor lattice are accelerated. This carrier motion is described as *drift*. As the carriers move within the crystal, the carriers collide with ionized impurity atoms and thermally agitated lattice atoms. These collisions are described with the term *carrier scattering*. The *mobility* of the carriers characterizes the "ease of carrier motion" within the semiconductor lattice, as described by Pierret in [23]. The higher the carrier mobility, the greater the current flow through a semiconductor. Carrier mobility is, therefore, an important parameter that greatly affects the performance of an MOS device.

Carrier scattering can be characterized by two separate effects. The first effect is the collision of carriers with thermally agitated lattice atoms, described as *lattice scattering*. The second effect is the collision of carriers with ionized impurity atoms, called *ionized impurity scattering*. Both effects decrease carrier mobility. For impurity atom concentrations below $10^{15}/cm^3$, the effect of ionized impurity scattering on mobility can be neglected. However, if the impurity concentration is well above $10^{15}/cm^3$, the ionized impurity scattering cannot be neglected and the mobility of carriers are dependent on the doping concentration. The dependency on doping is described by

$$\mu = \mu_{min} \frac{\mu_o}{1 + (N/N_{ref})^{\alpha}}, \tag{2.1}$$

where μ is the carrier mobility (μ_n for electrons and μ_p for holes) and N is the doping concentration (N_D for donors and N_A for acceptors). The parameters μ_{min}, μ_o, N_{ref}, and α are empirical parameters. Typical values of these empirical parameters for electron and hole carriers (at 300° K) are listed in Table 2.1 [23].

Table 2.1: Parameter values for electron and hole carriers at 300° K

Parameter	Electrons	Holes
μ_{min} $(cm^2/Vsec)$	92	54.3
μ_o $(cm^2/Vsec)$	1268	406.9
N_{ref} (cm^{-3})	$1.3 \cdot 10^{17}$	$2.35 \cdot 10^{17}$
α	0.91	0.88

In addition to doping concentration, carrier mobility also strongly depends upon temperature. The temperature controls the agitation of the atoms within the semiconductor lattice. An increase in temperature increases the thermal agitation of these atoms and the probability of the carriers colliding with the atoms. The mobility, therefore, decreases with increasing temperature. For impurity doping concentrations below $10^{14}/cm^3$, temperature variations are the primary cause of mobility variations. For doping concentrations greater than $10^{14}/cm^3$, the carrier mobility decreases with increasing temperature, but this effect is less severe due to the dominant effect of ionized impurity scattering.

2.1.3 Variations in the gate oxide thickness

Another device parameter that is susceptible to imperfections of the manufacturing process is the gate oxide thickness (t_{ox}). The oxide layer at the gate of a transistor is extremely thin, with a nominal value well below 100 angstroms (45 Å for a 0.25 μm CMOS process [22]). The slightest imperfections of the deposition process can create significant variations in t_{ox}. The gate oxide thickness determines the input capacitance of a transistor gate, as well as the characteristics of the channel inversion and carrier concentration within the channel. Therefore, variations in t_{ox} can introduce uncertainty both in the input signal that drives the gate of a transistor and in the output signal of a transistor that depends upon the current flow in the channel. These effects are discussed in greater detail below.

The delay of a signal driving a transistor depends upon the total input capacitive load seen at the transistor gate. A portion of this load is the gate parasitic capacitance C_g,

$$C_g = C_{ox}\frac{W}{L} = \frac{\varepsilon_{ox}}{t_{ox}}\frac{W}{L}. \tag{2.2}$$

In (2.2), C_{ox} is the gate oxide capacitance per unit area and equals $\frac{\varepsilon_{ox}}{t_{ox}}$, where ε_{ox} is the dielectric constant of silicon dioxide. W and L represent the gate width and drawn length ($L = L_{eff} + 2x_d$), respectively. As shown in (2.2), t_{ox} determines the input capacitance of a transistor gate and therefore the delay of the input signal driving the transistor.

The delay of the output signal depends upon the amount of current flowing through the gate of the transistor. A first order approximation of this current is given by the Shockley model [24] for each region of operation of a transistor.

$$I_D = \begin{cases} 0 & : \ (V_{GS} \leq V_{th}) \\ \mu\frac{\varepsilon_{ox}}{t_{ox}}\frac{W}{L}\left\{(V_{GS} - V_{th})V_{DS} - \frac{1}{2}V_{DS}^2\right\} & : \ (V_{GS} \geq V_{th})\&(V_{DS} \leq V_{DSAT}) \\ \frac{1}{2}\mu\frac{\varepsilon_{ox}}{t_{ox}}\frac{W}{L}(V_{GS} - V_{th})^2 & : \ (V_{GS} \geq V_{th})\&(V_{DS} \geq V_{DSAT}). \end{cases} \tag{2.3}$$

As shown in (2.3), the current flow through an on transistor is inversely proportional to the gate oxide thickness. Therefore, variations in t_{ox} may cause changes in the current flow and introduce delay uncertainty at the output of a transistor.

2.1.4 Threshold voltage variation

Equation (2.3) also demonstrates the effect of another device parameter on the current flow, that is the threshold voltage (V_{th}). Threshold voltage is one of the most important parameters that characterize the operation and behavior of a CMOS circuit. As shown in (2.3), V_{th} determines the current flow through a transistor and, consequently, the signal delay. In addition, V_{th} determines the noise margins of a digital gate [5] that define the tolerance of a system to signal noise. The discussion that follows about V_{th} dependencies and variations is in terms of an N-channel device but the results are applicable, with minor modifications, to P-channel devices as well.

There are four physical components of the threshold voltage [21] that should be considered for almost all practical purposes. These are:

i. The work function difference Φ_{GC} between the gate and the channel of a transistor. Φ_{GC} represents the built-in potential drop in a MOS system.

ii. The change of the surface potential (*i.e.*, the voltage that is applied to achieve surface inversion) $-2\phi_F$, where ϕ_F is the Fermi potential for silicon,

$$\phi_F = \frac{kT}{q} \ln \frac{n_i}{N_D}. \tag{2.4}$$

In (2.4), k denotes the Boltzmann constant, q is the electron charge, T is the temperature, N_D represents the substrate doping density, and n_i is the intrinsic carrier concentration.

iii. The potential (*i.e.*, the voltage drop across the gate oxide) that offsets the depletion region charge $\frac{Q_B}{C_{ox}}$, where C_{ox} is described in (2.2) and Q_B is the depletion region charge density,

$$Q_B = -\sqrt{2qN_D\varepsilon_{Si}|-2\phi_F + V_{SB}|}, \tag{2.5}$$

where ε_{Si} is the dielectric constant of silicon, and V_{SB} is the source-to-substrate voltage.

iv. The voltage component to offset the fixed positive charge density Q_{ox} at the interface between the gate oxide and the silicon substrate. This voltage component is $-\frac{Q_{ox}}{C_{ox}}$.

Therefore, the threshold voltage V_{th} can be analytically expressed by the summation of these four components:

$$V_{th} = \Phi_{GC} - 2\phi_F - \frac{Q_B}{C_{ox}} - \frac{Q_{ox}}{C_{ox}}. \tag{2.6}$$

In this one-dimensional model, the variation in the threshold voltage δV_{th} arises from four independent sources. These sources are: the substrate bias V_{SB} (*e.g.*, body effect), the gate oxide thickness t_{ox}, the substrate doping density N_D, and the effective channel length L_{eff}. The effect of each of these factors on the variation of the threshold voltage is discussed below.

The dependence of the threshold voltage on the substrate bias V_{SB} is described by (2.5) and (2.6). Qualitatively, the width of the depletion region formed under the channel increases when a negative back-bias voltage (V_{SB}) is applied at the substrate with respect to the source. The increased depletion region requires additional charge in order to maintain a constant channel conductivity [21]. Therefore, the voltage V_{GS} applied at the gate is increased which corresponds to an increase in the threshold voltage V_{th}. Alternatively, if the substrate potential with respect to the source increases, the depletion region below the channel is decreased and the threshold voltage decreases.

Variations of the V_{SB} voltage are very common in those cases where the source of the transistor is floating. An example of a floating source is the NMOS stack structure of a 3-input NAND gate, shown in Fig. 2.2 [25].

The effect of the gate oxide thickness on threshold voltage is described by the two components of V_{th}: $-\frac{Q_B}{C_{ox}}$ and $-\frac{Q_{ox}}{C_{ox}}$. The variation of gate oxide thickness therefore affects the charge at the depletion region in the channel as well as the charge density at the oxide-substrate interface. The importance of variations in

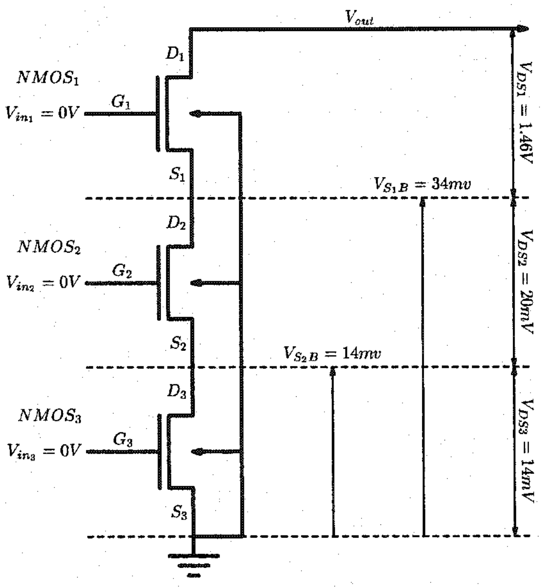


Fig. 2.2: V_{SB} variations due to floating sources along the N-channel tree portion of a three input NAND gate

the gate oxide thickness increases as the on-chip feature size is reduced into the deep submicrometer range. As shown in [26], the effect of t_{ox} variations on V_{th} is four times more significant in a 0.1 μm device as compared with a 1.0 μm sized transistor.

Experimental measurements of the standard deviations of V_{th} as a function of the gate oxide thickness are shown in Fig. 2.3. The closed and open circles show the experimental data for L_{eff}=0.5 μm and 0.3 μm CMOS technology respectively [27].

In addition to the effect of the gate oxide thickness, the effect of the effective channel length on the standard deviation of V_{th} is shown in Fig. 2.3. The data illustrated in Fig. 2.3 is another indication that as the on-chip feature size is decreased, variations in the device parameters becomes increasingly important. The nominal value of V_{th}, however, is found to monotonically decrease with decreasing

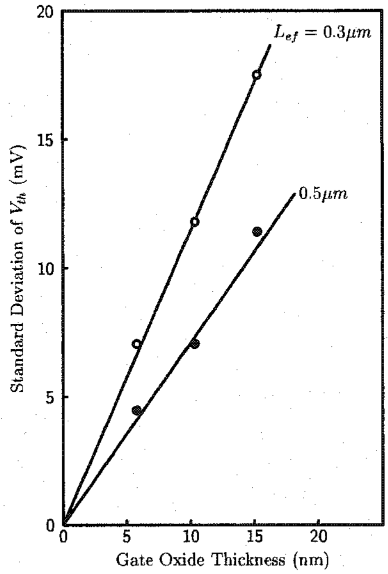


Fig. 2.3: Standard deviation of V_{th} as a function of t_{ox}

channel length. Qualitatively, a reduction in threshold voltage can be explained as follows [23]. In order to form an inversion layer, or channel, beneath the gate, the subgate region must first be depleted. In a short-channel device, the *pn* junctions formed between the source/drain regions and the silicon substrate assist in depleting the region under the gate. Thus, less charge is required at the gate to reach the state of channel inversion, therefore V_{th} decreases. As the L_{eff} becomes smaller, the source and drain *pn* junctions assist in depleting a greater percentage of charge under the gate, further reducing V_{th}.

Another process parameter that affects the threshold voltage is the substrate doping density N_D. As shown in (2.4) and (2.5), the substrate doping density contributes both to the surface potential (ϕ_F) and the charge density of the depletion region (Q_B). The substrate doping density is the controlling parameter of the threshold voltage in low power applications such as dual-threshold voltage systems that reduce standby dissipation caused by subthreshold leakage currents

[28]. Experimental results determined from varying the threshold voltage with respect to the substrate doping density are presented in [27]. The standard deviation of threshold voltage σV_{th} as a function of the average channel doping density is shown in Fig. 2.4 for two different channel lengths. As shown in Fig. 2.4, the change in σV_{th} with respect to the doping density is relatively small. This behavior explains the use of doping density to control the threshold voltage [27].

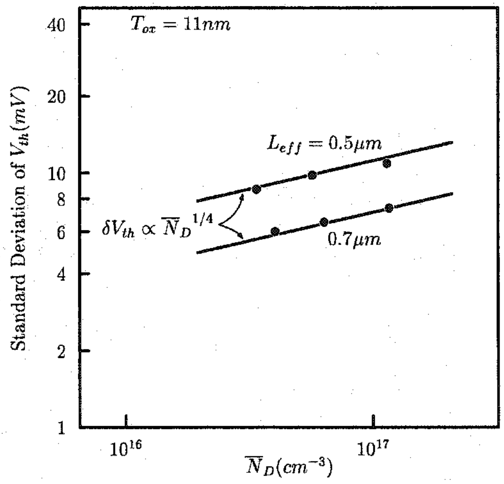


Fig. 2.4: Standard deviation of V_{th} as a function of doping concentration

2.2 Variations in interconnect parameters

As the on-chip feature size is reduced concurrently with increasing chip dimensions, the on-chip interconnect delay has become more significant than the gate delay. By decreasing the on-chip feature size, the geometric parameters of the interconnect lines have also been decreased. Reducing the interconnect wire width and the distance between the lines increases the per unit length wire resistance and interline capacitance. In addition, increasing die dimensions have resulted in interconnect lines running over longer distances across an IC. The interconnect component of the on-chip signal delay has, therefore, become increasingly significant.

The basic geometric parameters of the interconnect lines are shown in Fig. 2.5. The symbols w, h, d, L, and t_{ILD} are the line width, the metal thickness, the distance between two lines of the same metal level, the length of the line, and the thickness of the interlevel dielectric oxide, respectively. The variation of these parameters due to manufacturing process imperfections produces variations in the parasitic resistance and capacitance of a line. These impedances determine the

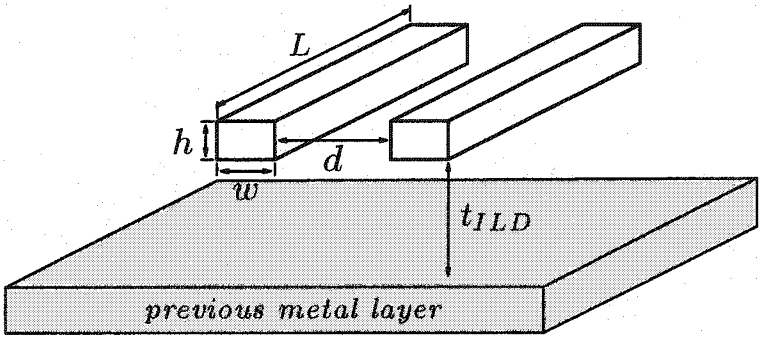


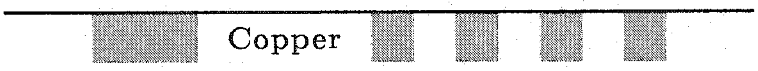
Fig. 2.5: Geometric parameters of the interconnect lines

signal propagation delay within a line. Variations in the geometric parameters, therefore, create delay uncertainty. The dominant role of interconnect on the total on-chip delay demonstrates the importance of delay uncertainty. The effects that create variation in each one of the interconnect geometric parameters are discussed below [25].

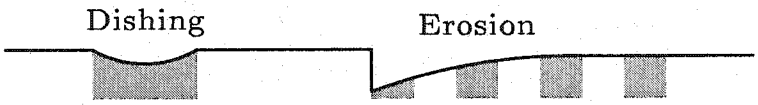
Line width and interline spacing. Line width variation arises primarily due to photolithography and etching imperfections. At smaller dimensions (*i.e.*, lower metal levels), line width variations occur due to proximity and lithographic effects. Etching effects which depend on the line width and local layout can also create non-uniformities of the deposited metal. The variation of the line width has a direct impact on the resistance of a line as well as on the capacitance between overlapping lines on different metal layers. In addition, line width variations also result in differences in interline spacing between lines of the same metal layer. These spacing variations can affect the line-to-line capacitance, thereby affecting the crosstalk and signal integrity.

Metal thickness. The deposition of metal wires and barriers is well controlled in an aluminum metal interconnect process. Small variations in metal thickness can occur on different dies across a wafer or among different wafers. However, in damascene (*e.g.*, copper) processes, the metal thickness of the patterned lines can vary significantly due to dishing and erosion effects, as shown in Fig. 2.6. The losses of line thickness depend upon the particular line patterns and are within the range of 10% to 20% [12].

Dielectric thickness. The thickness of deposited and polished dielectric oxide layers over metal layers can also vary significantly. The variation of the dielectric oxide thickness across a wafer is within the range of 5%. However, greater varia-



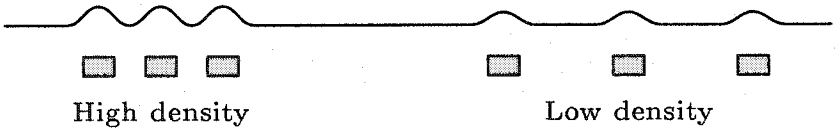
(a) Different patterns of copper interconnect lines



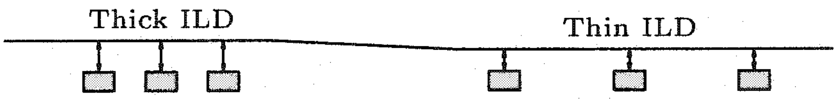
(b) Dishing and erosion effects after the polishing process

Fig. 2.6: Copper line thickness losses due to dishing and erosion effects

tions can occur due to pattern dependencies of the dielectric deposition process. For example, in high-density plasma (HDP) processes the dielectric thickness depends strongly upon the size and/or width of a deposited feature. Furthermore, in chemical mechanical polishing (CMP) processes the dielectric oxide thickness can vary significantly depending upon the effective density of the underlying interconnect lines, as shown in Fig. 2.7.



(a) Dielectric oxide deposition over aluminum lines



(b) Dielectric oxide thickness variation

Fig. 2.7: Variation in ILD thickness due to different pattern density of interconnect lines

Contact and via size. Variations in the etching process and the dielectric thickness can affect the size of a contact or via. The etching depth can vary significantly depending upon the location of the contact or via, resulting in variations in the lateral opening size. These variations can significantly change the resistance of a contact or via.

2.3 Interconnect noise

The trend in next generation integrated circuit technology is towards smaller on-chip feature sizes and higher operating speeds. The density of the on-chip interconnect lines has increased together with the switching rate of the signals propagating through these lines, resulting in increased on-chip interconnect noise. Interconnect noise is primarily introduced by electromagnetic effects such as capacitive and inductive crosstalk between/among interconnects. The contribution of these effects to signal delay uncertainty along interconnect lines is discussed in this section.

2.3.1 Capacitive interconnect coupling

With the transition to deep submicrometer technologies, shrinking geometries have led to a reduction in the self-capacitance of interconnect lines, while coupling capacitances between the lines have increased as the lines are placed physically closer. In current processes, the coupling capacitance can be as high as the capacitance-to-ground [8]. Furthermore, trends indicate that the role of coupling capacitances will become more dominant as the feature size continues to shrink [29]. The capacitive components between three parallel lines on two different metal levels are shown in Fig. 2.8.

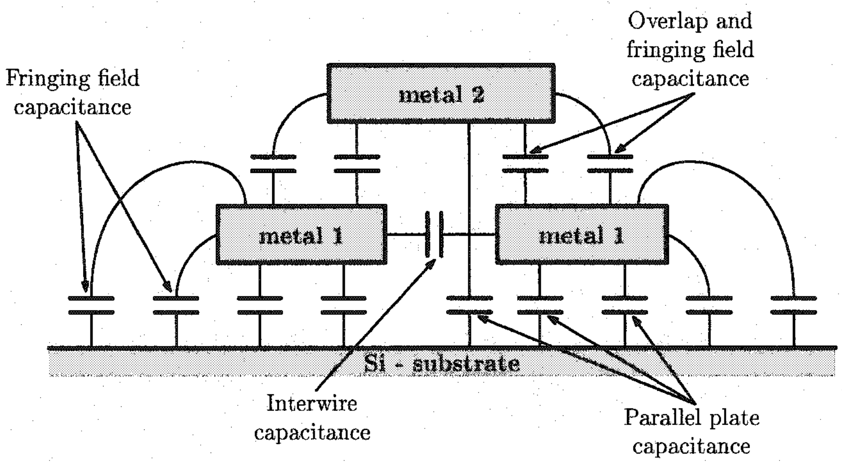

Overlap and fringing field capacitance
Fringing field capacitance
metal 2
metal 1
metal 1
Si - substrate
Interwire capacitance
Parallel plate capacitance

Fig. 2.8: Capacitive coupling among different interconnect layers

One of the important effects of coupling capacitances is unwanted voltage spikes in neighboring nets. A net (or wire) on which a switching event is generated is termed an *aggressor*, while a net (wire) on which that switching event produces a noise spike is referred to as a *victim*. This effect of coupling unwanted signals is known as *crosstalk*. Crosstalk can affect the behavior of circuits in one of two ways:

- Introducing unwanted noise on a quiescent line.
- Altering the delay of a switching transition.

In both of these cases, the switching of a line in a capacitively coupled net alters the effective capacitance of all of the other coupled lines.

Consider, for example, the simple capacitively coupled net shown in Fig. 2.9. The effective load capacitance driven by each of the CMOS inverters depends upon

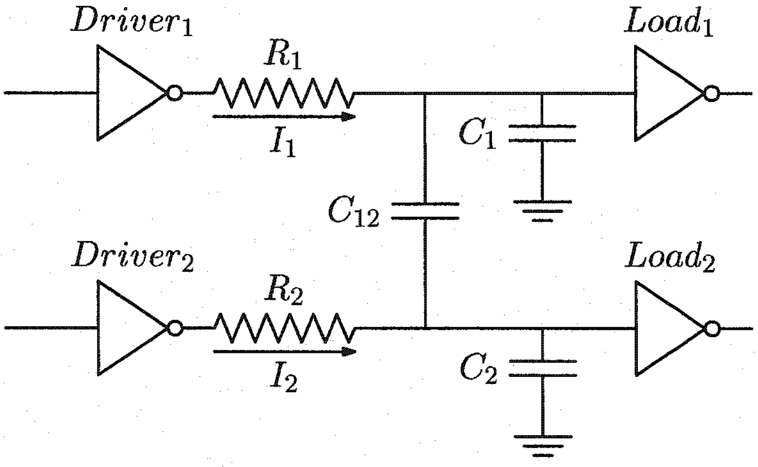


Fig. 2.9: Capacitively coupled net

the lines switching (i) in phase, (ii) out of phase, and (iii) one being active and the other remaining quiescent [7]. The uncertainty of the effective load capacitance due to the signal activity introduces uncertainty on the signal propagation delay within a coupled net. Further delay uncertainty is introduced by the coupling capacitance between two adjacent nets. This capacitance is proportional to the length along which the nets run close to each other [30]. Modeling and estimating the delay uncertainty due to crosstalk effects in large nets has therefore become a complicated and challenging process.

2.3.2 Inductance effects

Another effect that increases the deviation of the signal propagation delay from traditional interconnect models is inductance. While on-chip inductance has been relatively ignored in the past, with faster on-chip signal transition times and longer wire lengths, on-chip inductance has become increasingly important. Wide wires are frequently encountered in clock distribution networks and in upper metal layers. These wires are low resistance lines that can exhibit significant inductance effects. More accurate RLC models are therefore required for these global interconnect lines [9]. Furthermore, as performance requirements are accelerating the introduction of new materials (copper) for low resistive interconnect, inductance effects have become important for an increasing portion of on-chip lines.

Three different factors determine the inductance effects on an RLC line [9, 31]. These are (i) the signal impedance characteristics across the line, (ii) the transition time of the signal at the near end, and (iii) the time of flight of the signal propagating along the line. Based on these factors, the effect of inductance on the signal propagation delay increases as

- The attenuation of the signal along the line decreases (wider wires have less resistivity per unit length).
- The transition of the signal is faster (smaller driver resistance).
- The signal time of flight increases (longer wires).

The factors described above have been combined into a two-sided inequality [31] that determines the range of interconnect length for which inductance effects are important. This range is

$$\frac{t_r}{2\sqrt{LC}} \leq l \leq \frac{2}{R}\sqrt{\frac{L}{C}}, \tag{2.7}$$

where R, L, and C are the resistance, inductance, and capacitance of the line, respectively, l is the length of the line, and t_r is the transition time of the signal. This inequality is graphically illustrated in Fig. 2.10 [31].

Inequality (2.7) demonstrates that for long wire lengths, the RC time constant is sufficient due to high signal attenuation along the line. At intermediate

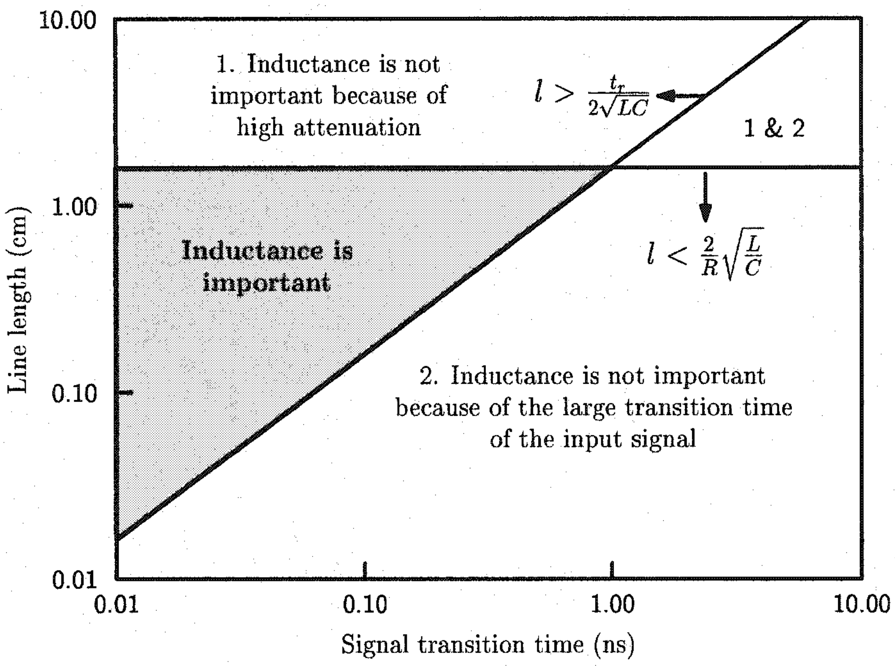


Fig. 2.10: Transition time (t_r) versus the length of the interconnect line (l). The shaded area denotes the region where inductance is important

wire lengths the ratio of the signal transition time over the signal time of flight determines the inductive behavior of a line. Lines with intermediate length are typical on many on-chip busses where the signal delay is affected by the mutual inductance among the lines. It is therefore important to accurately characterize the inductive behavior of these lines. For short wire lengths, inductance becomes important only for very fast signal transition times.

2.4 Variations in system parameters

In addition to the noise introduced by crosstalk and inductance effects, there are many other sources of delay uncertainty and noise originating at the system level or the surrounding environment of an IC. Such sources of uncertainty are power supply fluctuations (IR drops), electromagnetic interference (EMI), and system temperature. In this section, the effects of these noise sources on high performance digital circuits are discussed.

2.4.1 Power supply fluctuations - IR drops

One issue that has become an important source of delay uncertainty is the fluctuation of the power supply V_{DD} due to the resistance of the power distribution system, also known as *IR drops*. IR drops increase with larger currents flowing through the power grid. With narrow metal lines common in deep submicrometer (DSM) circuits, interconnect resistance has become a major factor in power distribution systems. Although wider lines are often used in power busses to reduce resistance, high operating frequencies require large buffers which draw large currents. When these currents flow from the power supply to the drivers, any resistance encountered in the power busses can cause the voltage to drop. This effect is shown in Fig. 2.11 [13]. The far end inverter shown in Fig 2.11 experiences a lower supply voltage than the initial inverter due to the IR drop along the supply line.

Since the power distribution network is composed of long interconnect lines running across a die, IR drops have a global effect within a die. The on-chip global structure that is primarily affected by IR drops is the clock distribution

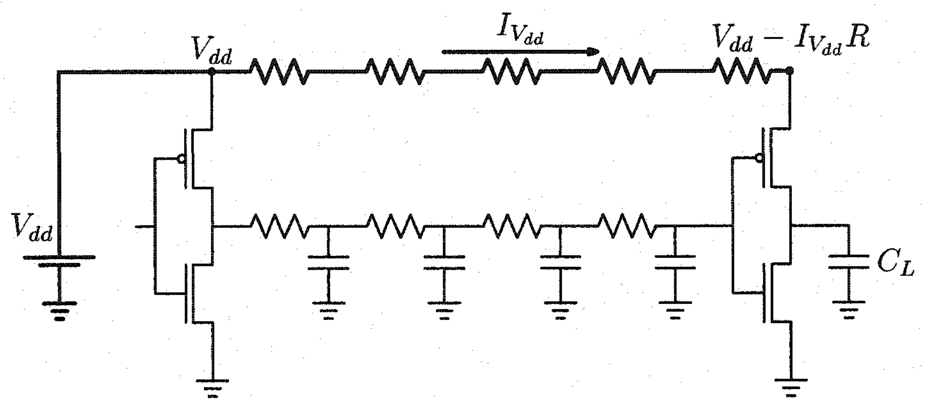


Fig. 2.11: Voltage drop due to the resistance within the power distribution network

network. Since *IR* drops effectively reduce the supply voltage, the clock buffers connected to the power grid provide less current, increasing the propagation delay of the clock signal.

Furthermore, voltage variations due to *IR* drops change transiently at each buffer as the clock signal changes. The current delivered to the buffers varies depending upon the instantaneous level of the *IR* drops. This transient fluctuation of the supply voltage causes variations in the clock delay, clock skew, and signal slew rates. As shown in [13], an *IR* drop can be translated directly into delay uncertainty. For example, a 10% voltage drop of the power supply can cause at most a 10% increase in delay. More accurately, a 10% *IR* drop increases the delay by 5% to 10%. The experimental results described in [13] demonstrate that an N% *IR* drop causes a delay change between N/2% to N%. This relation serves as a useful rule of thumb for quantifying the impact of *IR* drops on delay uncertainty.

In order to mitigate potential *IR* drops, wide metal lines are used in the topmost metal layers to reduce the interconnect resistance. In addition, the routing of these lines is often changed to reduce the transient *IR* drops. Other methods include the use of decoupling capacitors and ball-grid arrays. Key drawbacks of these advanced design techniques are the increased cost and complexity of the verification process for both the power and clock distribution networks.

2.4.2 Electromagnetic Interference (EMI) effects

With increasing operating frequencies and circuit densities, electromagnetic interference is becoming a crucial issue in the design of modern electronic systems. In particular, EMI has two distinct effects on digital devices [15]. The first effect is false switching or *static failure*, which occurs when the amplitude of the interference is sufficient to cause a change in the state of a static signal. The second effect is that of EMI-induced delays. It is shown in [15] that significant variations in the propagation delay of a device occur at much lower amplitudes of EMI than those that cause false switching. These variations lead to a violation of the critical timing constraints such as the minimum set-up and hold time of a flip flop. The violation of these constraints may cause *dynamic failures* which, unlike static failures, are dependent on the phase of the EMI with respect to the transition of a logic state [32].

When EMI is injected into a digital circuit, variations in the signal propagation delay are produced. This behavior is a result of EMI changing the time at which a logic transition crosses the switching threshold. The amount of EMI-induced delay can be positive or negative depending upon the phase of the EMI relative to the logic transition, as shown in Fig. 2.12 [15]. Note in Fig. 2.12 that at 0° and 180° of phase the effect of EMI is zero; therefore, no change in the propagation delay occurs. Also, for an EMI phase between 0° and 180°, the induced delay is negative, resulting in a decrease of the signal propagation delay. However, for an EMI phase between 180° and 360°, the induced delay is positive and the signal propagation delay increases.

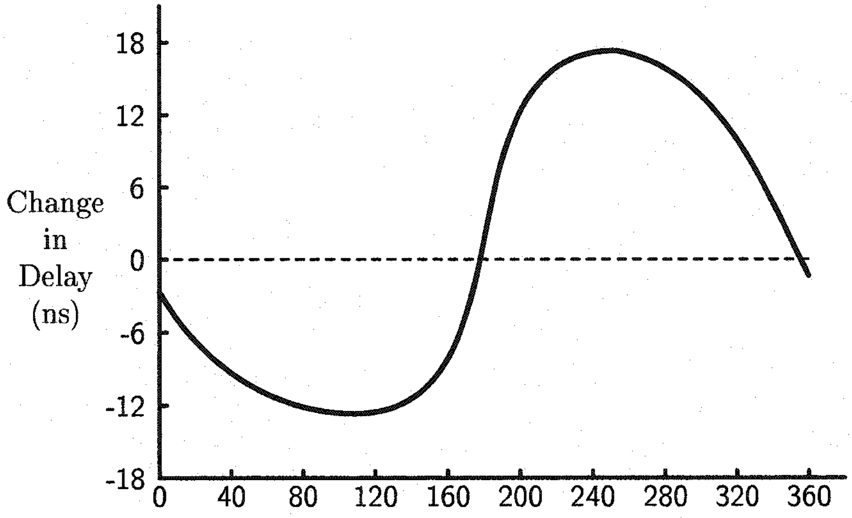


Fig. 2.12: EMI-induced delay versus EMI phase differences for a logic transition

Increased system immunity to EMI-induced delay uncertainty is achieved by increasing the *delay margins* within a system. The delay margins are defined in [15] as "the maximum allowable change in the timing of a signal transition for which a circuit will continue to operate reliably." The delay margins for a synchronous circuit depend upon the set-up (positive delay margin) and hold time (negative delay margin) constraints. As shown in [15], a larger delay margin leads to a greater immunity to EMI effects and for a given clock frequency, the immunity is maximized when both the positive and negative delay margins are equal.

2.4.3 Temperature variations

Temperature (T) variation is another factor that affects the performance of a circuit. Temperature variations have two primary components, the change in temperature due to the heat generated by the power dissipated on-chip and any changes in the ambient temperature. With increasing circuit densities and greater on-chip power dissipation, the temperature variations due to the heat generated on-chip has become significant. As the die size becomes larger, differences in the temperature occur across an IC. The effects of temperature on performance are therefore non-uniform across a die.

There are three primary circuit parameters that vary with temperature and introduce delay uncertainty [33]. These are the resistivity of the interconnect $\rho(T)$, the threshold voltage $V_{th}(T)$, and the carrier mobility $\mu(T)$. Temperature variations across an IC result in different buffer speeds and wire resistances, directly affecting the signal propagation delay. A simulation of the clock skew due to temperature variation is presented in [14]. The clock skew is based on an H-tree assuming a circular temperature gradient from the center of the IC to the edges. For a temperature variation of $\Delta T = 30$ K, an average increase in clock skew of 20% is demonstrated for a 0.25 μm CMOS technology.

2.5 Conclusions

The effects that introduce uncertainty in the signal propagation delay are presented in this chapter. The main sources of delay uncertainty within a CMOS device are variations in the effective channel length, gate oxide thickness, and threshold voltage. The uncertainty of the interconnect delay is caused by variations in the geometric parameters of a line due to imperfections in the manufacturing process. Noise from signal transitions produce crosstalk among interconnect lines which contributes to the delay uncertainty of the signals propagating along an interconnect line. Finally, variations in system level parameters introduce delay uncertainty to signals at different locations within a die. The effects of variations in the power supply voltage due to the resistance of the interconnect, electromagnetic interference, and temperature variations are specifically discussed.

Chapter 3

Performance Enhancements Through Clock Skew Scheduling

The effects that introduce delay uncertainty in various circuit elements are described in Chapter 2. As the device size is scaled and clock frequencies push deeper into the multi-gigahertz frequency levels, timing constraints have become much tighter and delay uncertainty has become increasingly significant. Deviations of the clock and data signals from the target delay can cause incorrect data to be latched within a register resulting in the system malfunctioning. The sensitivity of the circuit elements to these effects has therefore become an issue of fundamental importance to the problem of designing high speed digital integrated circuits.

Increasing the chip size and density adds to the on-chip power dissipation. High power dissipation penalizes the overall system since more advanced packaging and heat removal technology are necessary. Additionally, wider on-chip and off-chip power busses, larger on-chip decoupling capacitors, and more complicated power supplies are required. These factors increase the system size and cost. Furthermore, with the revolution of portable electronic devices, power dissipation has become a system performance metric, since the operation of these devices is limited by the battery life.

Design techniques and strategies that relax the tight timing constraints and reduce the on-chip power dissipation have been demonstrated on an industrial circuit and are presented in this chapter. To improve the timing margins of the data paths and the circuit speed, non-zero clock skew scheduling has been applied to specific circuit blocks of a high performance microprocessor. The application of this methodology is presented in section 3.1. In order to reduce the power dissipation, a technique that increases the delay of the non-critical data paths to exploit power savings has also been applied to this circuit and is discussed in section 3.2. Finally, some conclussions are reviewed in Section 6.3.

3.1 Improving the Timing Constraints and Speed

In this section, the effectiveness of the application of non-zero clock skew scheduling to improve performance and minimize the likelihood of race conditions is demonstrated. Background information about clock skew scheduling is presented in subsection 3.1.1. An algorithm to implement a non-zero clock skew schedule is discussed in subsection 3.1.2. Finally, the demonstration of the application of this technique on certain blocks of an industrial high performance microprocessor is presented in subsection 3.1.3.

3.1.1 Background on clock skew scheduling

A synchronous digital circuit is composed of a network of functional logic elements and globally clocked registers. Two registers, R_i and R_j, in a synchronous

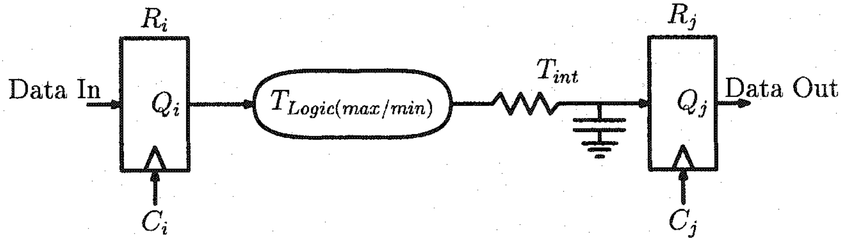


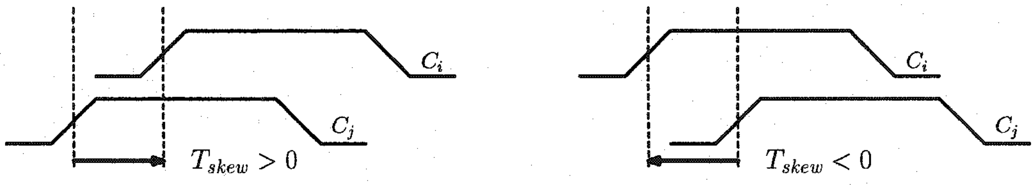
Fig. 3.1: A local data path.

digital circuit are considered sequentially-adjacent if there exists at least one sequence of logic elements and/or interconnect connecting the output of the initial register R_i to the input of the final register R_j. A pair of sequentially-adjacent registers together with a logic block and/or interconnect make up a local data path. A data path consisting of one or more local data paths is called a global data path. A local data path composed of two registers, R_i and R_j, driven by the clock signals, C_i and C_j, respectively, is shown in Fig. 3.1.

The difference in clock signal arrival times between two sequentially-adjacent registers is called *local clock skew* [1]. More specifically, given two sequentially-adjacent registers, R_i and R_j, the clock skew between these two registers is defined as

$$T_{skew} = T_{CD_i} - T_{CD_j}, \tag{3.1}$$

where T_{CD_i} and T_{CD_j} are the clock delays from the clock source to the registers, R_i and R_j, respectively. If the clock delay to the initial register T_{CD_i} is greater than the clock delay to the final register T_{CD_j}, the clock skew is described as positive. Similarly, if the clock delay to the initial register T_{CD_i} is less than the clock delay to the final register T_{CD_j}, the clock skew is described as negative. Waveforms exemplifying positive and negative clock skew for the local data path shown in Fig. 3.1 are illustrated in Fig. 3.2 [34].



(a) Positive clock skew

(b) Negative clock skew

Fig. 3.2: Examples of positive and negative clock skew.

The strategy of minimizing clock skew has been a central design technique for decades in synchronous digital circuit design methodologies. This technique is called *zero skew clock scheduling* and can be implemented in many different ways such as inserting distributed buffers within the clock tree [35], using symmetric distribution networks, such as H-tree structures [36] to minimize the clock skew, and applying zero skew clock routing algorithms [37, 38] to automatically layout high speed clock distribution networks. Zero (or minimum) clock skew scheduling techniques that require the clock delay from the clock source to each register of the system to be approximately equal, have been used in many high performance circuits. Intel Corporation applies a minimum clock skew methodology with localized tuning in the design of their latest microprocessors, including the Itanium^{TM}[†], the first processor in the Intel's IA-64 microarchitecture family [39, 40].

Further optimization of the circuit performance and reliability can be achieved by the application of *non-zero clock skew scheduling* in some (or all) of the local data paths, as described by Fishburn in [41]. The individual clock skew for each local data path is determined by satisfying specific timing relationships and conditions in order to minimize the system-wide clock period while avoiding all race

---

[†] Itanium^{TM} is a registered trademark of Intel Corporation

conditions. For the local data path from register R_i to register R_j, shown in Fig. 3.1, these timing relationships are listed below.

$$T_{CP} \geq T_{skew} + T_{PD_{max}}, \tag{3.2}$$

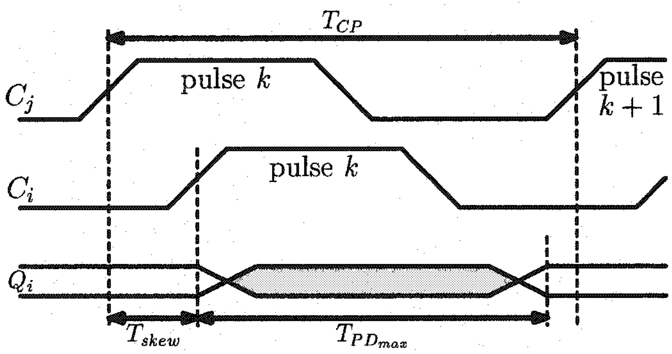
$$T_{PD_{min}} \geq T_{skew} + T_{hold}, \tag{3.3}$$

$$T_{PD_{max}} = T_{C-Q_i} + T_{Logic(max)} + T_{int} + T_{set-up}, \tag{3.4}$$

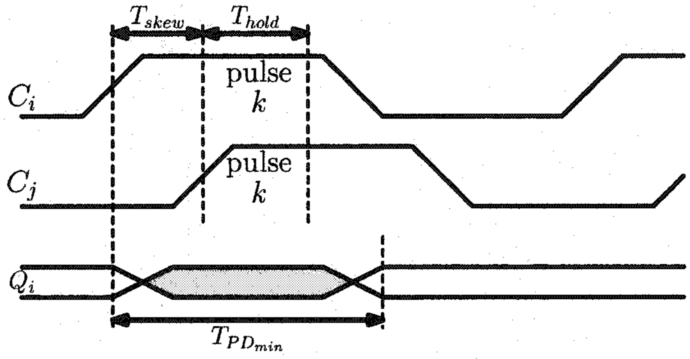
$$T_{PD_{min}} = T_{C-Q_i} + T_{Logic(min)} + T_{int} + T_{set-up}, \tag{3.5}$$

In the inequalities listed above, T_{skew} is the clock skew between registers R_i and R_j, as defined in (3.1). $T_{PD_{max}}$ ($T_{PD_{min}}$) is the maximum (minimum) propagation delay between registers, R_i and R_j, shown in (3.4) and (3.5), respectively. $T_{Logic(max)}$ ($T_{Logic(min)}$) is the maximum (minimum) propagation delay of the logic block between the registers R_i and R_j. T_{hold} is the time that the input data signal must be stable at register R_j once the clock signal changes state. T_{set-up} is the time required for the data signal to successfully propagate to and be latched within the register R_j. T_{C-Q_i} is the time required for the data signal to leave R_i once the register is enabled by the clock pulse C_i. T_{int} represents the temporal effect of the interconnect impedance on the path delay between the registers, R_i and R_j [42, 43]. T_{CP} is the minimum clock period.

From the timing inequalities, (3.2) guarantees that the data signal released from R_i is latched into R_j before the next clock pulse arrives at R_j, preventing *zero clocking* [41]. Also, (3.3) prevents latching an incorrect data signal into R_j by the clock pulse that latched the same data signal into R_i, or *double clocking* [41]. This race condition is created when the clock skew is negative and greater in magnitude than the path delay. If the clock skew is negative but smaller than the path delay, this effect can be used to improve circuit performance. This method



(a) To prevent zero clocking, $T_{CP} \geq T_{skew} + T_{PD_{max}}$



(b) To prevent double clocking, $T_{PD_{min}} \geq T_{skew} + T_{hold}$

Fig. 3.3: Preventing timing hazards in synchronous digital systems.

of improving performance is called *clock skew scheduling* [19, 34, 41, 44]. Timing relationships that prevent zero and double clocking are shown in Figs. 3.3(a) and 3.3(b), respectively.

For a given clock period T_{CP}, (3.2) and (3.3) determine a range within which each local clock skew T_{skew} can vary. This tolerance range is described here as the *permissible clock skew range* [43, 45] between the minimum permissible clock skew $T_{skew(min)}$ and the maximum permissible clock skew $T_{skew(max)}$, as shown in Fig. 3.4. The permissible clock skew range varies for different data paths since

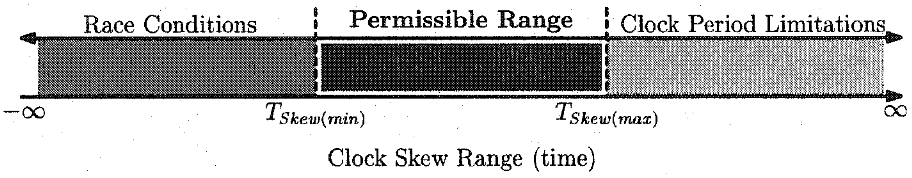


Fig. 3.4: Clock skew permissible range.

$T_{PD_{min}}$ and $T_{PD_{max}}$ depend on the delay characteristics of each local data path. $T_{skew(max)}$ is zero for those critical local data paths that limit the minimum clock period T_{CP} of the entire system. If a positive clock skew is applied to those paths the circuit speed is reduced.

The inequalities (3.2) and (3.3) are sufficient conditions to determine an optimal clock skew schedule, the associated minimum clock path delays, and the permissible range of the clock skew for each local data path. In this way, the minimum clock period is determined such that the overall circuit performance is maximized while eliminating any race conditions.

3.1.2 Optimal clock skew schedule algorithm implementation

The optimal clock scheduling problem has been described in [41] as a set of linear inequalities which can be solved with standard linear programming techniques. An algorithm for determining the minimum clock period based on the overlapping of permissible ranges of the clock skew between different data paths has been described in [43, 45] and is shown in Fig. 3.5. These concepts have been further enhanced, implemented as an algorithm, integrated into a software tool [19, 44, 46], and applied to a functional unit within a high performance microprocessor to determine an optimal clock skew schedule.

```
$T_{CP_{upper}} = max(T_{CP_{max}})$
$T_{CP_{lower}} = max(T_{CP_{max}} - T_{CP_{min}})$
**while** $(T_{CP_{max}} - T_{CP_{min}}) \geq \varepsilon$
    $T_{CP_{current}} = \frac{T_{CP_{upper}} + T_{CP_{lower}}}{2}$
    Determine the **local permissible skew range**
    Determine the **global permissible skew range**
    if ($\bigcap$ of local permissible skew range $\neq \emptyset$)
        $T_{CP_{possible}} = T_{CP_{current}}$
        $T_{CP_{upper}} = T_{CP_{current}}$
    else
        $T_{CP_{lower}} = T_{CP_{current}}$
end of **while** loop
$T_{CP_{optimum}} = T_{CP_{possible}}$
```

Fig. 3.5: The minimum clock period extraction algorithm

The development of a software tool to implement this optimum clock skew scheduling algorithm is described in [19, 43–45]. The input data to this tool are the minimum and maximum delays of each of the local data paths of the circuit. With this information, the software tool specifies an optimal clock skew schedule for the circuit; specifically, the minimum clock period that maximizes circuit performance and the associated clock path delays from the clock source to the individual registers that satisfy the target clock skew schedule. The steps of the implemented algorithm are as follows:

Step 1. A graph model of the circuit is produced that describes the input circuit C. Each vertex of the graph represents a register within C. Each arch of the graph connecting two vertices represents a local data path in C. There are two weights on each arch representing the maximum and minimum delays of the corresponding data paths.

Step 2. A current clock period for the circuit C is determined. The current clock period is the arithmetic mean of two bounding values. The upper bound is initially set equal to the maximum delay of all of the data paths belonging to C. The lower bound is initially set equal to the greatest difference between the maximum and minimum propagation delay of each local data path within C.

Step 3. Using the clock period specified from step 2, the permissible clock skew range is calculated from (3.2) and (3.3) for each pair of sequentially-adjacent registers in C.

Step 4. The permissible range of the clock skew of the global data paths is specified by the intersections of the permissible ranges of the local data paths calculated in the previous step. If the intersection is empty, no feasible clock skew schedule exists for the clock period specified in step 2.

Step 5. If a feasible clock skew schedule results from step 4, the algorithm iterates to step 2, and the current clock period specified in the previous iteration becomes the upper bound and is marked as a possible optimum solution. If a non-feasible clock skew schedule results from step 4, the algorithm iterates again to step 2 and the previously specified current clock period becomes the lower bound.

Iterations of the algorithm between steps 2 and 5 continue until the difference between the upper and lower bounds of the clock period is less than a specified positive number ε. The last clock period marked as a possible optimum solution is the minimum achievable clock period for the circuit C. The steps 2 to 5 implement the minimum clock period extraction algorithm shown in Fig. 3.5. Based on this clock period, (3.2), and (3.3) the clock skew between each pair of sequentially-adjacent registers within C is computed.

Step 6. The final step of the algorithm assigns the clock path delay to each of the registers within C. For each global data path, the individual clock delays from the clock source to the registers are calculated by first assigning the delay to the registers of the local data path with the largest clock skew value. The delays to the other registers are assigned by using the relative clock skew values among the remaining registers within the global data path.

The optimality of the solution depends solely upon the value of the constant ε that controls the number of approximating iterations executed by the algorithm. Reducing the value of ε reduces the distance between the minimum clock period determined by the algorithm and the minimum clock period set by (3.2) and (3.3). The choice of ε is a tradeoff between performance and the computational run time of the algorithm.

3.1.3 Experimental results from application of optimum clock skew scheduling

In a joint research project between the University of Rochester and Intel Corporation, the process of enhancing the speed and power dissipation [47] of an industrial circuit through the application of non-zero clock skew scheduling has

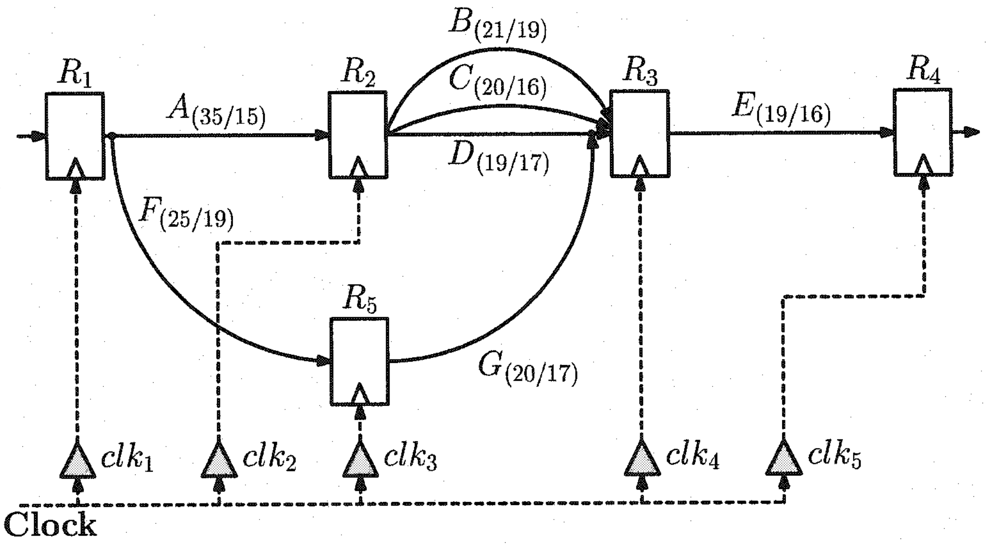


Fig. 3.6: Circuit graph of Itanium^{TM} FUB with normalized data path delays.

been investigated. Specifically, the application of clock skew scheduling to certain (highly tuned) functional blocks within a high performance microprocessor has been evaluated. It is shown here that the application of non-zero clock skew scheduling to these circuits yields a speed (or timing margin) improvement of up to 18% within the data paths of certain functional unit blocks (FUBs).

The clock scheduling tool described in Section 3.1.2 [19, 44] has been applied to specific FUBs within a high performance microprocessor. A circuit diagram of one of these FUBs is shown in Fig. 3.6 with normalized maximum and minimum local data path delays. All of the timing information in the following analysis is described in terms of these normalized path delays.

The initial clock period for the FUB shown in Fig. 3.6 is 35 tu (time units). By exploiting the differences in the maximum delays between data path A and the three parallel data paths, B, C, and D, the clock period can be reduced from 35 tu to 28 tu. This 20% performance improvement can be achieved through application of a negative clock skew of -7 tu to data path A by adding 7 tu to the clock path delay from the clock source to register R_2. In this case, the time available for the data signal to propagate along data path A is $T_{CP} + T_{skew} = 28 + 7 = 35$ tu. The time available for a data signal to propagate along the longest of the data paths between registers R_2 and R_3 (data path B) is 28 - 7 = 21 tu. Note that data paths F and G can also be synchronized by a clock period of 28 tu without violating any timing constraints. Thus, an approximately 20% improvement in circuit performance can be achieved by applying a non-zero clock skew schedule to this specific FUB.

Alternatively, the delay of 7 tu can be added to the clock signal arriving at register R_2 *without* reducing the clock period of the circuit. In this case the tight timing constraints of the critical data path A is relaxed significantly since additional 7 tu is available before the clock signal arrives at register R_2. The cost of this improvement is the reduction of the timing margins of the data paths B, C, and D by 7 tu. For the longest of these data paths (path B) the new timing margin is 7 tu which is long enough to compensate for a great amount of delay uncertainty.

The added delay to the path from the clock source to register R_2 is achieved by decreasing the size of the clock buffer (clk_2 shown in Fig. 3.6) that sources the clock signal that drives the register. This delay change is accomplished by replacing the clock buffer with a slower buffer from a predesigned cell library. In this way, the clock signal delay can be increased without requiring the redesign of the original clock buffer. Several different sizes of predesigned clock buffers that drive register R_2 have been evaluated. The variation of the clock signal delay to different clock buffer sizes is shown in Fig. 3.7

Buffer Number	Normalized buffer size	Normalized clock signal delay (tu)
1	1.00	24.93
2	1.43	20.14
3	1.71	17.79
4	2.05	16.27
5	6.07	10.90
6	10.47	9.67

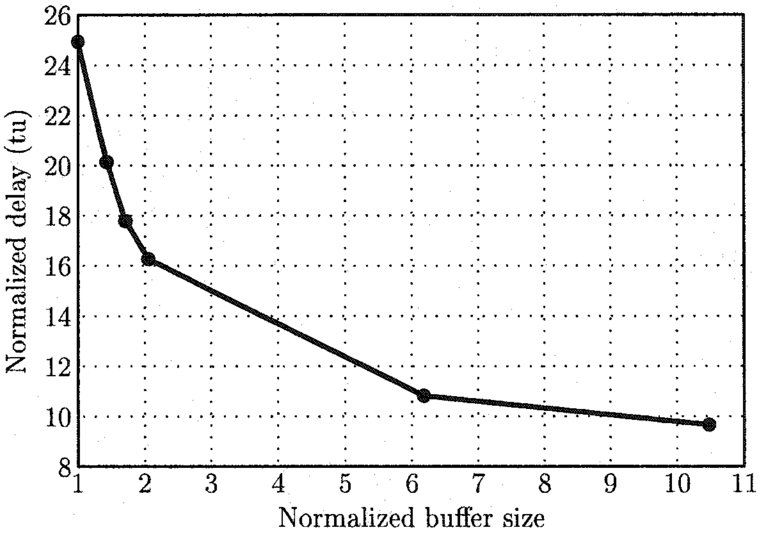


Fig. 3.7: Variation of clock signal delay to different clock buffer sizes.

As illustrated in Fig. 3.7, the clock delay from the clock source to a register is inversely proportional to the size of the clock buffer. This behavior is due to the increased output resistance of the smaller sized buffers, resulting in reduced current flow which introduces additional delay to the clock signal [48]. The clk_2 buffer that is initially used in the specific FUB (see Fig 3.6) is buffer No. 6 with a delay of 9.67 tu (see Fig 3.7). In order to produce an additional clock delay of 7 tu to drive register R_2, buffer No. 4 is used. The signal delay is $16.27 - 9.67 = 6.60$ tu, only a 5.7% error from the target value of 7 tu. The minimum clock signal period that is achieved with this clock skew schedule is 28.4 tu, producing an 18.8% improvement in speed.

Decreasing the size of the clock buffer in order to increase the delay of the clock line has an additional beneficial effect on the power dissipation, since the current flowing through the buffer is reduced. For the target circuit that contains the slower clock buffers, the power saving is approximately 1% of the total power consumed by this block.

3.2 Reducing Power in Non-Critical Data Paths

Two of the most popular techniques that are used to reduce power dissipation are supply voltage (V_{dd}) scaling and clock gating [49, 50]. V_{dd} reduction is an effective way for reducing power, since power dissipation is proportional to the square of V_{dd} as shown in (3.6).

$$P_{dyn} = C_{Load} V_{dd}^2 f, \tag{3.6}$$

The disadvantages of supply voltage scaling are effects such as sub-threshold and gate oxide leakage and increased sensitivity to noise [4]. Clock gating reduces the capacitance being switched by the clock distribution network [50]. The major disadvantages of clock gating are the increased complexity of the timing analysis and the increased transient currents when large blocks of logic are switched on and off.

Another technique to reduce the dissipated power is the use of smaller size circuit elements from predesigned cell libraries in order to achieve significant power savings. The smaller sized elements introduce smaller load capacitances albeit with a small delay penalty [49]. When this technique is applied to non-critical data paths, the delay penalty has no impact on the overall performance of the system. A demonstration of the application of this technique to an industrial circuit is presented here. It is shown that significant improvements in power dissipation can be achieved. Additionally, a methodology to expand this technique to slower (more critical) data paths is also discussed in this section.

The concept of the technique, the delay constraints, and the limitations in power savings are presented in subsection 3.2.1. The necessary conditions to apply this technique to slower data paths are described in subsection 3.2.2. Simulation results that demonstrate the power savings achieved on an industrial circuit are presented in subsection 3.2.3.

3.2.1 The general technique and related delay constraints

In a large high performance synchronous digital system, such as a microprocessor, the number of critical data paths is small as compared with the total number of data paths in the system. For example, in a specific system described in [19], less than 5% of the total data paths are within 20% of the maximum path delay

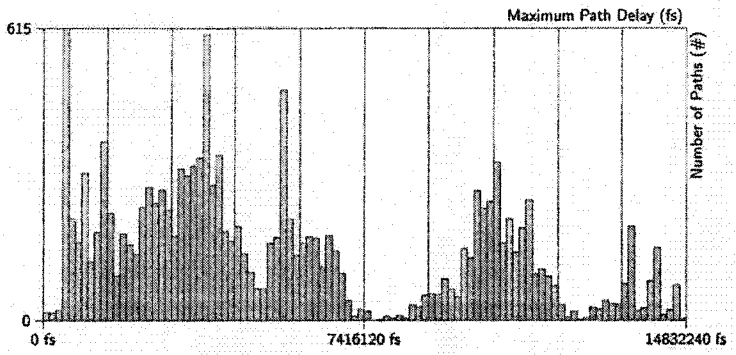
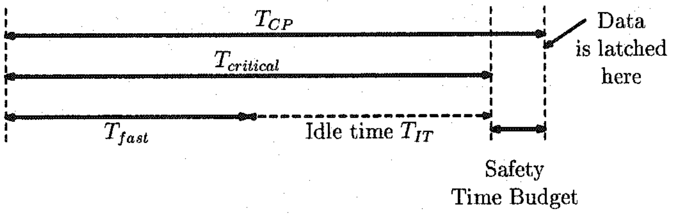


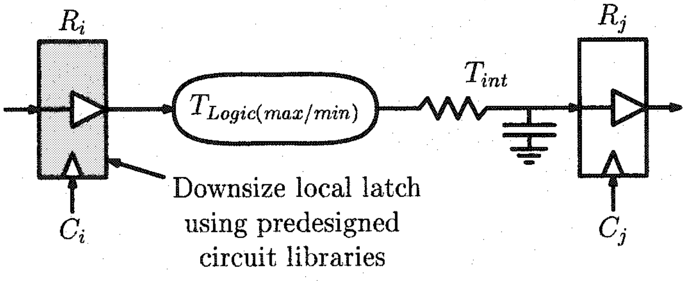
Fig. 3.8: Path delay distribution with zero clock skew scheduling

while more than 65% of the total data paths have path delays less than half of the maximum path delay. Alternatively, more than 65% of the local data paths are at least twice as fast as compared with the slowest local data paths. This distribution of data path delays is shown in Fig. 3.8. A similar distribution of path delays is common in the majority of high complexity circuits.

The fast data paths of a system are synchronized by the same clock signal that synchronizes the critical long data paths. Therefore, *idle time* (T_{IT}) exists in these short data paths since the data signal arrives at the final register well before the



(a) Short data path delay as compared to a critical long data path delay.



(b) A data path with a downsized latch to decrease the power of the fast data paths

Fig. 3.9: Increasing the delay of the fast data paths by downsizing the local latches that drive these paths.

clock signal arrives at the same register, as shown in Fig. 3.9(a). This idle time can be exploited to slow down these short data paths in order to save power. One way to accomplish this technique is by downsizing (*i.e.*, decreasing the geometric width) of the latch R_i that drives the data path as shown in Fig. 3.9(b), using smaller sized circuits from a predesigned cell library. By downsizing the latch the effective capacitance of the latch is decreased and the power required to drive the latch is reduced. Also, the geometric width of the output driver within the latch is decreased, thereby reducing the output current of the latch [48] and increasing the path delay. Therefore, this procedure results in a decrease in power consumption, albeit with an increase in the data path delay.

There are constraints, however, that limit the minimum size of an output driver and thereby the additional delay that can be introduced. One constraint is that the additional delay should not exceed the maximum permissible path delay constraint as shown in Fig. 3.10. The summation of the initial data path delay $T_{initial}$, the additional delay T_{add}, and the safety time budget T_{safe} should be less (or in the limit, equal) to the clock period T_{CP}.

$$T_{CP} \geq T_{initial} + T_{add} + T_{safe} \tag{3.7}$$

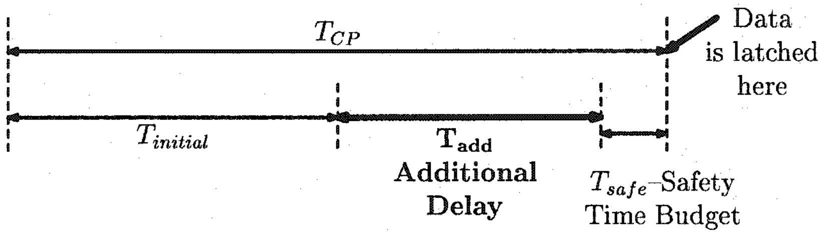


Fig. 3.10: The added delay of the fast data path should not violate the long path timing constraint.

Another constraint is that the introduction of smaller sized output drivers should not degrade the signal rise and fall times below some target level. Due to the reduced size of the output driver, the output signal transition time of the latch is slower, increasing the short-circuit power dissipation on the gates that are driven by the latch. The short-circuit power dissipation is due to the current that flows directly from the power supply to the ground of a CMOS gate when the input voltage is within the range V_{tn} and $V_{dd} + V_{tp}$ (when both the PMOS and NMOS transistors are on). When the transition time of the input voltage is longer, the time during which both transistors are on is also longer, increasing the short-circuit power dissipation. A close approximation of the short-circuit power dissipation is given by [51]

$$P_{SC} = \frac{1}{2} I_{peak} t_{base} V_{dd} f, \tag{3.8}$$

where I_{peak} depends on the size of the transistors of the driven gate, t_{base} is the input signal transition time, and f is the switching frequency of the input signal.

As shown by (3.8), as the size of the output buffer of the latch is decreased, the input signal transition time t_{base} increases, increasing the short-circuit power dissipated in the load gates. Therefore, there is a lower limit on decreasing the size of the output driver to achieve less power.

3.2.2 Application to critical data paths

The concept of slowing down fast data paths in order to save power can be further applied to slower, more critical data paths with the aid of *non-zero clock skew scheduling* [19, 41]. By applying negative clock skew to the slower, more critical data paths, the idle time in these data paths can be increased, permitting

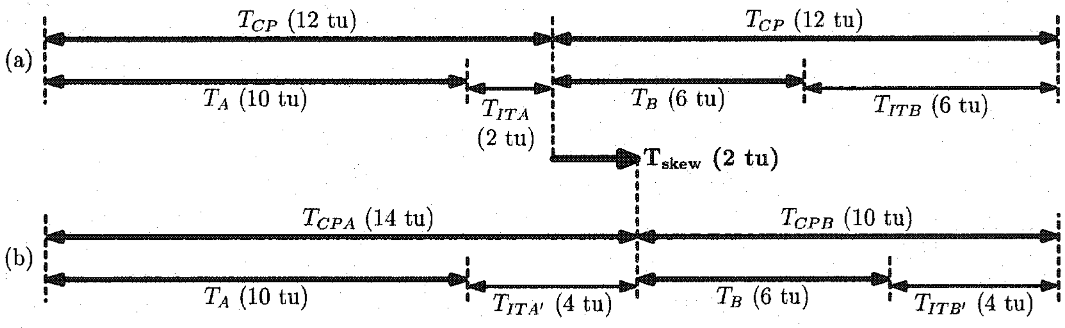


Fig. 3.11: Application of local clock skew to equalize the available idle time between the long and short delay data paths. (a) Initial timing of the data paths. (b) Timing of the data paths after the application of local clock skew

these paths to be slowed down further. However, there is one condition that must be satisfied for this concept to be feasible. This condition is that the data path that follows the slow data path should be sufficiently fast to satisfy the zero clocking timing constraint.

An example of the application of this concept to long data path delays is shown in Fig. 3.11. As shown in Fig. 3.11(a), data path A has a long delay of $T_A = 10$ tu and data path B has a short delay of $T_B = 6$ tu. The clock period of the system is $T_{CP} = 12$ tu. Because the delay of data path B is short as compared to the clock period, the clock signal that controls the latching operation of the register located between data paths A and B can be delayed by 2 tu, as shown in Fig. 3.11(b). This strategy delays the data signal propagating into data path B without creating any timing hazards, satisfying $T_{CPB} = T_{CP} - T_{skew} \geq T_B + T_{ITB'}$. Alternatively, delaying the arrival of the clock signal at the register delays the latching of the data signal that propagates into data path A, adding more idle time to data path A. Therefore, both data paths have sufficient idle time, permitting the drivers to be downsized so as to reduce the power dissipation of the overall circuit. If the slow data path A is not further slowed down, the application of negative clock skew can increase the safety margin of data path A which can be used to relax the strict timing constraints and make the circuit less sensitive to process parameter variations [45].

The approach presented above and illustrated in Figs. 3.11(a) and 3.11(b) provides an additional technique for saving power through the application of negative clock skew. The negative clock skew across data path A can be produced either by inserting a delay element along the clock signal path that distributes the clock signal to the final register of data path A, or by decreasing the transistor size of the clock buffer that drives this clock line. In the latter case, decreasing the size of the clock buffer results in less output current, which provides an additional savings in power.

3.2.3 Results on a demonstration circuit

The efficiency of this technique to decrease the power dissipation of non-critical data paths by changing the timing of these paths has been demonstrated on certain FUBs of a high performance industrial microprocessor. One of these FUBs is illustrated in Fig. 3.6.

The technique has been applied to the fast data paths, B, C, and D, of the FUB shown in Fig. 3.6. Each of these data paths is slowed down by downsizing the driving latch R_2 by using a different latch selected from a circuit library. The maximum and minimum delay of these data paths prior to and after decreasing

Table 3.1: Comparison between the original and the increased delay of data paths B, C, and D within the FUB illustrated in Fig. 3.6

Data path	Original max/min data path delay (tu)	Increased max/min data path delay (tu)	Increased Delay (%)
B	(21/19)	(25/21)	14.7
C	(20/16)	(25/20)	22.5
D	(19/17)	(25/21)	27.5
	Average Increase in Delay (%)		21.6

the size of the data path driver is listed in Table 3.1. It is shown in Table 3.1 that the delay of these data paths is increased on average by 21.6%.

The effect of downsizing the local latches that drive the data paths is to substantially reduce the power dissipated within the circuit block that contains these latches. As shown in Table 3.2, the total power dissipation of the circuit block is reduced by 82% by downsizing a total of 69 latches.

The remaining data paths within the FUB are unchanged. The effect of changing the latch on the data signal rise and fall times in data paths B, C, and D is

Table 3.2: Normalized power dissipation within the circuit block containing the latches

No optimization	Downsize latches w/o clock scheduling	Downsize latches w. clock scheduling
100	18	17.3

negligible. Also, no maximum data path delay constraint is violated since the larger maximum delay of the affected data paths (25 tu) is less than the maximum delay of the most critical data path (35 tu). Since the difference between the delay of data path A and the delays of the data paths, B, C, and D, is significant, the circuit performance can be improved with the application of non-zero clock skew scheduling, as described in Section 3.1. In this case, the clock period can be reduced to $T_{CP} = 25 + \frac{35-25}{2} = 30$ tu. The performance of the circuit can therefore be further enhanced by approximately 14%. Furthermore, the application of negative clock skew to downsize clock buffers results in an additional 4% decrease of the power dissipation as shown in Table 3.2. This decrease is due to the reduced capacitance of the clock buffer that drives the downsized latches.

3.3 Conclusions

Simulations of specific FUBs within a high performance commercial microprocessor demonstrate that improvements in the timing margin of the data paths can be achieved by applying non-zero clock skew. It is shown that in specific circuit blocks the timing margins can be increased by up to 18% by exploiting the differences in propagation delays between sequentially-adjacent data paths. The required clock delays from the clock source to the individual registers can be achieved by replacing the clock buffers that drive these registers with buffer cells from a predesigned cell library.

A non-zero clock skew scheduling software tool has also been developed [19, 46]. This tool has been evaluated on numerous industrial circuits [19], demonstrating the general utility of clock skew scheduling to improve the timing characteristics of a synchronous digital system.

A strategy for decreasing the power dissipation by reducing the size of the driving latches and increasing the delay of the non-critical data paths has also been demonstrated. The constraints, advantages, and disadvantages have been discussed. The application of non-zero clock skew scheduling to increase the idle time of the slower data paths has also been presented. Simulations on specific functional unit blocks within a high performance industrial microprocessor demonstrate that a substantial local power reduction of greater than 80% can be achieved by applying this strategy.

Chapter 4

Clock Tree Topological Enhancements to Reduce Delay Uncertainty

The effects that introduce uncertainty to the delay of a signal propagating within various circuit elements are described in Chapter 2. Chapter 3 discusses the application of non-zero clock skew scheduling to relax the timing constraints of the data paths and improve the system performance [19]. By relaxing the timing constraints of the data paths, the overall tolerance of a system to the delay uncertainty of the data signals is enhanced. Furthermore, the application of non-zero clock skew scheduling requires very accurate timing of the clock registers in a chip, therefore, the sensitivity of a system to the delay uncertainty of the clock signals increases. The delay of the clock signal from the clock source to a register should agree precisely with the target delay as specified by the non-zero clock skew schedule. Deviations of the clock signal from the target delay can cause incorrect data to be latched within a register, resulting in the system malfunctioning. Uncertainty of the clock signal delay is introduced by a number of factors that affect the clock distribution network, examples of which include process and environmental parameter variations [52, 53] and interconnect noise [54]. The sensitivity of a clock distribution network to these effects has therefore become an issue of fundamental importance to the synchronous design problem [34, 55, 56].

In this chapter, a polynomial time algorithm that improves the tolerance of a clock distribution network to delay uncertainty is presented. The algorithm focuses on the topological design of a clock distribution network and the manner in which the topology affects the sensitivity of the network to delay uncertainty. The algorithm extracts the clock tree topology based on the temporal criticality of the data paths. The concept of the algorithm is summarized in Section 4.1. The operation of the algorithm is described in Section 4.2. Incorporating non-zero clock skew scheduling within the topology extraction algorithm is discussed in Section 4.3. Experimental results for applying the algorithm to benchmark circuits that demonstrate topological enhancements are reviewed in Section 4.4. Finally, some conclusions are presented in Section 4.5.

4.1 Concept of the Algorithm

The most crucial effect of the uncertainty introduced in clock signal delays is the increased delay uncertainty between the arrival times of different clock signals that drive sequentially-adjacent registers connected by a combinational path. The more strict the setup and hold time constraints of a combinational data path, the more sensitive a data path is to delay uncertainty. A small difference in the clock signal delay can violate these constraints and cause a circuit to malfunction. Intuitively, the effects of process and environmental parameter variations (PEPV) on the common portion of the clock tree introduce identical delays to those clock signals driving sequentially-adjacent registers [19]. Alternatively, along the non-common portion of the clock tree, PEPV may introduce different clock delays and thereby cause a violation of the strict timing constraints of the critical data paths. This concept is illustrated in Fig. 4.1.

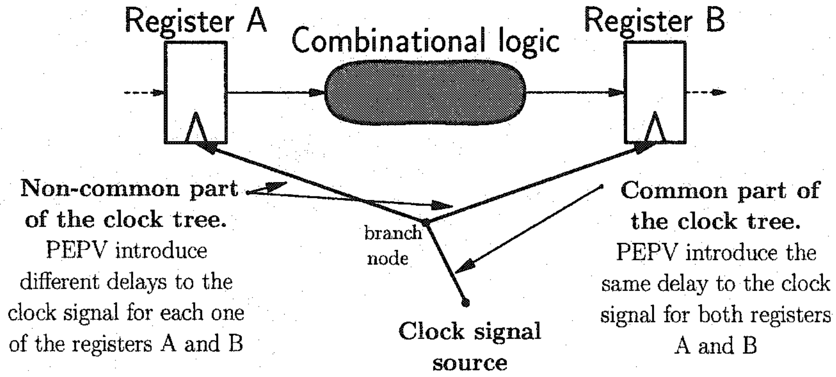

Register A
Combinational logic
Register B
Non-common part of the clock tree.
PEPV introduce different delays to the clock signal for each one of the registers A and B
branch node
Clock signal source
Common part of the clock tree.
PEPV introduce the same delay to the clock signal for both registers A and B

Fig. 4.1: Introduction of different clock signal delays to the non-common portions of the clock tree.

The clock tree topology (CTT) that specifies the hierarchy of the branch nodes within the clock tree can greatly affect the delay uncertainty introduced along the clock paths [43]. In particular, as the common portion of two paths in a clock tree increases, the delay uncertainty between the leaves of these paths is likely to decrease. The common portion of the two paths can be increased by separating these paths from a branch node *deeper* within the clock tree (closer to the leaf registers). The algorithm presented in this paper relies on this principle to generate a clock tree topology with improved tolerance to PEPV. The objective of the algorithm is to minimize the delay uncertainty of the sensitive data paths within a circuit. In this way, the overall tolerance of a circuit to delay uncertainty can be improved.

A synchronous digital circuit is represented in the algorithm as an edge-weighted graph G, which is called an *uncertainty graph.* An example of an uncertainty graph representation is shown in Fig. 4.2. Each node u in the graph G denotes a register in the circuit. Each edge $u \rightarrow v$ in G denotes a combinational path between the registers corresponding to u and v in the original circuit. The weight $w(u, v)$ of each edge represents the tolerance of the corresponding data path to PEPV which imposes a constraint on the delay uncertainty of the clock signals driving the registers u and v. In particular, for the circuit to function correctly, this uncertainty must not exceed $w(u, v)$. For example, the path corresponding to the edge 3→4 is critical, since the path can tolerate zero uncertainty in the clock delays from the clock source to the bounding registers. Alternatively, path 4→5 can tolerate up to 3 tu (time units) of delay uncertainty. Integer-valued time units are used in this example and in the rest of this chapter to improve the clarity of the presentation. In the implementation of the algorithm, the tolerance of the data paths to delay uncertainty is represented using real numbers.

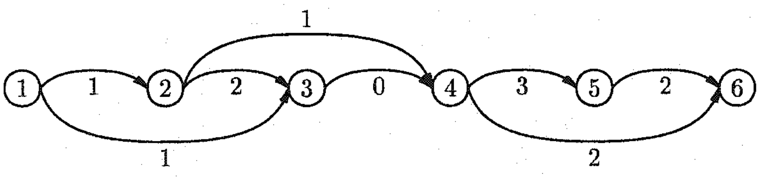


Fig. 4.2: Uncertainty graph representation of a circuit.

The algorithm relies on a topological delay uncertainty metric to generate clock trees that satisfy targeted uncertainty constraints. Specifically, given two paths, the delay uncertainty between the clock signals at the corresponding leaf nodes is assumed to be equal to the number of internal nodes within the tree (or branch nodes) in the non-common portions of the paths. The basic assumption underlying this metric is that as the number of non-common tree nodes between two paths increases, these paths share a smaller portion of the clock tree and the delay uncertainty between these paths therefore increases.

A clock tree topology that satisfies the delay uncertainty constraints of the graph shown in Fig. 4.2 is illustrated in Fig. 4.3. Consider, for example, the critical data path 3→4. This path can tolerate zero delay uncertainty between the clock signals arriving at the registers 3 and 4. The clock paths driving these registers split at the internal node 7 and arrive at registers 3 and 4. There are no non-common branch nodes between these two clock signal paths, therefore, these paths share the greatest portion of the clock tree. For data path 2→4, the clock paths split at the internal node 8. The number of non-common branch nodes of the two paths is one (node 7) which is equal to the delay uncertainty constraints for this data path. For data path 4→6, the paths from the clock source split at node 9, and the number of non-common branch nodes is two (nodes 7 and 8) which is also equal to the edge weight $w(4,6)$.

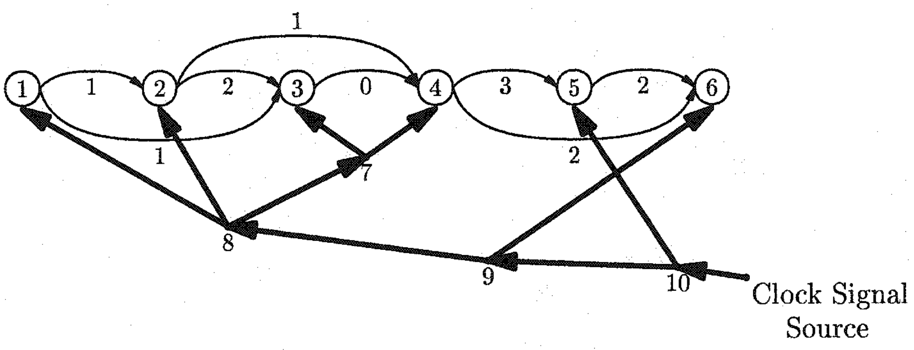


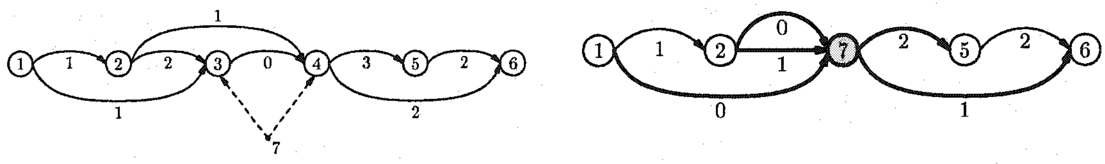
Fig. 4.3: Clock tree topology for the circuit shown in Fig. 4.2.

4.2 Description of the Algorithm

The algorithm presented in this section extracts the clock tree topology (CTT) by determining the hierarchy of the branch nodes of the tree such that the clocked elements of the most critical data paths share the greatest portion of the clock tree. The extracted topology can be further applied in the process of clock tree synthesis to generate a clock distribution network with improved tolerance to PEPV. Based on the information characterizing the topology of a clock tree, a physical layout can be developed which includes the interconnect, the placement of the branch nodes, and the location of the inserted clock buffers [34].

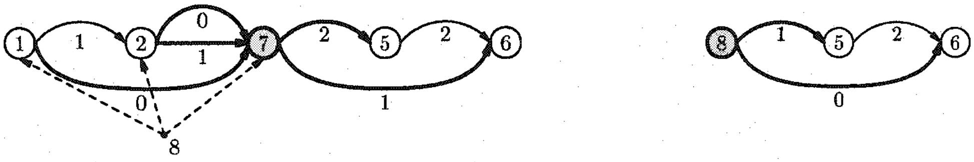
The algorithm starts by iteratively selecting from the uncertainty graph the registers of the data paths with the minimum tolerance to PEPV. These paths correspond to the edges with the minimum edge weight. In each iteration, a new branch node is introduced, and the clock signals are distributed from that node to the selected registers. These branch nodes replace the selected register nodes in the graph. The edges entering or leaving the replaced nodes are redirected to the introduced branch node, and the edge weights are adjusted to reflect the new tolerance of these edges. The algorithm continues until only one node remains in the graph. The clock tree topology that satisfies all of the uncertainty constraints is obtained by unfolding the computation to establish the connections among the hierarchically introduced branch nodes.

The execution of the proposed algorithm on the uncertainty graph shown in Fig. 4.2 is illustrated in Fig. 4.4. The algorithm starts with the graph shown in Fig. 4.4(a). The minimum-weight edge in this graph is between nodes 3 and 4. The clock signal is therefore distributed to these nodes from the new branch node



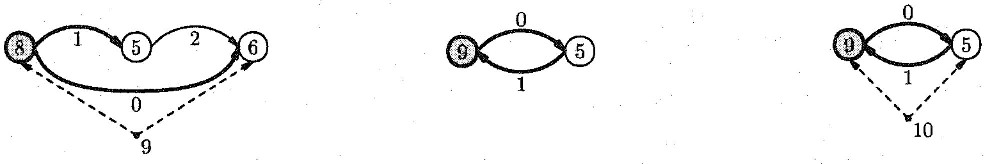
(a) **1st Iteration:** The clock signal is distributed to the registers of the most critical data path (3→4) from branch point 7.

(b) Branch point 7 replaces the nodes that it drives within the graph. The weights of the redirected arches (**bold**) are reduced by one.



(c) **2nd Iteration:** The clock signal is distributed to the registers of the most critical data path (1→7),(2→7) from branch point 8.

(d) Branch point 8 replaces the nodes that it drives within the graph. The weights of the redirected arches (**bold**) are reduced by one.



(e) **3rd Iteration:** The clock signal is distributed to the registers of the most critical data path (8→6), from branch point 9.

(f) Branch point 9 replaces the nodes that it drives within the graph. The weights of the redirected arches (**bold**) are reduced by one.

(g) **4th Iteration:** There is only one node (10) inserted in the graph. The iterations terminate.

Fig. 4.4: Iterations of the algorithm to reduce the input graph to a single node.

7. Node 7 is inserted in the uncertainty graph as shown in Fig. 4.4(b), replacing the selected nodes 3 and 4. The edges leaving or entering nodes 3 and 4 are redirected to node 7. The weights of these edges are reduced by 1 tu, the amount of uncertainty introduced by the branch node 7. The iterative application of this

basic procedure continues until only one node remains in the graph (node 10) as shown in Figs. 4.4(c) through 4.4(g). At this point, the algorithm extracts the final clock tree topology, which is shown in Fig. 4.3. Note that nodes 3 and 4 (corresponding to the most critical data path with zero tolerance to PEPV) share the greatest portion of the clock tree from the clock signal source to branch node 7. In the case of a less critical data path such as the path between nodes 4 and 6, the clock paths to the registers have in common a smaller portion of the clock tree.

The correctness of the CTT generation algorithm can be proved by an inductive argument showing that after each iteration, the portion of the clock tree that has been generated satisfies all relevant uncertainty constraints. The algorithm has polynomial complexity, terminating in $O(n^2)$ steps, where n is the number of nodes in the uncertainty graph. The number of iterations is n, and within each iteration, the number of updates is proportional to n.

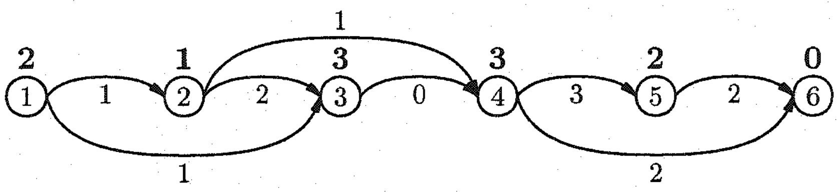
4.3 Incorporating non-zero clock skew scheduling

The ability to accurately control the clock signal delay from the clock source to the registers permits further improvements in the circuit performance and reliability. These improvements can be accomplished by the application of non-zero clock skew scheduling as described in [19, 41, 46, 57]. In non-zero clock skew scheduling, different arrival times of the clock signal at the registers are purposefully introduced, providing more time for the data signals to propagate along the slower data paths in a system. The differences in the arrival times are achieved by inserting delay elements (*e.g.*, active clock buffers) along the clock paths that drive the registers of sequentially-adjacent data paths. Several algorithms have been developed that generate an optimal clock skew schedule [19, 41, 43, 57]. The output of these algorithms is the value of the clock delay from the clock source to each of the registers which minimizes the clock period of a system or maximizes the timing margins of the most critical data paths.

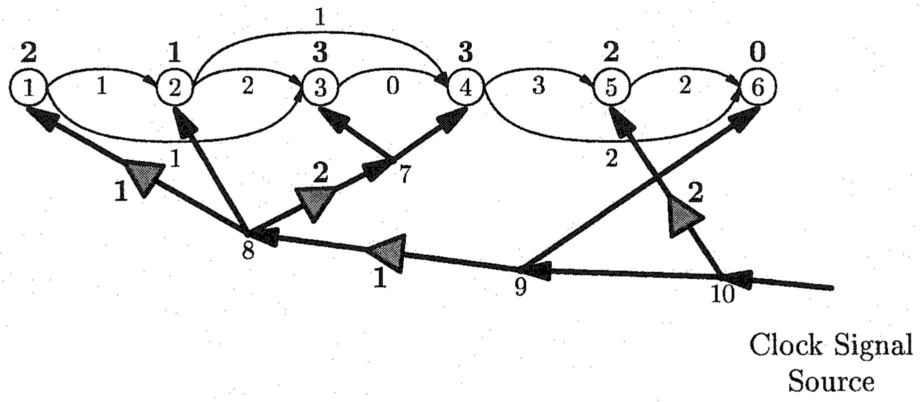
The clock tree topology extraction algorithm presented in Section 4.2 has been enhanced to incorporate non-zero clock skew scheduling. Given the clock delays to each of the registers, the algorithm extracts the clock tree topology for the circuit, including the assignment of the clock buffers to the branches of a clock tree. These clock buffers provide the specific clock signal delay in order to implement the target non-zero clock skew schedule. The synthesized clock tree topology enhances the overall tolerance of the clock distribution network to PEPV. The number of inserted clock delay buffers for this topology is also minimized.

The clock buffer insertion process is performed simultaneously with the construction of the clock tree topology, based on the uncertainty graph of the circuit. In this case, information that specifies the clock signal delay from the clock source to each of the registers is included in the uncertainty graph as shown in Fig. 4.5(a). The clock delay to each register is described by the numbers located above each graph node.

During the execution of the algorithm, each of the newly created branch nodes is assigned a clock signal delay equal to the *minimum* delay of the nodes that are replaced within the graph. An additional, higher than minimum delay is generated by assigning the appropriate number of buffers along the clock lines that drive these nodes. Therefore, the minimum delay of all of the branches



(a) Uncertainty graph containing clock signal delay information



(b) Clock tree topology including clock delay buffers

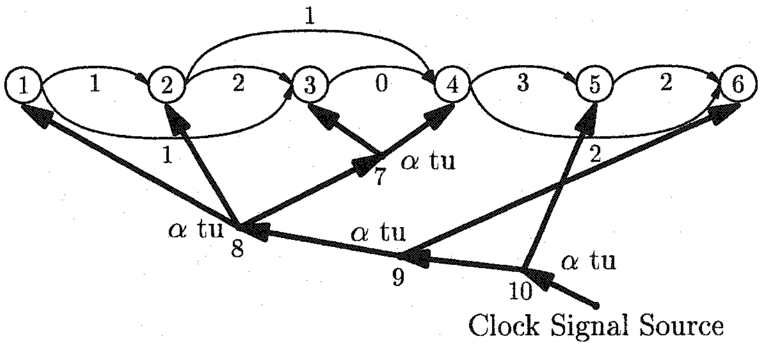
Fig. 4.5: Incorporating non-zero clock skew scheduling within the CTT algorithm

leaving a branch node is generated by a single buffer located prior to the branch node. In this way, the number of buffers required to implement a non-zero clock skew schedule is minimized. Additionally, the tolerance of a clock distribution network to PEPV is improved since the effects of PEPV on the common clock buffers prior to a branch node are the same for all of the branches following the branch node.

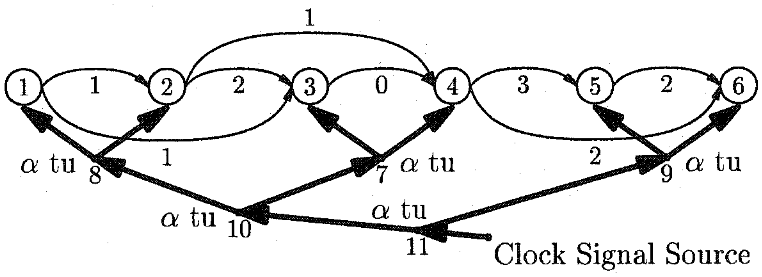
An example of an extracted clock tree topology with clock buffers is shown in Fig. 4.5(b). The number, written in bold next to each buffer, represents the amount of clock delay that is introduced by the buffer on the corresponding branch. Note in Fig. 4.5(b) that branch node 7 has a delay of 3 tu, the minimum clock delay from the clock source to nodes 3 and 4. Branch node 8, moreover, has a delay of 1 tu, the minimum clock delay from the clock source to nodes 1, 2, and 7. An additional clock delay of 2 tu from node 8 to node 7 is generated by assigning an appropriate delay buffer on the branch connecting these two nodes. An additional delay of 1 tu between node 8 and leaf node 1 is generated in a similar way.

4.4 Experimental results

The process in which the proposed algorithm reduces the non-common portion of the clock tree that drives the critical data paths of a circuit is illustrated in Fig. 4.6. The topology generated by the proposed algorithm is compared with a binary tree topology under the assumption that the delay uncertainty between the clock signals immediately following a branch node is constant and equal to α tu. In this case, as in the example shown in Fig. 4.6(a), the clock delay uncertainty for the data path $1 \rightarrow 3$ is 2α due to the branch nodes 7 and 8. In a binary clock tree as shown in Fig. 4.6(b), the corresponding delay uncertainty of the data path $1 \rightarrow 3$ is 3α due to the branch nodes 7, 8, and 10. A 33% reduction in delay uncertainty is therefore introduced to the clock signals that drive the data path $1 \rightarrow 3$. In the same way, the delay uncertainty for the data paths $2 \rightarrow 4$ and $4 \rightarrow 6$ is reduced by 33% and 25%, respectively. Therefore, the timing margins for those data paths in the circuit shown in Fig. 4.6(a) can be less strict than for the circuit shown in Fig. 4.6(b). In this example, the clock period is not decreased since the delay uncertainty for the most critical data path $3 \rightarrow 4$ is the same for both clock tree topologies.



(a) Algorithmically extracted clock tree topology for an arbitrary circuit



(b) Clock tree topology for the same circuit assuming a binary tree

Fig. 4.6: Comparison between the algorithmically extracted CTT and a binary tree.

The proposed algorithm has been tested on a number of benchmark circuits (see Table 4.1). In these tests, the average and worst case reduction in the range of delay uncertainty for a set of critical data paths has been determined. By reducing the delay uncertainty of these critical data paths, the overall timing constraints can be relaxed, thereby reducing the clock period and improving the overall circuit performance.

In evaluating the benchmark circuits it is assumed that the original clock tree topology is a balanced tree. The reduction in delay uncertainty is determined

Table 4.1: Reduction in delay uncertainty of the most critical data paths. BF describes the branching factor of the original binary tree.

Benchmark Files	Number of Registers	Avg. reduction (%) of delay uncertainty for a set of critical data paths				Delay uncertainty reduction (%) for the most critical data path			
		BF = 2	BF = 4	BF = 8	BF = 16	BF = 2	BF = 4	BF = 8	BF = 16
S27cp	20	33.3%	25%	0%	0%	66.6%	50%	0%	0%
S386	20	39.5%	18.7%	12.5%	0%	31.2%	12.5%	12.5%	0%
mm4a	23	72.9%	50%	37.5%	0%	72.9%	50%	37.5%	0%
S1196	46	83.3%	66.7%	50%	50%	83.3%	66.7%	50%	50%
S1238	46	60%	30%	0%	0%	58.3%	25%	0%	0%
mult16b	48	83%	66.7%	50%	50%	83.3%	66.7%	50%	50%
mult32a	66	84.5%	71.1%	58.9%	50%	85.7%	75%	66.7%	50%
S838_1	67	58.9%	35.7%	13.2%	5.9%	70.8%	50%	25%	0%
S953	68	66.4%	45.2%	23.8%	21.4%	66.7%	50%	0%	0%
S641	77	84.8%	71.8%	60.4%	50%	83.3%	66.7%	50%	50%
sbc	120	84.3%	70.2%	57.1%	50%	83.3%	66.7%	50%	50%
S9234	209	82%	66.3%	52.1%	44.6%	82.4%	65.6%	52%	43.7%
S5378	246	84.2%	70.3%	58.1%	48.8%	85.7%	75%	66.7%	50%
S38584_1	446	90%	80%	75%	66.7%	90%	79.9%	75%	66.7%
diffeq	454	87.7%	77.6%	65.6%	60.6%	88.8%	80%	66.7%	66.7%
dsip	644	88.7%	79.4%	66.7%	64.8%	88.7%	79.4%	66.7%	64.9%
bigkey	683	88.7%	79.4%	66.7%	64.8%	88.9%	80%	66.7%	66.7%

for four different branching factors (BF) of the balanced tree. The branching factor of a tree is the number of branches leaving from a branch node within the tree. The results of these experiments are listed in Table 4.1. It is shown that the delay uncertainty of the critical data paths can be reduced by up to 90%. Note in Table 4.1 that the smaller the branching factor of the original clock tree topology, the greater the reduction in delay uncertainty. This behavior is caused by the increased depth of the clock tree when the branching factor is small. As a tree becomes deeper, the clock paths that drive the leaf registers have a smaller common portion of the clock tree and therefore an increasing delay uncertainty. When the branching factor increases, the depth of the clock tree is reduced, permitting the clock paths to share a greater common portion of the clock tree.

As shown in Table 4.1, in certain smaller circuits the reduction in delay uncertainty is zero when the branching factor is high. In these specific cases, the clock paths driving the registers of the critical paths already share the greatest common portion of the clock tree, therefore a decrease in delay uncertainty is not possible. Alternatively, the larger circuits have a deep clock tree even in those cases where the branching factor is high. A significant reduction in delay uncertainty in these circuits is shown for all of the branching factors.

4.5 Conclusions

A methodology for generating a clock distribution network with high tolerance to process and environmental variations is presented. An algorithm that extracts a clock tree topology in order to minimize the delay uncertainty of the clock signals that drive the most critical data paths is presented. The hierarchy of the branch nodes of the clock tree is determined such that the clocked elements of the most critical data paths share the greatest portion of the clock tree. Algorithmic enhancements that incorporate non-zero clock skew scheduling are also presented. Experimental results from the application of the algorithm to benchmark circuits demonstrate significant improvements in the tolerance of a circuit to process and environmental variations.

Chapter 5

Buffer Sizing for Reduced Delay Uncertainty

Delay uncertainty can have a critical effect on the operation of high speed synchronous circuits. Uncertainty in the propagation delay of the clock signal can cause violations in the set-up and hold time constraints of the data path registers [19]. In addition, delay uncertainty of the data signals can also cause similar violations. To prevent these violations, either the tight timing constraints should be relaxed, or the uncertainty in the signal delay should be reduced. Relaxing the timing constraints, particularly the most critical paths of a circuit, increases the clock period, reducing the circuit performance. Alternatively, reducing delay uncertainty can prevent the violations of these timing constraints, thereby improving the robustness of a circuit and enhancing circuit performance. In order to develop design methodologies that reduce delay uncertainty, those effects that cause uncertainty in the signal propagation delay should be investigated. A brief review of these effects is presented below.

Delay uncertainty is introduced by a number of factors that affect the signal propagation delay, such as process and environmental parameter variations [12, 33, 52] and interconnect noise [54, 58]. Effects such as the non-uniformity of the gate oxide thickness [27] and imperfections in the polysilicon etching process [26] can cause variations of the current flow within a transistor, thereby introducing delay uncertainty. Environmentally induced parameter variations caused by changes in the ambient temperature [14] and external radiation [15] also introduce delay uncertainty. On-chip noise due to interconnect crosstalk [7] introduces additional delay uncertainty as the interconnect length increases and the wire-to-wire spacing becomes shorter.

Significant research effort has therefore focused on characterizing and reducing delay uncertainty [59]. A primary research target is the statistical characterization [60] of process parameter variations and delay uncertainty in order to specify a minimum time for synchronizing high speed circuits [61]. In addition, design methodologies for clock distribution networks [55] have been developed to reduce uncertainty in the clock signal delay [19, 45, 62]. A wire sizing strategy that reduces the sensitivity of the clock signal delay to interconnect parameter variations has been introduced in [63]. A methodology that extracts the topology of a clock distribution network to reduce the delay uncertainty of the clock signal at the most critical data paths is presented in [64]. Additionally, noise effects due to crosstalk among interconnect lines have been investigated [65, 66]. The focus of the research described in [65, 66], however, is primarily on the effect of noise as voltage and current level variations rather than uncertainty in the signal propagation delay.

In this chapter, the delay uncertainty of a signal propagating along a CMOS inverter due to process, environmental, and system parameter variations is investigated. In addition, the effects of interconnect crosstalk on delay uncertainty are discussed. It is shown that increasing the buffer size reduces the uncertainty in the signal delay of both the buffer and the interconnect. Variations in the current of an inverter due to device parameter variations are discussed in Section 5.1. It is shown that increasing the inverter size reduces the delay uncertainty. The effect of interconnect crosstalk on the delay is presented in Section 5.2. Delay uncertainty caused by coupling among the lines and can be reduced by increasing the buffer size. In section 5.3, the effect of increased buffer size on power dissipation is described. A new power efficiency metric is introduced in Section 5.3 to characterize the tradeoff between the reduction in delay uncertainty and the increase in power dissipation. Finally, some conclusions are presented in Section 6.3.

5.1 Delay uncertainty due to device parameter variations

The signal propagation delay along two points X and Y in a circuit is the time interval between the 50% point of the signal transition at point X to the 50% point of the signal transition at point Y [19]. In the case that the points X and Y correspond to the input and output of a logic gate, the signal delay from X to Y is the propagation delay of a gate. The signal propagation delay of a logic gate depends upon the gate size and the magnitude of the current flowing within the gate. Variations of the device parameters can cause the magnitude of the current flow to change, introducing delay uncertainty. Therefore, reducing the effect of the device parameters on the current flow can reduce delay uncertainty. In particular, the delay uncertainty of the active components of the most critical data paths and interconnect lines within a circuit should be reduced to prevent tight timing constraints from being violated.

In the case of a CMOS inverter, a first order relationship between the signal propagation delay and the current through an inverter is presented in [67] as

$$t_D = \left(\frac{1}{2} - \frac{1 - v_T}{1 + \alpha}\right) t_T + \frac{C_L V_{DD}}{2 I_{D0}}, \tag{5.1}$$

where

$$v_T = \frac{V_{TH}}{V_{DD}}, \tag{5.2}$$

and t_D is the signal propagation delay, α is the velocity saturation index described in [67], t_T is the transition time of the signal at the input of the inverter, V_{DD} is the supply voltage, and C_L is the capacitive load at the output of a CMOS inverter. V_{TH} is the threshold voltage of the active transistor during a signal transition and I_{D0} is the drain current flowing through that transistor (defined at $V_{GS} = V_{DS} = V_{DD}$). I_{D0} is often used as an index of the current drive of a MOSFET transistor and depends upon the transistor size,

$$I_{D0} = \frac{W}{L_{eff}} P_d, \tag{5.3}$$

where W and L_{eff} represent the width and the effective channel length of a transistor, respectively, and P_d represents the effect of the device parameters on the current flow.

In order to evaluate the effect of the variation of the device parameters on the inverter propagation delay, I_{D0} is substituted from (5.3) into (5.1), yielding

$$t_D = \left(\frac{1}{2} - \frac{1 - v_T}{1 + \alpha}\right) t_T + \frac{C_L V_{DD}}{2 \frac{W}{L_{eff}} P_d}. \tag{5.4}$$

Differentiating (5.4) with respect to P_d,

$$\frac{\partial t_D}{\partial P_d} = -\frac{C_L V_{DD}}{2 \frac{W}{L_{eff}} {P_d}^2} . \tag{5.5}$$

According to MOSFET transistor models first proposed by Shockley in [24] and Sakurai and Newton in [67], the primary factors that affect the drain current within a transistor are the carrier mobility (μ_o), the gate oxide capacitance (C_{ox}), the threshold voltage (V_{TH}), and the power supply voltage (V_{DD}). The dependence of these factors on process, environmental, and system parameters is shown in Fig. 5.1. These parameters are temperature ($Temp$), gate oxide thickness (t_{ox}), substrate doping density (N_A), and power supply voltage (V_{DD}). A variation of these parameters causes a change in the magnitude of the current flow within a transistor and, eventually, uncertainty in the propagation delay.

As shown in (5.5), the sensitivity of the delay of an inverter on variations of the device parameters $\frac{\partial t_D}{\partial P_d}$ is inversely proportional to the inverter size. The delay uncertainty due to variations in the drain current can be reduced by increasing the size of the inverter. In addition, as shown in (5.4), the propagation delay of an inverter also decreases with increasing inverter size. The reduction in delay is shown

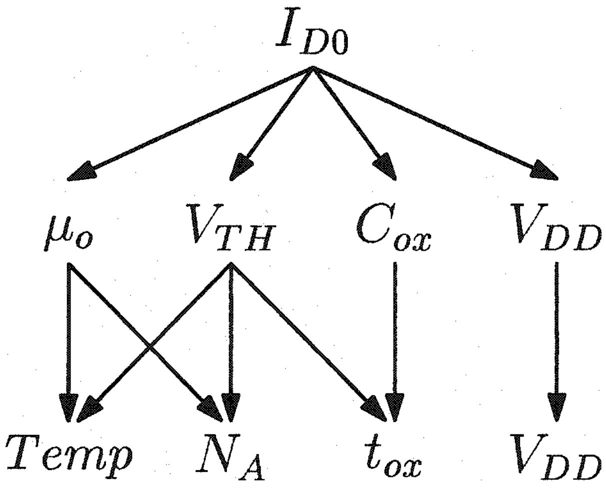


Fig. 5.1: Dependence of device parameters on process, environmental, and system parameters

in Fig. 5.2 where the inverter size is increased by up to five times. The inverter delay is calculated analytically from (5.1) and simulated using Spectre®simulator* [68]. The difference between the calculated and simulated delay values is caused by only I_{D0} of the active transistor during a signal transition being considered. The short-circuit current flowing through the inactive transistor is ignored and therefore the calculated delay is smaller. The delay reduction, however, is similar in both the calculated and the simulated propagation delay, as shown in Fig. 5.2.

The effect of device parameter variations on the inverter propagation delay is shown in Fig. 5.3(a) for a high-to-low signal transition. Note that the variation of different device parameters has a different effect on the delay uncertainty. Increasing the buffer size reduces the sensitivity of the delay, as described by (5.5). As shown in Fig. 5.3(a), the variation of the power supply voltage (V_{dd}) is the dominant effect that introduces delay uncertainty, followed by variations in the gate oxide thickness (t_{ox}) and temperature (*temp*). In addition to these parame-

---

*Spectre®is a registered trademark of Cadence Design Systems Incorporated.

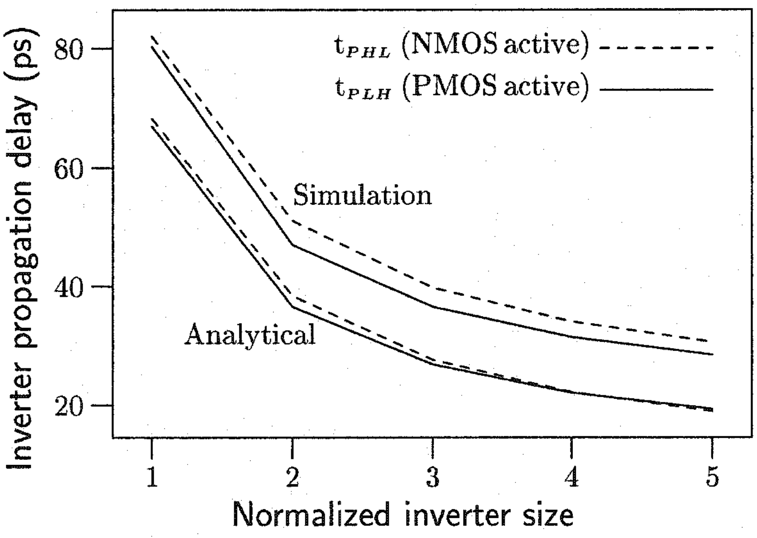
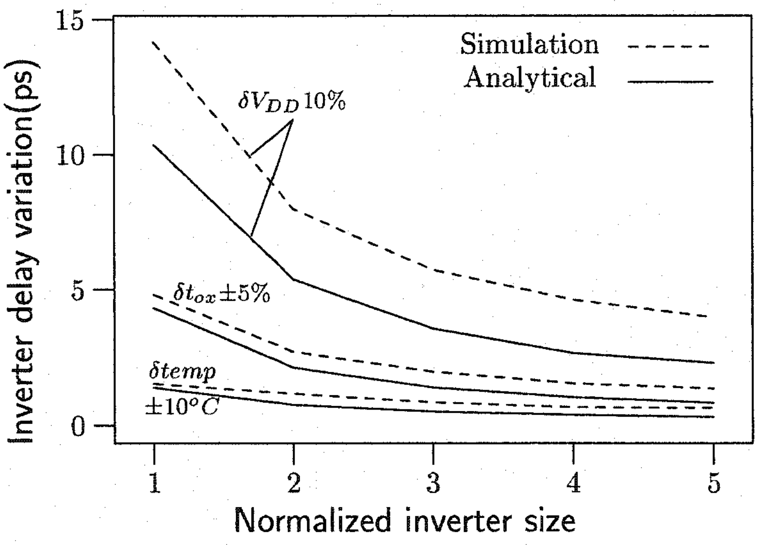
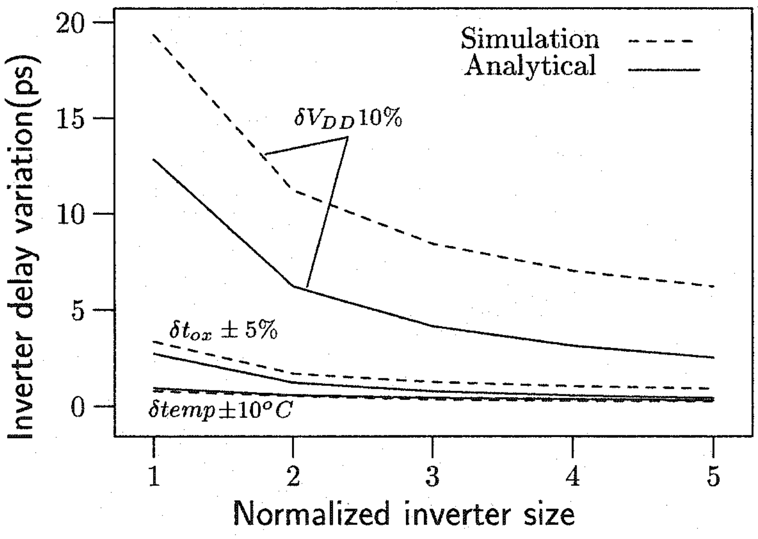


Fig. 5.2: Reduction in delay with increasing inverter size



(a) Delay uncertainty due to device parameter variations for a high-to-low signal transition



(b) Delay uncertainty due to device parameter variations for a low-to-high signal transition

Fig. 5.3: Uncertainty in inverter delay due to process, environmental, and system parameter variations

ters, the variation of the channel doping concentration (N_A) by ±5% has also been investigated, where the delay uncertainty is shown to be negligible. The case of a low-to-high signal transition is shown in Fig. 5.3(b), where the delay uncertainty is introduced for the same variations in device parameters.

As shown in (5.1), the propagation delay of a signal transition also depends upon the transition time t_T of the input signal. The effect of a variation in t_T on the delay uncertainty is illustrated in Fig. 5.4, where the delay uncertainty due to a variation of V_{dd} of 10% is shown for a variation of t_T between 30 ps and 80 ps. Increasing the input signal transition time increases the delay uncertainty; however, the effect of increasing the inverter size is dominant, and the overall delay uncertainty is decreased. Similar delay uncertainty trends are shown in

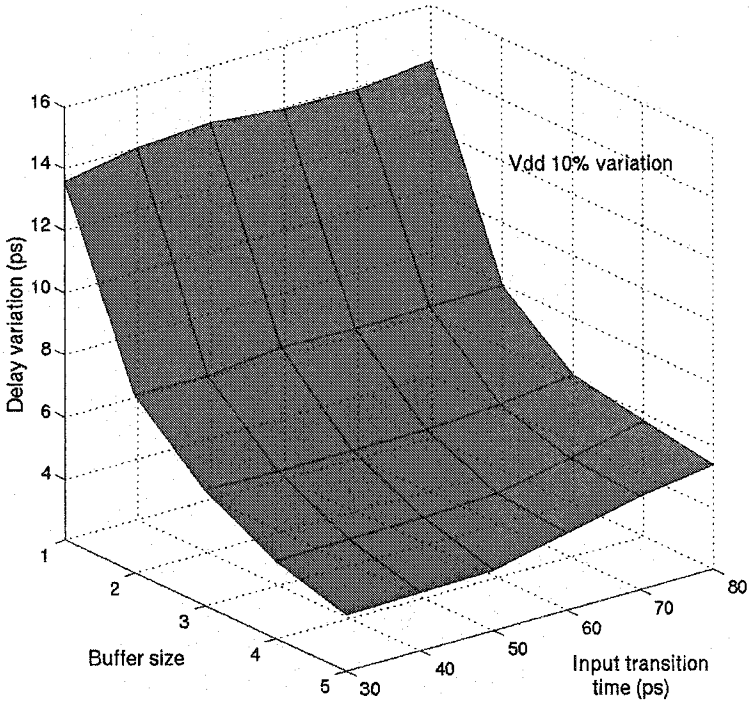
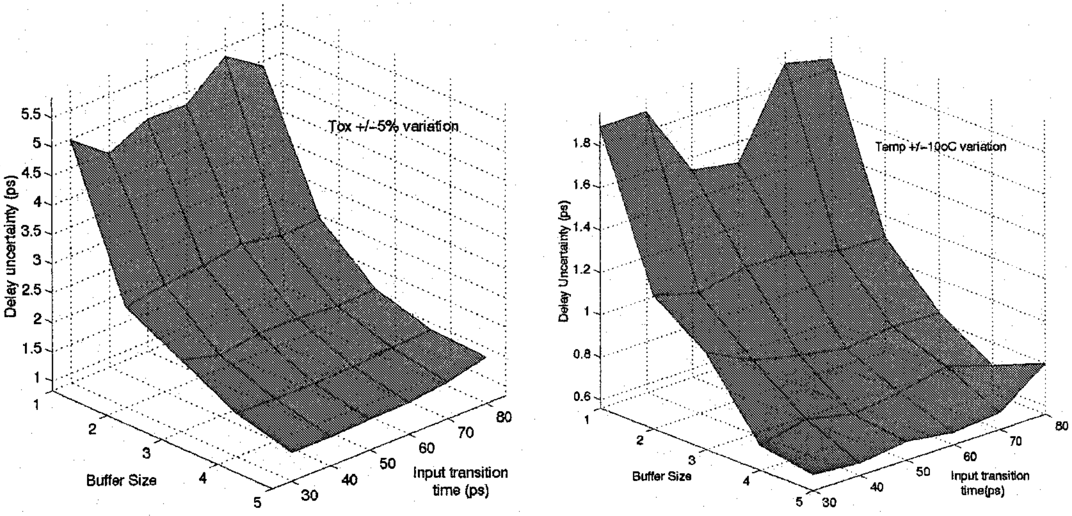


Fig. 5.4: Delay uncertainty due to 10% V_{dd} variation for different input transition times

Figs. 5.5(a) and 5.5(b) for a ±5% variation in t_{ox} and a $\pm 10^o$ C variation in temperature, respectively.



(a) Delay uncertainty due to ±5% t_{ox} variation for different input transition times

(b) Delay uncertainty due to $\pm 10^o$ C variation in temperature for different input transition times

Fig. 5.5: Effect of input signal transition time t_T on delay uncertainty

5.2 Delay uncertainty due to interconnect crosstalk

On-chip feature size scaling and increased circuit density have aggravated the importance of crosstalk among interconnect lines [7,66]. Scaling of the on-chip feature size reduces the routing distance between the interconnect lines. In addition, the wire aspect ratio increases with each technology generation. These effects result in a significant increase in the coupling capacitance and crosstalk among interconnect lines. Increased circuit density and functionality increases the number of interconnect lines that are closely routed together, thereby increasing the crosstalk and the design complexity of a coupled interconnect system.

With increasing interconnect length, both the propagation delay of the interconnect and the delay uncertainty of a signal propagating along a global wire become significant. One of the most critical global signals is the clock signal that is distributed across the entire die. Due to the global nature of a synchronous clock distribution network, crosstalk effects from different sources across a die can introduce significant uncertainty to the clock signal delay. Reducing delay uncertainty in a clock distribution network is therefore a challenging task, particularly for those clock paths that drive the registers of the most critical data paths. In order to control the signal delay on these networks, interconnect buffers and signal repeaters [20] are inserted along the global interconnect structures. In this section, the effect of changing the size of these buffers on the interconnect crosstalk is investigated.

Capacitive coupling among interconnect lines introduces uncertainty in the effective capacitive load of the buffers driving these lines. The variation in the effective load creates uncertainty in the signal propagation of the buffers as well as the interconnect. The delay uncertainty of a buffer due to variations in the effective capacitive load can be determined by differentiating (5.1) with respect to C_L,

$$\partial t_D = \frac{V_{DD}}{2I_{D0}} \partial C_L. \tag{5.6}$$

As shown in (5.6), increasing the buffer size (*i.e.*, increasing I_{D0}) reduces the effect of the capacitive load variation on the signal propagation delay.

An example of a pair of capacitively coupled interconnects is shown in Fig. 5.6. The propagation delay from point A to point C along the *victim line* shown in Fig. 5.6 is affected by the switching of the capacitively coupled *aggressor line.*

The signal delay on the victim line can be divided into two parts: the buffer delay from point A to point B as shown in Fig. 5.6, and the interconnect delay between points B and C. The uncertainty of the signal transition of the aggressor line introduces delay uncertainty in the signal propagating along the victim line. The possible switching activities of the aggressor line during a signal switch on the victim line are

i) switch in-phase (*i.e.* the same direction) as the victim line,

ii) switch out-of-phase (*i.e.* the opposite direction) as the victim line,

iii) no switching (*i.e.* remain at a steady state, either high or low).

In order to evaluate the delay uncertainty of the signal on the victim line, the coupled interconnect structure as shown in Fig. 5.6 is simulated using Spectre for any possible switching activity of the aggressor line. The interconnect structure being evaluated is illustrated in Fig. 5.7. Four different coupling scenarios are

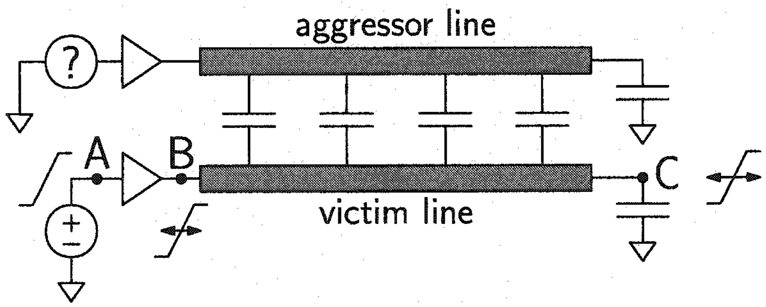


Fig. 5.6: Capacitive coupling between two interconnect lines

- Metal layer : m1
- Feature size $\lambda = 0.09\mu m$
  - minimum line width $W_{min} = 3\lambda = 0.27\mu m$
  - minimum line spacing $S_{min} = 3\lambda = 0.27\mu m$
- Line length $L = 400\mu m$
- Line resistivity $R_L = 296\mu\Omega/\mu m$
- Line capacitance:

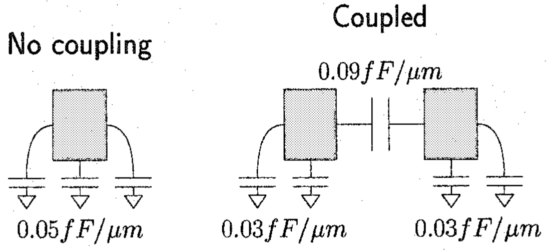


Fig. 5.7: Simulation of capacitively coupled interconnect

considered, depending upon the length of the coupling between the interconnect lines:

1. Low coupling: the lines are coupled along $\frac{1}{4}$ of the total length,
2. Medium coupling: the lines are coupled along $\frac{1}{2}$ of the total length,
3. High coupling: the lines are coupled along $\frac{3}{4}$ of the total length,
4. Heavy coupling: the lines are coupled along the entire wire length.

The uncertainty of the propagation delay of the buffer driving the victim line is shown in Fig. 5.8 for *low coupling* between the lines and different switching patterns of the aggressor line. The size of the buffer is the same for both the victim and aggressor lines. The signal on the victim line switches from high-to-low and the delay uncertainty due to different switching activities of the aggressor line is 23.8 ps. In Fig. 5.8, the buffer delay for the case of an uncoupled line is also shown for comparison.

Increasing the size of the buffer driving the victim line reduces the delay uncertainty as shown in Fig. 5.9(a). The size of the buffer driving the aggressor

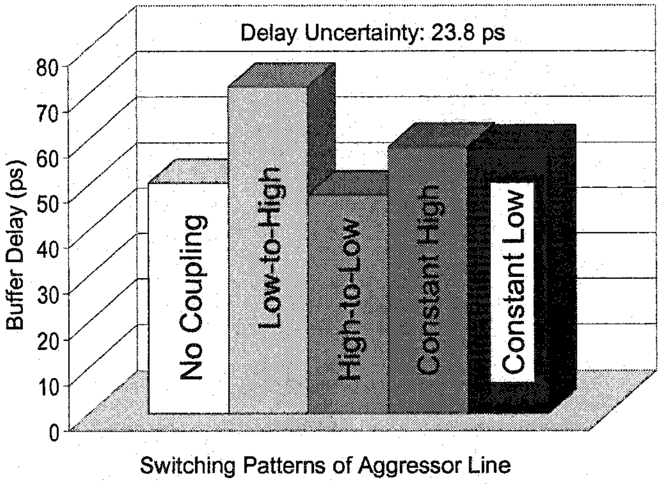
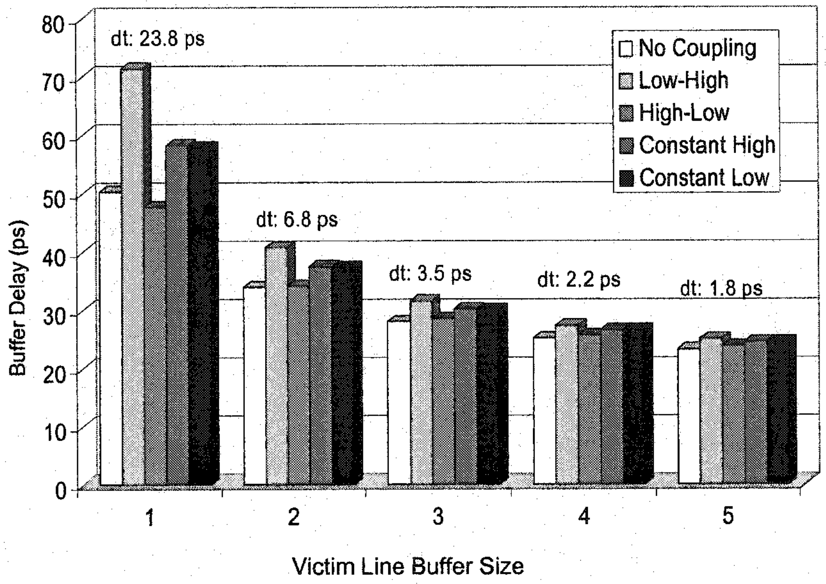


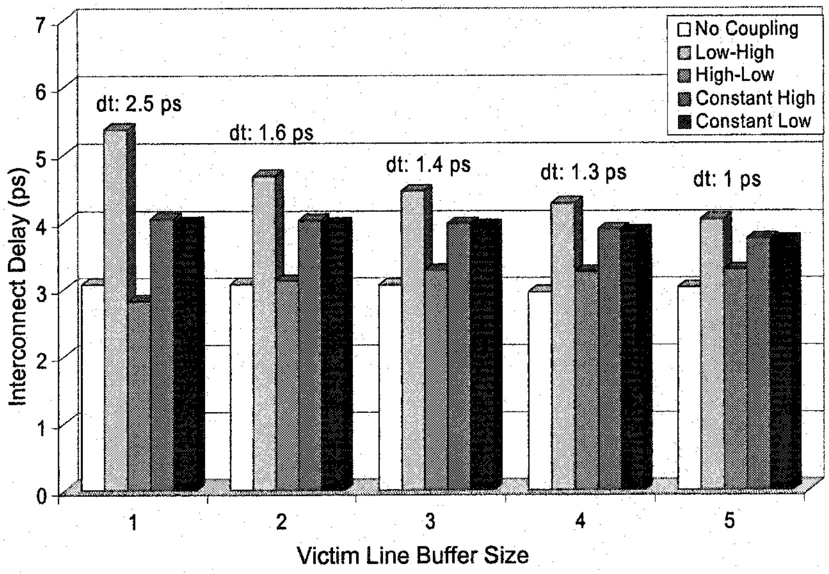
Fig. 5.8: Uncertainty of the signal delay of the buffer driving the victim line due to different switching activities of the aggressor line. Low coupling between the interconnect lines is considered

line equals one while the size of the victim line buffer is increased by up to five times. As described in (5.6), increasing the buffer size reduces the sensitivity of the buffer delay to variations in the load capacitance. In addition, as shown in Fig. 5.9(b), the increased current flowing through the buffer reduces the delay uncertainty of a signal propagating along the victim line, between points B and C, as shown in Fig. 5.6.

The effect of reducing the delay uncertainty by increasing the buffer size is more significant as the coupling between the aggressor and victim line increases. As shown in Fig. 5.10(a), the uncertainty of the buffer delay increases proportionally with increased capacitive coupling among the lines. The delay uncertainty, however, decreases exponentially with increasing buffer size. Similar trends in the uncertainty of the interconnect delay are illustrated in Fig. 5.10(b). The delay uncertainty increases proportionally with coupling between the lines and is reduced with increasing buffer size.

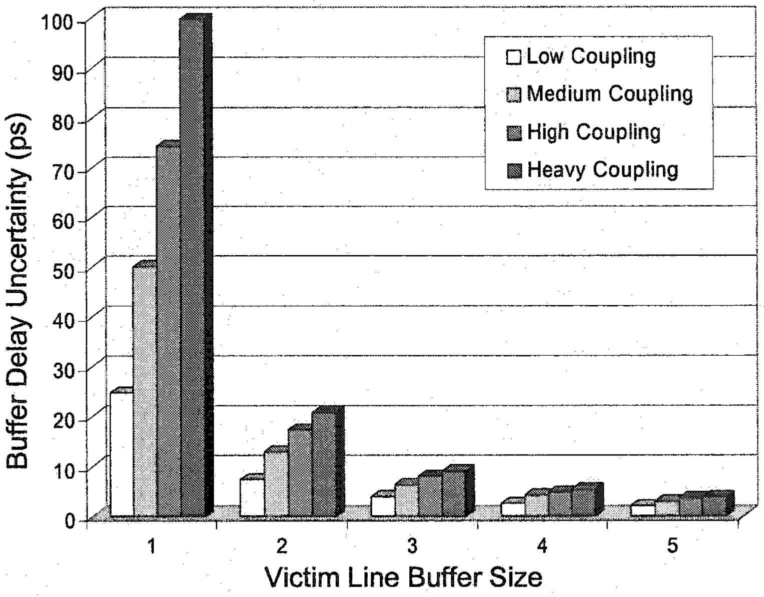


(a) Reduction in delay uncertainty of a buffer with increasing buffer size

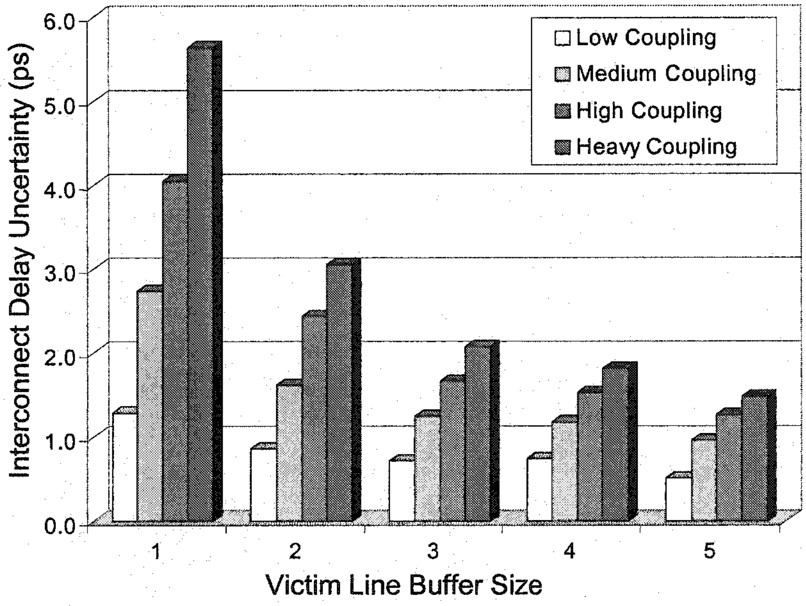


(b) Reduction in delay uncertainty of an interconnect line with increasing buffer size

Fig. 5.9: Reduction in delay uncertainty along the victim line with increasing buffer size



(a) Change in delay uncertainty of a driving buffer with increasing coupling and buffer size



(b) Change in delay uncertainty of an interconnect line with increasing coupling and buffer size

Fig. 5.10: Delay uncertainty increases proportionally with capacitive coupling among the lines

5.3 Power dissipation tradeoffs

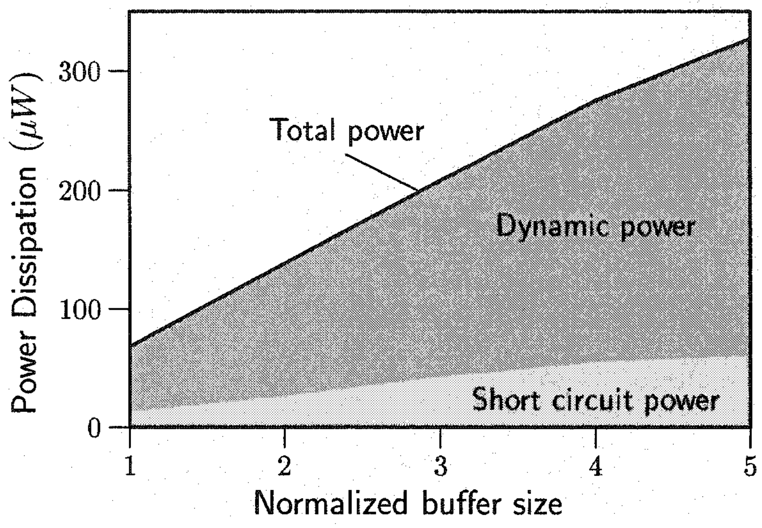
Increasing buffer size reduces the delay uncertainty due to variations in the transistor device parameters and interconnect crosstalk. Increasing the buffer size, however, also increases the requirements in buffer area and power dissipation. With increasing die area and scaling of on-chip feature sizes, increased buffer area is not a primary concern. The increase in power dissipation, however, is a significant effect that imposes practical constraints on the buffer size. In this section, design tradeoffs between the power dissipation and delay uncertainty are discussed.

The effect of the buffer size on the power dissipation characteristics is listed on Table 5.1. In the second and third columns of Table 5.1, the sizes of the PMOS and NMOS transistors, respectively, are listed. The power dissipation for a high-to-low and low-to-high signal transition is listed in columns four and five, respectively. The dynamic power is dissipated while charging and discharging the gates of the buffers. Short-circuit power is dissipated due to the current that flows directly from V_{dd} to ground during a signal transition, when both the PMOS and

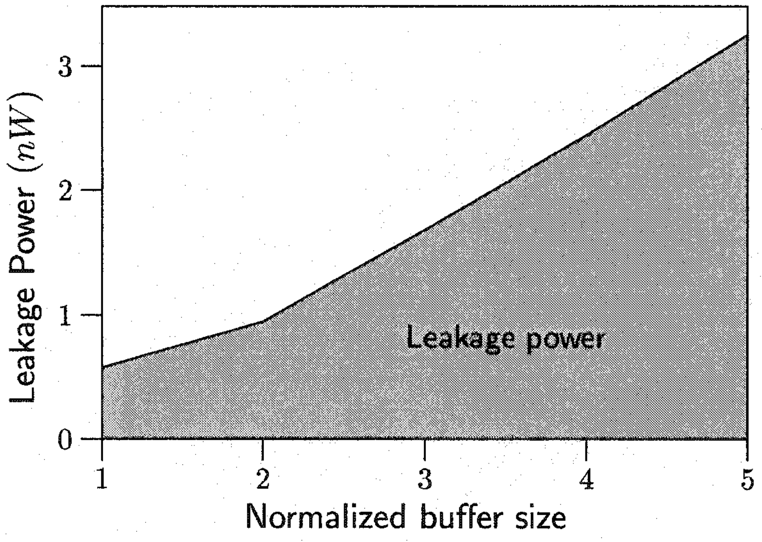
Table 5.1: Transistor size and power dissipation components for different buffer sizes.

Buffer Size	PMOS transistor size	NMOS transistor size	Dynamic power (μW)	Short-circuit power (μW)	Leakage power (nW)	Total power (μW)
1	0.9 (μm)	2.34 (μm)	55.58	12.71	0.58	68.29
2	1.8 (μm)	4.68 (μm)	112.25	26.22	0.94	138.47
3	2.7 (μm)	7.02 (μm)	165.52	42.78	1.68	208.31
4	3.6 (μm)	9.36 (μm)	220.35	55.02	2.44	275.38
5	4.5 (μm)	11.7 (μm)	266.976	60.37	3.25	327.25

NMOS transistors are on. The increase in short-circuit and dynamic power with buffer size is illustrated in Fig. 5.11(a). The leakage power listed in column six of Table 5.1 represents the power dissipated due to the leakage current flowing through the buffer transistors while the buffer input is maintained at a steady state



(a) Short-circuit, dynamic, and total power dissipation with increasing buffer size



(b) Increase in leakage power with increasing buffer size

Fig. 5.11: Increase in power dissipation with buffer size

voltage. The increase in leakage power with increasing buffer size is illustrated in Fig. 5.11(b). The total power dissipation is listed in column seven of Table 5.1 and is illustrated in Fig. 5.11(a).

The tradeoff between the reduction in delay uncertainty and increased power dissipation with increasing buffer size can be described by the figure of merit, *Power-Delay-uncertainty-Product* (PD_UP). The PD_UP is introduced to describe a *decrease* in the rate of delay uncertainty with respect to the *increase* in the rate of power dissipation as the buffer size is increased. Decreasing PD_UP with increasing buffer size indicates that the reduction in delay uncertainty is higher than the increase in power dissipation. Alternatively, an increase in PD_UP with increasing buffer size demonstrates a greater increase in power dissipation than the decrease in delay uncertainty.

The PD_UP with increasing buffer size is illustrated in Fig. 5.12 for delay uncertainty introduced by device parameter variations and interconnect coupling. The total PD_UP can be determined by considering the delay uncertainty due to both of these effects. The PD_UP figure of merit can be used to determine the effectiveness of increasing the buffer size to reduce delay uncertainty. For example, as shown in Fig. 5.12, increasing the buffer size is a power efficient way to reduce delay uncertainty due to interconnect coupling. Decreasing PD_UP indicates that the power dissipation increases more slowly than the delay uncertainty is reduced. In the case of reducing delay uncertainty due to device parameter variations, however, increasing PD_UP demonstrates that the increase in power dissipation is higher than the decrease in delay uncertainty. In this case, increasing buffer size is a non-efficient way, in terms of power dissipation, for reducing delay uncertainty.

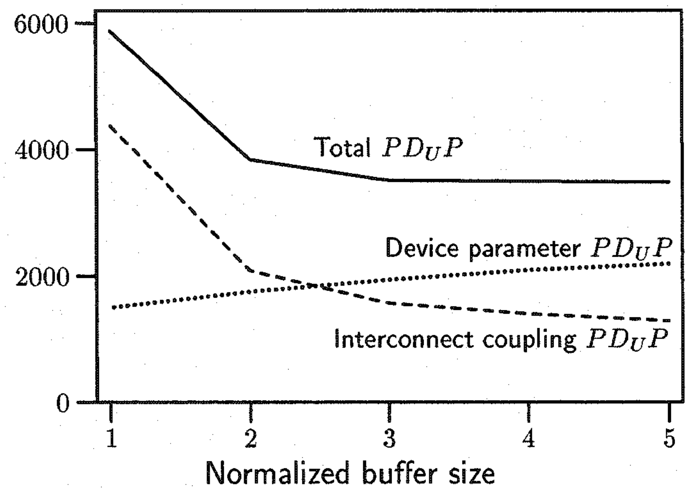


Fig. 5.12: Power-Delay uncertainty Product (PD_UP) for device parameter variations, interconnect coupling, and the sum of both effects

5.4 Conclusions

The effects of device parameter variations on the delay uncertainty of a signal propagating through a CMOS inverter are investigated. It is demonstrated that delay uncertainty is inversely proportional to the transistor current. Increasing the inverter size, therefore, reduces the delay uncertainty caused by device parameter variations. In addition, the delay uncertainty due to capacitive coupling among interconnect lines is examined. It is shown that delay uncertainty increases with coupling among lines and decreases with larger buffer size. The effect of increasing the buffer size on the power dissipation characteristics is also investigated. A power efficiency figure of merit, the Power-Delay-uncertainty-Product (PD_UP), is introduced to quantify the tradeoff between the reduction in delay uncertainty and the increase in power dissipation. It is shown that increasing buffer size to reduce delay uncertainty is more power efficient in the case of interconnect capacitive coupling than in the case of device parameter variations.

Chapter 6

Clock Tree Buffer Insertion and Layout Enhancements

The delay uncertainty of a clock signal is strongly dependent upon the geometric characteristics and spatial location of the clock lines, as described in Chapter 2. The wire width and thickness of a metal line determine the effect of process variations on the signal propagation delay [63]. In addition, crosstalk noise depends upon the interconnect length and the spacing among the lines [65, 66]. Furthermore, the spatial location of the registers and the branching nodes within a clock tree determine the wire length and the impedance of each clock path. Therefore, in order to estimate and reduce delay uncertainty, physical layout information should be incorporated within the clock distribution network design process.

Physical layout information is necessary to properly choose the location to insert a buffer along a clock distribution network. Buffer insertion can reduce the signal propagation delay. In addition, as shown in Chapter 5, increasing the size of the interconnect buffers can reduce delay uncertainty. Increasing buffer size reduces the uncertainty in the buffer delay caused by variations in the device parameters. Increased current flow also reduces the effect of coupling among interconnect lines. The effects of buffer insertion and buffer sizing on the delay uncertainty of the clock signal and the power dissipation of the clock tree are discussed in Section 6.1.

As described in Chapter 4, the delay uncertainty of the clock signal arriving at sequentially-adjacent registers can be decreased by reducing the non-common portion of the clock tree that drives those registers [64]. A methodology to apply this concept to the physical layout of a clock tree is described in Section 6.2. Reducing the non-common portion of the clock paths that drive the registers of the critical data paths decreases the delay uncertainty and relaxes the tight timing constraints of these registers. Additionally, both the number and size of the buffers within a clock tree are reduced, decreasing the overall power dissipated by a clock distribution network. Some conclusions are presented in Section 6.3.

6.1 Buffer insertion and sizing

The delay of a signal propagating along an interconnect structure depends upon the interconnect resistance, capacitance, and inductance [9]. In an *RC* line, the resistance and capacitance of a line are proportional to the wire length; therefore, the signal propagation delay increases quadratically with the length of a line [18]. Inserting buffers along an interconnect line alleviates the quadratic dependence of delay on the line length, permitting a line to be modeled as a simple capacitive line rather than an *RC* line [48]. Buffer insertion can therefore reduce the delay required for a signal to propagate along an interconnect line while also reducing the signal transition time. Additionally, in the case of a clock distribution network, buffer insertion can be used to apply a divide-and-conquer approach for driving hundreds of thousands of clocked elements [19].

Buffer insertion along an interconnect line affects the signal propagation delay. Device parameter variations change the current flow within a buffer, thereby introducing uncertainty in the buffer delay. Additionally, as described in Chapter 5, interconnect crosstalk causes variations in the effective load of a buffer, thereby changing the buffer delay and introducing additional delay uncertainty. The overall delay uncertainty of a signal propagating along an interconnect structure, however, can be controlled by changing the buffer size. As described in Chapter 5, increasing the size of a buffer reduces the delay uncertainty of both the buffer and the interconnect, albeit with an increase in power dissipation. The tradeoff between the reduction in delay uncertainty and the increase in power dissipation can be described by the Power-Delay-uncertainty-Product (PD_UP).

To investigate the effect of increasing buffer size on the delay uncertainty and power dissipation of a clock signal, a buffer insertion and sizing tool has been developed. Buffers are inserted along a clock tree. The location of these buffers depends upon the combined load of the clocked nodes and the wire capacitance. The size of the inserted buffers is determined by the delay uncertainty constraints of the clock signal along the clock paths that drive the most critical data paths of the circuit. The reduction in delay uncertainty and the resulting increase in power dissipation are determined for increasing buffer size.

The input to the buffer insertion tool is a minimal rectilinear Steiner tree that represents the layout of a clock tree. An example of such a tree is shown in Fig. 6.1. The source of the clock signal is located at the center of the tree plane. The numbered nodes shown in Fig. 6.1 represent the location of the clock registers within the circuit.

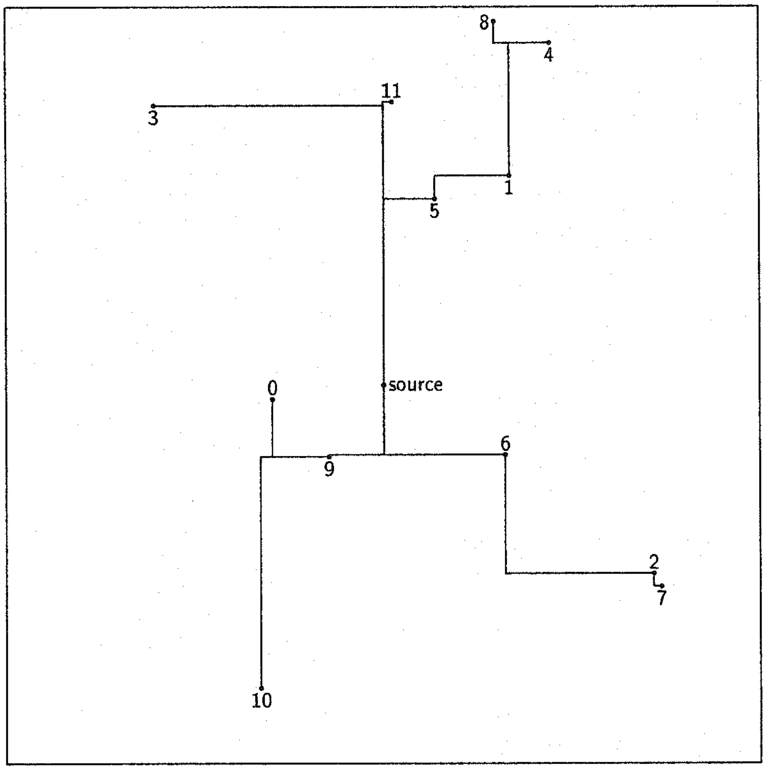


Fig. 6.1: An example of a minimal rectilinear Steiner clock tree

The first step of this tool is to insert buffers within the clock tree. Clock buffers are inserted in a bottom-up approach, starting from the tree leaves (*i.e.* the clocked elements) at the lowest level and advancing towards the root of the tree. When an intermediate node in the tree is reached, the total load from that node to the bottom of the tree is the summation of the capacitive load of the interconnect lines and the clocked elements. A clock buffer is inserted at a node when the downstream capacitive load of that node exceeds a particular threshold value. The magnitude of the downstream load determines the size of the inserted buffer.

When a buffer is inserted in a clock tree, a *subtree* is defined as the set of all tree nodes below the inserted buffer that are driven *directly* from that buffer. A node n in a tree is driven *directly* from a buffer B if no other buffer B' exists along the downstream path from buffer B to node n. If a buffer B' exists along that path, node n is driven directly from node B'. A buffered clock tree produced for the example shown in Fig. 6.1 is illustrated in Fig. 6.2. There are three buffers A, B, and C inserted in this clock tree. The relative size of each dot shown in Fig. 6.2 represents the size of the corresponding buffer. Three subtrees are defined in the clock tree shown in Fig. 6.2, each one rooted at one of the buffers A, B, and C.

The second step of the buffer insertion tool is to calculate the delay uncertainty between the clock signals that drive the sequentially-adjacent registers of the critical data paths. The delay uncertainty is introduced to the clock signals arriving

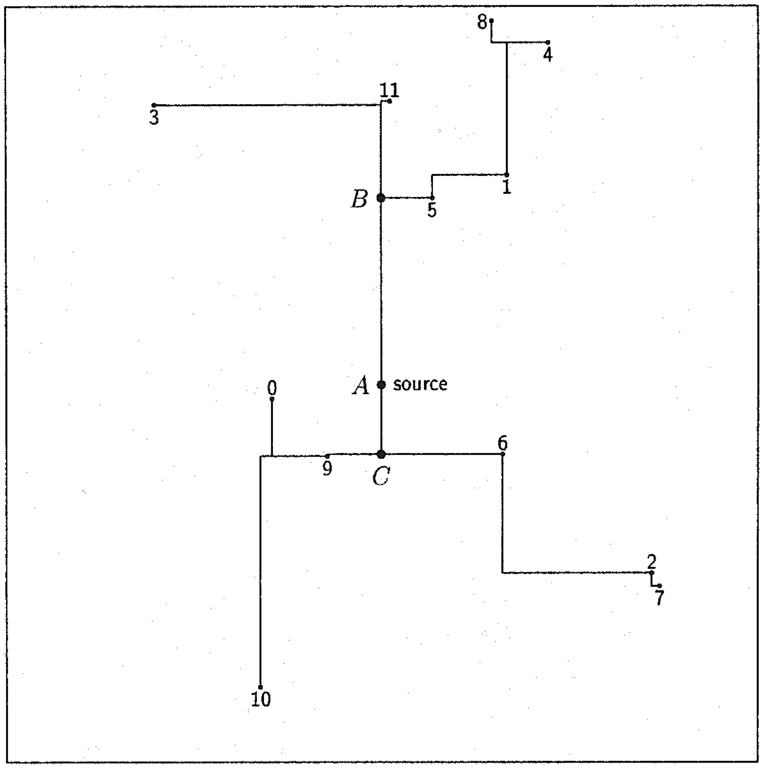


Fig. 6.2: Buffer insertion in the clock tree shown in Fig. 6.1

at those registers through different, non-common portions of the clock tree. The uncertainty in the signal delay is calculated for each of the subtrees on the signal path. In each subtree, the following three components of delay uncertainty are considered.

i) *Interconnect delay uncertainty due to crosstalk.* The uncertainty of the interconnect delay due to crosstalk is proportional to the wire length of a clock path within a subtree. It is inversely proportional to the size of the buffer driving a subtree.

ii) *Buffer delay uncertainty due to crosstalk.* Buffer delay uncertainty increases with line-to-line crosstalk. It is therefore proportional to the total wire length of a subtree driven by a buffer. It is inversely proportional to the size of the buffer driving a subtree.

iii) *Buffer delay uncertainty due to device parameter variations.* It is inversely proportional to the size of a buffer.

The total delay uncertainty of a signal propagating along a clock path can be calculated from the above components, given the wire length of a path, the wire length of the subtrees, and the size of the buffers driving those subtrees. The worst case delay uncertainty between the signals that drive sequentially-adjacent clock registers equals the sum of the delay uncertainties in each of the non-common clock paths [69]. If the resulting delay uncertainty is greater than the delay uncertainty constraints for the particular clock registers, the size of the buffers located along the non-common clock paths are increased to reduce the delay uncertainty. The delay uncertainty between the clock registers is calculated again using the new buffer sizes. The size of the inserted buffers is increased iteratively until the delay uncertainty constraints are satisfied.

In the example shown in Fig. 6.2, the critical registers are located at nodes 5 and 6. The clock paths driving those registers from the tree source are illustrated in Fig. 6.3. The delay uncertainty between the clock signals arriving at those nodes depends upon the size of the buffers A, B, and C (see Fig. 6.3). Increasing the size of these buffers further reduces the delay uncertainty between the clock signals. The increased sizes of those buffers that satisfy the delay uncertainty constraints for nodes 5 and 6 are shown in Fig. 6.3.

As demonstrated above, increasing buffer size can reduce the delay uncertainty of the clock signal while satisfying certain timing constraints. Increasing the size

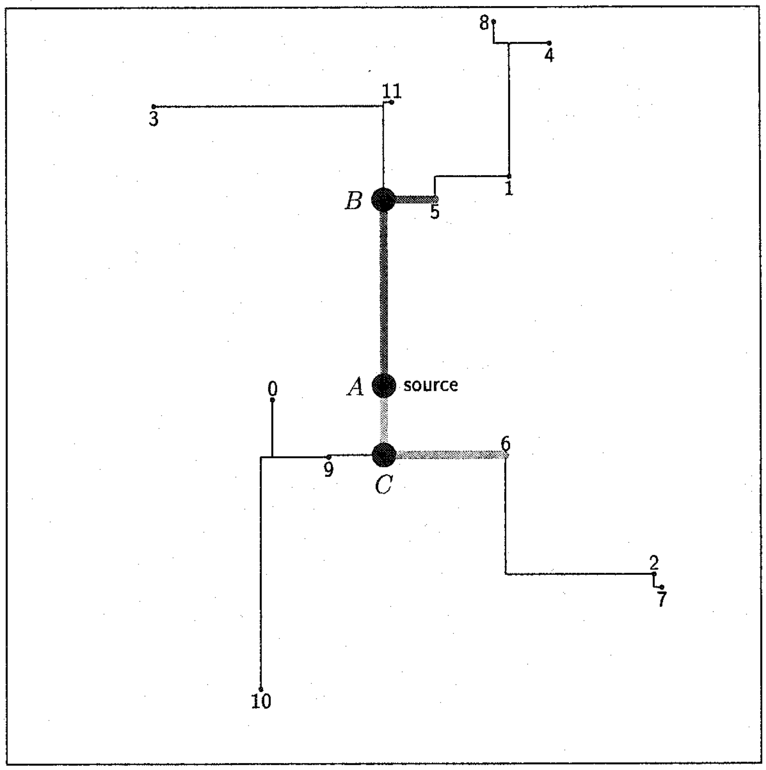


Fig. 6.3: Increasing buffer size to reduce delay uncertainty

of the buffers, however, increases the power dissipated by those buffers. The increase in power dissipation with increasing buffer size is listed in Table 6.1 for different clock distribution networks. The example shown in Figs. 6.1, 6.2, and 6.3 is listed in the second row of Table 6.1 (Circuit # 2). To satisfy the delay uncertainty constraints between the critical buffers 5 and 6, the size of the buffers A, B, and C has been increased by six times. The aggregated buffer size therefore is increased from 3 to 18, as listed in Table 6.1. The increase in the size of the buffers causes a 40% increase in the power dissipated by the clock tree for this circuit, as listed in the second row of Table 6.1.

Table 6.1: Increase in power dissipation with increasing buffer size.

Circuit	Number of registers	Aggregated size of initial buffers	Aggregated size of final buffers	Increase in buffer size (%)	Initial power (μW)	Final power (μW)	Increase in power (%)
1	11	4	21	425	361	499	38
2	12	3	18	500	303	425	40
3	17	5	29	480	444	638	43
4	23	6	34	466	536	763	42
5	28	8	30	275	647	825	27
6	36	8	25	212	745	882	18
7	42	8	15	87.5	802	923	15

6.2 Dedicated minimal clock tree driving the critical path registers

As described in Section 6.1, uncertainty in the delay of the clock signals is introduced as those signals propagate through different, non-common clock paths. In addition, it is demonstrated that the longer these non-common clock paths, the greater the delay uncertainty introduced between the clock signals. A design approach for reducing delay uncertainty should therefore aim in reducing the non-common portion between the clock paths. In particular, the non-common portion between the clock paths driving the registers of the most critical data paths should be reduced to relax the tight timing constraints on those registers. This approach is applied to extract the topology of a clock tree, as described in Chapter 4. An application of this approach to reduce the delay uncertainty by changing the layout of the clock tree is presented in this section. The effect of this approach on the power dissipation and interconnect area of the clock tree is also discussed.

A significant portion of the power dissipated within a clock distribution network is due to the capacitive load of the interconnect. Reducing the wire length of the clock tree, therefore, decreases the power dissipated by the interconnect. In addition, less wire length requires smaller routing area. In a minimal wire length tree, however, the length of the path connecting two arbitrary nodes is not always minimal. In the case where these nodes are the registers of a critical data path, the non-common portion of the clock paths driving these registers is not a minimum. The delay uncertainty introduced to the clock signals at those registers is therefore also not a minimum.

As discussed in Section 6.1, the delay uncertainty can be reduced by increasing the size of the buffers driving the non-common clock paths, which also increases the power dissipation. An example of this approach is illustrated in Fig. 6.3, where the delay uncertainty between registers 5 and 6 is reduced within the required range, although the non-common clock paths driving those registers are not minimized. Reducing delay uncertainty is possible by increasing the size of the buffers A, B, and C shown in Fig. 6.3. The increase in buffer size results in a 40% increase in the power dissipation, as listed in the second row of Table 6.1.

An alternative approach utilizes a dedicated minimal clock tree to distribute the clock signal to the registers of the critical data paths. In this approach, the delay uncertainty introduced due to the non-common clock paths is decreased. The delay uncertainty can be further reduced to satisfy the design constraints by increasing the size of the buffers that drive this dedicated clock tree. Due to the smaller load of those buffers, the delay uncertainty constraints are satisfied with a lower increase in the buffer size and power dissipation as compared to the buffer insertion and sizing approach presented in Section 6.1. Once the delay uncertainty at the critical nodes is controlled, the clock signal is distributed to the remaining clock registers through a minimal wire length clock tree.

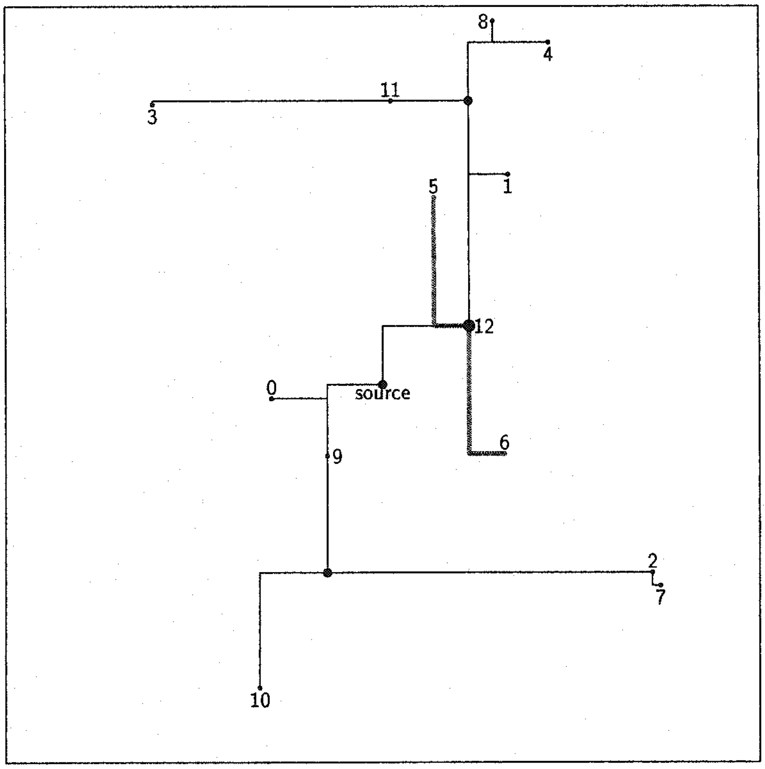


Fig. 6.4: Dedicated clock tree and buffers to drive the critical registers in the circuit shown in Fig. 6.1

An example of a dedicated minimal clock tree driving the critical registers of a circuit is shown in Fig. A.14 for the circuit example illustrated in Fig. 6.1. The dedicated minimal tree is driven by buffer No. 12. Note that the size of the clock buffers shown in Fig. A.14 is smaller than the size of the buffers in the circuit shown in Fig. 6.3. The total area of the clock tree, however, is increased when a dedicated minimal clock tree for the critical nodes is used.

The proposed approach utilizing a dedicated minimal clock tree for the critical registers is compared with the buffer insertion and sizing approach described in Section 6.1. The tradeoff between power dissipation and clock tree area is listed in Table 6.2. The reduction in power dissipation is due to the reduction of the aggregate buffer size of each of the clock trees listed in Table 6.3.

Table 6.2: Tradeoff between the reduction in power dissipation and the increase in clock tree area.

Circuit	Number of Registers	Power Dissipation (μW)			Clock tree area (μm)		
		Buffer Insertion	Dedicated Tree	Reduction (%)	Buffer Insertion	Dedicated Tree	Increase (%)
1	11	499	488	2	3065	3930	28
2	12	425	356	16	2348	2706	15
3	17	638	531	16	3382	4167	23
4	23	763	656	14	3823	5005	31
5	28	825	737	10	4490	5403	20
6	36	882	851	3	4901	5911	20
7	42	923	831	10	4901	5167	5

The clock trees produced by the buffer sizing approach and the dedicated minimal clock tree approach for each of the circuits listed in Tables 6.1, 6.2, and 6.3 are illustrated in Appendix A. The aggregate buffer size, power dissipation, and clock tree area for each approach are also listed in Appendix A.

Table 6.3: Comparison between the reduction in power dissipation and the reduction in the aggregate buffer size.

Circuit Number	Number of Registers	Power Dissipation (μW) Buffer Insertion	Power Dissipation (μW) Dedicated Tree	Power Dissipation (μW) Reduction (%)	Aggregate Buffer Size Buffer Insertion	Aggregate Buffer Size Dedicated Tree	Aggregate Buffer Size Reduction (%)
1	11	499	488	2	21	10	52
2	12	425	356	16	18	5	72
3	17	638	531	16	29	7	75
4	23	763	656	14	34	8	76
5	28	825	737	10	30	9	70
6	36	882	851	3	25	10	60
7	42	923	831	10	15	9	40

6.3 Conclusions

Clock buffer insertion improves the delay and signal transition characteristics of a clock signal. The effect of buffer insertion and sizing on the delay uncertainty of a clock signal is investigated in this chapter. It is demonstrated that increasing the size of a clock buffer reduces the uncertainty of the clock signal delay, albeit with an increase in power dissipation. An alternative approach changes the layout of the clock tree and introduces a dedicated minimal tree that only drives the critical registers in a circuit. The clock delay uncertainty is reduced in the dedicated tree by changing the size of the clock buffers driving that tree. A smaller increase in the total power dissipation is demonstrated as compared to the clock buffer insertion and sizing approach. The total area of the clock distribution network however increases with a dedicated clock tree.

Chapter 7

Conclusions

The rapid scaling of on-chip geometric dimensions supports the system-on-a-chip integration of multiple subsystems, greatly increasing the functionality of an integrated circuit. This effect has resulted in hundreds of thousands of elementary operations synchronized by a periodic reference signal, namely the clock signal. The continuous quest for higher circuit performance has pushed clock frequencies deep into the gigahertz frequencies range, reducing the period of the clock signal well below a nanosecond. The resulting constraints demonstrate the requirement for tight timing control of the arrival times of the clock signal at the many clocked elements throughout an integrated circuit. Deviations of the clock signal from the target delay can cause incorrect data to be latched within a register, resulting in a system malfunctioning. These deviations of a signal delay from a target value are described as delay uncertainty.

The important task of distributing the clock signal within an integrated circuit is performed by the clock distribution network. Due to the large number of clocked elements in a circuit and the tight timing constraints, the design of a clock distribution network represents one of the most challenging tasks in the integrated circuit design process. Reducing uncertainty in the clock signal delay is one of the primary issues in the design of a high performance clock distribution network.

The uncertainty of the clock signal delay is caused by a number of factors that affect a clock distribution network, examples of which include process and environmental parameter variations. As described in Chapter 2, the variation of different parameters affect different elements in a clock distribution network. For instance, the variation of the device parameters affects the current flow within a clock buffer and therefore the delay of the clock signal through that buffer. In addition, variations in the geometric parameters of the interconnect wires introduce uncertainty in the propagation delay. Furthermore, crosstalk among interconnects also affects the delay of the clock signal. The sensitivity of a clock distribution network to these effects has become an issue of fundamental importance to the design of high performance synchronous systems.

The research effort described in this dissertation has focused on reducing the effects that introduce uncertainty into the delay of the clock signal, thereby enhancing the tolerance of a system to delay uncertainty. One approach to improve the tolerance to delay uncertainty is by relaxing the timing constraints, particularly the most critical data paths within a circuit. Non-zero clock skew scheduling can be exploited to change the arrival times of the clock signals at the registers of these data paths. Delaying the arrival time of a clock signal can provide additional time for the data to propagate within a data path, arrive and latch into the final register, thereby relaxing the overall timing constraints. A demonstration of this approach on an industrial microprocessor is presented in Chapter 3. The timing constraints of the most critical data paths in this circuit are relaxed by up to 18%. Furthermore, the application of non-zero clock skew scheduling to provide additional time for the non-critical data paths is also discussed, resulting in a reduction of the power dissipated by these paths.

A methodology to reduce the uncertainty in the delay of the clock signal arriving at the most critical data paths of a circuit is also developed. The key concept of this methodology is that no delay uncertainty is introduced among the clock signals that propagate along common portions of a clock distribution network. Uncertainty in the delays of these signals is introduced in the non-common portions of the clock paths. Reducing the non-common portions of the clock paths therefore reduces the delay uncertainty of the signals arriving at sequentially-adjacent registers. An algorithm that implements this methodology to extract the topology of a tree structured clock distribution network is presented in Chapter 4. Application of this algorithm to a suite of benchmark circuits demonstrates a significant reduction in the delay uncertainty of the clock signals arriving at the registers of the most critical data paths.

The delay of the clock signal propagating along a clock distribution network can be controlled by inserting a clock buffer within the signal path. The variation of the device parameters, however, can change the current flow through that buffer, thereby introducing uncertainty in the buffer delay. It is shown in Chapter 5 that the uncertainty in the buffer delay can be reduced by increasing the buffer size. Increasing the buffer size reduces the effect of device parameter variations on the current flowing through a buffer, thereby reducing delay uncertainty. Furthermore, it is demonstrated that increasing the buffer size reduces the delay uncertainty caused by crosstalk coupling among interconnect lines. The primary drawback of increasing the clock buffer size is the increase in power dissipation. The tradeoff between reducing the delay uncertainty and increasing the power dissipation is considered by introducing the Power-Delay-uncertainty-Product (PD_UP). It is shown that increasing the buffer size to reduce delay uncertainty is more power efficient in the case of interconnect crosstalk than in the case of device parameter variations.

The delay uncertainty of a clock signal is strongly dependent upon the geometric characteristics and the spatial location of the clock lines and registers. Therefore, in order to estimate and reduce delay uncertainty, physical layout information should be incorporated into the clock distribution network design process. Two different strategies are described in Chapter 6 for synthesizing the clock tree layout. Both strategies reduce the delay uncertainty of the clock signals arriving at the registers of the most critical data paths. Clock buffer insertion and sizing is utilized in one approach, exploiting the reduction in delay uncertainty with increasing buffer size. In the second approach, the clock signal is distributed to the registers of the critical paths by a dedicated portion of the clock tree, which increases the common path among the clock signals. To satisfy the delay uncertainty constraints, the size of the clock buffer driving the dedicated portion of the clock tree is increased. A tradeoff between power dissipation and wire length is demonstrated by the application of these two strategies to the synthesis of a clock layout for a set of benchmark circuits.

In summary, the research presented in this dissertation introduces a design methodology for enhancing the tolerance of a circuit to the uncertainty of the clock signal delay. This methodology either relaxes the timing constraints at the registers of the most critical data paths, or reduces the delay uncertainty between the clock signals that drive those registers. The proposed methodology is applied to determine a non-zero clock skew schedule, extract the topology of a clock tree, and synthesize the clock tree layout. Enhancing the tolerance of a clock distribution network to delay uncertainty is demonstrated by the application of the proposed methodologies on a design example presented in Appendix B.

Chapter 8

Future Research

The effects that introduce delay uncertainty are aggravated with continuous scaling of the on-chip feature size. In addition, with increasing clock frequencies, the timing constraints at the registers of a circuit become tighter. The importance of uncertainty in the signal propagation delay therefore increases in current and future synchronous intergrated circuits. Significant research effort is already focused on characterizing and reducing delay uncertainty. Technology trends [6], however, indicate that greater effort is required to develop a system level design methodology that will control the effects that introduce delay uncertainty.

Reducing delay uncertainty at the system level requires characterizing early in the circuit design process those effects that introduce delay uncertainty. Models of delay uncertainty should therefore be developed at each stage in the circuit design process. These models would characterize delay uncertainty early in the design flow, based on the available design information. Techniques for incorporating delay uncertainty models into the design flow are presented in Section 8.1. In each stage of the design process, delay uncertainty models can be used to determine the most critical design parameters. Proceeding into the next stage, new design information can be extracted and back annotated into an earlier design stage to optimize the design and re-evaluate these critical design parameters.

Modeling delay uncertainty early in the circuit design process allows the development of techniques that reduce the effects that introduce uncertainty at each level of the design flow. These techniques leverage the design criteria provided by the delay uncertainty models, while optimizing the available circuit resources. Some of those techniques are presented in Section 8.2. The target of these techniques is to minimize delay uncertainty at the most critical portions of a circuit.

As technology evolves, new design trends and manufacturing capabilities emerge, enhancing the function and performance of an integrated circuit. Together with these advantages, however, new design challenges emerge as well. Some of these new challenges in tolerating delay uncertainty are discussed in Section 8.3.

8.1 Incorporating delay uncertainty into the design flow

At the beginning of the design cycle, many of the design and implementation details have yet to be determined. There is, therefore, a great amount of design freedom, providing a broad selection of design choices to determine the optimal tradeoffs among the many circuit resources in order to satisfy the target circuit specifications. These design tradeoffs are important in determining the primary sources of noise and delay uncertainty. It is therefore important to obtain the proper information to minimize these effects early in the design cycle. The time-to-market can thereby be shortened and the reliability of a circuit enhanced.

Incorporating information about the delay uncertainty early in the design process requires the development of delay uncertainty models at different levels of the design flow. During the early stages of a design, little circuit information is available and the models provide only an approximation of the effects that introduce delay uncertainty. Later in the design flow, however, more design and implementation details become available, improving the accuracy of the delay uncertainty models. A process for incorporating these models to estimate the delay uncertainty at different stages of the circuit design process is illustrated in Fig. 8.1.

As shown in Fig. 8.1, at each stage of the circuit design process, the delay uncertainty can be back annotated into previous stages of the design process. An earlier stage can use the resulting information about delay uncertainty, permitting the final circuit to be more highly optimized. For example, as shown in Fig. 8.1, the critical data paths of a circuit are determined during the Circuit Design stage. This information is used in the following stage, Circuit Optimization, to obtain an optimal clock skew schedule and extract the topology of the clock tree in order to minimize the delay uncertainty of the most critical data paths. When the clock tree topology is extracted, however, the criticality of some data paths may change. It is, therefore, possible that another set of critical data paths can be determined with a smaller tolerance to delay uncertainty. Information describing the extracted topology can be back annotated into the Circuit Design stage and, using static timing analyses, the criticality of the data paths can be determined. Finally, note in Fig. 8.1 that information about the delay uncertainty, experimentally extracted

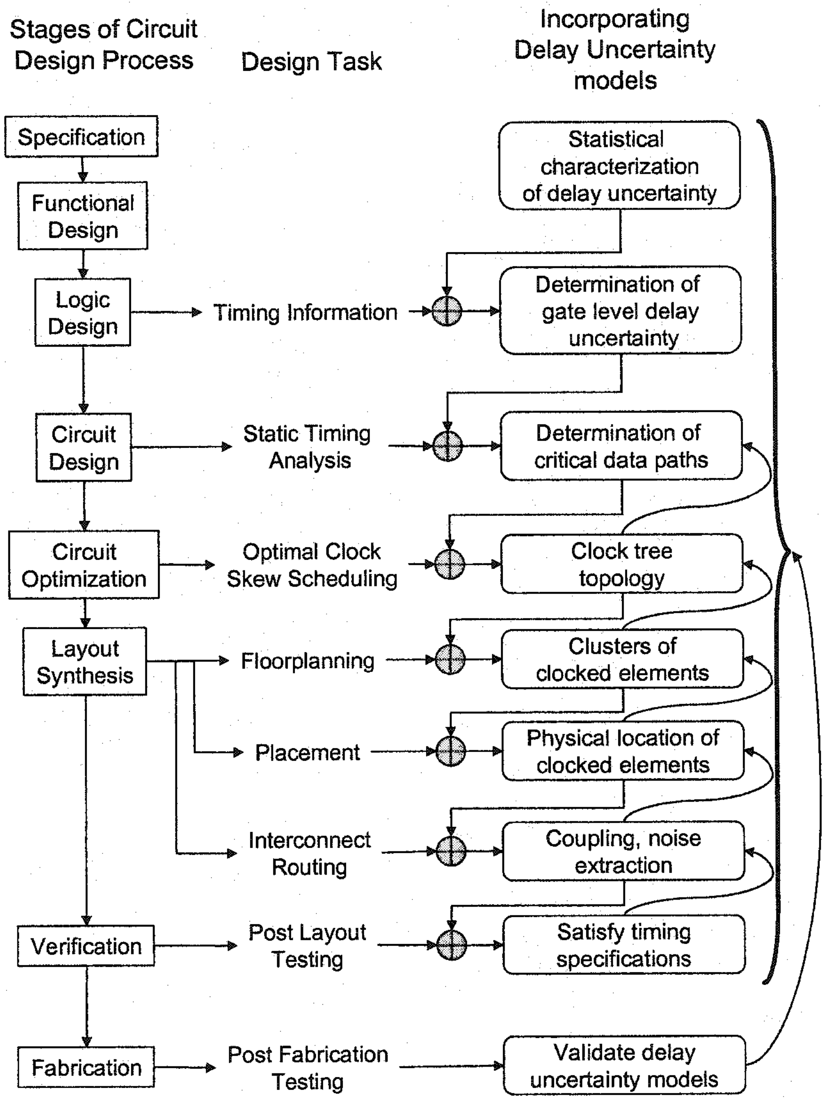

Stages of Circuit Design Process
Design Task
Incorporating Delay Uncertainty models
Specification
Functional Design
Logic Design
Timing Information
Circuit Design
Static Timing Analysis
Circuit Optimization
Optimal Clock Skew Scheduling
Layout Synthesis
Floorplanning
Placement
Interconnect Routing
Verification
Post Layout Testing
Fabrication
Post Fabrication Testing
Statistical characterization of delay uncertainty
Determination of gate level delay uncertainty
Determination of critical data paths
Clock tree topology
Clusters of clocked elements
Physical location of clocked elements
Coupling, noise extraction
Satisfy timing specifications
Validate delay uncertainty models

Fig. 8.1: Incorporating delay uncertainty models into the design flow to estimate delay uncertainty earlier in the circuit design process

from a test circuit, can be used to validate the models of delay uncertainty at each stage of the circuit design process.

8.2 Design techniques for reducing delay uncertainty

Using the models derived for incorporating delay uncertainty into the circuit design flow, design techniques can be developed at each level of abstraction of the design process to minimize delay uncertainty. The insertion of tapered clock buffers is described in section 8.2.1. The use of shield lines to reduce crosstalk among interconnects is discussed in section 8.2.2.

8.2.1 Tapered buffer insertion for delay balancing

In the placement process during the Layout Synthesis stage as shown in Fig. 8.1, the physical locations of the clocked registers in a circuit are determined. In addition, grouping these registers into clusters as well as determining an appropriate clock tree topology are based on information from previous design stages. The fanout of each branch of a clock tree can be calculated during the placement process and divided into smaller segments, if necessary, by hierarchically inserting clock buffers along the tree. The impedance of each of these segments, however, may be different due to variations in the number of registers within a cluster. These load variations can cause the clock signal to arrive at each register at different times within different tree segments, thereby creating clock skew. To minimize or control the clock skew, the inserted clock buffers can be sized in proportion to the load. The current flow through a buffer should be chosen such that the delay of the clock signal along different tree segments is either the same, or will satisfy a target clock skew schedule [35].

Most synchronous digital systems utilize a non-inverting clock distribution scheme, where the clock signal has the same phase along the entire clock distribution network. To achieve a non-inverting clock signal, the clock buffers inserted along a clock path should have an even number of inverting stages. A common approach is to insert a two stage tapered buffer as a non-inverting clock buffer. Each of the inverter stages in these buffers can be sized individually so as to match the upstream and downstream signal delays among parallel clock paths, as shown in Fig. 8.2. The inverters of the first stage in buffers B, C, and D, shown in Fig 8.2, are sized so as to balance the delay D_1 of the three paths driven by the upstream buffer A. The inverters in the second stage are sized so as to balance the delay D_2 of the clock signal along the segments driven by each of the buffers B, C, and D.

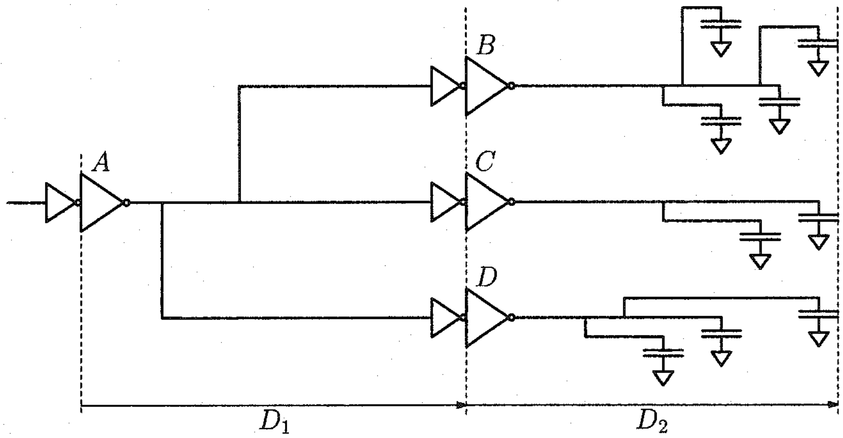


Fig. 8.2: Sizing of tapered buffers per stage to match upstream and downstream delays

Once the location and size of these buffers are determined, the next step in the Layout Synthesis process is to route the interconnect wires. The total load of a tapered buffer, including the load of the interconnects, can be back annotated into the previous level of the Layout Synthesis process. In the previous level, the size of the tapered buffers can be determined to achieve a balanced or managed clock delay.

8.2.2 Shielding of clock lines

Capacitive coupling and mutual inductance among interconnect lines become more significant as the distance between lines is decreased. In addition, the effects of on-chip inductance have become more significant with higher signal transition times and lower resistivity interconnect lines.

One way to reduce delay uncertainty introduced by these effects is through the application of shielding [70, 71]. Shield lines can be inserted around those interconnects that are more sensitive to delay uncertainty, such as the lines within a critical path. Inserting a shield around the clock lines can occur within the interconnect routing step of the layout synthesis process. After routing the interconnect lines is complete, interconnect crosstalk noise information can be extracted and the most affected lines can be identified. A shielding technique specifies those sensitive victim lines and inserts power supply lines, such as V_{dd} and ground lines, among the sensitive lines, as shown in Fig 8.3. In this way, a more accurate estimate of the capacitive and inductive impedances of these lines is achieved, permitting the delay uncertainty of the critical signals to be reduced.

The noise injected on a victim line increases significantly as the signal coupling occurs farther from the line driver, as shown in Fig. 8.4 [72]. It is assumed that

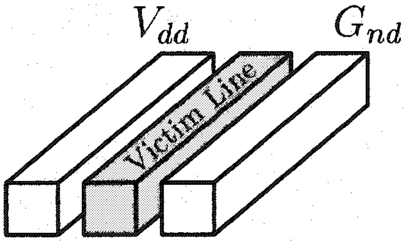
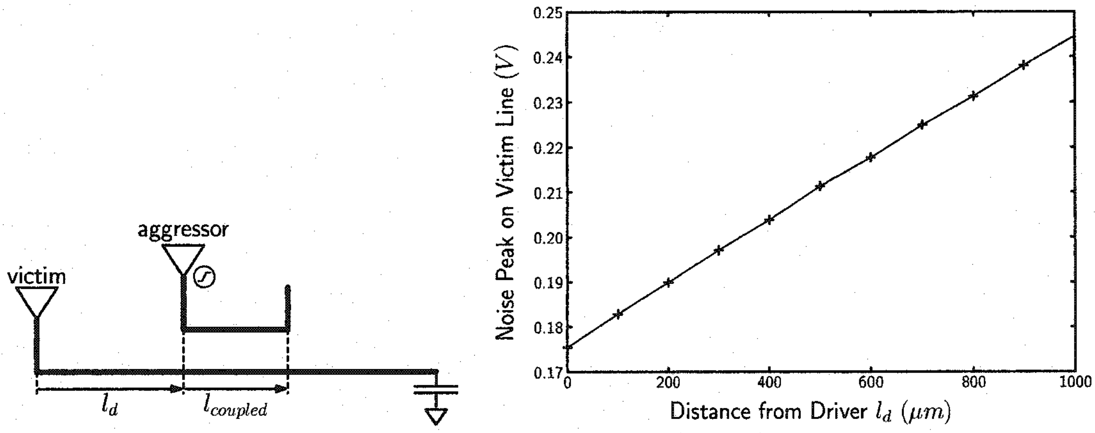


Fig. 8.3: Shielding a victim line with power supply lines

a victim line is coupled to an aggressor line along a portion $l_{coupled}$ of the victim line, as shown in Fig. 8.4(a). As an aggressor line is moved farther from the driver of the victim line, the peak noise at the receiver of the victim line increases by up to 40%, as shown in Fig. 8.4(b). It is therefore preferable to introduce shield lines closer to the line receiver. This shielding strategy increases the efficiency of the shield line since a greater reduction in noise is achieved with a limited use of metal resources.



(a) Coupling between aggressor and victim lines

(b) Variation of peak noise with distance from the driver

Fig. 8.4: Peak noise increases as coupling occurs farther from the driver

8.3 New design trends and challenges

High synchronous performance is the result of reducing the period of the clock signal, which tightens the timing constraints within a circuit. The tolerance of a circuit to noise effects that introduce delay uncertainty is thereby reduced. Additionally, operating in a low power mode reduces the power supply voltage, thereby decreasing circuit noise margins. Both high speed and low power operation reduce the tolerance of a circuit to noise effects and increase the significance of uncertainty in the signal delay.

Tolerance to noise is therefore a new design criterion, marking the beginning of a design trend for *low power, low noise, and high speed* integrated circuits. Tolerance to noise requires sufficient power supply voltages and relaxed timing constraints. These requirements introduce a new category of design tradeoffs, providing the opportunity for the development of novel design techniques and methodologies, making the design of future integrated circuits more exciting than ever.

References

[1] L. Gwennap, "Birth of a Chip," *BYTE magazine*, December 1996.

[2] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, pp. 114–117, April 1965.

[3] R. H. Dennard, F. H. Gaensslen, H. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions," *IEEE Journal of Solid State Circuits*, Vol. SC-9, No. 5, pp. 256–258, October 1974.

[4] K. Chen and C. Hu, "Performance and V_{dd} Scaling in Deep Submicrometer CMOS," *IEEE Journal of Solid State Circuits*, Vol. SC-33, No. 10, pp. 1586–1589, October 1998.

[5] J. R. Hauser, "Noise Margin Criteria for Digital Logic Circuits," *IEEE Transactions on Education*, Vol. 36, No. 4, pp. 363–368, November 1993.

[6] S. I. A., "The National Technology Roadmap for Semiconductors," Technical Report, Semiconductor Industry Association, 2001.

[7] K. T. Tang and E. G. Friedman, "Delay and Noise Estimation of CMOS Logic Gates Driving Coupled Resistive-Capacitive Interconnections," *Integration, the VLSI Journal*, Vol. 29, No. 2, pp. 131–165, September 2000.

[8] K. T. Tang and E. G. Friedman, "Peak Crosstalk Noise Estimation in CMOS VLSI Circuits," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, pp. 1539–1542, September 1999.

[9] Y. I. Ismail and E. G. Friedman, "Effects of Inductance on the Propagation Delay and Repeater Insertion in VLSI Circuits," *IEEE Transactions on Very*

Large Scale Integration (VLSI) Systems, Vol. 8, No. 2, pp. 195–206, April 2000.

[10] R. A. Lawes, "Future Trends in High-Resolution Lithography," *Journal of Applied Surface Science*, Vol. 154-155, No. 1-4, pp. 519–526, February 2000.

[11] M. Yoshizawa and S. Moriya, "Comparative Study of Resolution Limiting Factors in Electron Beam Lithography Using the Edge Roughness Evaluation Method," *Journal of Vaccum Science and Technology B (Microelectronices and Nanometer Structures)*, Vol. 19, No. 6, pp. 2488–2493, November 2001.

[12] V. Mehrotra, S. L. Sam, D. Boning, A. Chandrakasan, R. Vallishayee, and S. Nassif, "A Methodology for Modeling the Effects of Systematic Within-Die Interconnect and Device Variation on Circuit Performance," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 172–175, June 2000.

[13] R. Saleh, S. Z. Hussain, S. Rochel, and D. Overhauser, "Clock Skew Verification in the Presence of IR-Drop in the Power Distribution Network," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No. 6, pp. 635–644, June 2000.

[14] S. Sauter, D. Schmitt-Landsiedel, R. Thewes, and W. Weber, "Effect of Parameter Variations at Chip and Wafer Level on Clock Skews," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 13, No. 4, pp. 395–400, November 2000.

[15] J. F. Chappel and S. G. Zaky, "EMI Effects and Timing Design for Increased Reliability in Digital Systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 44, No. 2, pp. 130–142, February 1997.

[16] G. E. Moore, "Progress in Digital Integrated Circuit," *Proceedings of the IEEE International Electron Devices Meeting*, pp. 11–14, December 1975.

[17] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*, Prentice Hall Inc., 1995.

[18] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publishing Company, 1990.

[19] I. S. Kourtev and E. G. Friedman, *Timing Optimization Through Clock Skew Scheduling*, Kluwer Academic Publishers, Norwell Massachusetts, 2000.

[20] V. Adler and E. G. Friedman, "Uniform Repeater Insertion in RC Trees," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 47, No. 10, pp. 1515–1523, October 2000.

[21] S. M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits. Analysis and Design.*, McGraw-Hill Companies, Second Edition, 1999.

[22] H. Veendrick, *Deep-Submicron CMOS-ICs. From Basics to ASICs*, Kluwer BedrijfsInformatie b.v., Deventer, The Netherlands, 1998.

[23] R. F. Pierret, *Semiconductor Device Fundamentals*, Addison-Wesley Publishing Company Inc., 1996.

[24] W. Shockley, "A Unipolar "Field-Effect" Transistor," *Proceedings of the Institute of Radio Engineers*, pp. 1365–1377, November 1952.

[25] A. Chandrakasan, W. J. Bowhill, and F. Fox, (Eds.), *Design of High-Performance Microprocessor Circuits*, IEEE Press, Piscataway, New Jersey, 2001.

[26] R. Sitte, S. Dimitrijev, and H. B. Harrison, "Device Parameter Changes Caused by Manufacturing Fluctuations of Deep Submicron MOSFET's," *IEEE Transactions on Electron Devices*, Vol. 41, No. 11, pp. 2210–2215, November 1994.

[27] T. Mizuno, J. Okamura, and A. Toriumi, "Experimental Study of Threshold Voltage Fluctuation Due to Statistical Variation of Channel Dopant Number in MOSFET's," *IEEE Transactions on Electron Devices*, Vol. 41, No. 11, pp. 2216–2221, November 1994.

[28] J. T. Kao and A. P. Chandrakasan, "Dual-Threshold Voltage Techniques for Low Power Digital Circuits," *IEEE Journal of Solid State Circuits*, Vol. 35, No. 7, pp. 1009–1018, July 2000.

[29] M. B. Anand, H. Shibata, and M. Kakumu, "Optimization Study of VLSI Interconnect Parameters," *IEEE Transactions on Electron Devices*, Vol. 47, No. 1, pp. 178–186, January 2000.

[30] S. S. Sapatnekar, "A Timing Model Incorporating the Effect of Crosstalk on Delay and its Application to Optimal Channel Routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No. 5, pp. 550–559, May 2000.

[31] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of Merit to Characterize the Importance of On-Chip Inductance," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 7, No. 4, pp. 442–449, December 1999.

[32] J. J. Laurin, S. G. Zaky, and K. G. Balmain, "EMI-Induced Delays in Digital Circuits: Prediction," *Proceedings of the IEEE Symposium in Electromagnetic Compatibility*, pp. 443–448, August 1992.

[33] P. Zarkesh-Ha, T. Mule, and J. D. Meindl, "Characterization and Modeling of Clock Skew with Process Variations," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 441–444, May 1999.

[34] E. G. Friedman, (Ed.), *Clock Distribution Networks in VLSI Circuits and Systems*, IEEE Press, Piscataway, New Jersey, 1995.

[35] E. G. Friedman and S. Powell, "Design and Analysis for a Hierarchical Clock Distribution System for Synchronous Standard Cell/Macrocell VLSI," *IEEE Journal of Solid State Circuits*, Vol. SC-21, No. 2, pp. 240–246, April 1986.

[36] H. B. Bakoglou, J. T. Walker, and J. D. Meindl, "A Symmetric Clock-Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits," *Proceedings of the IEEE International Conference on Computer Design*, pp. 118–122, October 1986.

[37] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, K. D. Boese, and A. B. Kahng, "Zero Skew Clock Routing with Minimum Wirelength," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 39, No. 11, pp. 799–814, November 1992.

[38] A. B. Kahng and G. Robins, *On Optimal Interconnections for VLSI*, Kluwer Academic Publishers, Boston, Massachusetts, 1995.

[39] U. Desai, S. Tam, R. Kim, and J. Zhang, "Itanium Processor Clock Design," *Proceedings of the ACM/SIGDA International Symposium on Physical Design*, pp. 94–98, April 2000.

[40] S. Rusu and S. Tam, "Clock Generation and Distribution for the First IA-64 Microprocessor," *Proceedings of the IEEE International Solid State Circuits Conference*, pp. 176–177, February 2000.

[41] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, Vol. 39, No. 7, pp. 945–951, July 1990.

[42] J. L. Neves and E. G. Friedman, "Design Methodology for Synthesizing Clock Distribution Networks Exploiting Non-Zero Clock Skew," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. VLSI-4, No. 2, pp. 286–291, June 1996.

[43] J. L. Neves and E. G. Friedman, "Buffered Clock Tree Synthesis with Non-Zero Clock Skew Scheduling for Increased Tolerance to Process Parameter Variations," *Journal of VLSI Signal Processing*, Vol. 16, No. 2/3, pp. 149–161, June/July 1997.

[44] I. S. Kourtev and E. G. Friedman, "Synthesis of Clock Tree Topologies to Implement Non-Zero Skew Schedule," *IEE Proceedings-Circuits, Devices and Systems*, Vol. 146, No. 6, pp. 321–326, December 1999.

[45] J. L. Neves and E. G. Friedman, "Optimal Clock Skew Scheduling Tolerant to Process Variations," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 623–628, June 1996.

[46] I. S. Kourtev and E. G. Friedman, "Clock Skew Scheduling for Improved Reliability via Quadratic Programming," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 239–243, November 1999.

[47] D. Velenis, K. T. Tang, I. S. Kourtev, V. Adler, F. Baez, and E. G. Friedman, "Demonstration of Speed and Power Enhancements through Application of Non-Zero Clock Skew Scheduling," *Proceedings of the ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 58–63, December 2000.

[48] V. Adler and E. G. Friedman, "Repeater Design to Reduce Delay and Power in Resistive Interconnect," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. CAS II-45, No. 5, pp. 607–616, May 1998.

[49] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F. Baez, "Reducing Power in High-Performance Microprocessors," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 732–737, June 1998.

[50] L. Benini, P. Siegel, and G. De Micheli, "Saving Power by Synthesizing Gated Clocks for Sequential Circuits," *IEEE Design & Test of Computers*, Vol. 11, No. 4, pp. 32–41, Winter 1994.

[51] V. Adler and E. G. Friedman, "Delay and Power Expressions for a CMOS Inverter Driving a Resistive-Capacitive Load," *Analog Integrated Circuits and Signal Processing*, Vol. 14, No. 1/2, pp. 29–39, September 1997.

[52] S. Natarajan, M. A. Breuer, and S. K. Gupta, "Process Variations and Their Impact on Circuit Operation," *Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 73–81, November 1998.

[53] S. R. Nassif, "Delay Variability: Sources, Impacts and Trends," *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 368–369, February 2000.

[54] K. T. Tang and E. G. Friedman, "Interconnect Coupling Noise in CMOS VLSI Circuits," *Proceedings of the ACM International Symposium on Physical Design*, pp. 48–53, April 1999.

[55] E. G. Friedman, (Ed.), *High Performance Clock Distribution Networks*, Kluwer Academic Publishers, Norwell, Massachusetts, 1997.

[56] M. Nekili, Y. Savaria, and G. Bois, "Design of Clock Distribution Networks in Presence of Process Variations," *Proceedings of the IEEE Great Lakes Symposium on VLSI*, pp. 95–102, February 1998.

[57] J. L. Neves and E. G. Friedman, "Topological Design of Clock Distribution Networks based on Non-Zero Clock Skew Specifications," *Proceedings of the*

IEEE Midwest Symposium on Circuits and Systems, pp. 468–471, August 1993.

[58] A. Vittal, L. H. Chen, M. Marek-Sadowska, K.-P. Wang, and S. Yang, "Crosstalk in VLSI Interconnections," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 18, No. 12, pp. 1817–1824, December 1999.

[59] S. R. Nassif, "Modeling and Analysis of Manufacturing Variations," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 223–228, May 2001.

[60] M. Orshansky, J. C. Chen, and C. Hu, "Direct Sampling Methodology for Statistical Analysis of Scaled CMOS Technologies," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 12, No. 4, pp. 403–408, November 1999.

[61] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani, "Analysis of the Impact of Process Variations on Clock Skew," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 13, No. 4, pp. 401–407, November 2000.

[62] M. Nekili, C. Bois, and Y. Savaria, "Pipelined H-Trees for High-Speed Clocking of Large Integrated Systems in Presence of Process Variations," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 5, No. 2, pp. 161–174, June 1997.

[63] S. Pullela, N. Menezes, and L. T. Pileggi, "Post-Processing of Clock Trees via Wiresizing and Buffering for Robust Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 6, pp. 691–701, June 1996.

[64] D. Velenis, M. C. Papaefthymiou, and E. G. Friedman, "Reduced Delay Uncertainty in High Performance Clock Distribution Networks," *Proceedings of the IEEE Design Automation and Test in Europe Conference*, pp. 68–73, March 2003.

[65] M. Kuhlmann and S. S. Sapatnekar, "Exact and Efficient Crosstalk Estimation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, No. 7, pp. 858–866, July 2001.

[66] T. Sakurai, "Closed-Form Expressions for Interconnection Delay, Coupling and Crosstalk in VLSI's," *IEEE Transactions on Electron Devices*, Vol. 40, No. 1, pp. 118–124, January 1993.

[67] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid State Circuits*, Vol. 25, No. 2, pp. 584–593, April 1990.

[68] *Affirma Spectre Circuit Simulator User Guide*, Cadence Design Systems, Inc., 2000.

[69] A. L. Fisher and H. T. Kung, "Synchronizing Large VLSI Processor Arrays," *IEEE Transactions on Computers*, Vol. 34, No. 8, pp. 734–740, August 1985.

[70] J. D. Z. Ma, A. Parihar, and L. He, "Pre-Routing Estimation of Shielding for RLC Signal Integrity," *Proceedings of the IEEE International Conference on Computer Design*, pp. 553–556, September 2001.

[71] K. M. Lepak, I. Luwandi, and H. Lei, "Simultaneous Shield Insertion and Net Ordering Under Explicit RLC Noise Constraint," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 199–202, June 2001.

[72] J. Cong, D. Z. Pan, and P. V. Srinivas, "Improved Crosstalk Modeling for Noise Constrained Interconnect Optimization," *Proceedings of the ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 14–20, December 2000.

Appendix A

Clock Tree Layouts

The clock tree layouts for the circuit examples listed in Tables 6.1, 6.2, and 6.3 are illustrated in this appendix. These layouts are produced by the buffer insertion and sizing approach and the dedicated minimal clock tree approach described in Sections 6.1 and 6.2 respectively. For each example, the aggregate buffer size, power dissipation, and clock tree area produced by those approaches are listed together with the percent change.

The physical locations of the clock buffers are represented by the dots shown on the clock trees. The relative size of each dot represents the size of the corresponding buffer.

Circuit # 1 Number of registers: 11	Aggregate buffer size	Power Dissipation (μW)	Clock tree area (μm)
Buffer insertion and sizing	21	499	3065
Dedicated minimal clock tree	10	488	3930
Change (%)	- 52 %	- 2 %	+ 28 %

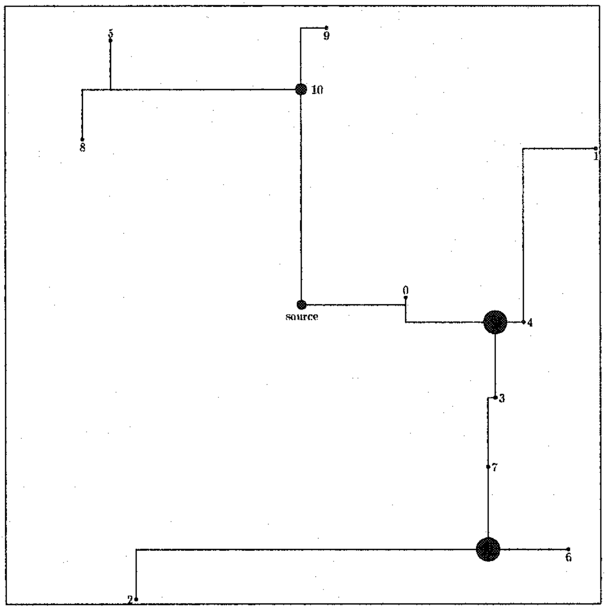


Fig. A.1: Circuit # 1 - buffer insertion and sizing

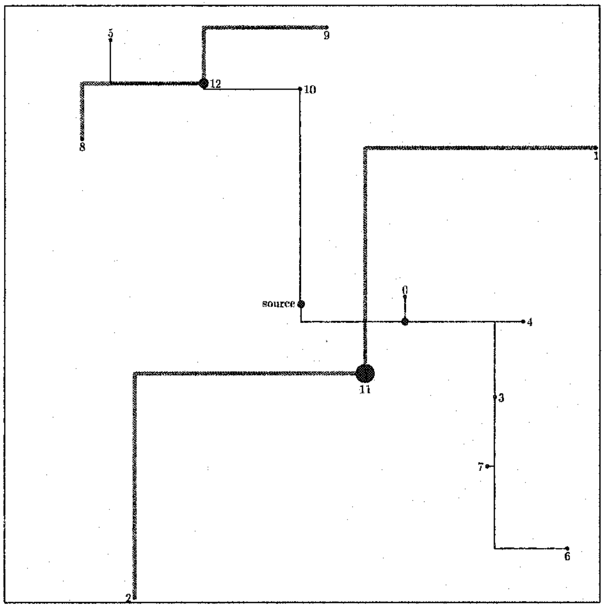


Fig. A.2: Circuit # 1 - dedicated minimal clock tree

Circuit # 2 Number of registers: 12	Aggregate buffer size	Power Dissipation (μW)	Clock tree area (μm)
Buffer insertion and sizing	18	425	2348
Dedicated minimal clock tree	5	356	2706
Change (%)	- 72 %	- 16 %	+ 15 %

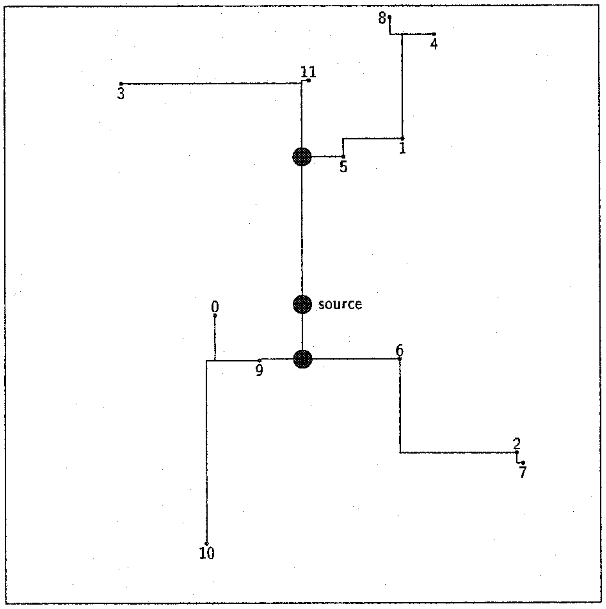


Fig. A.3: Circuit # 2 - buffer insertion and sizing

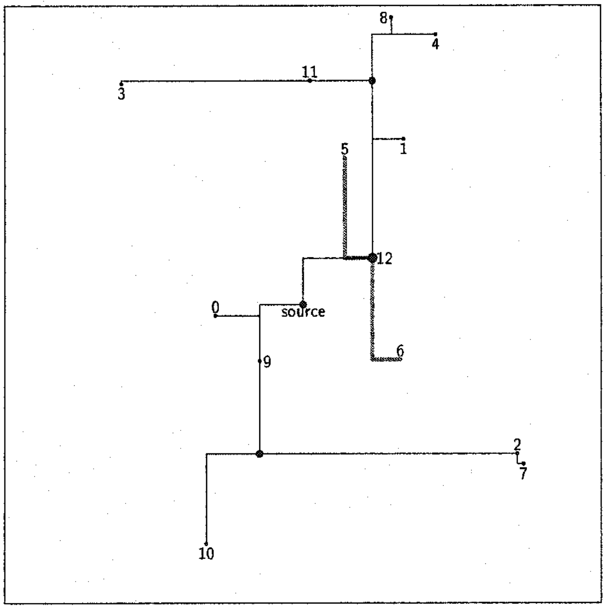


Fig. A.4: Circuit # 2 - dedicated minimal clock tree

Circuit # 3 Number of registers: 17	Aggregate buffer size	Power Dissipation (μW)	Clock tree area (μm)
Buffer insertion and sizing	29	638	3382
Dedicated minimal clock tree	7	531	4167
Change (%)	- 75 %	- 16 %	+ 23 %

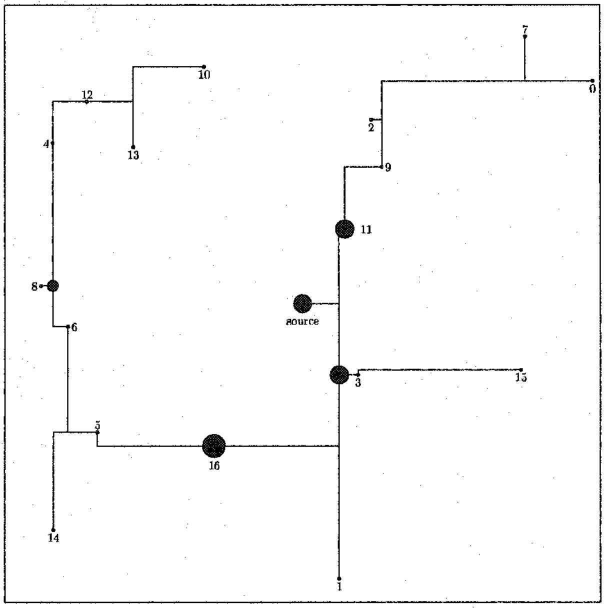


Fig. A.5: Circuit # 3 - buffer insertion and sizing

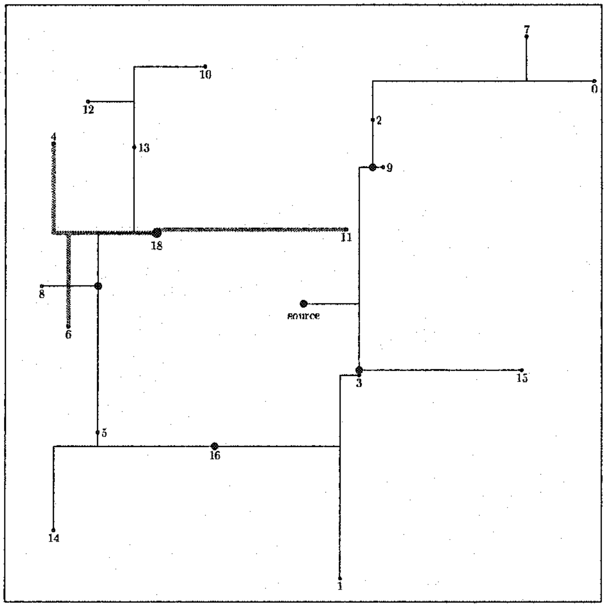


Fig. A.6: Circuit # 3 - dedicated minimal clock tree

Circuit # 4 Number of registers: 23	Aggregate buffer size	Power Dissipation (μW)	Clock tree area (μm)
Buffer insertion and sizing	34	763	3823
Dedicated minimal clock tree	8	656	5005
Change (%)	- 76 %	- 14 %	+ 31 %

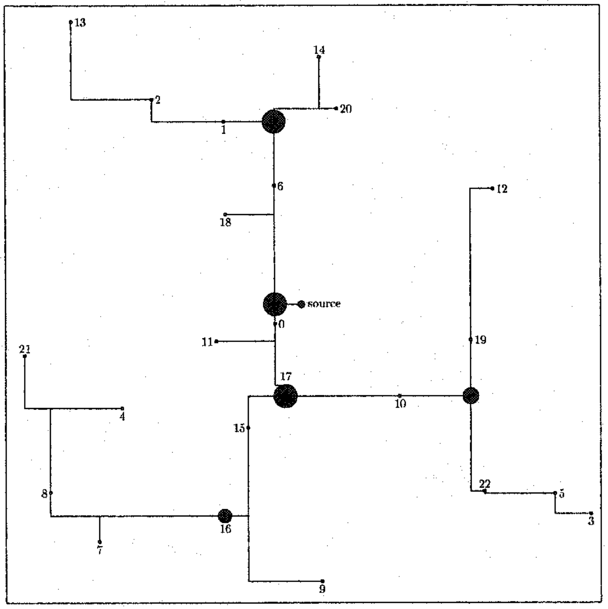


Fig. A.7: Circuit # 4 - buffer insertion and sizing

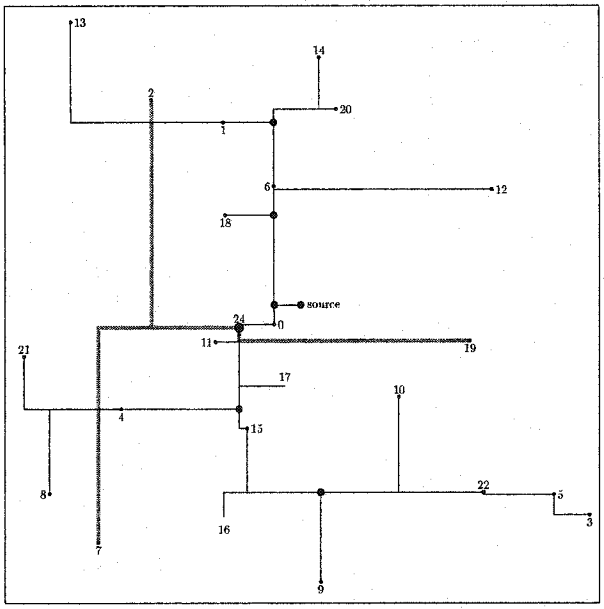


Fig. A.8: Circuit # 4 - dedicated minimal clock tree

Circuit # 5 Number of registers: 28	Aggregate buffer size	Power Dissipation (μW)	Clock tree area (μm)
Buffer insertion and sizing	30	825	4490
Dedicated minimal clock tree	9	737	5403
Change (%)	- 70 %	- 10 %	+ 20 %

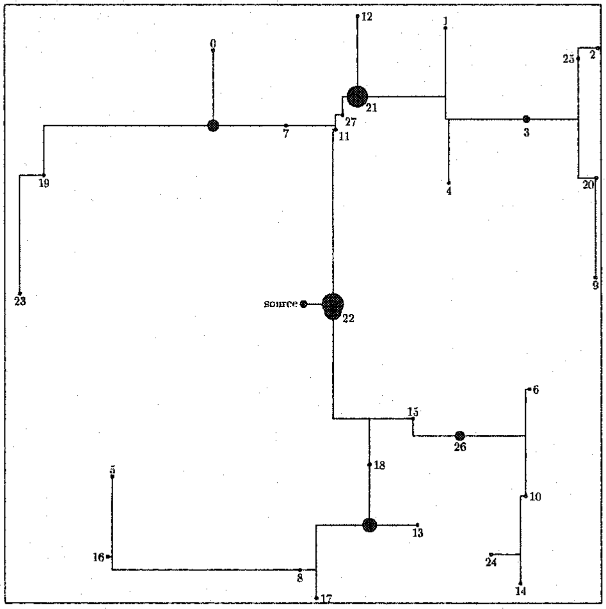


Fig. A.9: Circuit # 5 - buffer insertion and sizing

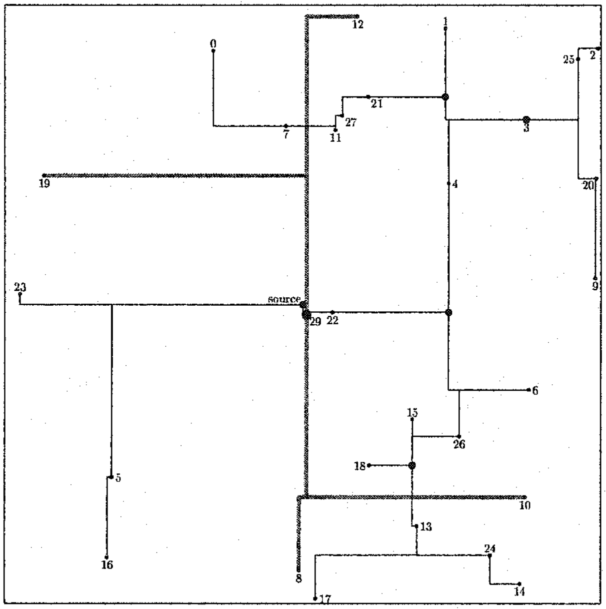


Fig. A.10: Circuit # 5 - dedicated minimal clock tree

Circuit # 6 Number of registers: 36	Aggregate buffer size	Power Dissipation (μW)	Clock tree area (μm)
Buffer insertion and sizing	25	882	4901
Dedicated minimal clock tree	10	851	5911
Change (%)	- 60 %	- 3 %	+ 20 %

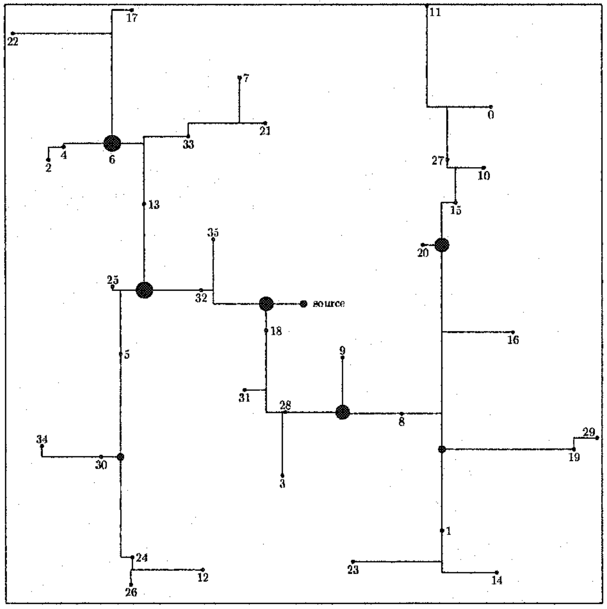


Fig. A.11: Circuit # 6 - buffer insertion and sizing

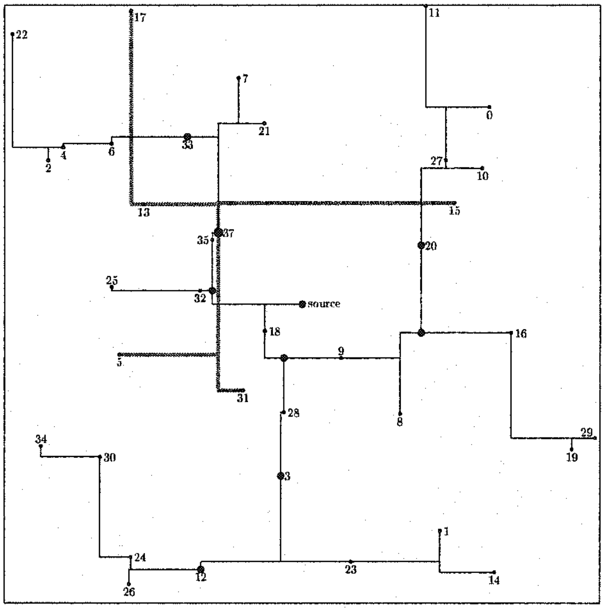


Fig. A.12: Circuit # 6 - dedicated minimal clock tree

Circuit # 7 Number of registers: 42	Aggregate buffer size	Power Dissipation (μW)	Clock tree area (μm)
Buffer insertion and sizing	15	923	4901
Dedicated minimal clock tree	9	831	5167
Change (%)	- 40 %	- 10 %	+ 5 %

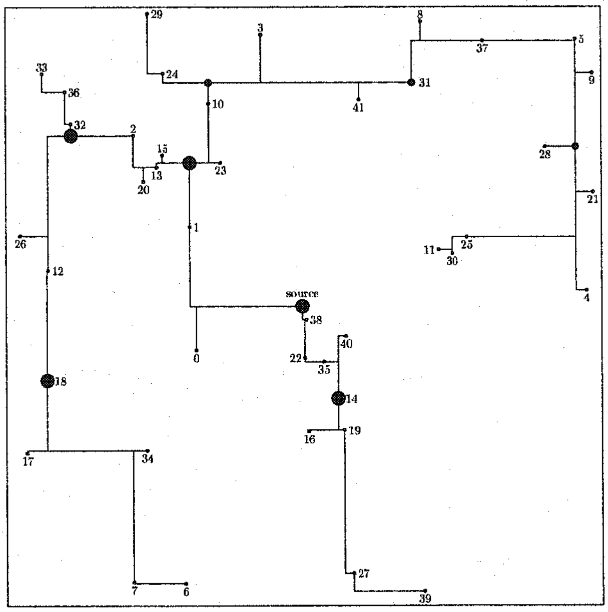


Fig. A.13: Circuit # 7 - buffer insertion and sizing

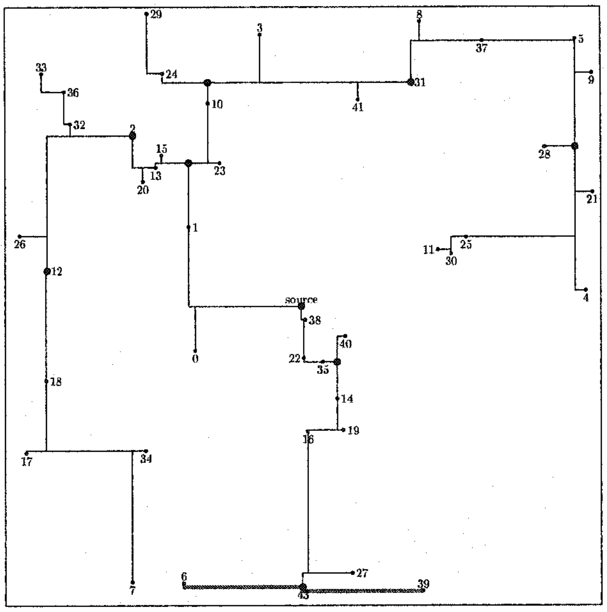


Fig. A.14: Circuit # 7 - dedicated minimal clock tree

Appendix B

Design Example

The research presented in this dissertation introduces a design methodology for enhancing the tolerance of a circuit to the uncertainty of the clock signal delay. Different design techniques are developed that manage delay uncertainty at different stages of the circuit design process. The application of these techniques on different design stages of a simple circuit is demonstrated in this Appendix.

The developed methodology is applied on an example circuit represented by the circuit graph shown in Fig. B.1. The nodes of the graph represent the clock registers of the circuit, the arches between the nodes represent the data paths, and the weight of each arch represents the delay of the corresponding data path.

The design target in this example is to tolerate delay uncertainty of up to 25% of the clock period. This target is achieved by applying the developed design

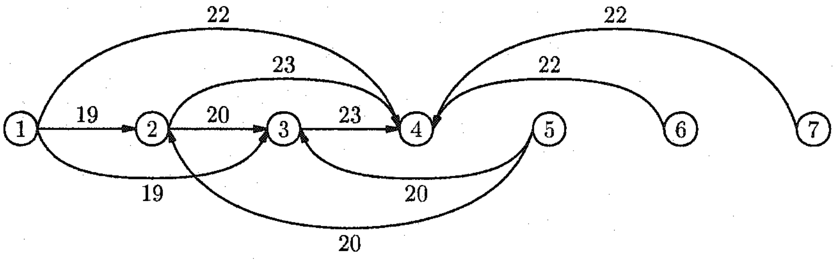


Fig. B.1: Example of circuit data paths

Table B.1: Application of different design techniques to manage delay uncertainty.

Method	**Reduction (%)**	**Delay uncertainty (% clock period)**		
Relax timing constraints	Time budget 8.6 %	25 %		
Clock tree topology extraction	33 %	16.8 %	**Power increase**	**Area increase**
Clock buffer insertion & sizing	84 %	2.6 %	73 %	0 %
Dedicated tree	69 %	5.1 %	34 %	47 %
Dedicated tree & buffer sizing	86 %	2.3 %	54 %	

techniques at different stages of the design process to manage and reduce the delay uncertainty of the clock signal. The resulting reduction in delay uncertainty is listed in Table B.1 for each of the applied techniques.

Initially, non-zero clock skew scheduling is applied to relax the timing constraints of the most critical data paths. A safety time budget of 8.6% of the clock period is introduced at the most critical data paths without penalizing the circuit performance. This safety time budget can tolerate clock signal delay uncertainty of up to 8.6% of the clock period. Therefore, the design target becomes reducing the delay uncertainty from 25% of the clock period, to less than 8.6%.

The topology of the clock tree is extracted in the next step of the design process to minimize the non-common portion of the clock signals that drive the registers of the most critical data paths. The delay uncertainty of the clock signal is reduced at those data paths by 33%, from 25% of the clock period, to 16.8%, as listed in Table B.1.

The next step in the circuit design process, following the clock tree topology extraction, is the synthesis of the clock tree layout. Two different approaches are applied to produce a clock tree layout for the circuit illustrated in Fig. B.1. Applying the buffer insertion and sizing approach reduces the delay uncertainty of the clock signal by 84%, from 16.8% of the clock period to 2.6%. Note that the uncertainty of the clock signal delay is below 8.6% of the clock period, thereby the design target is achieved. However, due to the increased size of the clock buffers the power dissipation on the clock tree is increased by 73%, as compared to a minimal rectilinear Steiner tree.

Alternatively, the dedicated minimal clock tree approach can be applied to synthesize the clock tree layout. Using a dedicated minimal clock tree to drive only the registers of the most critical data paths reduces the uncertainty of the clock signal by 69%, from 16.8% of the clock period to 5.1%, as listed in Table B.1. This approach increases the power dissipation by 34% and the clock tree wire length by 47%, as compared to a minimal rectilinear Steiner tree. Furthermore, inserting and sizing clock buffers reduces delay uncertainty by 86%, to 2.3% of the clock period. In this case, the power dissipation of the clock distribution network increases by up to 54%.

Appendix C

Publications

1. D. Velenis, M. C. Papaefthymiou and E. G. Friedman, "Reduced Delay Uncertainty in High Performance Clock Distribution Networks," *Proceedings of the IEEE Design, Automation and Test in Europe Conference*, pp. 68-73, March 2003.

2. D. Velenis, K. T. Tang, I. S. Kourtev, V. Adler, F. Baez, and E. G. Friedman, "Demonstration of Speed and Power Enhancements on an Industrial Circuit through Application of Clock Skew Scheduling," *Journal of Circuits, Systems and Computers*, Vol. 11, No. 3, pp. 231-245, June 2002.

3. D. Velenis, K. T. Tang, I. S. Kourtev, V. Adler, F. Baez, and E. G. Friedman, "Demonstration of Power Enhancements on an Industrial Circuit Through Delay Management of Non-Critical Data Paths," *Proceedings of the IEEE ASIC Conference*, pp. 30-33, September 2001.

4. D. Velenis, K. T. Tang, I. S. Kourtev, V. Adler, F. Baez, and E. G. Friedman, "Demonstration of Speed Enhancements on an Industrial Circuit Through Application of Non-Zero Clock Skew Scheduling," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, pp. 1021-1025, September 2001.

5. D. Velenis, E. G. Friedman and M. C. Papaefthymiou, "A Clock Tree Topology Extraction Algorithm for Improving the Tolerance of Clock Distribu-

tion Networks to Delay Uncertainty," *Proceedings of the IEEE International Symposium on Circuits and Systems*, Vol IV, pp. 422-425, May 2001.

6. D. Velenis, K. T. Tang, I. S. Kourtev, V. Adler, F. Baez, and E. G. Friedman, "Demonstration of Speed and Power Enhancements through Application of Non-Zero Clock Skew Scheduling," *Proceedings of the ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pp. 58-63, December 2000.