

INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 8923070

Performance limitations in synchronous digital systems

Friedman, Eby Gershon, Ph.D.

University of California, Irvine, 1989

Copyright ©1989 by Friedman, Eby Gershon. All rights reserved.

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

UNIVERSITY OF CALIFORNIA

Irvine

Performance Limitations in Synchronous Digital Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Engineering

by

Eby Gershon Friedman

Committee in charge:


Professor James H. Mulligan, Jr., Chair


Professor Jose B. Cruz, Jr.

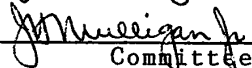
Professor Fadi J. Kurdahi

1989

The dissertation of Eby Gershon Friedman is approved,
and is acceptable in quality and form for
publication on microfilm:







Committee Chair

University of California, Irvine

1989

ii

© 1989

EBY GERSHON FRIEDMAN

ALL RIGHTS RESERVED

DEDICATION

to my parents, Joseph and Helen Friedman

my memory of them inspires my life

and ennobles my endeavors

CONTENTS

List of Tables.....	ix
List of Figures.....	x
Acknowledgments.....	xiii
Curriculum Vitae.....	xv
Abstract.....	xviii
Chapter 1: Introduction.....	1
Chapter 2: Review of Interconnect Delay in Integrated Circuits.....	7
1) Interconnect Capacitance.....	8
2) Interconnect Delay.....	13
3) Effect of Interconnect Delay on Performance.....	16
4) Summary.....	17
Chapter 3: Clock Distribution Networks.....	19
1) General Overview of Synchronous Systems.....	19
2) Delay Components of a Synchronous Digital System.....	20
3) Maximum Data Path/Clock Skew Constraint Relationship.....	23
4) Minimum Data Path/Clock Skew Constraint Relationship.....	24
5) Effects of Clock Distribution on Performance.....	27
6) Summary.....	27

Chapter 4: A Theory for Bounding Clock Skew.....	29
1) Definition of Clock Distribution Parameters.....	31
2) General Delay Equations for a Clock Path.....	34
Interconnect Delay.....	35
Buffer Delay.....	38
3) Example of Clock Skew Bounding Algorithm.....	43
4) Summary.....	48
Chapter 5: Latching Characteristics of Bistable Register.....	50
1) Bistable NAND Gate Configuration.....	50
2) Latching of Data into Register.....	52
3) Regions of Operation of Bistable Register.....	54
Region 1.....	55
Region 2.....	56
Region 3.....	60
Region 4.....	67
Register Output Waveform.....	69
4) Conditions for Latching.....	70
Necessary and Sufficient Conditions for Latching.....	71
Limiting Requirement for Latching..	72
Maximum Clock Frequency.....	74
5) Summary.....	75

Chapter 6: High Speed Synchronous Data Paths.....	76
1) Overview of Integrated Systems.....	76
2) A Data Path of a General Synchronous System.....	78
Single Stage Logic Circuit.....	80
Delay of an N Stage Cascaded Data Path.....	82
The Output of the Nth Logic Stage to the Register.....	83
Maximum and Minimum Bounds on the Data Path Delay.....	84
Determination of Output Ramp Response.....	84
3) Register.....	85
Input Latching Conditions of Register.....	85
Data-to-Clock Timing Skew.....	86
Positive Data-to-Clock Skew.....	88
Negative Data-to-Clock Skew.....	89
4) Integrated Synchronous System.....	90
Positive and Negative Clock Skew...	91
Relationship Between Clock Skew and Data-to-Clock Skew.....	91
5) Summary.....	92
Chapter 7: Data Throughput and Clock Frequency in Pipelined Data Paths.....	93
1) Data Path Delay Components.....	96
2) Definition of Data Throughput Parameters.	98

3) Design Paradigm for Pipelined Synchronous Systems.....	101
4) Design Equations Describing Data Throughput.....	107
5) Performance Cost of Latency.....	109
6) Optimal Number of Logic Stages.....	111
7) Effect of Clock Skew on Optimal Number of Logic Stages.....	114
8) Example of Algorithm.....	115
Ad Hoc Approach 1.....	118
Ad Hoc Approach 2.....	118
Algorithmic Approach.....	119
9) Maximum Performance of Optimized Data Path.....	120
10) Theoretical Maximum Clock Frequency.....	124
11) Summary.....	127
Chapter 8: Application of Theoretical Results.....	129
1) Representative Design Problem.....	131
2) Systematic Design Approach.....	132
Partitioning the Data Paths.....	133
Determining the Logic Path Delays.....	135
Determining the Register Delay Characteristics.....	136
Determining the Clock Skew Characteristics.....	137
Shaping of Register Input Waveforms.....	139
Analyzing Synchronous Data Paths...	140

3) Illustrative Design Examples.....	142
Example 1: Use of Design Procedure to Analyze Synchronous Data Paths.	143
Example 2: Performance Advantage of Integrated Synchronous Design Approach.....	146
Example 3: Derivation of Clock Frequency for a Specified Data Throughput.....	149
Example 4: Derivation of Data Throughput for a Specified Clock Frequency.....	151
Example 5: Use of Negative Clock Skew to Improve Maximum Clock Frequency.....	153
4) Summary.....	154
Chapter 9: Directions for Future Research.....	156
1) Extension of Class of Applicable Systems.....	156
2) More Sophisticated Models.....	160
3) Implementation of Algorithms in Design Tools.....	162
Chapter 10: Conclusions.....	164
References.....	170
Appendix A: BASIC Program for Generating Register Output Waveform.....	182

LIST OF TABLES

Table	Page
4.1 Comparison of SPICE Delay with Clock Delay Bounds.....	47
4.2 Range of Clock Skew Values Between Paths i, j, and k.....	47
5.1 Example of Limiting Latch Condition.....	74
7.1 Comparison of Pipelining Approaches in Example Data Path.....	119
8.1 Comparison of Algorithmic Results with SPICE Simulation.....	145

LIST OF FIGURES

Figure	Page
2.1 A Metal Line Above a Ground Plane.....	9
2.2 Three Interconnect Lines Above a Ground Plane..	12
2.3 Driver Interconnect Delay Components.....	14
3.1 Synchronous Data Path.....	20
3.2 Timing Diagram of Clocked Data Path.....	22
3.3 Clock Timing Diagrams.....	24
3.4 K-Bit Shift Register.....	26
4.1 Clock Distribution Network.....	31
4.2 Clock Distribution Network with Notational Information.....	33
4.3 Example of a Clock Distribution Network.....	44
5.1 Bistable NAND Gate Circuit Configuration.....	51
5.2 CMOS Implementation of Bistable Register.....	52
5.3 Circuit Diagram of Upper NAND Gate in Region 1.....	55
5.4 Region 1 Timing Diagram.....	56
5.5 Region 2 Circuit Configuration.....	57
5.6 Small Signal Model of Region 2.....	57
5.7 Region 2 Timing Diagram.....	58
5.8 Circuit Diagram of Region 3.....	61
5.9 Small Signal Model of Device A in Region 3.....	62

5.10	Small Signal Model of Device B in Region 3.....	62
5.11	Region 3 Timing Diagram.....	65
5.12	Region 4 Timing Diagram.....	68
5.13	Transient Response of Bistable Register.....	70
5.14	Timing Diagram of Limiting Condition for Latching.....	72
6.1	Synchronous Data Path with N Stages of Logic...	79
6.2	First Stage of N Stage Data Path.....	80
6.3	Positive and Negative Data-to-Clock Timing Skew.....	87
6.4	Positive Data-to-Clock Timing Skew.....	88
6.5	Negative Data-to-Clock Timing Skew.....	89
7.1	Synchronous Data Path with N Stages of Logic...	96
7.2	Design Paradigm for Pipelined Synchronous Systems.....	102
7.3	Effect of Positive Clock Skew and Technology on Design Paradigm.....	105
7.4	Effect of Negative Clock Skew and Technology on Design Paradigm.....	106
7.5	30 Stage Data Path.....	117
7.6	Maximum Frequency as a Function of Relative Clock Skew.....	122
7.7	Normalized Optimal Clock Frequency.....	123
7.8	Example of Theoretically Maximum Clock Frequency.....	126
8.1	Representative Synchronous Digital System.....	130

8.2	Example of an Integrated Synchronous Data Path.....	144
8.3	Example of Design Paradigm with Constraining Maximum Data Throughput.....	150
8.4	Example of Design Paradigm with Constraining Maximum Clock Frequency.....	152
8.5	Example Circuit with Two Parallel Data Paths...	154

ACKNOWLEDGMENTS

My advisor, Professor James H. Mulligan, Jr., deserves the highest praise and gratitude. In addition to his technical excellence and attention to details, he has molded my technical and professional skills into an attitude and way of thinking that should prove to have a significant impact on my life and career. I do thank you for the massive amount of time and energy you invested in me to help me develop as a professional and as a person. I know it to be a very special gift.

I would also like to express my gratitude to Professors Stubberud, Zuleeg, Healey, Cruz, Kurdahi, and the late Professor Hostetter. Their guidance and support as members of my Ph.D. committees have sharpened my technical skills and shaped my academic perspective.

I would like to acknowledge Hughes Aircraft Company for their financial and professional support under the Hughes Aircraft Ph.D. Fellowship program and in particular to express my thanks to Ron Finnilla and George Persky for supporting my involvement in this program and to my friends and coworkers at Hughes for tolerating my change of focus.

Lastly, I would like to thank my family. My wife, Laurie Ann, whose support, advice, and love guided me through this long and difficult program. I hope to replace the many lonely days and nights with years of shared happiness and achievements. And finally, to my son Joseph Shimon, whose smile lights my darkest moments, and to any future children, I hope that this effort provides a source of inspiration for you as you apply yourselves to future endeavors.

CURRICULUM VITAE

Eby Gershon Friedman

August 10, 1957	Born in Jersey City, New Jersey
1979	B.S. in Electrical Engineering Lafayette College
1979-1983	Member of Technical Staff Hughes Aircraft Company Newport Beach, California
1981	M.S. in Electrical Engineering University of California, Irvine
1983-1984	Technical Supervisor Hughes Aircraft Company Carlsbad, California
1984-1988	Section Head Hughes Aircraft Company Carlsbad, California
1988-1989	Department Manager Hughes Aircraft Company Carlsbad, California
1989	Ph.D. in Electrical Engineering University of California, Irvine Dissertation: "Performance Limitations in Synchronous Digital Systems"

PUBLICATIONS

1. E. Friedman and G. Yacoub, "A Two Level Metal, Software Compatible, CMOS/SOS Gate Array Family," Microelectronics Journal, Vol. 14, No. 6, pp. 117-118, November/December 1983 and presented at 1983 SOS/SOI Technology Workshop, October 1983.

2. S. Powell, E. Iodice, and E. Friedman, "An Automated, Low Power, High Speed Complementary PLA Design System for VLSI Applications," Microelectronics Journal, Vol. 15, No. 4, pp. 47-54, July/August 1984 and Proceedings of 1984 IEEE International Conference on Computer Design, pp. 314-319, October 1984.

3. E. Friedman, W. Marking, E. Iodice, and S. Powell, "Parameterized Buffer Cells Integrated into an Automated Layout System," Proceedings of 1985 IEEE Custom Integrated Circuits Conference, pp. 389-392, May 1985.
4. E. Friedman, "Feedback in Silicon Compilers," IEEE Circuits and Devices Magazine, Vol. 1, No. 3, pp. 15-20, May 1985.
5. E. Friedman, G. Yacoub, and S. Powell, "A CMOS/SOS VLSI Design System," Journal of Semicustom ICs, Vol. 2, No. 4, pp. 5-11, June 1985.
6. E. Friedman and S. Powell, "Design and Analysis of 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.
7. E. Friedman, "A Partitionable Clock Distribution System for Sequential VLSI Circuits," Proceedings of 1986 IEEE International Symposium on Circuits and Systems, pp. 743-746, May 1986.
8. E. Friedman et. al., "A Signal Tracking Chip Utilizing a VHSIC CMOS/SOS Structured Custom Design Methodology," Proceedings of 1986 Government Microcircuit Applications Conference, pp. 217-222, November 1986.
9. G. Yacoub and E. Friedman, "An Environment Sensitive Circuit Design Technique for Modeling VLSI Interconnect Impedances," Proceedings of 1987 Government Microcircuit Applications Conference, pp. 391-394, October 1987.
10. G. Yacoub, H. Pham, M. Ma, and E. Friedman, "A System for Critical Path Analysis Based on Back Annotation and Distributed Interconnect Impedance Models," Microelectronics Journal, Vol. 19, No. 3, pp. 21-30, May/June 1988.

PRESENTATIONS

1. E. Friedman, G. Yacoub, and S. Powell, "A Hierarchical VLSI Design System for Synthesizing CMOS/SOS Integrated Circuits," 1984 IEEE SOS/SOI Technology Workshop, October 1984.

2. E. Friedman, W. Marking, E. Iodice, and S. Powell, "Generating Parameterized Cells Using Application Specific Feedback," 1985 IEEE Physical Design Workshop, January 1985.

3. E. Friedman, "A Hierarchical Design Technique for Minimizing Clock Skew in VLSI Circuits," 1986 IEEE Physical Design Workshop, March 1986.

4. E. Friedman, "Designing a High Performance Digital Signal Processor," 1989 IEEE Physical Design Workshop on Module Generation and Silicon Compilation, May 1989.

ABSTRACT OF THE DISSERTATION

Performance Limitations in Synchronous Digital Systems

by

Eby Gershon Friedman

Doctor of Philosophy in Engineering

University of California, Irvine, 1989

Professor James H. Mulligan, Jr., Chair

Synchronous digital systems consist of functional blocks operating under the influence of a global clock signal. Fundamental performance limitations exist within these systems due to the necessary requirements of propagating data signals through logic and interconnect and synchronizing the data flow between functional blocks. These limitations depend upon the properties of the device and interconnect technologies as well as the design approach. The underlying principles necessary for the optimum design of high performance synchronous digital systems are based on these properties and have been applied to representative engineering problems.

The underlying design principles were developed by analyzing the signal transport properties of interconnect and device technologies, the transient response and latching characteristics of data registers, and the relations in time between data and clock signals. In the course of this research, these elements were investigated in detail and analytic equations were developed describing their behavior. These results were applied to the systems level problem of optimal data throughput in high speed pipelined data paths.

In order to determine the fundamental performance limitations of the complete synchronous digital system, the interdependent elements were analyzed as a single integrated system; specifically, how the clock distribution network, the registers, and the data path affect the performance of each other. This permitted the

development of an integrated approach for designing and analyzing high performance synchronous systems and represents one of the fundamental results of this research effort.

Thus, this research describes:

1) the inherent limitations of technology and design methodology on maximum synchronous performance.

2) guidelines for designing clock and data timing relationships to maximize synchronous performance.

3) an approach for designing high performance synchronous digital systems by applying the characteristics of the interconnect delay, clock distribution network, logic path, and register latching conditions to both determine and optimize the data throughput and clock frequency.

In summary, the research results presented in this dissertation provide quantitative insight into how the performance of a synchronous digital system is limited and how to design a system in order to maximize its data throughput and clock frequency.

CHAPTER 1
INTRODUCTION

Modern computers represent the most conspicuous example of the common use of digital systems. Synchronous digital systems consist of functional blocks operating under the influence of a global clock signal. Within these synchronous digital systems, there are many cost and complexity advantages to moving data as quickly as possible. Fundamental limitations to data throughput exist, however, due to the necessary requirements of propagating data signals through logic and interconnect and synchronizing the data flow between functional blocks. These limitations depend upon the properties of the device and interconnect technology as well as the design approach. In this dissertation, these properties have been incorporated into the underlying principles necessary for the optimum design of high performance synchronous systems and have been applied to representative engineering problems.

This research describes

- 1) the inherent limitations of technology and design methodology on maximum synchronous performance.

2) an approach for designing high performance synchronous digital systems by applying the characteristics of the interconnect delay, clock distribution networks, logic path, and register latching conditions in order to optimize the data throughput and clock frequency of the system.

3) guidelines for designing clock and data timing relationships to maximize synchronous data flow

Three factors affect the speed of movement of data through a synchronous digital system. These are

1) the transient response and latching characteristics of a bistable register

2) the propagation characteristics of the interconnect media and device technology

3) coincidence of the data signal and the synchronization of its waveforms

Therefore, it is desirable to analyze signal transport properties of interconnect, the transient response and latching characteristics of data registers, and the effect of relations in time between data and clock signals.

In the course of this research, each of these elements was investigated in detail and quantitative relationships were developed which consider each individually. These results were then applied to the

systems level paradigm of optimal data throughput and clock frequency in high speed pipelined data paths.

In order to determine the fundamental performance limitations of the complete synchronous digital system, each of these interdependent systems must be analyzed not just separately but as a whole; specifically, how each system interacts with each other so as to form an integrated high performance system. Therein lies the performance limitations of a synchronous digital system. Not only should the independent operations of the registers, logic and interconnect, and clock distribution networks be individually optimized but an optimally designed integrated design approach is necessary for developing high performance synchronous systems. This design methodology represents one of the fundamental results of this research effort.

In this dissertation, the effects of interconnect and device delay on the clock distribution network are examined in Chapters 2 through 4. Specifically, the propagation characteristics of sections of interconnect on signal degradation and delay are investigated in Chapter 2. Emphasis is placed on how data and clock waveforms are degraded and delayed in time, thereby skewing them from other clock and data signals.

The analysis of interconnect delay provides insight into the design and analysis of clock distribution networks and clock skew and describes how the movement of data is constrained through sequential data paths. Constraint relationships are developed for different data path/clock skew regimes. These topics are discussed in Chapter 3.

In order to apply the results developed in Chapter 3, clock distribution networks must be analyzed in terms of the interconnect and buffer characteristics of each clock delay path. Algorithms have been developed which bound the minimum and maximum clock delay of each clock signal path, thereby defining bounds on the clock skew for every clock signal path in the system. These results are described in Chapter 4.

The results presented in Chapters 2 through 4 are merged with an analysis of the register and logic path in Chapters 5 and 6, respectively, in order to analyze the total integrated synchronous system. These results are then applied to the entire synchronous digital system in Chapter 7. Specifically, in Chapter 5, the effects of the clock and data signals on the transient output response of a bistable register are discussed in detail. Special emphasis is placed on the limiting conditions for latching data into a bistable register for a set of physical and input signal conditions.

The waveform characteristics of the data signal are analyzed in Chapter 6. How the data signal is delayed and degraded through the logic and interconnect stages and latched into the final register is analyzed. Data-to-clock skew is also discussed in terms of its limiting effect on synchronous performance.

Chapter 7 contains an analysis of the data throughput/clock frequency tradeoff in terms of the fundamental characteristics of a pipelined data path. These results are described in terms of a design paradigm, relating clock frequency and data throughput to the level of pipelining. This perspective permits design equations to be presented which partition signal paths into pipelined data paths by optimizing the effects of increased latency and clock rate on data throughput. From these results, the maximum possible clock frequency is discussed in terms of the fundamental limitations of the synchronous digital system.

The application of these theoretical results is discussed in Chapter 8. The design principles developed in this dissertation are integrated and applied to representative example circuits, thereby describing how the results of this research can be used to analyze and design high performance synchronous digital systems.

In Chapter 9, possible areas of future research are described. Finally, in Chapter 10 the research results are summarized and some final conclusions are made.

CHAPTER 2

REVIEW OF INTERCONNECT DELAY IN INTEGRATED CIRCUITS

In an effort to satisfy the ever increasing demands for improved system performance, integrated circuit (IC) technology has been continually developed to improve density, speed, and power dissipation. As technology improves, IC die size, complexity, and device density increase. New lithographic and etching techniques permit reduction in minimum feature sizes. Improvements in material properties permit greater device densities and higher levels of integration. According to scaling theory [1], these smaller dimensions should improve device speed which should improve overall integrated system speed. The interconnect RC time constant, based on scaling theory, remains constant as feature sizes decrease. This occurs since the cross sectional areas of the interconnect decrease, which increases interconnect line resistance and, to first order, decreases interconnect line capacitance. However, for large circuits, interconnect time delays increase since distances between devices are much greater which can significantly degrade circuit performance [2,3]. This is particularly true for certain global signals such as clock lines.

As dimensions are scaled, chip size as well as circuit density increases, forcing the global block-to-block interconnect as well as the local interconnect to increase in length. At micron and submicron design rules, the capacitive coupling between adjacent lines and fringing capacitance to the substrate begin to dominate the parallel plate component of the total line capacitance [4-7]. When this occurs, the capacitance no longer decreases with decreasing line width and the RC time constant for a fixed length interconnect increases with decreasing physical interconnect width and spacing and dielectric thickness.

1) Interconnect Capacitance

One of the earliest efforts to model metal line interconnect capacitance was by Chang [8,9] in 1976. He developed equations from approximate conformal mapping techniques instead of the heretofore used techniques of numerical analysis. He derived equations for two physical configurations: 1) the interconnect capacitance for a finite thickness metal line over a conducting ground plane and 2) the capacitance of the same metal line with an additional conducting metal line above it. Both of these formulas were shown to be accurate within 1% for a metal line whose width, W , is equal to the dielectric thickness, h , (see Figure 2.1) and become more accurate for metal

lines whose width exceeds its surrounding dielectric thickness. Thus, his results are satisfactory (less than 1% error) for the condition, $W/h > 1$. Chang was therefore

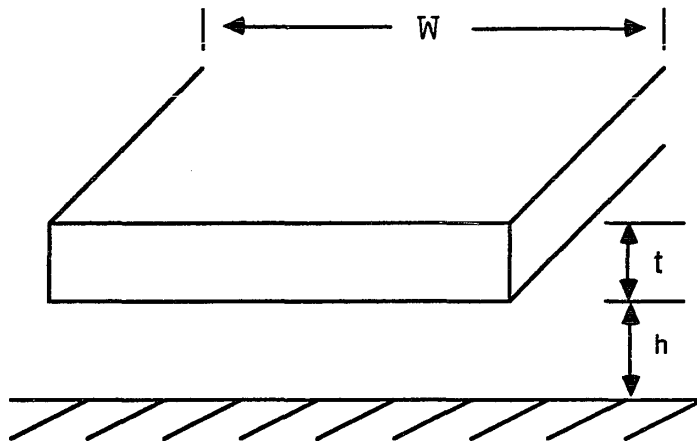


Figure 2.1: A Metal Line Above a Ground Plane

able to develop both efficient and accurate equations for computing metal interconnect capacitance. He did not extend his work to consider the effects of lines whose width to height ratio is small (less than one) and line-to-line fringing capacitance, thereby decreasing the general applicability of his equations.

Elmasry [10] developed a simple empirical equation for the capacitance of an interconnect line as given by equation (2.1),

$$C/C_1 = 1 + 2 \frac{h}{W} \ln(1 + \frac{t}{h}) + 2 \frac{t}{W} \ln(1 + \frac{W/2}{h+t}) \quad (2.1)$$

where C_1 is the conventional parallel plate capacitance given by equation (2.2) and the other physical dimensions are depicted in Figure 2.1.

$$C_1 = \frac{\epsilon_0 \epsilon_{ox} LW}{t_{ox}} \quad (2.2)$$

Equation (1) is composed of three terms, 1) the conventional parallel plate capacitance, 2) the two sided side wall capacitance, and 3) the capacitance originating from the top of the conductor. It doesn't consider the effects of parallel or crossover lines or the location of the ground plane (other than the thickness of the oxide to the silicon substrate, t_{ox} , in the parallel plate capacitor equation). The percentage error of this closed form model of the capacitance of a single interconnect line is shown to be less than 5% for W/h ratios greater than seven. For small W/h (less than seven) and t/h approximately equal to one, the percentage error increases dramatically.

Sakurai and Tamaru [11] develop closed form equations for the line capacitance of various two- and three-dimensional geometric configurations. These are empirical models but provide physical interpretation since they separate out the effects of line-to-ground and line-to-line capacitance. They show that the line capacitance is a function of interconnect width, spacing, thickness, and distance to the ground plane. For the case of three lines above a ground plane, the capacitance of the middle line per unit length is shown in equation (2.3).

$$C_3/\epsilon_{ox} = 1.15\left(\frac{W}{h}\right) + 2.80\left(\frac{T}{h}\right)^{0.222} + \left[0.06\left(\frac{W}{h}\right) + 1.66\left(\frac{T}{h}\right) - 0.14\left(\frac{T}{h}\right)^{0.222}\right]\left(\frac{S}{h}\right)^{-1.34} \quad (2.3)$$

where W is the width of the interconnect line, h is the thickness of the insulating oxide, T is the thickness of the interconnect line, and S is the spacing between interconnect lines, as shown in Figure 2.2. The relative error of this model for interconnect capacitance, as compared with two-dimensional numerical analysis, is less than 10% for physical configurations which satisfy the following conditions, $0.3 < W/h < 10$, $0.3 < T/h < 10$, and $0.5 < S/h < 20$.

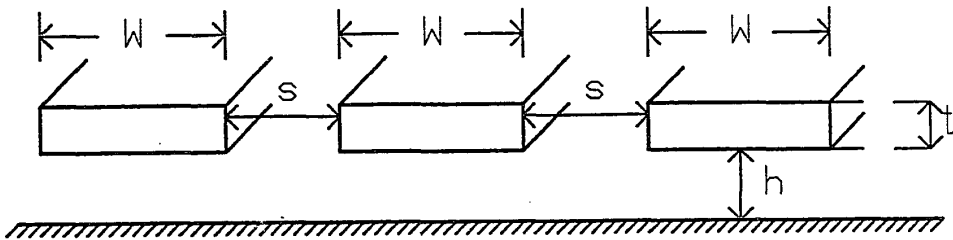


Figure 2.2: Three Interconnect Lines Above a Ground Plane

These aforementioned papers [8-11] describe physically oriented models which all improve upon the parallel plate formula for modeling interconnect line capacitance. Closed form analytic solutions are provided which can be used to model the interconnect capacitance of common geometric configurations. These empirical models replace the CPU intensive numerical analysis techniques that are commonly used to model interconnect capacitance [12-15]. The accuracy of these formulas are typically within 10%, permitting their use in estimating the interconnect capacitance of a signal path when calculating interconnect delays occurring within data paths and clock distribution networks.

2) Interconnect Delay

This section discusses interconnect transit times resulting from distributed capacitive and resistive impedances. Models are described which consider the effects of conductor and insulator materials and geometries such as height, width, thickness, and spacing on interconnect delay. Physical dimensions such as minimum feature size, chip area, and interconnect length also play an important role in the development of the models described within this section.

Sakurai in [16] describes a model for interconnect delay using lumped circuit approximations which has less than 4% error over the entire range of parameters. He describes the interconnect delay as shown in equation (2.4) below:

$$T_{int} = 1.02R_{int}C_{int} + 2.21[C_{tr}R_{tr} + C_{tr}R_{int} + R_{tr}C_{int}] \quad (2.4)$$

where R_{tr} represents the resistance of the driving transistor, C_{tr} represents the capacitance of the load transistor, and R_{int} and C_{int} represent the interconnect resistive and capacitive impedance, respectively (see Figure 2.3). The assumption that R_{tr} is constant is clearly inaccurate; however, a worst case resistance of $1/\text{maximum drain conductance}$ is used by Sakurai to model the nonlinear resistance of the transistor.

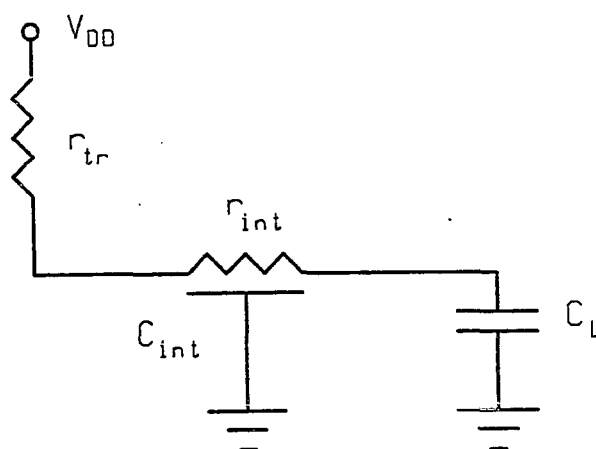


Figure 2.3: Driver Interconnect Delay Components

Bakoglu and Meindl [17,18] develop models for interconnect delay in VLSI circuits. They show that local interconnect impedance ($R_{tr} \gg R_{int}$, typically interconnect within a low level cell) remains constant with scaling; however, long distance interconnect impedance ($R_{int} > R_{tr}$, interconnect between high level functional blocks) increases quadratically with scaling. The capacitance per unit length of interconnect approaches a minimum of approximately 2 pf/cm as interconnect width and spacing decrease, assuming silicon dioxide as the dielectric material. Thus, as interconnect width and spacing and dielectric thickness scale, two-dimensional fringing capacitance begins to dominate. Their model for interconnect delay, equations (2.5) and (2.6), is similar to the model described previously by Sakurai where C_L is the input load capacitance.

$$T_{int} = 1.0R_{int}C_{int} + 2.3[R_{tr}C_{int} + R_{tr}C_L + R_{int}C_L] \quad (2.5)$$

$$\cong [2.3R_{tr} + R_{int}]C_{int} \quad (2.6)$$

Interconnect delay with repeaters placed periodically along a long interconnect line effectively transforms the interconnect impedance into a capacitive load, $T \cong 2.3R_{tr}C_{int}$, by making the interconnect resistance small with respect to the driver resistance, $R_{int} \ll 2.3R_{tr}$. The ultimate lower limit for the interconnect time delay is defined by the propagation speed of a signal in a lossless transmission line and this limit is approached as parasitic resistances are eliminated [18].

It is interesting to note that lower limits for capacitance per length (2 pf/cm) and time delay per length (0.5 ns/cm) are mentioned by Bakoglu and Meindl [18] and Yuan, Lin, and Chiang [19], respectively. Assuming a metal line resistivity of 0.1 ohm per square and a metal width of two to three micrometers (2.5 micrometers would be exact), both of these independently derived lower limits agree. Thus, for low resistivity interconnect lines, e.g., less than 0.1 ohms per square in aluminum, the propagation delay approaches 0.5 ns./cm., a fundamental limitation to moving data through an interconnect dominated circuit.

3) Effect of Interconnect Delay on Performance

In order to accurately analyze and interpret the behavior of signals in interconnect lines, the impedance of these lines must be accurately determined [20-22]. As the magnitude of these interconnect impedances approach and in many cases exceed the magnitude of the device related impedances of the signal path, it becomes of paramount importance that these RC time constants are modeled accurately.

Much effort has been extended in developing analytical and empirical models for estimating resistive and capacitive impedances in interconnect lines. Initially, isolated line [23,24] and then line-to-line [4,5,7,25] effects were explored and modeled. Two- and three-dimensional analyses were made of interconnect lines to determine their resistive [26] and capacitive [15] characteristics as a function of these added dimensions.

While local interconnect impedance ($R_{int} \ll R_{tr}$) is shown to remain constant with scaling, long distance interconnect impedance ($R_{int} > R_{tr}$) increases quadratically with decreasing feature sizes [18]. Thus, ever decreasing dimensions and increasing die size will exacerbate the effects of parasitic interconnect impedance on high performance synchronous digital systems.

Circuit design techniques, such as cascaded repeaters [27], have been developed to partially diminish the effects of interconnect impedances. However, the trend toward interconnect lines dominating the total data path delay is expected to remain due to the increase of interconnect delay with scaling.

4) Summary

Specific analytical capacitance models are described which define interconnect capacitance as a function of the width of the interconnect line, the spacing between adjacent interconnect lines, the thickness of the insulator, the thickness of the interconnect line, and the distance to the ground plane. Equation (2.3) considers fringing field effects between adjacent lines and has a relative error, as compared with two-dimensional numerical analysis, of less than 10% for physical configurations in which $0.3 < W/h < 10$, $0.3 < T/h < 10$, and $0.5 < S/h < 20$.

Sakurai describes a model for interconnect delay, equation (2.4), using lumped circuit approximations which has less than 4% error over its entire range of parameters. It is interesting to note that multiple sources derive limits of interconnect delay approaching 0.5 ns./cm., defining a fundamental limitation to moving data through an interconnect dominated circuit. With these analytical models for interconnect delay, accurate

estimates of clock skew and data-to-clock skew can be derived, permitting the improved design of clock distribution networks and data paths for high performance synchronous digital systems.

CHAPTER 3
CLOCK DISTRIBUTION NETWORKS

1) General Overview of Synchronous Systems

In a synchronous digital system, the global clock signal is used to define a relative time reference for all movement of data within that system [28-47]. Most synchronous digital systems consist of cascaded banks of sequential registers with combinatorial logic between each set of registers. The functional requirements of the digital system are satisfied by the combinatorial logic while the global performance and local timing requirements are satisfied by the careful insertion of pipeline registers into equally spaced time windows to satisfy critical worst case timing constraints [32-34,40] and the proper design of the clock distribution system to satisfy critical timing requirements as well as ensure that no race conditions can occur [40].

Each data signal typically is stored in a latched state within a bistable register awaiting the incoming clock signal to define when the data should leave the register. Once the enabling clock signal reaches the register, the data will leave the bistable register and propagate through the combinatorial network and, for a properly working system, enter the next register and be

fully latched into that register before the next clock signal appears [29,34]. This synchronous system is pictured in Figure 3.1, where C_i and C_f represent the clock to the initial register and to the final register, respectively, and both originate from the same clock signal source.

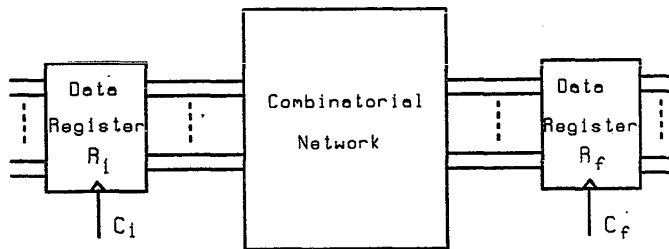


Figure 3.1: Synchronous Data Path

2) Delay Components of a Synchronous Digital System

The minimum requirement for a system to correctly operate is for the data signal to latch into the final register of its data path before the next clock signal appears. This latching requirement is discussed in detail in Chapter 5 and supplies many of the quantitative relationships necessary to provide insight into how performance becomes limited within the registers, inherent to all synchronous digital systems.

The delay components that make up a general synchronous system are composed of three individual subsystems:

- A) the memory elements
- B) the data path elements
- C) the clocking circuitry and distribution

Each of these subsystems are composed of individual delay terms which are described below:

Memory Element

- 1) T_{c-Q} - the clock-to-Q delay of the initial register element, R_i .
- 2) T_{set-up} - the set-up time of the final register element, R_f .
- 3) T_{hold} - the hold time of the register element.

Data Path

- 1) T_{int} - the delay due to the passive RC interconnect sections of the data path.
- 2) T_{logic} - the delay through the active functional logic of the data path.

Clocking Circuitry and Distribution

- 1) T_{skew} - the time difference between clock signals in a sequential data path (e.g., between C_i and C_f).

The minimum allowable clock period between two registers in a sequential data path is given by equation (3.1) below:

$$\text{Clock Period (min)} \geq T_{PD} + T_{SKEW} \quad (3.1)$$

where

$$T_{PD} = T_{C-Q} + T_{logic} + T_{int} + T_{set-up} \quad (3.2)$$

and T_{SKEW} can be positive or negative depending on whether C_f leads or lags C_i , respectively. A timing diagram depicting each delay component in equation (3.2) in terms of the clock period is shown in Figure 3.2. These waveforms show the timing requirement of equation (3.1) being barely satisfied.

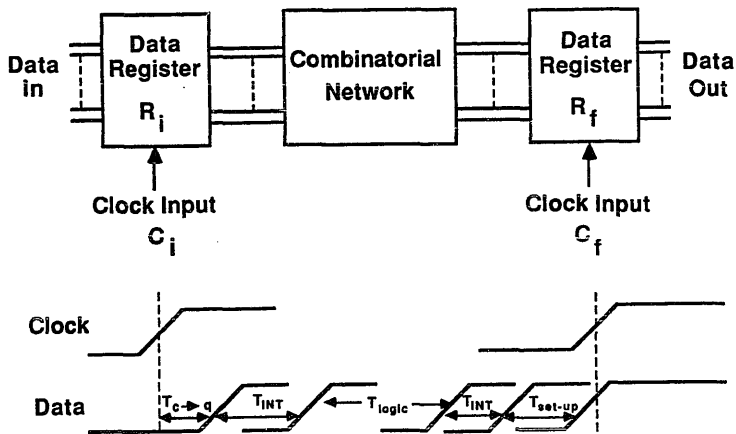


Figure 3.2: Timing Diagram of Clocked Data Path

3) Maximum Data Path/Clock Skew Constraint Relationship

For a design to meet its specified timing requirements, the greatest collective propagation delay of any data path between a pair of data registers, R_i and R_f , being synchronized by a clock distribution system must be less than the inverse of the circuit's maximum clock frequency as shown below:

$$T_{PD} + T_{SKEW} \leq T_{\text{clock period}} = 1/f_{\text{clk}} \quad (3.3)$$

When C_f leads C_i in time (see Figure 3.3A), henceforth referred to as a positive clock skew, a maximum constraint on the data path delay occurs [34,40]. In the positive clock skew case, clock and data signals feed from opposite directions. From equations (3.1), (3.2), and (3.3), the maximum permissible positive clock skew is defined below:

$$T_{SKEW} \leq T_{\text{clock period}} - (T_{c-Q} + T_{\text{int}} + T_{\text{logic}} + T_{\text{set-up}}) \quad (3.4)$$

where C_f leads C_i . This situation is the typical critical path analysis requirement commonly seen in most high performance digital synchronous systems. In circuits where positive clock skew is significant and equation (3.4) is not satisfied, the clock and data signals should be run in the same direction thereby forcing C_f to lag C_i .

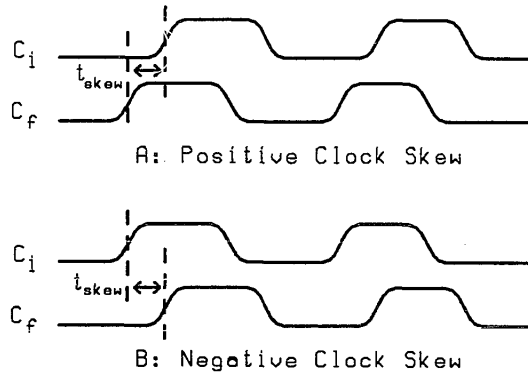


Figure 3.3: Clock Timing Diagrams

4) Minimum Data Path/Clock Skew Constraint Relationship

When C_f lags C_i in time (see Figure 3.3B), henceforth referred to as a negative clock skew, a potential minimum constraint can occur [40]. In this case, the clock skew when C_f lags C_i must be less than the time required for the data to leave the initial register, propagate through the interconnect and combinatorial logic, and set-up in the final register (see Figure 3.1). If this condition is not met before the data stored in register R_f can be shifted out of R_f , it is overwritten by the data that had been stored in register R_i and had propagated through the combinatorial logic. Correct operation requires that R_f

latches data which corresponds to the data R_i latched during the previous clock period. This constraint on clock skew is shown below:

$$|T_{\text{SKEW}}| \leq T_{\text{PD}} = T_{\text{C-Q}} + T_{\text{int}} + T_{\text{logic}} + T_{\text{set-up}} \quad (3.5)$$

where C_f lags C_i . In the negative clock skew case, clock and data signals are fed from the same direction.

An important example in which this minimum constraint occurs is in designs which use cascaded registers, such as a serial shift register or n-bit counter. As depicted in Figure 3.4, T_{logic} is equal to zero and T_{int} approaches zero. If C_f lags C_i (i.e., negative clock skew), then

$$|T_{\text{SKEW}}| \leq T_{\text{C-Q}} + T_{\text{set-up}} \quad (3.6)$$

and all that is necessary for maloperation of the system is a poor relative placement of the flip flops or a highly resistive connection between C_i and C_f . In a circuit configuration, such as a shift register or counter, where negative clock skew is a more serious problem than positive clock skew, the data and clock signals need to be run in opposing directions, forcing C_f to lead C_i , unlike the example shown in Figure 3.4.

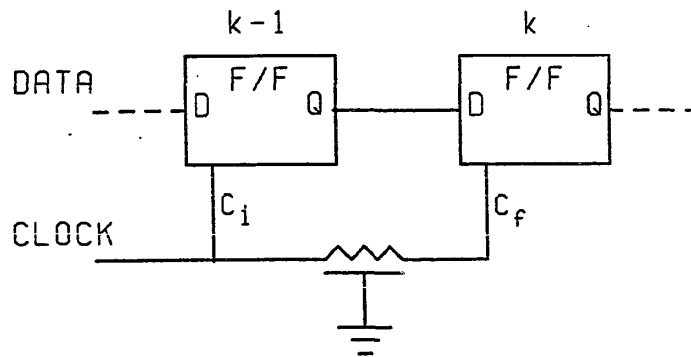


Figure 3.4: K-Bit Shift Register

As dimensions are scaled, device dependent delays such as T_{c-Q} , T_{set-up} , and T_{logic} scale as $1/S$ while interconnect delays such as T_{skew} remain constant to first order and if fringing capacitance is considered, actually increase with decreasing dimensions [18,37]. Therefore, equations (3.5) and (3.6) increase in importance as dimensions are scaled and the problem of negative clock skew becomes more significant.

Finally, as chips become functionally larger, on-chip testability is necessary. Data registers, configured in the form of serial set/scan chains during test mode, are a

common example of a built-in test design technique [48]. The placement of these circuits is typically optimized around the normal operational data flow, not when operating in the set/scan test mode. Thus, poor relative placement of the registers can occur when operating in the test mode and potentially, equation (3.6) would no longer be satisfied.

5) Effects of Clock Distribution on Performance

As was shown in sections three and four of this chapter, the magnitude and direction of the clock skew can have a significant effect on the ability to successfully move data and represent a potentially fundamental limitation to performance in a synchronous digital system. The distributed interconnect impedances seen by the clock distribution network between a set of cascaded registers create a difference in clock delay to each register. This difference, or clock skew, is a measure of the propagation characteristics between registers along the clock distribution path, as discussed in Chapter 2.

6) Summary

Clock distribution systems were analyzed in terms of their data path timing requirements. Clock skew was shown to affect performance by both the lead/lag relationship of a clock waveform to its adjacent clock waveforms along a data path as well as by its magnitude. Data path/clock

skew constraint relationships were developed for both the positive clock skew case, equation (3.4), and the negative clock skew case, equation (3.5). From these specific constraint relationships, recommended design procedures have been offered to eliminate the deleterious effects of clock skew on the maximum performance of synchronous digital systems.

CHAPTER 4
A THEORY FOR BOUNDING CLOCK SKEW

Since clock distribution networks are tree structured and contain cascaded buffers as well as distributed resistive and capacitive interconnect sections, the problem of determining the propagation delay from the clock input to all sequential elements (i.e., clock delay) and thereby determining the minimum and maximum clock skew between all sequential elements is a superset of the RC tree network problem analyzed by Penfield and Rubinstein [49,50]. These bounded interconnect delay values can be combined with the bounded gate delay values along a particular clock path to provide upper and lower time delay bounds for each clock signal path. From these bounds, the clock skew of each data path can be determined.

By analyzing the clock distribution tree network in a manner similar to the RC tree network, concise definitions of each of the clocking parameters can be derived. These definitions of clock delay and clock skew, described in section 1, can be used as a basis for the analysis of clock distribution networks in synchronous systems.

As in the Penfield-Rubinstein algorithm, time constants have been defined for specific path dependent delay elements. These augmented time constants, however, are specific to the clock distribution problem and include the effects of the cascaded buffer elements as well as the distributed interconnect sections. Upper and lower bounds on clock delay are therefore possible for any given node within a clock distribution network, thereby permitting bounds on clock skew between any pair of sequential registers. Details of this clock skew bounding algorithm are described in section 2.

These clock skew bounds can be related to the data path delays of a synchronous digital system as described in Chapter 3, thereby satisfying the minimum and maximum timing constraints of the local data path with respect to the local clock skew. An example of the clock skew bounding algorithm is provided in section 3. Finally, some concluding remarks are made in section 4.

1) Definition of Clock Distribution Parameters

The following designations of clocking parameters are used:

$$\text{Clock Delay} \triangleq T_{CD} \quad (4.1)$$

$$\text{Interconnect Delay} \triangleq T_{INT} \quad (4.2)$$

$$\text{Buffer Delay} \triangleq T_B \quad (4.3)$$

$$\text{Clock Skew} \triangleq T_{SKEW} \quad (4.4)$$

The circuit depicted in Figure 4.1 is an example of a general clock distribution configuration, where the resistor/capacitor symbol represents a distributed RC interconnect section. No constraints have been placed on the relative symmetry of the network or on the signal polarity at any of the output nodes. Nodes i, j, and k represent sequential registers being synchronized by the clocking network.

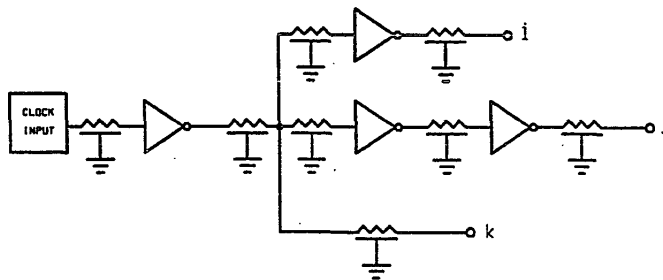


Figure 4.1: Clock Distribution Network

The clock delay, T_{CD} , is the delay from the clock input signal to a specific clocked register within a path of a clock distribution network. The clock delay can be represented as the sum total of all of the individual buffer delays along a given path added to the sum total of all of the individual interconnect delays along that same path. This relationship is expressed in equation (4.5), where T_{INT} and T_B are the interconnect delay and the buffer delay, respectively, as represented in equations (4.2) and (4.3).

$$T_{CDi} = \sum_a T_{Ba} + \sum_b T_{INTb} \quad \text{along path } i \quad (4.5)$$

Clock skew, T_{SKEW} , is the difference in the clock delay of any two nodes, i and j , within the same clock distribution network and is defined in equation (4.6).

$$T_{SKEWij} = T_{CDi} - T_{CDj} \quad (4.6)$$

Using a notation similar to that used by Penfield and Rubinstein, T_{CDi} is the clock delay from the clock input to node i and T_{CDj} is the clock delay from the clock input to node j , as shown in Figure 4.2. The sign of T_{SKEW} is dependent on the lead/lag relationship between nodes i and j . This relationship has significant consequences when the clock signals are related to the flow of the data signals as discussed in Chapter 3.

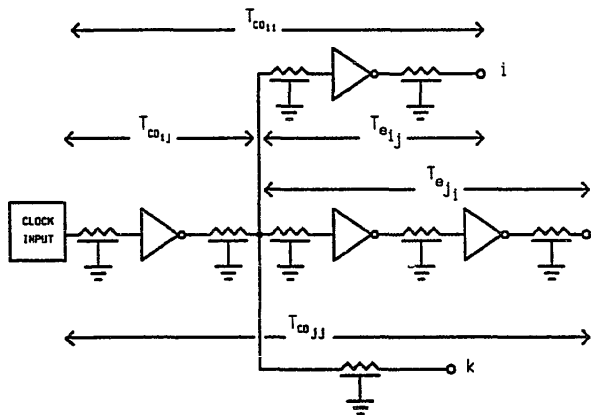


Figure 4.2: Clock Distribution Network with Notational Information

Continuing this same notational format, T_{CDij} is the clock delay in common with paths i and j and is represented in Figure 4.2 by the initial "trunk" of the tree making up the clock distribution system. A new delay element is defined, T_{Eij} , which is the difference in delay between the clock delay of path i , T_{CDii} , and the common clock delay, T_{CDij} , as shown in equation (4.7). Alternatively, T_{Eij} is the time delay peculiar to the specific clock path i and independent of any delay elements of any other clock path j in which clock skew information is being inferred.

$$T_{Eij} = T_{CDii} - T_{CDij} \quad (4.7)$$

Thus, the total clock skew between any two nodes, i and j , is the difference in their respective T_E terms, as shown in equation (4.8).

$$\begin{aligned} T_{SKEW} &= T_{CDii} - T_{CDjj} \\ &= T_{CDij} + T_{Eij} - (T_{CDij} + T_{Eji}) \\ &= T_{Eij} - T_{Eji} \end{aligned} \quad (4.8)$$

This agrees with an intuitive perspective which implies that when two paths branch out from the same node, their signal delay will differ by the amount of asymmetry between those two paths.

2) General Delay Equations for a Clock Path

As shown in equation (4.9), the total delay along a distributed clock path i , T_{CDii} , can be modeled as the summation of its individual interconnect and buffer delays.

$$T_{CDii} = \sum_a T_{Ba} + \sum_b T_{INTb} = T_{Bi} + T_{INTi} \quad \text{along path } i \quad (4.9)$$

where

$$T_{INTi} = \sum_{b=1}^m T_{INTb} = T_{INT1} + T_{INT2} + \dots + T_{INTm} \quad \text{along path } i \quad (4.10)$$

and

$$T_{Bi} = \sum_{a=1}^n T_{Ba} = T_{B1} + T_{B2} + \dots + T_{Bn} \quad \text{along path } i \quad (4.11)$$

Each T_{INT} term represents the cumulative interconnect delay between each buffer and each T_B term represents the delay of one buffer driving a specific interconnect impedance. The total number of buffers along a path i is represented by n , while m is the total number of distributed interconnect impedances between buffer stages along the same path i .

Interconnect Delay

Each interconnect term in equation (4.10) represents the total delay along a clock path composed of distributed interconnect sections between two serial buffers. Therefore, each of these interconnect impedance components represents an individual distributed interconnect problem and can be solved by applying the Penfield-Rubinstein algorithm. Once each term has been determined, the total interconnect delay along a clock path can be derived by summing each term as shown in equation (4.10).

As described in [50], upper and lower bounds of an RC delay can be derived based on specific path dependent time constants [these are shown in equations (4.12) - (4.14)], where each of the terms making up these time constants have been redefined for our clock distribution problem.

$$T_P = \sum_k R_{kk} C_k \quad (4.12)$$

$$T_{Di} = \sum_k R_{ki} C_k \quad (4.13)$$

$$T_{Ri} = \frac{\sum_k R_{ki}^2 C_k}{R_{ii}} \quad (4.14)$$

i and k represent arbitrary nodes of a path in which the next buffer or a sequential memory element in the clock distribution path appears. Node i is the node to which the incremental delay is measured.

R_{ii} is the resistance along the unique path between the previous buffer and node i .

R_{kk} is the resistance along the unique path between the previous buffer and node k .

R_{ki} is the resistance of that portion of the unique path between the previous buffer and node i that is common with the unique path between the previous buffer and node k .

C_k is the lumped capacitance at node k , representing the interconnect and device load capacitance at that node.

With these definitions, upper and lower bounds on the time delay can be developed for each incremental distributed interconnect impedance. These are shown below where equations (4.16) and (4.18) represent a tighter bound than equations (4.15) and (4.17), respectively, if the constraint equation on $V_i(t)$ is satisfied.

Upper Bound on Time

$$t \leq \frac{T_{Di} - T_{Ri}}{1 - V_i(t)} \quad (4.15)$$

$$t \leq T_P - T_{Ri} + T_P \ln \frac{T_{Di}}{T_P [1 - V_i(t)]} \quad \text{for } V_i(t) \geq 1 - T_{Di}/T_P \quad (4.16)$$

Lower Bound on Time

$$t \geq T_{Di} - T_P [1 - V_i(t)] \quad (4.17)$$

$$t \geq T_{Di} - T_{Ri} + T_{Ri} \ln \frac{T_{Ri}}{T_P [1 - V_i(t)]} \quad \text{for } V_i(t) \geq 1 - T_{Ri}/T_P \quad (4.18)$$

where t is the incremental interconnect delay, T_{INTb} , from the buffer output to node i and is always a non-negative quantity and $V_i(t)$ is the normalized voltage at node i .

Thus, from equations (4.15) - (4.18), one can determine the time delay of each distributed interconnect impedance component between a pair of buffers or a buffer and a memory element. These incremental delays can then be summed as shown in equation (4.10) to give the minimum and maximum total interconnect delay of a specific clock path, T_{CDii} .

Buffer Delay

As represented in equation (4.11), the total time delay of a clock path composed of cascaded buffers can be modeled as the sum of its individual buffer delays. An algorithm describing the delay through cascaded MOS devices has been developed by Lee and Soukup [51] and is given below:

$$T_D = \frac{C_o}{G_o} \sum_{n=1}^N R_n + N \frac{C_{in} + C_w}{G_o} \left(\frac{C_L}{C_i} \prod_{n=1}^N R_n \right)^{1/N} \quad (4.19)$$

where G_o is the incremental output conductance, C_o is the local parasitic capacitance at the output, C_{in} is the input capacitance, C_w is the local interconnect capacitance, C_L is the load capacitance at the output of the gate, N is the number of stages, and R_n is the number of serial devices within a gate. For a single stage buffer repeater used to drive a portion of the clock distribution network, N is one and R_n is one and therefore equation (4.19) becomes

$$\begin{aligned} T_d = T_{Ba} &= \frac{C_o}{G_o} + \frac{C_{in} + C_w}{G_o} \frac{C_L}{C_{in}} \\ &= \frac{1}{G_o} (C_o + C_L + C_L C_w) \end{aligned} \quad (4.20)$$

The ratio C_L/C_i is used to estimate the local wiring capacitance between cascaded stages. Therefore, for a single stage buffer, C_w represents only the local wiring capacitance intrinsic to the buffer and not the interconnect impedance between buffer stages as discussed previously.

In order to provide bounds on the time delay through a buffer stage for a given capacitive load environment, minimum and maximum values of output conductance must be determined [16]. The maximum delay through a buffer occurs when one of its two devices is dominant and is saturated ($V_{GS} - V_T \leq V_{DS}$) and the minimum delay occurs when both devices are operating in the linear region ($V_{GS} - V_T \geq V_{DS}$). This assumes that for a given output waveform polarity, the saturated device output conductance $G_o(\text{sat})$ is dominated by either the N-channel or the P-channel conductance while the linear output conductance $G_o(\text{lin})$ is the sum of the P-channel conductance G_{oP} and the N-channel conductance G_{oN} . These two conditions provide the smallest and largest conductance values, thereby bounding the buffer output conductance. This bounding technique also assumes that the P-channel and N-channel devices are ratioed to provide equal output conductances (and equal rise times and fall times).

In the linear region,

$$I_{DS} = K'S[2(V_{GS} - V_T)V_{DS} - V_{DS}^2] \quad (4.21)$$

where K' is the transconductance parameter of the device and S is the ratio of the channel width to the channel length, W/L .

In the saturation region,

$$I_{DS} = K'S(V_{GS} - V_T)^2(1 + \lambda V_{DS}) \quad (4.22)$$

where the channel-length modulation parameter $\lambda = 1/V_a$ is the reciprocal of the Early voltage and represents the effect of V_{DS} on the drain current during saturation. Typical values of λ are in the range 0.01 to 0.1 V^{-1} [52].

By taking the derivative of I_{DS} with respect to V_{DS} , the incremental channel conductance can be derived for a specific operating point within each region.

$$G_o = \left. \frac{dI_{DS}}{dV_{DS}} \right|_{\text{operating point}} = \begin{cases} 2K'S(V_{GS} - V_T - V_{DS}) & \text{linear region} \quad (4.23) \\ K'S\lambda(V_{GS} - V_T)^2 & \text{saturation region} \quad (4.24) \end{cases}$$

Thus, the channel conductance G_o , when operating in an ON mode, has a value whose range can be estimated to be between the incremental saturation conductance and the sum of the P-channel and N-channel incremental linear conductances where $G_o(\text{lin}) = G_{oP} + G_{oN}$ and each G_o term is derived from equation (4.23).

$$G_o(\text{sat}) \leq G_o \leq G_o(\text{lin}) \quad (4.25)$$

The estimate of the saturated or the total linear output conductance G_o inserted into equation (4.20) provides a first order approximation of the maximum and minimum delay of a single stage CMOS buffer. These values are then individually summed as shown in equation (4.11) to provide the total buffer related time delay of a specific clock path, T_{CDii} .

The effect of interconnect resistance between buffer stages on the upper and lower buffer delay bounds requires comment. Equations (4.19) and (4.20) assume that the resistive portion of the total output RC impedance is represented by the incremental output conductance G_o and the output load is purely capacitive. As described by Sakurai [16], the effect of interconnect resistance on device delay is dependent upon the magnitude of the resistive ratio R_w/R_o , where R_w is the line resistance and R_o is the output device resistance. The smaller the

magnitude of the resistive ratio, the more accurate equations (4.19) and (4.20) become. For the purpose of this analysis, the upper bound and the lower bound need to be considered separately.

1) Lower Bound (Minimum Delay)

- a theoretical minimum buffer delay occurs when R_w is equal to zero, as assumed in equation (4.19). Thus, the ratio R_w/R_o represents a lower bound when equal to zero and no constraint on R_w occurs when determining the minimum buffer delay.

2) Upper Bound (Maximum Delay)

- since the output conductance, as defined in equation (4.24), is small, the output resistance is large. Therefore, the ratio R_w/R_o is typically small for small values of R_w . However, in order for equation (4.20) to accurately represent the maximum delay through a CMOS buffer, the following constraint is placed on R_w when determining the buffer related upper bound.

$$R_w \ll 1/G_o(\text{sat}) \quad (4.26)$$

Different ratios of R_w/R_o have been tabulated by Sakurai as a function of acceptable error (for the delay of a single buffer driving an RC load) and are provided in [16].

The total delay of any given clock path can then be derived by adding the total interconnect delay to the total buffer delay of that path as described in equation (4.9). This summation represents an approximate model since it does not consider the shape of the signal waveform. However, from this information the clock skew between any two clock paths, i and j , is readily available by taking the difference in delay between any two paths i and j .

3) Example of Clock Skew Bounding Algorithm

In order to describe the use of the aforementioned clock skew bounding algorithm, an example clock distribution network is shown in Figure 4.3. Nodes i , j , and k represent sequential registers being synchronized by the clock distribution network. Minimum and maximum clock delays have been derived for each clock path and compared with delay values for each path generated from SPICE [53] using Level 1 Shichman-Hodges device equations [54].

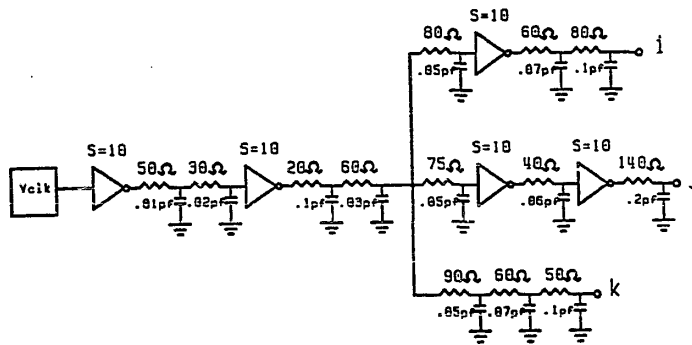


Figure 4.3: Example of a Clock Distribution Network

The following parameter values were used to characterize the transistor devices:

$$K' = 2.158 \times 10^{-5} \text{ Amperes/volt}^2$$

$$\lambda = 0.05 \text{ Volts}^{-1}$$

$$U_0 = 500 \text{ cm}^2/\text{Volt-second}$$

$$L = 2 \text{ micrometers}$$

$$W = 20 \text{ micrometers}$$

$$C_0 = 0.005 \text{ picofarads}$$

$$C_w = 0.001 \text{ picofarads}$$

C_i from equation (4.20) is shown in equation (4.27) and for the cited example has a value of 0.035 pf. From equations (4.23) and (4.24), $G_o(\text{lin})$ and $G_o(\text{sat})$ have values of 1.500×10^{-3} mhos. and 1.726×10^{-4} mhos.,

$$C_i = \frac{2K' WL}{U_o} \quad (4.27)$$

respectively. This example does not consider the variation of device parameters such as K' and mobility on the time response of the buffers and interconnect. One can use such worst case parameters for the maximum clock delay and best case parameters for the minimum clock delay to further expand the delay bounds and thereby encompass known process variations.

By using equation (4.20) to determine the individual buffer delays, the total maximum and minimum buffer delays for a given path can be derived. For path i in Figure 4.3, which contains three buffers, the following values are derived:

$$\begin{aligned} T_{Bi}(\text{max}) &= T_{B1\text{max}} + T_{B2\text{max}} + T_{B3\text{max}} & (4.28) \\ &= 2.769 \text{ ns} \end{aligned}$$

$$\begin{aligned} T_{Bi}(\text{min}) &= T_{B1\text{min}} + T_{B2\text{min}} + T_{B3\text{min}} & (4.29) \\ &= 0.319 \text{ ns} \end{aligned}$$

Thus, the total buffer delay along path i ranges between 0.319 ns. to 2.769 ns.

Each of the individual interconnect delay components of path i in Figure 4.3 is individually summed as described in equations (4.15) - (4.18), where a value of $V_i = 0.5$ is used. The total interconnect delay for path i in the example shown in Figure 4.3 is given below:

$$\begin{aligned} T_{INTi}(\max) &= T_{INT1\max} + T_{INT2\max} + T_{INT3\max} & (4.30) \\ &= 0.073 \text{ ns} \end{aligned}$$

$$\begin{aligned} T_{INTi}(\min) &= T_{INT1\min} + T_{INT2\min} + T_{INT3\min} & (4.31) \\ &= 0.014 \text{ ns} \end{aligned}$$

Thus, the total maximum and minimum delay for path i in the clock distribution network in Figure 4.3 is given below in equations (4.32) and (4.33), respectively.

$$T_{CDi}(\max) = \sum_a T_{Ba\max} + \sum_b T_{INTb\max} \quad (4.32)$$

$$T_{CDi}(\min) = \sum_a T_{Ba\min} + \sum_b T_{INTb\min} \quad (4.33)$$

Bounded clock delays for each path, i , j , and k , in Figure 4.3 are provided in Table 4.1 and compared with SPICE Level 1 delay results, defined at the 50% point. Thus, upper and lower bounds on the clock delay for each path described in Figure 4.3 have been derived. It is

clearly seen from Table 4.1 that $T_{CD}(\min)$ and $T_{CD}(\max)$ form lower and upper bounds, respectively, on the clock delay as compared with the SPICE generated clock delay of each path.

Path	$T_{CD}(\text{SPICE})$	$T_{CD}(\text{Min})$	$T_{CD}(\text{Max})$
i	0.97 ns.	0.33 ns.	2.84 ns.
j	1.15 ns.	0.43 ns.	3.62 ns.
k	0.58 ns.	0.29 ns.	2.60 ns.

Table 4.1: Comparison of SPICE Delay with Clock Delay Bounds

With this information, maximum negative/positive clock skew can be derived, as shown in Table 4.2. This information can be used in equations (3.4) and (3.5) to assist in the design and analysis of high performance clock distribution networks. The type of information represented in Table 4.2 and applied to equations (3.4)

<u>Paths</u>	<u>Bounded Clock Skew Values</u>
ij	-2.41 ns. to +3.29 ns.
ik	-2.55 ns. to +2.27 ns.
jk	-3.33 ns. to +2.17 ns.

Table 4.2: Range of Clock Skew Values Between Paths i, j, and k

and (3.5) can also be utilized in timing analysis tools to automatically flag timing problems related to clock distribution networks in high performance synchronous circuits [55-59]. Timing analysis tools generally ignore clock skew and therefore cannot accurately account for the effect of clock skew on critical timing paths (positive clock skew) and race conditions (negative clock skew). Therefore, the reliability of timing analyzers in determining critical paths and race conditions is limited by the absence of and can be improved by the addition of clock skew bounds such as described within this chapter.

4) Summary

This chapter describes an algorithm which bounds the delay of a clock distribution tree, thereby defining the minimum and maximum clock delay of each clock signal path. From this information, the lead/lag behavior and the magnitude of the clock skew of each local data path are easily derived.

The clock distribution problem has been shown to be a superset of the RC tree network problem analyzed by Penfield and Rubinstein. Concise definitions of each of the parameters belonging to a clock distribution network have been developed and a general algorithm for generating theoretical upper and lower bounds on clock skew has been derived. An example was given describing the use of the

clock skew bounding algorithm for determining maximum positive and negative clock skew.

The bounding of clock delays, and therefore clock skew, is directly applicable to the design and analysis of synchronous digital systems by permitting the efficient evaluation of clock skew and its effect on critical worst case timing paths, in the positive clock skew case, and on the proper design of cascaded registers, in the negative clock skew case. The relative clock skew of each path can also be utilized to enhance the accuracy of timing analysis tools by permitting these tools to consider the effect of clock skew on each data path when evaluating their specific timing requirements. Thus, with a theoretical basis for analyzing clock distribution networks and practical examples describing their use, these results will provide a more formal, systematic methodology for developing and analyzing high performance clock distribution networks.

CHAPTER 5

LATCHING CHARACTERISTICS OF BISTABLE REGISTER

The specific emphasis of this research is the analysis of the fundamental limitations of moving data through a synchronous digital system. The minimum functional requirement of all synchronous digital systems is the ability to latch data into a register element. A fundamental form of a register element is the bistable latch configuration which can be constructed from either two NAND gates or two NOR gates. Either circuit performs the basic latching operation upon which other more complicated types can be constructed [60,61].

1) Bistable NAND Gate Configuration

The NAND gate implementation of the bistable latch has been chosen (shown in Figure 5.1) instead of the NOR gate version, since it performs better in CMOS technology (due to the higher mobilities of the serial N-channel devices than of the serial P-Channel devices) and is therefore more commonly used. However, all physical theory and algorithmic solutions described in this chapter are easily applied to a NOR gate implementation of the bistable register.

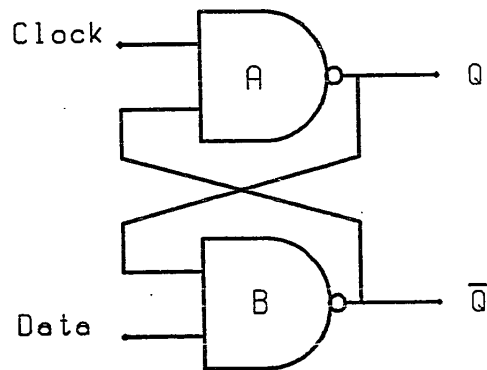


Figure 5.1: Bistable NAND Gate Circuit Configuration

A CMOS implementation of the bistable NAND gate structure has been chosen to evaluate how data is fundamentally performance limited by the ability to latch data into a register. The CMOS bistable register circuit is shown in Figure 5.2. The clock signal drives the input of one NAND gate, A, and the data signal is the input to the second NAND gate, B. The other input of each NAND gate is furnished by the output signal of its complementary NAND gate. Given an initial voltage at V_1 and its complement at V_2 , the input data and clock signal are chosen so as to maintain or flip the output logic states. Once the new state of the register is defined, the input data is considered to have been latched into the register.

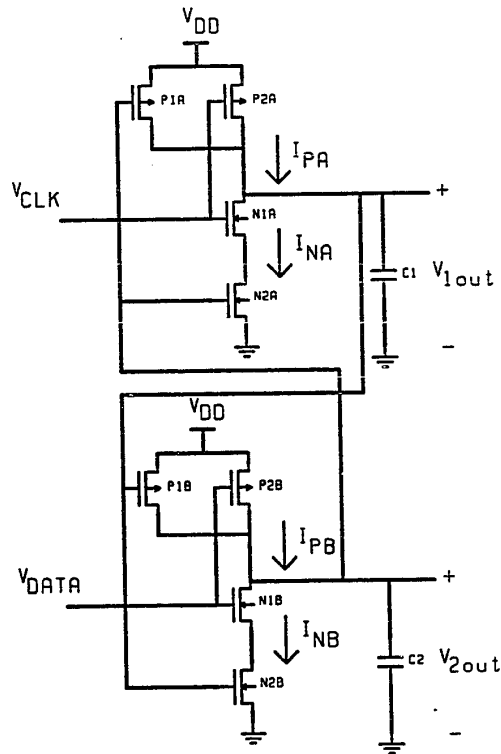


Figure 5.2: CMOS Implementation of Bistable Register

2) Latching of Data into Register

In order to latch data into a register, the clock and data signals must appear at the input of the register at the correct relative time and at their correct voltage magnitudes. These time and voltage requirements are

described in analytical form in sections three and four of this chapter.

The initial conditions of $V_1(0) = 0$ volts and $V_2(0) = 5$ volts have been chosen to exemplify the latching phenomenon. Assuming the clock signal, V_{CLK} , is at 5 volts and the data signal, V_{DATA} , is at 0 volts, the circuit exists in a restoring equilibrium state. In order to change the polarity of the output voltages at V_1 and V_2 , both the clock and the data input signals must switch. The clock signal must decrease from 5 volts and the data signal must increase from ground.

As seen in Figure 5.2, as V_{CLK} decreases from V_{DD} to $V_{DD} + V_{TP}$, no current will flow in the top device since both the P-channel devices remain in cutoff. Once V_{CLK} equals $V_{DD} + V_{TP}$, P_{2A} turns on and enters the saturation region. This permits current to flow within the top NAND gate. Once I_{PA} becomes greater than I_{NA} , V_{1out} will increase. When V_{1out} equals V_{TN} , it turns on the lower N-channel device of the lower NAND gate. If V_{DATA} is greater than V_{TN} of the top device plus V_{DS} of the lower device, the lower NAND gate can also conduct current. Assuming these conditions exist and V_{CLK} continues to decrease (thereby increasing V_{1out}), the bistable NAND gate register will enter the regenerative latch mode. Thus, as V_{1out} increases above V_{TN} and assuming V_{DATA} is above V_{TN} , the N-channel tree of the lower device will

sink current to ground. Once I_{NB} becomes greater than I_{PB} , V_{2out} will decrease from its equilibrium potential of V_{DD} volts. As V_{2out} decreases below $V_{DD} + V_{TP}$, P_{1A} turns on and this further accelerates the rising voltage at V_{1out} which in turn further decreases V_{2out} . This closed loop regenerative action permits the bistable register to quickly respond to its changing input signals and to latch the input data and in effect to change the state of the register. As V_{1out} increases toward V_{DD} , V_{DS} across both of the P-channel devices becomes very small and the amount of output voltage change due to a change in input voltage decreases until the regenerative loop is broken. The final region of operation is a non-regenerative open loop in which the P-channel devices charge the output capacitor C_1 up to V_{DD} . These four regions of operation are quantitatively described in the next section.

3) Regions of Operation of Bistable Register

The response of the bistable register to its changing input signals can be broken up into four separate regions. Each region represents the bistable register operating under different circuit conditions and therefore each region has a different output response.

Region 1

As V_{CLK} decreases from V_{DD} to $V_{DD} + V_{TP}$, no current can flow through the upper NAND gate (see Figure 5.3) since 1) both P-channel devices are in cutoff, P_{2A} due to $V_{CLK} \geq V_{DD} + V_{TP}$ and P_{1A} due to $V_{2out} \geq V_{DD} + V_{TP}$, and 2) V_{1out} is at the same potential as the sources of the two N-channel devices. Thus, region 1 represents the time

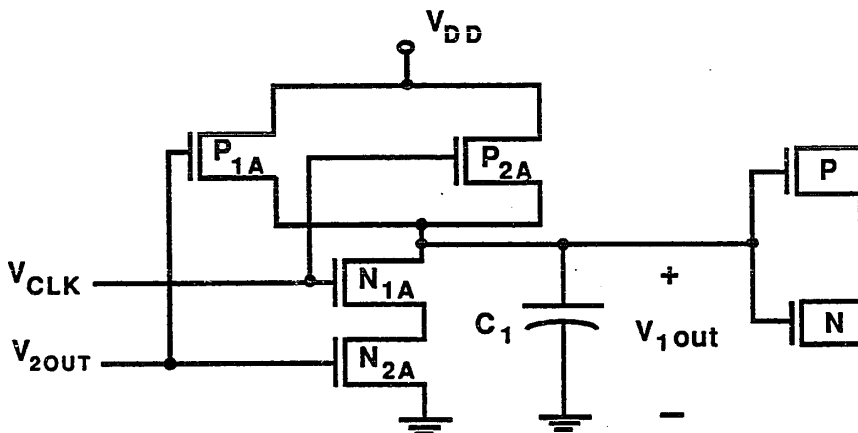


Figure 5.3: Circuit Diagram of Upper NAND Gate in Region 1

required for V_{CLK} to reach $V_{DD} + V_{TP}$ and turn on P_{2A} , thereby permitting current to flow. Throughout this region V_{1out} remains at 0 volts, as shown in Figure 5.4, and the time delay, T_1 , of this region is given by equation (5.1) below:

$$T_1 = |V_{TP}/k_c| \quad (5.1)$$

where k_c is the fall time of the clock signal in volts/second and V_{TP} is the threshold voltage of the P-channel devices and for an enhancement mode device is negative.

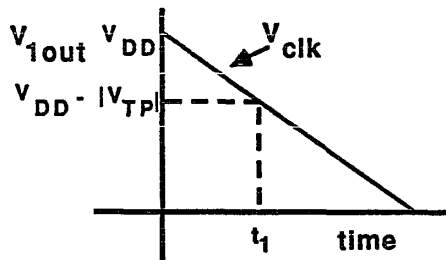


Figure 5.4: Region 1 Timing Diagram

Region 2

Once V_{CLK} decreases below $V_{DD} + V_{TP}$, current will flow between V_{DD} and ground. As V_{CLK} decreases further, the current supplied by P_{1A} will become greater than the current sunk by the N-channel tree. Once this occurs, $V_{1out}(t)$ will begin to rise. In this region, the bistable NAND gate register can be represented by a single NAND gate with one changing input, (since $V_{2out} = V_{DD}$) as shown in Figure 5.5.

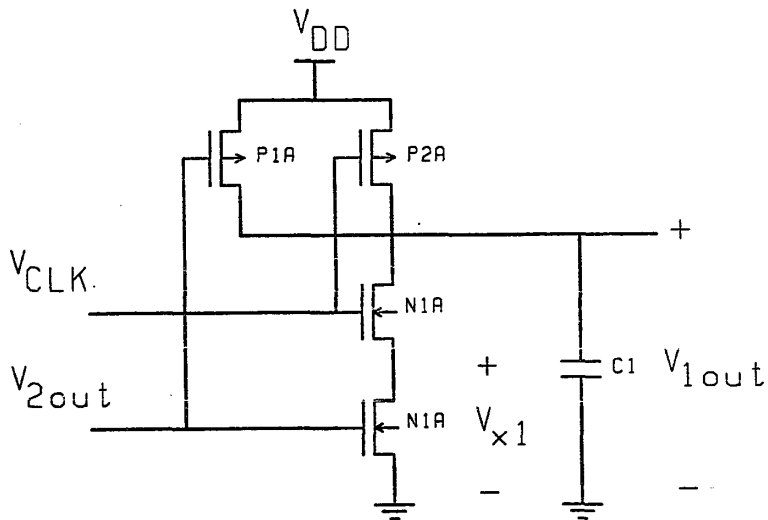


Figure 5.5: Region 2 Circuit Configuration

The circuit shown in Figure 5.5 can be represented by the small signal model shown in Figure 5.6 where v_c represents the incremental change in V_{CLK} and v_{1out} represents the incremental change in V_{1out} .

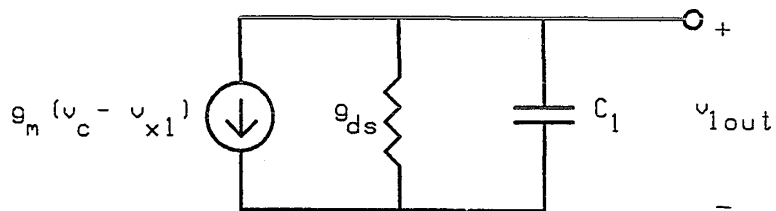


Figure 5.6: Small Signal Model of Region 2

From this model, $V_{1out}(t)$ can be determined based on a ramp input clock signal decreasing at a rate of k_c volts/second. $V_{1out}(t)$ for region 2 is shown in equation (5.2) below:

$$V_{1out}(t) = k_c[AC_1/B^2][\exp(-Bt/C_1) + Bt/C_1 - 1] \quad (5.2)$$

Note that $V_{1out}(t)$ increases from 0 and terminates at V_{TN} volts within this region, as shown in Figure 5.7. Region 2 ends at V_{TN} since, assuming $V_{DATA} > V_{TN}$, the circuit will

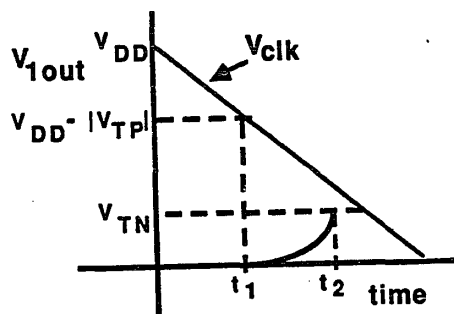


Figure 5.7: Region 2 Timing Diagram

enter the regenerative region of operation. A and B represent the transconductance and output conductance of the single NAND gate, respectively, in region 2 and are given below:

$$A = g_m' + g_{mp} - \frac{g_m' g_{mn}}{g_{n1} + g_{n2} + g_{mn}} \quad (5.3)$$

$$B = g_{ds} - \frac{g_m' g_{n1}}{g_{n1} + g_{n2} + g_{mn}} \quad (5.4)$$

where

$$g_m' = \frac{g_{mn} g_{n2}}{g_{n1} + g_{n2}} \quad (5.5)$$

$$g_{ds} = \frac{g_{n1} g_{n2}}{g_{n1} + g_{n2}} \quad (5.6)$$

g_{mn} and g_{mp} represent the transconductance of the P-channel tree and the N-channel tree, respectively, of the input clock signal and g_{n1} and g_{n2} are the output conductances of the two serial N-channel transistors.

The operating point for region 2 from which the small signal parameter values can be derived is approximately halfway between 0 and V_{TN} as shown in equation (5.7). V_{x1}

$$V_{1out}(\text{Region 2 operating point}) = V_{TN}/2 \quad (5.7)$$

(see Figure 5.5), the potential at the common source/drain between N_{1A} and N_{2A} can be derived from equation (5.8) where $V_{1out} = V_{TN}/2$.

$$V_{x1} = \frac{(V_{Clk} - V_{TN}) + (V_{2out} - V_{TN})}{2} \quad (5.8)$$

$$- \sqrt{\frac{[(V_{CLK} - V_{TN}) + (V_{2out} - V_{TN})]^2 - (V_{CLK} - V_{TN})V_{1out} + \frac{V_{1out}^2}{2}}{4}}$$

Note that $V_{2out} = V_{DD}$ and $V_{CLK} = V_{DD} + 2V_{TP}$ around the operating point of $V_{1out} = V_{TN}/2$. From the above information, the process dependent parameters \bar{K}_p' and \bar{K}_n' , and the geometric W/L ratios of each of the P- and N-channel transistors, S_p and S_n , the output response $V_{1out}(t)$ for a decreasing ramp clock input signal can be determined for region 2.

Region 3

Once V_{1out} reaches V_{TN} volts, N_{2B} is turned on (see Figure 5.2). If V_{DATA} is also greater than V_{TN} of the top N-channel device plus V_{DS} of the lower N-channel device, N_{1B} will also turn on; with both on, current can flow between V_{DD} and ground. At some point in time, depending upon the magnitude of V_{DATA} and V_{1out} , the N-channel tree will sink more current than the P-channel tree will source. At this point, $V_{2out}(t)$ will decrease from V_{DD} volts. Once V_{2out} decreases below $V_{DD} + V_{TP}$, P_{1A} will turn on and source additional current, furthering the rate of increase of V_{1out} . This in turn will improve the ability of the N-channel tree of the lower NAND gate to

sink more current, further decreasing V_{2out} . Herein lies the closed loop regenerative mode of operation inherent to the bistable NAND gate circuit configuration and fundamental to the latching behavior of a register. Note that the circuit is a two time constant system.

At a certain operating point, the data will be fully latched into the register and the clock input signal can be returned to V_{DD} and the state of the register will still enter its correct state ($V_{1out} = V_{DD}$ and $V_{2out} = 0$). This irreversible latching point represents the limiting ability to latch data into a register and will be further defined in section four of this chapter.

The circuit configuration of region 3 is shown in Figure 5.8. This regenerative circuit can be represented by the small signal models depicted in Figures 5.9 and 5.10, where Figure 5.9 represents the small signal model for the upper NAND gate, A, and Figure 5.10 represents the small signal model for the lower NAND gate, B.

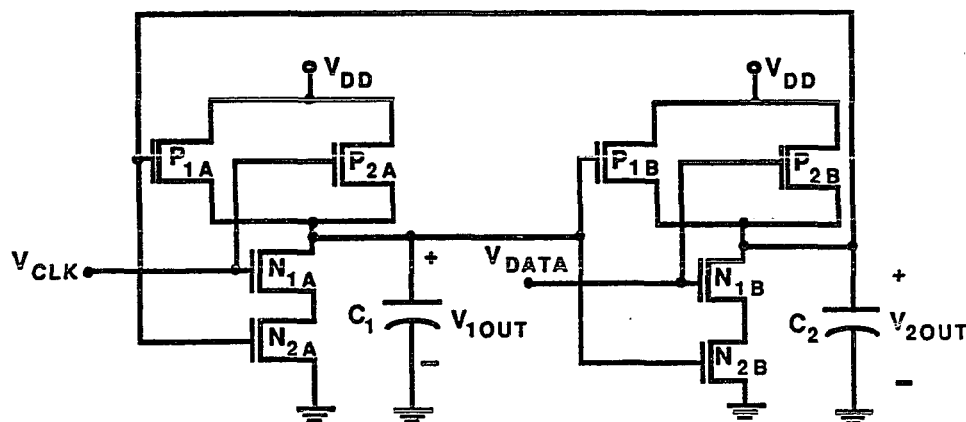


Figure 5.8: Circuit Diagram of Region 3

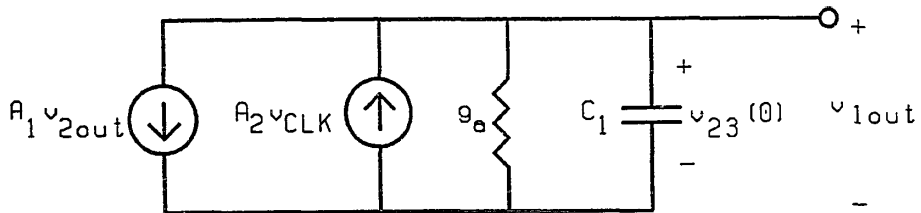


Figure 5.9: Small Signal Model of Device A in Region 3

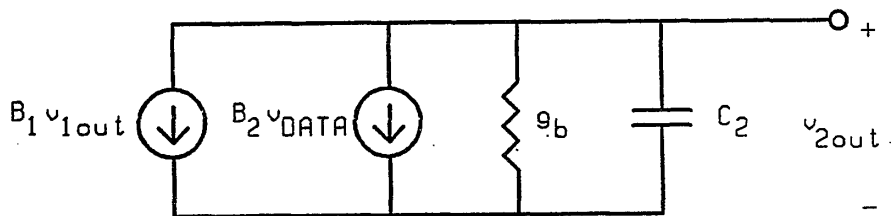


Figure 5.10: Small Signal Model of Device B in Region 3

In the regenerative mode of region 3 with a constantly decreasing clock input signal, $v_{1out}(t)$ is composed of three terms: one due to the initial condition of region 3, $v_{23}(0)$, one due to the input clock signal, and the third due to the input data signal.

$$V_{iout}(t) = V_{ioutA} + V_{ioutC} + V_{ioutD} \quad (5.9)$$

Each term is described individually below:

$$V_{ioutA}(t) = V_{23(0)}[R_{1A} + R_{2A} + R_{3A}] \quad (5.10)$$

$$R_{1A} = \frac{(\alpha_1 - D)\exp(-\alpha_1 t)}{\alpha_1(\alpha_2 - \alpha_1)} \quad (5.11)$$

$$R_{2A} = \frac{(\alpha_2 - D)\exp(-\alpha_2 t)}{\alpha_2(\alpha_1 - \alpha_2)} \quad (5.12)$$

$$R_{3A} = \frac{D}{\alpha_1 \alpha_2} \quad (5.13)$$

$$V_{ioutC}(t) = \frac{-A_2 k_1 [(R_{1C} + R_{2C} + R_{3C})U(t) - (R_{1C} + R_{2C} + R_{3C}) \Big|_{t = E/k_1}]}{C_1} \quad (5.14)$$

$$R_{1C} = \frac{(D - \alpha_1)\exp(-\alpha_1 t)}{\alpha_1^2(\alpha_2 - \alpha_1)} \quad (5.15)$$

$$R_{2C} = \frac{(D - \alpha_2)\exp(-\alpha_2 t)}{\alpha_2^2(\alpha_1 - \alpha_2)} \quad (5.16)$$

$$R_{3C} = \frac{\alpha_1 \alpha_2 (1 + Dt) - D(\alpha_1 + \alpha_2)}{\alpha_1^2 \alpha_2^2} \quad (5.17)$$

$$V_{\text{IoutD}}(t) = \frac{-A_1 B_1 k_2 [(R_{1D} + R_{2D} + R_{3D})U(t) - (R_{1D} + R_{2D} + R_{3D})U(t - E/k_2)]}{C_1 C_2} \Big|_{t = E/k_2} \quad (5.18)$$

$$R_{1D} = \frac{\exp(-\alpha_1 t)}{\alpha_1^2 (\alpha_2 - \alpha_1)} \quad (5.19)$$

$$R_{2D} = \frac{\exp(-\alpha_2 t)}{\alpha_2^2 (\alpha_1 - \alpha_2)} \quad (5.20)$$

$$R_{3D} = \frac{\alpha_1 \alpha_2 t - (\alpha_1 + \alpha_2)}{\alpha_1^2 \alpha_2^2} \quad (5.21)$$

$$D = g_b / C_2 \quad (5.22)$$

$$\alpha_1 = \frac{-(g_a / C_1 + g_b / C_2) + Q}{2} \quad (5.23)$$

$$\alpha_2 = \frac{-(g_a / C_1 + g_b / C_2) - Q}{2} \quad (5.24)$$

$$Q = \sqrt{(g_a / C_1)^2 + (g_b / C_2)^2 - 2 \frac{g_a g_b}{C_1 C_2} + 4 \frac{A_1 B_1}{C_1 C_2}} \quad (5.25)$$

Thus, equations (5.9) - (5.25) represent the output voltage across C_1 , $V_{1out}(t)$, when operating in a closed loop regenerative mode during region 3. Figure 5.11 describes the $V_{1out}(t)$ waveform in region 3.

Each of the transconductance and output conductance terms shown in Figures 5.9 and 5.10 require definition. A_1 and B_1 represent the transconductance of the two feedback output voltages, V_{2out} and V_{1out} , respectively. A_2 and B_2 are the transconductances of the input clock and data signals, respectively. A_1 , B_1 , A_2 , and B_2 in terms of their small signal parameters are shown below in equations (5.26) - (5.29):

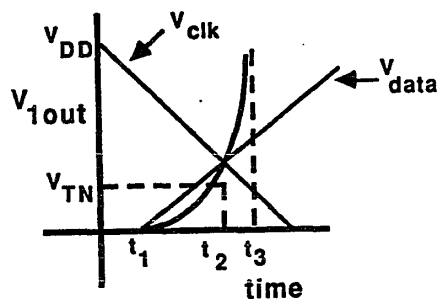


Figure 5.11: Region 3 Timing Diagram

$$A_1 = \frac{g_{mn1a}g_{mn2a}}{g_{n2a} + g_{mn1a}} - g_{mp1a} \quad (5.26)$$

$$B_1 = \frac{g_{mn1b}g_{mn2b}}{g_{n2b} + g_{mn1b}} - g_{mp1b} \quad (5.27)$$

$$A_2 = \frac{g_{mn1a}g_{n2a}}{g_{mn1a} + g_{n2a}} - g_{mp2a} \quad (5.28)$$

$$B_2 = \frac{g_{mn1b}g_{n2b}}{g_{mn1b} + g_{n2b}} - g_{mp2b} \quad (5.29)$$

The output conductances of device A and device B are given in equations (5.30) and (5.31).

$$g_a = 0 \quad (5.30)$$

$$g_b = g_{p1b} + g_{p2b} \quad (5.31)$$

Since in region 3 all of the ON transistors in the upper device are saturated, g_a is equal to zero.

The operating point for region 3 from which the small signal parameters can be derived is approximately halfway between the end points of region 3 as shown in equation (5.32).

$$V_{1out}(\text{Region 3 operating point}) = V_{DD}/2 \quad (5.32)$$

V_{x1} and V_{x2} , the common source/drain nodes between N_{1A} and N_{2A} and N_{1B} and N_{2B} , respectively, in region 3 can be derived from equations (5.33) and (5.34) given below.

$$V_{x1} = \frac{(V_{CLK} - V_{TN}) + (V_{2out} - V_{TN})}{2} \quad (5.33)$$

$$- \sqrt{\frac{[(V_{CLK} - V_{TN}) + (V_{2out} - V_{TN})]^2 - (V_{CLK} - V_{TN})V_{1out} + \frac{V_{1out}^2}{2}}{4}}$$

$$V_{x2} = \frac{(V_{DATA} - V_{TN}) + (V_{1out} - V_{TN})}{2} \quad (5.34)$$

$$- \sqrt{\frac{[(V_{DATA} - V_{TN}) + (V_{1out} - V_{TN})]^2 - (V_{DATA} - V_{TN})^2}{4}}$$

From these values, each of the small signal parameters in equations (5.26) - (5.31) can be determined.

Region 4

As V_{1out} increases toward V_{DD} and as V_{2out} decreases toward ground, V_{DS} across P_{1A} , P_{2A} , N_{1B} , and N_{2B} becomes very small and therefore A_1 and A_2 both approach zero. This breaks the regenerative loop of region 3 and the bistable register becomes once again an open loop single time constant system in which the P-channel devices charge

the capacitor C_1 up to V_{DD} (see Figure 5.12). $V_{1out}(t)$ for region 4 is shown below where $V_{34}(0)$ is the initial condition of region 4.

$$V_{1out}(t) = V_{DD} - [V_{DD} - V_{34}(0)] \exp(-g_{a4}t/C_1) \quad (5.35)$$

where

$$g_{a4} = g_{14a} + g_{24a} \quad (5.36)$$

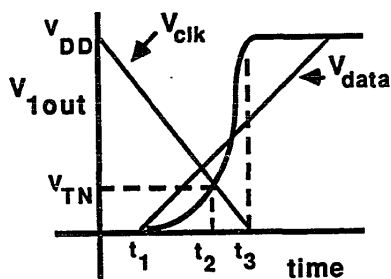


Figure 5.12: Region 4 Timing Diagram

The operating point for region 4 at which the small signal parameters, g_{14a} and g_{24a} , should be derived is given in equation (5.37).

$$V_{1out}(\text{Region 4 operating point}) = V_{DD} + V_{TP}/2 \quad (5.37)$$

V_{x1} can be determined from equation (5.38).

$$V_{x1} = \frac{(V_{CLK} - V_{TN}) + (V_{2out} - V_{TN})}{2} \quad (5.38)$$

$$- \sqrt{\frac{[(V_{CLK} - V_{TN}) + (V_{2out} - V_{TN})]^2}{4} - \frac{(V_{CLK} - V_{TN})^2}{2}}$$

Register Output Waveform

The output voltage waveform of the bistable register for an input clock signal decreasing at 1 volt/ns. and a data signal increasing at 1 volt/ns. skewed from the clock signal by 1 ns. is shown in Figure 5.13. This analytically derived output waveform has been compared to a waveform generated from the SPICE circuit simulator program [53] using Level 1 Shichman-Hodges device equations [54] with the same circuit, geometric, and process characteristics. Close agreement within each region is apparent. A BASIC program which generates the output waveform for any clock signal fall time, data signal rise time, data-to-clock timing skew, as well as K_p' , K_n' , and geometric W/L ratio is provided in Appendix A.

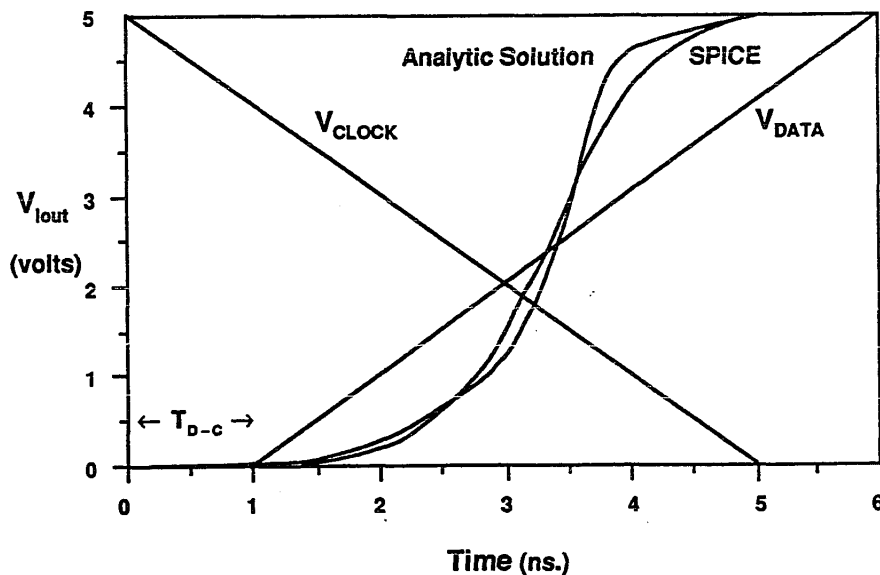


Figure 5.13: Transient Response of Bistable Register

4) Conditions for Latching

As mentioned previously, a series of necessary and sufficient conditions are required to irreversibly latch data into a bistable register. From these conditions, fundamental limiting relationships have been derived which define whether, for a given combination of clock and data input signals, K_p' , K_n' , S_n , and S_p parameters, the bistable register will latch. Once a register has irreversibly latched, the clock signal can be returned to

V_{DD} and still the register will maintain its correct state. This minimum clock signal represents the smallest required clock period or alternatively, the greatest clock frequency possible for a given set of input, geometric, and process conditions.

Necessary and Sufficient Conditions for Latching

The necessary and sufficient conditions for latching are given below:

$$1: V_{CLK} < V_{DD} + V_{TP} \quad (5.39)$$

$$2: V_{DATA} > V_{TN} + V_{x2} \quad (5.40)$$

$$3: A_1 V_{2out} + A_2 V_{CLK} > 0 \quad (5.41)$$

$$4: B_1 V_{1out} + B_2 V_{DATA} > 0 \quad (5.42)$$

The terms A_1 , A_2 , B_1 , and B_2 represent the transconductance parameters described in region 3 and given as equations (5.26) - (5.29). Equation (5.41) states that the P-channel tree in device A sources more current than the N-channel tree in device A sinks, thereby increasing V_{1out} . Equation (5.42) states that the N-channel tree in device B sinks more current than the P-channel tree in device B sources, thereby decreasing V_{2out} . If these four conditions are satisfied for all operating points within region 3, the device will latch.

Limiting Requirement for Latching

Equation (5.41) provides the fundamentally limiting condition for latching. As $V_{CLK(min)}$ is reached (see Figure 5.14) and the clock signal is returned to V_{DD} ,

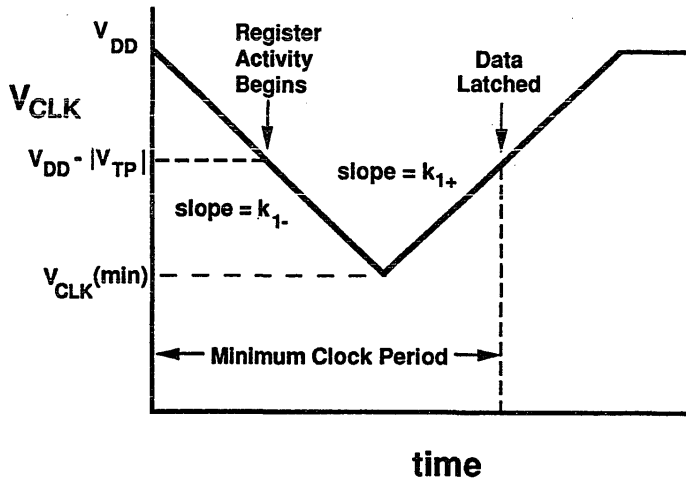


Figure 5.14: Timing Diagram of Limiting Condition for Latching

V_{2out} decreases further from V_{DD} . If at the operating point, $V_{CLK} = V_{DD} + V_{TP}$, (P_{2A} becomes cutoff), the current supplied by P_{1A} (and driven by V_{2out}) is larger than the current sunk by the N-channel tree of device A, thereby maintaining a monotonically increasing voltage at V_{1out} , the device will latch. This condition is shown below:

$$A_1 V_{2out} > A_2 V_{CLK} \text{ and } A_1 V_{2out} \text{ is a positive quantity} \quad (5.43)$$

In terms of its small signal parameters, equation (5.43) can be presented in the form of equation (5.44).

$$g_{m1a} \geq g_{mn} = \frac{g_{mn1a} g_{mn2a}}{g_{n2a} + g_{mn1a}} \Bigg|_{V_{CLK+} = V_{DD} + V_{TP}} \quad (5.44)$$

where V_{CLK+} represents the operating point at which $V_{CLK} = V_{DD} + V_{TP}$ after it had reached its minimum value and has risen to $V_{DD} + V_{TP}$. Equation (5.44) represents the ultimate limiting condition for latching data into a bistable register. Table 5.1 describes an example circuit which operates just at the latch breakpoint. One parameter, V_{DATA} , was varied to exemplify the limiting nature of equation (5.44). The other circuit characteristics were kept constant and are listed below:

$$V_{CLK(\min)} = 1.7 \text{ volts}$$

$$K_p', K_n' = 4.316 \times 10^{-5} \text{ Amperes/volt}^2$$

$$k_{1-}, k_{1+} = 1 \text{ volt/nanosecond}$$

$$S_n, S_p = 5 \text{ (W/L ratio)}$$

V_{DATA}	ξ_{mp}	ξ_{mn}	Latch?
2.55 volts	$6.578 \times 10^{-4} \text{ V}$	$6.603 \times 10^{-4} \text{ V}$	no
2.60 volts	$7.907 \times 10^{-4} \text{ V}$	$7.907 \times 10^{-4} \text{ V}$	breakpoint
2.65 volts	$1.548 \times 10^{-3} \text{ V}$	0	yes

Table 5.1: Example of Limiting Latch Condition

Maximum Clock Frequency

If equation (5.44) is satisfied, then the maximum clock frequency at which this digital synchronous system can operate is given in equation (5.45).

$$f_{\max} = 1/\text{minimum clock period} \quad (5.45)$$

where the minimum clock period begins when V_{CLK} initially decreases from V_{DD} and ends when it reaches $V_{DD} + V_{TP}$, after having passed its minimum value, as shown in Figure 5.14.

As can be seen from the results described within this chapter, the magnitude and transition times of the input clock and data signals, as well as the skew between these two signals, have a direct influence on whether or not the bistable register will latch. The strategy for determining the effects of these characteristics on latching are discussed in Chapter 6.

5) Summary

Closed form analytic solutions of each of the four regions of operation of a bistable register have been developed. Close agreement with a SPICE generated output waveform for an equivalent circuit was made and is shown in Figure 5.13. Necessary and sufficient conditions for latching data into a bistable register were developed and are presented in equations (5.39) - (5.42). From these conditions, the limiting condition for latching has been defined. This limiting condition is presented in equation (5.44) and is repeated below:

$$g_{m1a} \geq g_{mn} = \frac{g_{m1a}g_{m2a}}{g_{n2a} + g_{m1a}} \Bigg|_{V_{CLK+} = V_{DD} + V_{TP}} \quad (5.44)$$

This result was corroborated by testing equation (5.44) at $V_{CLK+} = V_{DD} + V_{TP}$ with breakpoint conditions and the result, shown in Table 5.1, fully agreed with expectations.

A minimum clock period is defined by equation (5.45). From these results, fundamentally limiting conditions for latching data into a bistable register are quantitatively defined.

CHAPTER 6
HIGH SPEED SYNCHRONOUS DATA PATHS

1) Overview of Integrated Systems

This dissertation is intended to describe research results which consider the problem of moving data through a synchronous digital system as quickly as possible. These results provide quantitative relationships between the key circuit elements and parameters of a synchronous digital system so as to permit the design of a system operating at its maximum possible performance. In order to develop these results, the complete synchronous digital system has been decomposed into each of its separate elements.

A synchronous digital system is typically composed of data paths in which data is moved from a register through some logical functions and into a second register. As described in Chapter 3, the synchronization of the data flow between the initial and final registers is coordinated by a single control signal, typically called the clock signal. Thus, a synchronous digital system is composed of three interrelated systems:

1) the clock distribution system which generates the synchronizing clock pulse and defines when data can flow from one register to the next

2) the registers which store the data signals awaiting the synchronizing clock pulse

3) the data paths which contain the logical circuitry of the digital system

The total delay of a data path is determined by the time required to leave the initial register once a clock signal arrives, T_{c-Q} , the time necessary to propagate through the logic and interconnect, $T_{logic} + T_{int}$, and the time required to successfully propagate to and latch within the final register of the data path, T_{set-up} . This relation was described in Chapter 3 and is repeated below:

$$T_{PD} = T_{c-Q} + T_{logic} + T_{int} + T_{set-up} \quad (3.2)$$

Thus, equation (3.2) describes the individual delay components making up the total delay, T_{PD} , of any data path. The data paths whose total delay plus any positive clock skew are greatest represent the critical worst case timing requirements of a digital system and the delay and clock skew of these paths must be minimized in order to maximize the performance of the entire digital system. Thus, in a high performance synchronous digital system, the critical paths constrain and define the maximum performance of the entire system. Therefore, the goal is to minimize each delay component in equation (3.2) as well as utilize any possible advantages (and minimize any possible disadvantages) of the clock distribution

circuitry which will increase the speed of operation of the critical data paths.

This chapter describes how the signal delay through the logic and interconnect is determined and how the signal waveform at the output of the final logic stage should be designed so as to optimally satisfy the latching conditions of the final register; thereby moving the data through the logic and into the register as quickly as possible. The data path delay is also described in terms of the clock delay at the final register; thereby, describing quantitative relationships for latching the correct data into the final register as quickly as possible.

2) A Data Path of a General Synchronous System

Digital systems are designed so as to satisfy some functional requirement. The functional circuitry in a digital system is composed of logic gates configured in a specific fashion. Every different system utilizes a different arrangement of logic gates. Therefore, in order to investigate the general problem of moving data through a digital system, one cannot constrain the circuitry to any specific logic function and the problem must be analyzed from a general perspective. Thus, in this investigation, no specific types of logical circuits are

assumed and therefore, these results are applicable to any logical configuration.

A general form of a data path is shown in Figure 6.1, where an initial register R_i begins the data path and is followed by N stages of logic and $N+1$ stages of interconnect, ending in a final register R_f .

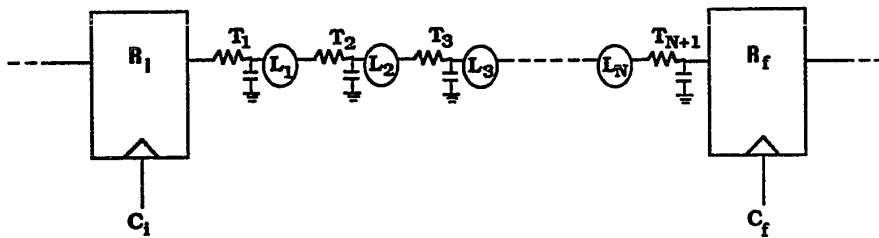


Figure 6.1: Synchronous Data Path with N Stages of Logic

This analysis assumes the output of R_i to be a ramp of magnitude k_1 volts/second. The output of each i th logic stage, L_i , is assumed to be a ramp of magnitude k_{i+1} volts/second where the ramp is determined by the device and circuit characteristics of the logic circuit. Each single pole interconnect time constant is designated by T_i where i represents each logic and interconnect stage and N is the total number of logic stages. Thus, in a data path composed of N logic stages there are $N+1$ interconnect time constants.

Single Stage Logic Circuit

In order to determine the delay through a critical path composed of N stages, the delay through each stage is individually determined, permitting each stage to then be appropriately summed. Thus, an algorithm to determine the delay through a single stage is described. Once developed, this algorithm is applied along the entire data path.

As shown in Figure 6.2, the output of the initial register R_i is a ramp of magnitude k_1 driving an interconnect time constant T_1 and logic stage L_1 . $V_1(t)$, the node voltage at the input to L_1 , is given by equation (6.1).

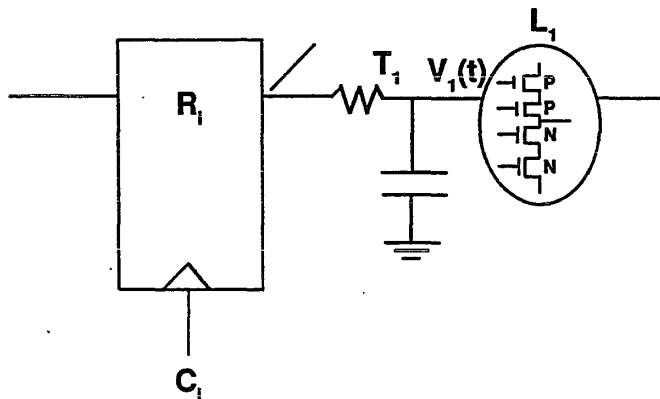


Figure 6.2: First Stage of N Stage Data Path

$$V_1(t) = k_1 T_1 [\exp(-t/T_1) + t/T_1 - 1] u(t) \quad (6.1)$$

$$-k_1 T_1 [\exp(-(t+V_{DD}/k_1)/T_1) + (t+V_{DD}/k_1)/T_1 - 1] u(t-V_{DD}/k_1)$$

Let us define a new term, $V_{1\log i}$, to be the threshold voltage at which an increasing ramp signal will bias the N-channel tree of the i th logic stage such that current can flow at the output of the logic stage. $V_{1\log i}$ is a function of the threshold voltage of the N-channel transistor it is driving and the number R of serially connected N-channel transistors between the driven transistor and ground. This relation is shown in equation (6.2). Note that R is one for an inverter and $V_{1\log}$ is equal to V_{Tn} for this simplified circuit configuration.

$$V_{1\log i} = V_{Tn} + \sum_{j=1}^R V_{DSj} \quad (6.2)$$

The delay through a single logic stage is described as the time required for its input waveform to reach the turn-on threshold voltage of the logic stage, as defined by equation (6.2). This delay per stage, T_{fi} , is summed for each stage to represent the total delay through N stages. $V_1(t)$, defined in equation (6.1), can be configured into the form of equation (6.3), if one assumes a region of operation as defined by equation (6.2). This regional form of equation (6.1) is given below:

$$V_1(t) = k_1 T_1 [\exp(-t/T_1) + t/T_1 - 1] u(t) \quad (6.3)$$

The delay of the first stage, T_{f1} , is calculated from equation (6.3) when $V_1(t)$ reaches the logical threshold of L_1 . This is shown in equation (6.3) where $t = T_{f1}$, the

$$V_{\log 1}(t=T_{f1}) = k_1 T_1 [\exp(-T_{f1}/T_1) + T_{f1}/T_1 - 1] \quad (6.4)$$

delay through the first stage. This transcendental equation in T_{f1} , equation (6.4), can be solved using interpolation techniques or it can be approximated. The exponential can be approximated as in equation (6.5) to give a closed form general solution of T_{fi} , the delay through the i th stage, described by equation (6.6).

$$\exp(-t/T_i) \cong 1 - t/T_i + t^2/T_i^2 \quad (6.5)$$

$$T_{fi} \cong \sqrt{T_i V_{\log i}/k_i} \quad (6.6)$$

Delay of an N Stage Cascaded Data Path

Thus, the total delay from the output of the initial register R_i to the output of the N th logic stage is the sum of the individual T_{fi} terms along that data path as shown in equation (6.7) below:

$$T_{\text{logic}} + T_{\text{int}} = \sum_{i=1}^N T_{fi} \quad (6.7)$$

Equation (3.2) can be redefined as shown below for an N stage data path where the $N+1$ st interconnect will be added to $T_{\text{set-up}}$ as will be discussed later. Equation (6.8) states that the total delay of a data path is

$$T_{PD} = T_{C-Q} + \sum_{i=1}^N T_{fi} + T_{set-up} \quad (6.8)$$

composed of the time required to leave the initial register, T_{C-Q} , the time to enter and latch into the final register, T_{set-up} , and the time required to propagate through N stages of logic and interconnect.

The Output of the Nth Logic Stage to the Register

As shown in Figure 6.1, all that remains to move the data signal from the initial register to the final register of the data path is to successfully move the signal from the output of the N th logic stage into the final register. In this case, the logic threshold is replaced by the latching characteristics of the register and T_{fN+1} is the time required to satisfy V_{logRf} , the threshold voltage of the final register, given by the transcendental equation (6.9) or the approximated solution for T_{fN+1} in equation (6.10) below.

$$\exp(-T_{fN+1}/T_{N+1}) + T_{fN+1}/T_{N+1} > 1 + V_{logRf}/(k_{N+1}T_{N+1}) \quad (6.9)$$

$$T_{fN+1} = \sqrt{T_{N+1} V_{logRf}/k_{N+1}} \quad (6.10)$$

Further details which consider the latching conditions of the register are discussed in the next section describing the register.

A by-product of equation (6.9) is an interesting design equation which quantifies how the final logic stage should be designed so as to satisfy a specific T_{fN+1} assuming $V_{\log Rf}$ and T_{N+1} is known. This design equation is given below:

$$k_{N+1} = \frac{V_{\log Rf}}{T_{N+1}[\exp(-T_{fN+1}/T_{N+1}) + T_{fN+1}/T_{N+1} - 1]} \quad (6.11)$$

Thus, equation (6.11) defines the magnitude of the ramp at the output of the Nth logic stage which will satisfy the logic threshold of the final register of the data path within a specified time T_{fN+1} .

Maximum and Minimum Bounds on the Data Path Delay

In order to represent the maximum delay of a data path, one must assume a) a minimum k_i , implying a poor logic response, b) T_i is large, and c) $V_{\log i}$ is a maximum. A minimum delay of a data path assumes that a) k_i is at its maximum, preferably approaching a step response, b) T_i is small, and c) $V_{\log i}$ is a minimum, approaching V_{Tn} .

Determination of Output Ramp Response

In order to determine k_i , one of three approaches can be used:

1) k_i of each stage can be assumed equal over the entire data path.

2) for data paths containing a statistically large value of N , k_i can be estimated to be the mean waveform response.

3) k_i can be calculated independently for each individual logic stage and handled individually.

The output ramp, k_i , can be determined from equation (6.12) where I_{Di} is the current at the output of the previous logic stage, L_i , and C_{i+1} is the total load capacitance being driven by L_i .

$$k_i = I_{Di}/C_{i+1} \quad (6.12)$$

3) Register

As described previously, the purpose of the register in the data path is to temporarily store the data signal as it awaits a synchronizing clock pulse. Once this clock pulse appears, then ideally all of the parallel data signals appearing at the output of the parallel data registers will be synchronized in time.

Input Latching Conditions of Register

In order to store data in a register, the data signal must successfully latch into the register. Necessary and sufficient latching conditions to store a rising data signal into a register were discussed in Chapter 5 and are repeated below using the notation developed in this chapter.

$$A) V_{CLK} < V_{DD} + V_{Tp} \quad (6.13)$$

$$B) V_{DATA} > V_{logRf} \quad (6.14)$$

$$C) I_{PA} > I_{NA} \quad (6.15)$$

$$D) I_{NB} > I_{PB} \quad (6.16)$$

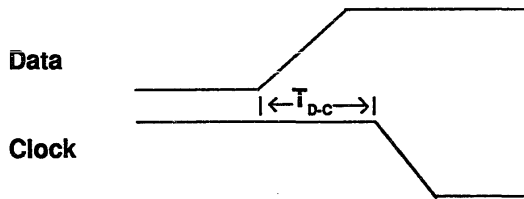
Thus, the set-up time referred to in equation (3.2) is the time required for the output waveform of the Nth logic stage to propagate through the T_{N+1} interconnect, reach V_{logRf} of the register, and satisfy the latching conditions C and D listed above and described in Chapter 5.

$$T_{set-up} = T_{fN+1} + T_{latch} \quad (6.17)$$

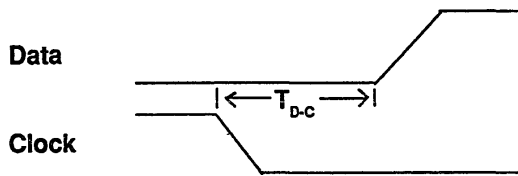
Data-to-Clock Timing Skew

Since no constraints are placed on the number of logic stages of a data path or on the size of the clock distribution tree, the total delay of the data signal through the data path, T_{pD} , with respect to the delay of the clock signal at the input of R_f , can vary significantly on a path by path basis. Therefore, a new term must be defined. T_{D-C} is the difference in time between the data and the clock signal (i.e., the data-to-clock timing skew) referenced at the 0% point for the data signal and at the 100% point for the clock signal (see

Figure 6.3). T_{D-C} can be either positive or negative depending upon whether the data signal leads or lags the clock signal, respectively.



A: Positive Data-to-Clock Skew



B: Negative Data-to-Clock Skew

Figure 6.3: Positive and Negative Data-to-Clock Timing Skew

Positive Data-to-Clock Skew

For the case where the data signal arrives at the register before the clock does, i.e., positive T_{D-C} , then the data signal must wait a time, $T_{\text{set-upD}}$, to fully set-up (i.e., propagate to and latch into) the register. In addition to the waveforms reaching their required respective thresholds (see Figure 6.4), an additional time, T_{QVTn} , must be added. T_{QVTn} is the time required for the output of the top NAND gate, V_{Iout} (or Q), to reach V_{Tn} volts once V_{CLK} reaches $V_{DD} + V_{Tp}$, thereby permitting current to flow. T_{QVTn} can be determined from equation (6.18) below:

$$T_{QVTn} = C_1 V_{Tn} / I_{DA} \quad (6.18)$$

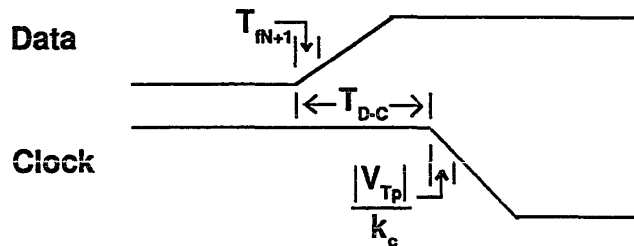


Figure 6.4: Positive Data-to-Clock Timing Skew

Thus, the set-up time for the positive data-to-clock skew case is given below:

$$T_{\text{set-upD}} \geq |T_{\text{D-C}} + |V_{\text{Tp}}|/k_c - T_{\text{fN+1}} + T_{\text{QVTn}}| \quad (6.19)$$

where k_c is the fall time of the clock signal in volts per second.

Negative Data-to-Clock Skew

For the case where the data signal arrives after the clock signal at the final register input, i.e., negative $T_{\text{D-C}}$, then the clock signal must wait a time, $T_{\text{set-upC}}$, in order to correctly latch the data into the register, thereby insuring correct operation of the system. $T_{\text{set-upC}}$ can be interpreted as the hold time of the clock signal. This waveform configuration is shown in Figure 6.5.

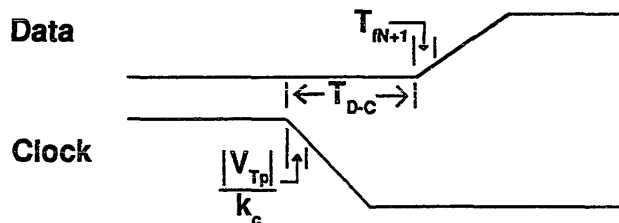


Figure 6.5: Negative Data-to-Clock Timing Skew

It is assumed in the negative data-to-clock skew case that

$$|T_{D-C}| + T_{fN+1} > |V_{Tp}|/k_c + T_{QVTn} \quad (6.20)$$

which states that the clock is indeed leading the data, thereby ensuring that $T_{\text{set-up}C}$ is independent of T_{QVTn} . Therefore, in this case the set-up time is

$$T_{\text{set-up}C} \geq \left| |T_{D-C}| + T_{fN+1} - |V_{Tp}|/k_c \right| \quad (6.21)$$

Thus, the time required for the output of the final logic stage to propagate through the T_{N+1} interconnect and latch into the register, considering a system dependent data-to-clock skew, is given by either equation (6.19) or (6.21), depending upon whether the data leads or lags the clock, respectively.

4) Integrated Synchronous System

As was discussed in Chapter 3, the magnitude and lead/lag behavior of the clock skew between C_i and C_f of the initial and final registers of a data path directly affects the minimum possible clock period or maximum possible clock frequency for that data path.

$$T_{PD} + T_{SKEW} \leq T_{\text{Clock Period}} = 1/f_{\text{CLKmax}} \quad (3.3)$$

Positive and Negative Clock Skew

Thus, from equation (3.3), if C_f leads C_i (i.e., positive clock skew), the maximum clock rate is decreased while if C_f lags C_i (i.e., negative clock skew), the maximum frequency is increased. The maximum permissible negative clock skew of any data path is dependent upon the previous data path, since the earlier C_i is for a given data path, the later that same clock signal, now C_f , is for the previous data path. Therein lies the difficulty of using negative clock skew to increase maximum performance. It can, however, be used and quite successfully, but its effect must be carefully considered for each individual data path.

Relationship Between Clock Skew and Data-to-Clock Skew

As described in Chapter 4, the time for the clock signal to appear at the input of R_i is T_{CDii} and the time for the clock signal to appear at the input of R_f is T_{CDff} . The time when the data appears at the input of the final register R_f from the original clock signal is $T_{CDii} + T_{PD}$. Thus, an equation describing the interrelationship between the data-to-clock skew, the data path delay, and the clock skew is given below:

$$\begin{aligned}
 T_{D-C} &= \text{Data Delay} - \text{Clock Delay} \\
 &= T_{CDii} + T_{PD} - T_{CDff} \\
 &= T_{SKEW} + T_{PD} \qquad (6.22)
 \end{aligned}$$

Equation (6.22) presents quantitatively how the data path and the clock distribution network interact. Thus, if T_{SKEW} is negative and greater in magnitude than T_{PD} , thereby making T_{D-C} negative without compensating by increasing $T_{set-upC}$ as defined in equation (6.21), the incorrect data will latch into the final register. This incorrect operation was described previously in Chapter 3 and quantitatively defined by the constraint equation (3.5).

5) Summary

Thus, with the results developed in Chapters 3, 4, 5, and within this chapter, specific quantitative relationships among the critical data path, the register, and the clock distribution network have been investigated and described. These results provide guidance in the design and analysis of high performance synchronous digital systems by developing the fundamental relationships between these interdependent subsystems.

CHAPTER 7

DATA THROUGHPUT AND CLOCK FREQUENCY IN PIPELINED DATA PATHS

In a synchronous digital system, the data throughput is defined as the total time required to move a particular data signal from the input of a system to its output (i.e., in order to process the signal). The minimum data throughput occurs when the data path is composed of only logic stages and is the time required to propagate a data signal through these logic stages. The sample period for this data path is equal to the time required to process one data sample, the data throughput. If the rate at which the data signals are sampled at the system input is more significant than the time required to process a particular data sample, registers can be inserted into the data path. This increases the frequency at which the data signals are sampled at the system input but degrades the data throughput. Thus, systems can be designed which minimize data throughput, maximize the clock frequency (i.e., data sample rate), or optimize both the data throughput and the clock frequency. This chapter investigates these topics and describes quantitative relationships for designing and analyzing each kind of system.

The data throughput is proportional to the latency, the total number of registers along a data path, and the sampling period (the clock period) at which data is moved from one register to the next. This chapter quantifies the effect of added latency on data throughput and relates this effect to increased clock frequency. Each of the relevant components and terms used in this chapter are defined in sections 1 and 2 while in section 3, a design paradigm relating data throughput and clock frequency as a function of the level of pipelining is described for studying the performance behavior of synchronous systems.

Most synchronous digital systems are designed to satisfy specific performance requirements such as minimum clock frequency or maximum data throughput. Thus, in these systems the design problem becomes either one of maximizing the clock frequency while not exceeding a maximum data throughput or minimizing the data throughput while meeting a specific clock frequency. These systems represent constrained design problems and are analyzed in section 4. In certain systems, neither the data throughput or the clock frequency ultimately constrains the design problem. In these unconstrained design problems, the level of pipelining must be chosen to optimize both the data throughput and the clock frequency. These systems are investigated in sections 5 and 6.

In many synchronous systems, the primary performance attribute is the sampling rate (alternatively, the clock frequency); therefore, these digital systems are designed to maximize f_{clk} [62]. This is apparent from the careful attention placed on minimizing the delay of each of the components of the critical paths of a system, that is, those data paths which constrain and define the maximum operating clock frequency. Often, however, little attention is placed on the effect of increased pipelining on the data throughput of the system through increased latency [63-68]. An arbitrarily defined cost/benefit figure of merit is developed in section 5 from which the optimal number of logic stages between registers (and the optimal number of pipeline registers within a serially cascaded set of data paths) is derived for a high speed synchronous digital system. This algorithm for determining the optimal number of logic stages between registers is applied in sections 7 through 9.

In section 10, the theoretically maximum clock frequency, for restricted classes of synchronous circuits, is analyzed. It is shown that for these circuits the maximum clock frequency is only limited by the resolution (i.e., control of circuit parameters) of the device and design technologies and the speed at which the input data signal is changing.

1) Data Path Delay Components

A general data path of a synchronous digital system is shown in Figure 7.1. It is composed of an initial register and a final register and N logic levels between them. The time interval for the data signal to appear at the output of the initial register upon arrival of the clock signal is the clock-to-Q delay T_{c-Q} . The time required for the data signal to propagate through each distributed RC interconnect section T_i and logic stage L_i is T_{fi} . Finally, the time required for the signal at the output of the final logic stage to propagate through the $N+1$ st interconnect section and latch into the final register is the set-up time T_{set-up} .

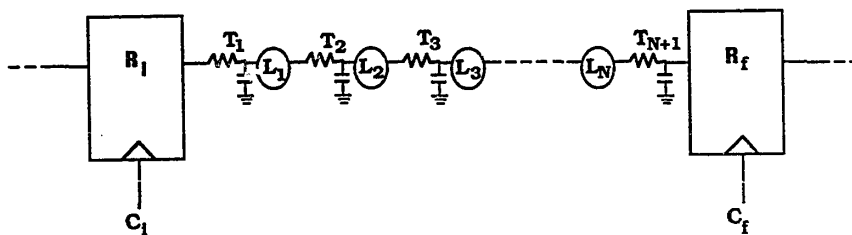


Figure 7.1: Synchronous Data Path with N Stages of Logic

Assuming N stages in the logic path, as discussed in Chapter 6 one can express the delay through a data path as

$$T_{PD} = T_{c-Q} + \sum_{i=1}^N T_{fi} + T_{set-up} \quad (6.8)$$

Equation (6.8) is composed of the delay required to get out of and into the initial and final registers, respectively, and the time required to propagate through N stages of logic and $N + 1$ sections of interconnect. If T_{REG} represents the total register related delay, then

$$T_{REG} = T_{c-Q} + T_{set-up} \quad (7.1)$$

and equation (6.8) can be written as

$$T_{PD} = T_{REG} + \sum_{i=1}^N T_{fi} \quad (7.2)$$

Thus, the total time to move a data signal through a data path is composed of the overhead requirements to get in and out of the register as well as the time required to perform the logical operations.

The maximum clock frequency at which a synchronous digital system can move data is defined by equation (3.3), where T_{SKEW} is the clock skew between C_i and C_f and its magnitude can be either positive or negative as discussed in Chapter 3. The data path with the greatest $T_{PD} + T_{SKEW}$ represents the critical path of the system.

$$T_{PD} + T_{SKEW} \leq T_{clock\ period} = 1/f_{clk} \quad (3.3)$$

2) Definition of Data Throughput Parameters

The throughput of a synchronous digital system is dependent upon the nature of its data paths. A generalized example of a single data path is shown in Figure 7.1. If a data path is composed of serially cascaded registers and logic paths, it is defined in this dissertation to be a global data path, typically representing a signal path from the input of the system to its output. Each individual data path within a global data path is described as a local data path and is composed of an initial and final register and typically, n logical stages between them. Note that each register within each local data path performs double duty, serving as the initial (final) and final (initial) register of the current and previous (next) local data path, respectively.

In order to analyze the problem of data throughput in pipelined data paths, the following terms require definition:

D_T - the time required to move a data signal from the input of the system to its output.

f_{clk} - the clock frequency at which data is moved from the initial register of one local data path to the initial register of the next serially connected local data path. Alternatively, it is the rate at which the input data signals are sampled.

L - the latency of the system is the number of registers from the input of a global data path to its output. Alternatively, it is the number of clock periods required to move a particular data signal from the input of the system to its output.

N - the number of logic stages per global data path.

n - the number of logic stages per local data path.

M - the number of local data paths per global data path.

T_{fN} - the average delay of all of the logic and interconnect stages per data path.

P_e - the pipelining efficiency which is a measure of the data throughput performance penalty incurred by inserting a single register into a global data path.

N_{opt} - the optimal number of logic stages per data path where the optimality criteria are discussed in sections 5 and 6.

R_{opt} - the optimal number of additional registers to insert into a global data path as defined by N_{opt} .

f_{clkopt} - the clock frequency of a local data path composed of N_{opt} stages between registers.

The maximum permissible negative clock skew in equation (3.3) can be represented by equation (7.3) where T_e is the aggregate delay due to the initial and final registers and the clock distribution network of a local data path. T_e can be used to represent the margin of error or the acceptable tolerance of negative clock skew for each local data path.

$$T_e = T_{REG} + T_{SKEW} \quad (7.3)$$

Note that when T_{SKEW} is zero, T_e equals T_{REG} . Also note that T_e is typically positive for most circuit configurations. Section 10 of this chapter discusses the special case where T_e can be made negative ($-T_{SKEW} > T_{REG}$) and describes how this approach can further improve performance but only for restricted applications.

T_e can be described as the total effective delay of the registers and clock distribution network per local data path. Each local data path within a global data path provides its own T_{ek} where k is the k th local data path. The average T_{ek} for a global data path is defined as T_{eM} , where M is the number of serially connected cascaded data paths.

Thus, the data throughput is the summation of the total delay through the global data path due to the N logic and interconnect stages, the L registers, and the M local clock distribution networks.

$$\begin{aligned}
 D_T &= \sum_{i=1}^N T_{fi} + \sum_{k=1}^M T_{ek} \\
 &= N T_{fN} + M T_{eM}
 \end{aligned}
 \tag{7.4}$$

The minimum data throughput of a system occurs when no registers exist in a global data path and is the summation of the N individual logic delays as shown by equation (7.5).

$$D_{Tmin} = \sum_{i=1}^N T_{fi} = N T_{fN}
 \tag{7.5}$$

3) Design Paradigm for Pipelined Synchronous Systems

Registers are inserted into global data paths in order to increase the clock frequency of a digital system with, albeit, an increase in the data throughput. This tradeoff between clock frequency and data throughput is graphically described in Figure 7.2. In this figure both the data throughput and the clock period are shown as a function of the number of pipeline registers M inserted into a global data path. Thus, as M increases, the throughput increases by T_{eM} for each inserted register and the maximum possible clock frequency increases, since the critical path is shortened (since there are less logic and interconnect stages per local data path).

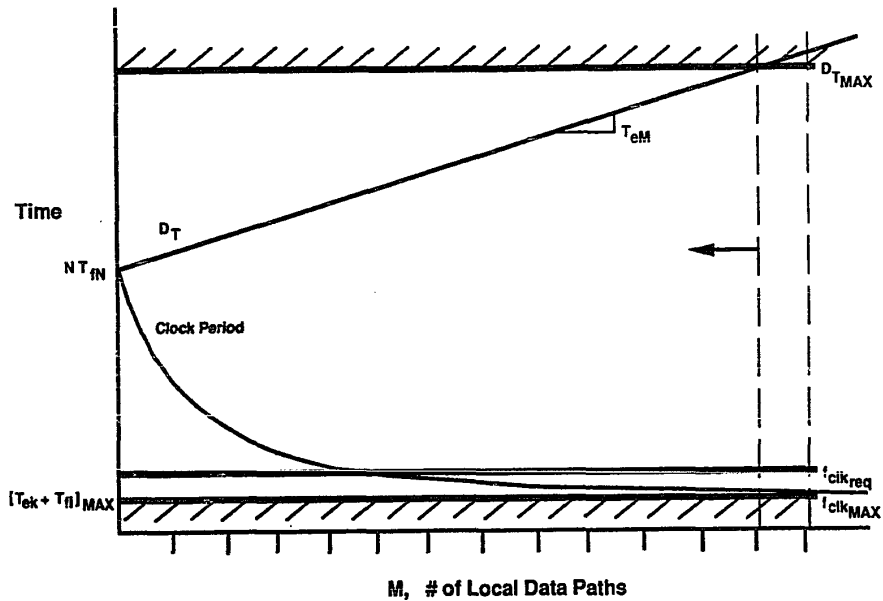


Figure 7.2: Design Paradigm for Pipelined Synchronous Systems

If no registers are inserted into the data path, the minimum data throughput D_{Tmin} is the summation of the individual logic delays, $N T_{fn}$, as shown by equation (7.5). As each register is inserted into the global data path, D_T increases by T_{eM} . Thus D_T increases linearly with M as shown by equation (7.4) and depicted in Figure 7.2.

The average number of logic stages per local data path n is given by equation (7.6) below:

$$n = N/M \quad (7.6)$$

From equations (7.2), (3.3), (7.3), and (7.6), the clock period can be expressed as

$$T_{\text{clock period}} \geq T_{\text{REG}} + n T_{fN} + T_{\text{SKEW}} \quad (7.7a)$$

$$= \begin{cases} N T_{fN} & \text{for } M = 0 \quad (7.7b) \\ T_{eM} + N T_{fN}/M & \text{for } M \geq 1 \quad (7.7c) \end{cases}$$

This inverse proportionality with M is depicted in Figure 7.2, where the maximum practical clock frequency occurs when n equals one as defined by $T_{ek} + T_{fi}$. This assumes that logical operations must be performed (i.e., not a simple shift register). The MAX subscript is used to emphasize that the critical local data path constrains the minimum clock period (maximum clock frequency) of the total global data path.

Most design requirements must satisfy some specified maximum time for data throughput while satisfying or surpassing a required clock frequency. The design constraints due to D_{TMAX} and f_{clkMAX} are shown in Figure 7.2 by the vertical dashed lines. Thus, for a given D_{TMAX} , the recommended maximum clock frequency and pipelining is defined by the intersection of D_T and D_{TMAX} . If D_{TMAX} is not specified and the desire is to make the clock frequency as high as possible, then the recommended f_{clk} is defined by the intersection of the clock period and the f_{clkMAX} line. Thus, for a specified D_T and f_{clk} , the possible design space is indicated by the horizontal

arrow. If D_T and f_{clk} are both of importance and no D_{TMAX} or f_{CLKMAX} is specified, then some optimal level of pipelining is required to provide a "reasonably high" frequency while maintaining a "reasonable" data throughput. This design choice is represented by a particular value of M , defining an application specific f_{CLK} and D_T .

These design choices have been investigated and quantitative design equations are presented which relate the latency of a system to its data throughput and clock frequency. Figure 7.2 depicts both the possible design space and the constraints that exist in determining the correct amount of pipelining in a synchronous digital system.

The effect of clock skew and technology on data throughput and clock period is graphically demonstrated by Figures 7.3 and 7.4. If the clock skew is positive or if a poorer technology (i.e., slower) is used, as shown in Figure 7.3, then T_{eM} increases and D_T quickly reaches D_{TMAX} . In addition, the minimum clock period increases (decreasing the maximum clock frequency) which, for large positive skew or a very poor technology, eliminates any possibility of satisfying a specified clock frequency f_{CLKreq} and decreases the entire design space as defined by the intersection of D_T and D_{TMAX} .

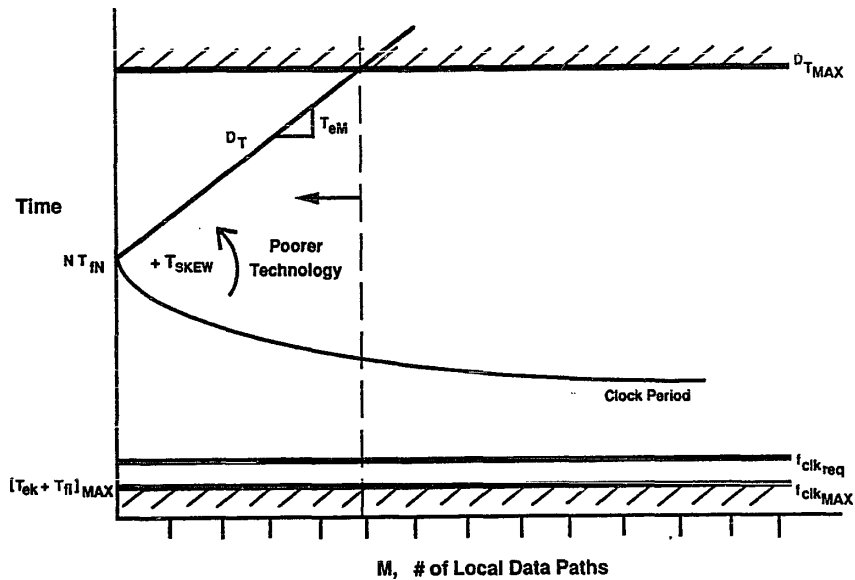


Figure 7.3: Effect of Positive Clock Skew and Technology on Design Paradigm

If the clock skew is negative or a better technology (i.e., faster) is used, as shown in Figure 7.4, T_{eM} decreases, permitting the data throughput to be less dependent on M . In addition, the minimum clock period decreases, satisfying f_{CLKreq} more easily and f_{CLKMAX} with minimal pipelining. The optional design space, represented by the intersection of D_T and D_{TMAX} , is much larger, permitting higher levels of pipelining if very high clock rates are desired.

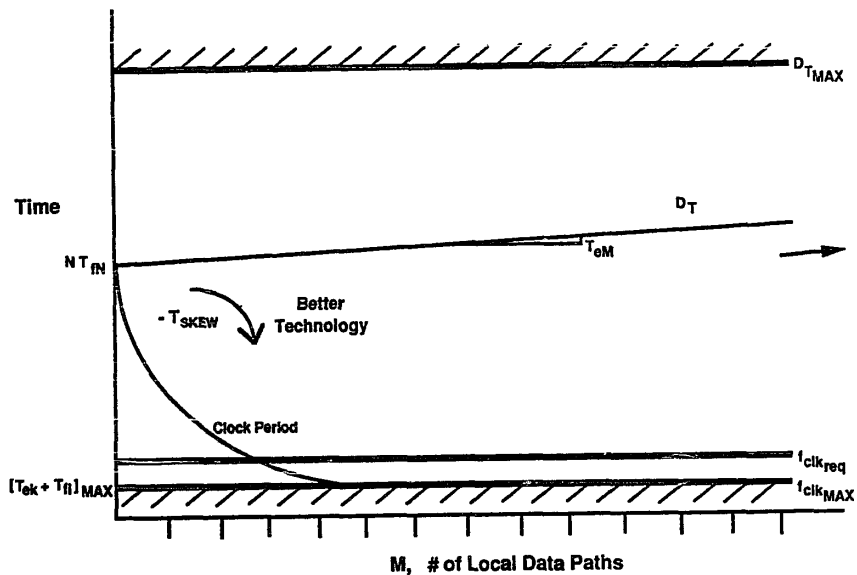


Figure 7.4: Effect of Negative Clock Skew and Technology on Design Paradigm

It should be noted, as mentioned in section 2, that T_e can be less than or equal to zero for restricted circuits. In this case, D_T would be flat for T_e equal to zero and actually have a negative slope for T_e less than zero. This result is elaborated on in greater detail in section 10 of this chapter. Thus, Figures 7.3 and 7.4 graphically describe how both clock skew and technology affect both the data throughput and the maximum clock frequency of a pipelined synchronous digital system.

4) Design Equations Describing Data Throughput

The data throughput of a synchronous digital system can be described as the time required to move data from the input of a system through cascaded local data paths to its output. For a pipelined global data path D_T is given by the relation

$$D_T = L/f_{clk} \quad (7.8)$$

The latency of a global data path can also be described by equation (7.9) below, where M is the number of local data paths per global data path and the plus one term includes an added register at the beginning of the data path.

$$L = M + 1 \quad (7.9)$$

If one writes

$$\sum_{i=1}^N T_{fi} = N T_{fN} \quad (7.10)$$

where T_{fN} is the average stage delay of all the stages within the data path, then the clock frequency of a pipelined data path is given by

$$f_{clk} \leq \frac{1}{NT_{fN}/M + T_e} \quad (7.11)$$

By combining equations (6.8), (3.3), (7.9), and (7.10), the total data throughput of a partitioned global data path can be expressed as

$$D_T = [M + 1] \cdot [nT_{fN} + T_e] \quad (7.12)$$

In applications where the maximum data throughput of a system is specified and D_{TMAX} constrains the design space, the clock frequency and latency can be determined from equations (7.13) and (7.14), respectively, where T_{eM} is the average $T_{REG} + T_{SKEW}$ delay of all of the M pipelined data paths of an N stage data path.

$$f_{clk} \geq \frac{D_T - N T_{fN}}{T_{eM} D_T} \quad (7.13)$$

$$M \leq \frac{D_T - N T_{fN}}{T_{eM}} \quad (7.14)$$

In applications where the maximum clock frequency is specified and f_{CLKMAX} constrains the design space, the latency and data throughput can be determined from equations (7.15) and (7.4), respectively.

$$M = \frac{NT_{fN}}{T_{Clock\ Period} - T_{eM}} \quad (7.15)$$

Thus, as shown in Figure 7.2 for a given maximum data throughput (maximum clock frequency) and knowledge of the average logic, register, and clock delay characteristics of a global data path, the minimum clock frequency (data throughput) and the required latency of a pipelined synchronous data path can be directly determined.

5) Performance Cost of Latency

Every register added to a data path decreases the data throughput of a system by the added time required to move the data signal in and out of the register, T_{REG} . This is typically accepted in order to increase the system clock frequency [43,62,65-67]. However, as added stages of pipelining are inserted into a data path, the marginal utility of the increased clock rate is approached. This occurs when the time required to move data in and out of the register becomes comparable to or greater than the time incurred performing the logic functions.

$$T_{REG} \geq \sum_{i=1}^n T_{fi} \quad (7.16)$$

Each register added increases D_T by T_{REG} and decreases the maximum clock period by the decreased logic delay. Equation (7.16) represents the point where the increase in latency costs the system (in increased data throughput time) more than the increase in clock frequency benefits the system. In order to quantify this, an arbitrary performance criterion is defined to describe the performance cost of latency or the efficiency of pipelining, P_e . P_e is a measure of the relative performance penalty incurred for n stages of logic per one pipelined local data path and describes the cost in performance (increased data throughput time) incurred by

the insertion of a single additional pipeline register. This normalized function is the ratio of the total logic delay to the total path delay, thereby defining what percentage of the data path delay is logic related and what percentage is register related. As n increases, the ratio of the total logic delay to the total data path delay increases toward one and reaches one when n is infinite (or practically, when the total logic delay is much greater than the register delay).

$$P_e = \frac{\sum_{i=1}^n T_{fi}}{T_{PD}} = \frac{\sum_{i=1}^n T_{fi}}{\sum_{i=1}^n T_{fi} + T_{REG}} \quad (7.17)$$

The benefit of inserting a register into a data path is increased clock frequency as described by equation (3.3) and redefined below:

$$f_{clk} \leq \frac{1}{T_{REG} + \sum_{i=1}^n T_{fi} + T_{SKEW}}$$

$$= \frac{1}{\sum_{i=1}^n T_{fi} + T_e} \quad (7.18)$$

Thus, the cost/benefit of inserting registers into an N stage data path can be represented by the function $P_e f_{clk}$, where P_e increases for increasing n and f_{clk} decreases for

increasing n . P_{efclk} is thus a figure of merit for representing the performance advantages and disadvantages of pipelining. A different function could be applied if the effects of increased area were also of significant importance [65-67,69]. However, this result emphasizes an optimal data throughput and clock frequency over area/speed optimization. If equation (7.10) is combined with equations (7.17) and (7.18), then the figure of merit P_{efclk} can be described as

$$P_{efclk} = \frac{nT_{fN}}{(nT_{fN} + T_{REG})} \cdot \frac{1}{(T_{REG} + nT_{fN} + T_{SKEW})} \quad (7.19)$$

The clock frequency and data throughput of a pipelined data path can be described in terms of its performance efficiency and delay and skew characteristics, as given by equations (7.20) and (7.21).

$$f_{clk} = \frac{1}{P_e T_{PD} + T_{REG} + T_{SKEW}} = \frac{1}{P_e (T_{REG} + nT_{fN}) + T_e} \quad (7.20)$$

$$D_T = [N/n + 1] \cdot [P_e (T_{REG} + nT_{fN}) + T_e] \quad (7.21)$$

6) Optimal Number of Logic Stages

An optimal number of logic stages N_{opt} in terms of maximizing the product P_{efCLK} is obtained from equation (7.22).

$$\frac{d(P_{efclk})}{dn} = 0 \quad (7.22)$$

Equation (7.22) represents the point where the cost/benefit of pipelining in a high performance synchronous system reaches a maximum. From equations (7.19) and (7.22), the optimal number of logic stages between registers N_{opt} which maximizes the function $P_e \times f_{clk}$ can be derived. Thus N_{opt} is the optimal number of logic stages per local data path for a high speed design where both clock frequency and data throughput are of importance and no constraints on D_{TMAX} or f_{CLKMAX} are specified.

$$N_{opt} = \frac{1}{T_{fN}} \sqrt{T_{REG}(T_{REG} + T_{SKEW})} \quad (7.23)$$

Note that T_{REG} is defined in equation (7.1), T_{fN} is the average stage delay of the entire data path, and T_{SKEW} can be zero, negative (assuming $T_{SKEW} \leq T_{REG}$), or positive.

Under the condition of an ideal clock distribution network with zero clock skew, equation (7.23) simplifies to equation (7.24).

$$N_{opt} = T_{REG}/T_{fN} \quad (7.24)$$

N_{opt} is simply a ratio of the register delay overhead to the average stage delay of the data path. If $T_{REG} \ll T_{fN}$, which occurs when the stage is a large high level function, then the cost of inserting registers is small and N_{opt} should be as small as feasible (since N_{opt} is discrete, its smallest realizable value is one stage) or

one should pipeline as often as the system permits. If $T_{REG} \gg T_{fN}$, which is more common when operating at the level of individual logic stages, then the cost of inserting registers is high and N_{opt} is some large number defined by equation (7.24). Another interpretation of equation (7.24) is that the optimal number of logic stages between registers occurs when the total logic path delay NT_{fN} equals the total register delay T_{REG} , thereby maximizing P_{efclk} .

Thus, knowing N_{opt} and the average gate delay T_{fN} , the optimal frequency at which a particular data path (and system) should operate at is given by equation (7.25) where N_{opt} in an actual application should be rounded to an integer value.

$$f_{clkopt} \leq \frac{1}{T_{REG} + N_{opt} T_{fN} + T_{SKEW}} \quad (7.25)$$

The number of additional registers R_{opt} which should be inserted into a high performance global data path (composed of two registers, R_i and R_f , as shown in Figure 7.1) is given by equation (7.26) below:

$$R_{opt} = N/N_{opt} - 1 \quad (7.26)$$

where N is the total number of stages, N_{opt} is given in equation (7.23), and the minus one is due to the condition that the original global data path is assumed to have two

registers before the insertion of any additional pipeline registers. Also, N_{opt} represents the maximum number of logic stages within a critical data path and therefore if N/N_{opt} is not an integer then some of the data paths should contain $N_{opt} - 1$ logic stages.

7) Effect of Clock Skew on Optimal Number of Logic Stages

Clock skew, the difference in delay between two sequentially adjacent clock paths, is an important determinant of the maximum operating frequency of a synchronous digital system. As shown in Figure 7.1 and described in Chapters 3 and 4, bounds on the minimum and maximum clock delay of each clock path determine the lead or lag nature of the clock skew as well as its magnitude for any particular synchronous data path. If the time of arrival of the clock signal at the final register of a data path (C_f) leads that of the clock signal at the initial register of the same sequential data path (C_i), then the clock skew is defined as positive; this condition degrades the maximum attainable operating frequency. If C_f lags C_i , the clock skew is defined to be negative; this can be used to improve the maximum performance of a synchronous system. This is shown by equation (3.3) in which T_{PD} is the total delay of the critical data path from the initial register to the final register. The maximum permissible negative clock skew of any data path,

however, is dependent upon the previous data paths, since the earlier C_i is for a given data path, the earlier that same clock signal, now C_f , is for the previous data path.

T_{SKEW} in equation (7.23) can be zero, negative, or positive with the constraint that if T_{SKEW} is negative, then its magnitude must be less than or equal to T_{REG} . It is interesting to note that the effect of clock skew on N_{opt} is relative to T_{REG} and T_{fN} . Thus, if T_{REG} is large with respect to T_{SKEW} , the relation essentially reduces to equation (7.24). Also, positive clock skew adds directly to T_{REG} and increases the cost of pipelining, thereby increasing the recommended number of logic stages between registers and quantifying how the clock distribution network affects the optimal design of a high speed data path.

8) Example of Algorithm

In the design of most current synchronous digital systems, one of two approaches is used to partition a high speed data path into multiply cascaded data paths, each isolated by a pipeline register. The first approach assumes there is a target specification for clock frequency, thus defining a maximum clock period. Time is allocated to move the data signal in and out of the register and account for any positive clock skew, leaving some remaining time to do useful logical manipulations of

the data signal. As many stages of logic are then included as possible, thereby defining where the next register should be inserted.

The second approach assumes there is no target clock frequency other than a goal of "as high a frequency as is reasonably possible." In this approach, the overall data path is usually partitioned into cascaded data paths that are functionally convenient and a reasonable multiple of the necessary register overhead. The time allocated for the logic stages is typically one to two times the total register delay, representing a heuristically defined acceptable cost, typified by 3 to 4 logic stages or 1 to 2 bits of addition between registers [63,70].

The algorithmic approaches discussed here describe how each of the key design parameters interacts and define an appropriate level of pipelining for synchronous applications. The design equations discussed in this chapter are compared to the aforementioned ad hoc approaches in the following example.

Figure 7.5 depicts a 30 stage global data path in which each stage is of varying delay. In this example, the following characteristics are assumed:

$$T_{C-Q} = 3 \text{ ns.}$$

$$T_{\text{set-up}} = 3 \text{ ns.}$$

$$T_{\text{SKEW}} = 2 \text{ ns.} \quad (\text{positive skew})$$

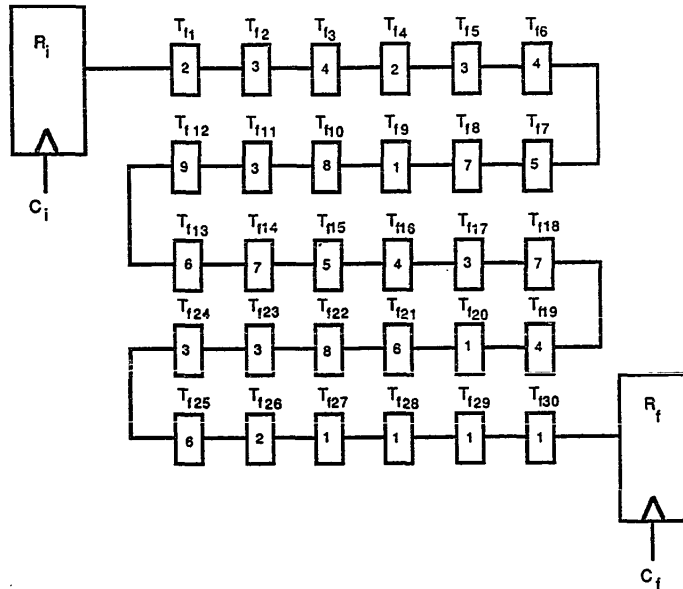


Figure 7.5: 30 Stage Data Path

The delay of each stage T_{fi} , for $i=1$ to 30 (shown in Figure 7.5), encompasses both the local interconnect delay and the delay through the logic stage. It should be noted that the classic ad hoc approaches typically assume positive clock skew and therefore, this is assumed in the example. Also, no data throughput or clock frequency requirement has been specified and therefore the optimal choice of D_T and f_{CLK} will be determined by the approach described in section 6.

Ad Hoc Approach 1

Assuming a specified clock frequency of 50 Mhz., 12 ns. per local data path is allocated to perform the logical operations. From perusal of Figure 7.5, the global data path is partitioned into 13 local data paths, requiring 12 additional registers.

If a 100 Mhz. clock frequency were assumed, only 2 ns. would remain per local data path. In order to meet this performance requirement, many of the logic stages would require multiple individual subdivisions and a total of 67 local data paths would be required for this global data path to operate at 100 Mhz., providing a data throughput of 670 ns. One can see the significance of P_e in this example (Note that $T_{REG} \gg N T_{fN}$).

Ad Hoc Approach 2

If the goal is as "high a frequency as is reasonably possible," then each local data path would typically be composed of three to four stages. This would decompose the global data path into eight local data paths, thereby requiring seven additional registers and operating at a maximum clock frequency of 38.5 Mhz. since the minimum clock period is $18 \text{ ns.} + T_{REG} + T_{SKEW} = 26 \text{ ns.}$

Algorithmic Approach

Since the total delay of the 30 logic stages is 120 ns., the average delay per stage T_{fN} is 4 ns./stage. Using equation (7.23) and the aforementioned register and clock skew characteristics, N_{opt} equal 1.56 stages per local data path. Thus, a logic delay of 6.3 ns. per local data path optimally trades off the maximum clock frequency against the data throughput efficiency. Thus, based on the characteristics of the global data path, a clock frequency f_{clkopt} of 70 Mhz. is recommended for this system.

These results are summarized in Table 7.1 below:

	f_{clk}	nT_{fN}	M	D_T
Ad Hoc Approach 1				
A:	50 Mhz.	12 ns.	13	224 ns.
B:	100 Mhz.	2 ns.	67	656 ns.
Ad Hoc Approach 2	38.5 Mhz.	18 ns.	8	184 ns.
Algorithmic Approach	69.9 Mhz.	6.3 ns.	18	264 ns.

Table 7.1: Comparison of Pipelining Approaches
in Example Data Path

Thus, for a design problem in which the design space is unconstrained (no D_{TMAX} or f_{CLKMAX} is specified), the algorithmic approach provides a technique for determining the appropriate level of pipelining M which optimizes both

f_{CLK} and D_T and in terms of the specific performance characteristics of each global data path. As shown in Table 7.1, the algorithmic approach, optimized for speed efficiency, provides a pipelined data path with relatively high clock frequency (70 Mhz.) while yet maintaining reasonable data throughput (264 ns.)

9) Maximum Performance of Optimized Data Path

The performance of a pipelined synchronous system can be maximized when the cost of the registers is minimal, thereby permitting the frequent insertion of pipeline registers and a higher overall system clock rate. As shown in equation (7.23) and Figure 7.4, if T_{SKEW} is negative, the overall performance cost of pipelining decreases.

$$N_{opt} = \frac{1}{T_{fN}} \sqrt{T_{REG}(T_{REG} + T_{SKEW})} \quad (7.23)$$

The maximum permissible negative clock skew in equation (7.23) can be represented by T_e , where T_e is the margin of error or acceptable tolerance as defined by equation (7.3). Equation (7.23) can then be rewritten as equation (7.27) below:

$$N_{opt} = \frac{\sqrt{T_{REG}T_e}}{T_{fN}} \quad (7.27)$$

Since N_{opt} must be an integer number of logic stages, \bar{N}_{opt} is used to designate a rounded integer value of equation (7.27). The optimal clock frequency can now be represented by equation (7.28).

$$f_{clkopt} \leq \frac{1}{\bar{N}_{opt}T_{fN} + T_e} \quad (7.28)$$

The following example is helpful in the interpretation of these results.

Assume

$$T_{REG} = T_{c-Q} + T_{set-up} = 6 \text{ ns.}$$

$$T_{fN} = 2 \text{ ns.}$$

Using an ad hoc approach and assuming an ideal clock distribution network, (i.e., $T_{SKEW} = 0 \text{ ns.}$), the maximum clock frequency is 83.3 Mhz. for a three stage logic path and 125 Mhz. for a single stage fully pipelined data path with the aforementioned delay characteristics. This compares to the algorithmic approach which, for $T_e = 0.5 \text{ ns.}$ ($T_{SKEW} = -5.5 \text{ ns.}$), \bar{N}_{opt} equals one and, from equation (7.26), a maximum clock frequency of 400 Mhz. is possible.

Depending upon the magnitude and lead/lag behavior of T_{SKEW} , T_e can range from a small positive number to a very large value, severely limiting the maximum clock frequency

of the system. Figure 7.6 depicts the maximum clock frequency as a function of T_e for specific values of T_{REG} and T_{fN} . Note that if negative clock skew is used, the maximum clock frequency becomes significantly larger. For the previously cited example, by using negative clock skew the maximum clock frequency increased from 83 Mhz. to 400 Mhz. (500 Mhz. with no tolerance, i.e., $T_e = 0$). Two curves are provided, one discontinuous since \bar{N}_{opt} is an integer and the second continuous since a continuous N_{opt} is assumed.

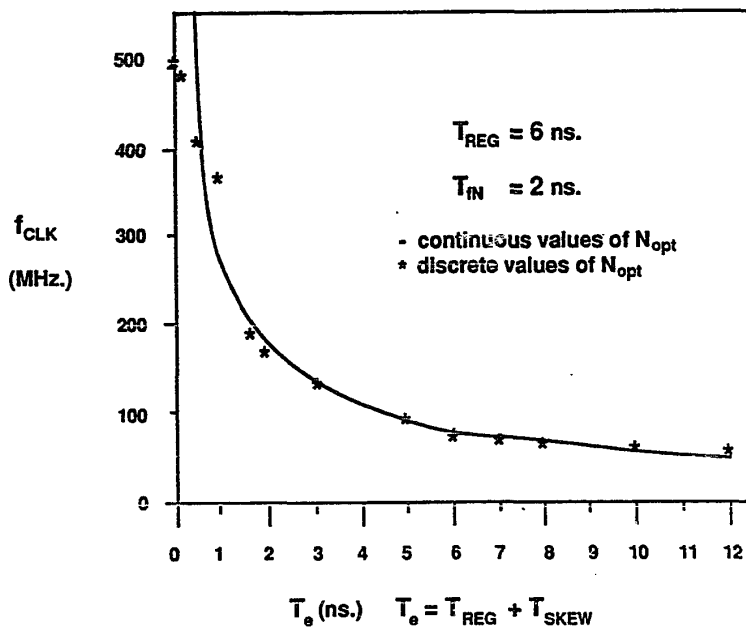


Figure 7.6: Maximum Frequency as a Function of Relative Clock Skew

When $T_e = 0$, equation (7.28) represents the clock frequency, f_{clk0} , when $T_{REG} = -T_{SKEW}$. Thus, f_{clkopt}/f_{clk0} represents the normalized optimal clock frequency and can be described as solely a function of the acceptable tolerance T_e and the register delay T_{REG} . This function is shown in equation (7.29) and plotted in Figure 7.7.

$$f_{clkopt}/f_{clk0} = \frac{1}{1 + \sqrt{T_e/T_{REG}}} \quad (7.29)$$

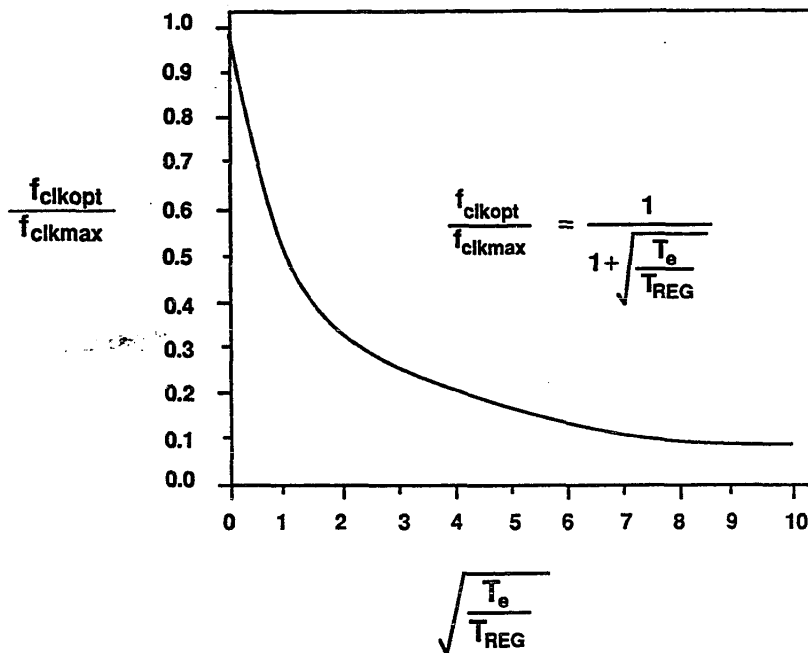


Figure 7.7: Normalized Optimal Clock Frequency

10) Theoretical Maximum Clock Frequency

The absolute maximum frequency of a data path in a synchronous digital system is only constrained by the resolution of its device and design technologies and the speed at which the input data signal is changing. If T_{SKEW} is designed to be negative and just less than T_{PD} of each local data path, the absolute maximum frequency is only limited by how close in negative magnitude each local T_{SKEW} can be designed to the individual T_{PD} of each local data path. If any feedback exists between the local data paths, this approach would be compromised in order to consider the additional feedback paths. Also, for signals which feed out to multiple ports, this design approach must account for the variation in T_{PD} between data signals. However, in this case the critical worst case path could be improved with negative clock skew with the constraint that the fastest signal path of each local data path would have a negative clock skew smaller than the signal path with the smallest T_{PD} . Also, additional delay could be added to the fastest path of each local data path to minimize this minimum constraint relationship [see equation (3.57)].

Assuming these conditions are observed, the maximum clock frequency of a data path is given by the following relation:

$$f_{\text{clkmax}} = \lim_{e+ \rightarrow 0} 1/e+ \quad (7.30)$$

where

$$e+ = T_{\text{PD}} + T_{\text{SKEW}} \quad (7.31)$$

and $e+$ is the time difference between T_{PD} and the negative clock skew and it must be positive for the circuit to operate correctly [see equations (3.5) and (3.6)]. The minimum value of $e+$ is established by the practical tolerances of the device and design technologies being used to implement the synchronous system. The closer the negative T_{SKEW} approaches T_{PD} , the higher the probability that maloperation of the system will occur.

The clock delays of this system would become large as negative clock skews are accumulated along each local data path; however, the clock frequency of a data path can be made infinitely high with infinitely good resolution. If the performance of a system were measured solely by its clock frequency, this system would be considered to be operating at extremely high performance levels.

This technique of designing in negative clock skew to approach T_{PD} is described in the following example. The global data path shown in Figure 7.8 consists of three local data paths connected serially. The clock signal driving each register is designed in such a way that C_1 always leads C_f , forcing the clock skew to be consistently negative. In this circuit, if C_1 reaches R_1 at 0 ns., C_2

at R_2 at 49 ns., C_3 at R_3 at 94 ns., and C_4 at R_4 at 129 ns., the clock skew of each local data path would be 1 ns. less than its individual T_{PD} . Thus, this system could be continuously clocked every 1 ns., providing a clock rate of 1 Ghz., even though the critical path with zero clock skew would imply a maximum clock frequency of only 20 Mhz. If T_{SKEW} could be designed to approach even closer to T_{PD} , the maximum frequency would further increase. If the negative clock skew could be reliably designed to approach T_{PD} within 0.1 ns., the maximum frequency at which data could be moved through that data path would be 10 Ghz.

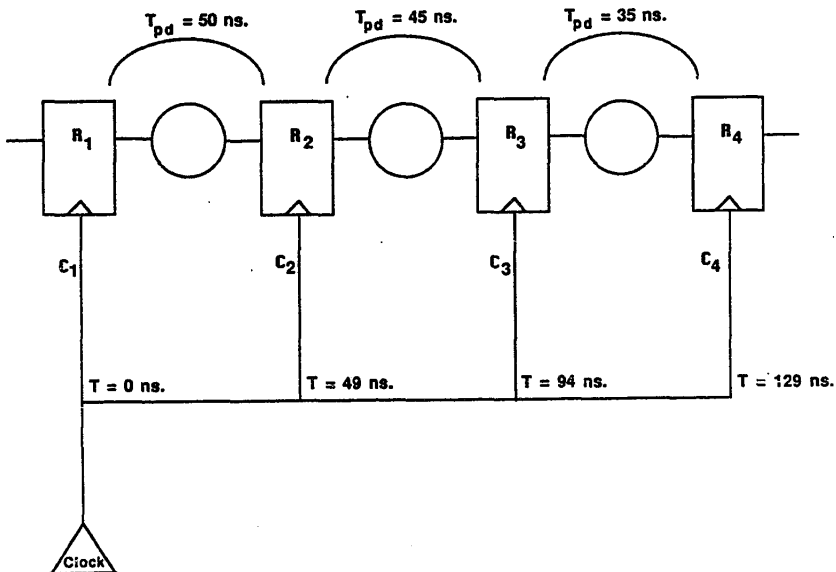


Figure 7.8: Example of Theoretically Maximum Clock Frequency

The disadvantages of this approach, however, are 1) the accumulated clock delays, 2) the severe tolerance requirements on the device technology, 3) the design precision and design time required, 4) the requirement that each cascaded local data path cannot have any feedback, and 5) the requirement that each of the data signals within a data path must be designed to have minimal differences in delay between their minimum and maximum T_{pd} .

11) Summary

In the design of high speed synchronous digital systems, global data paths are often partitioned into local data paths, thereby decreasing the delay of the critical paths and increasing the clock frequency, albeit with an increase in system latency. Data throughput is therefore compromised for increased clock rate. This chapter reports on an investigation of this tradeoff and a design paradigm is described which analyzes how the performance behavior of a synchronous system is affected by its degree of pipelining. This perspective permits the development of analytical design equations for describing pipelined digital systems in terms of the logic and register delays, clock skew, the performance efficiency of pipelining, and the total number of logic stages per local data path.

Three types of problems are described which can be solved using these design equations: 1) D_{TMAX} constrains the design space, 2) f_{CLKMAX} constrains the design space, and 3) the design space is unconstrained and an optimal choice of D_T and f_{CLK} must be determined. Design equations have been described which permit each type of problem to be analyzed and a solution determined. In order to solve the unconstrained design problem, an algorithm for partitioning global data paths into local pipelined data paths is developed which optimizes the effects of increased latency and increased clock frequency on data throughput. This algorithm defines an optimal number of logic stages between pipeline registers in terms of the average logic stage delay of a data path, the delays inherent to the register, and the clock distribution skew characteristics. Examples are provided which quantify how this algorithm is used to design optimal high speed data paths. Finally, a limiting case is described showing that for restricted classes of circuits, the maximum clock frequency is only limited by the resolution of its device and design technologies and the rate at which the input signal is changing.

CHAPTER 8
APPLICATION OF THEORETICAL RESULTS

The two primary goals of this research are to develop the underlying principles and relationships describing the integrated synchronous system composed of the logic path, the registers, and the clock distribution network and to develop a design approach for building high performance synchronous digital systems based on these underlying principles.

This chapter describes a design procedure, using the various theories, algorithms, and equations developed in Chapters 2 through 7, for analyzing and designing high performance synchronous digital systems. This design procedure, which embodies these principles, assumes the following conditions:

A) the digital system is composed of multiple parallel data paths, each composed of a variable number of logic stages with a single initial register and final register at the ends of each data path, as represented by Figure 8.1.

B) the desired functional requirements of the system are known and understood.

C) the timing is fully synchronous (i.e., there is a single global clock pulse defining a time reference for the entire system).

D) the system must be optimized to meet extremely high performance goals.

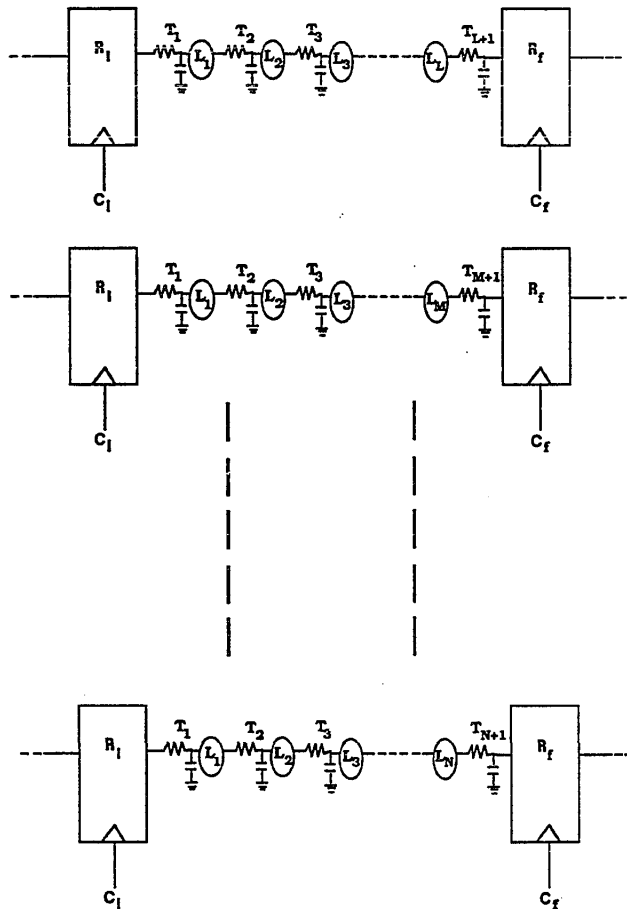


Figure 8.1: Representative Synchronous Digital System

1) Representative Design Problem

As described throughout this dissertation, the focus of all high performance synchronous digital systems is to move data as quickly as possible from one register through some logical elements and latched correctly into the next register. All data paths can be represented by the simple diagram shown in Figure 8.1, no matter where the data signal originated, where its destination is, or what combination of logical circuitry it propagates through.

The design problem described in this dissertation can be described at different levels of functional abstraction:

1) system design - to partition the functional blocks of each system into appropriate data paths which accomplish the functional and reliability goals while satisfying the performance (data throughput and clock frequency) requirements of the system.

2) logic design - to select the logic functions that implement the functional requirements of the system while optimizing the flow of data signals from the initial register to the final register of every critical data path.

3) circuit level - to optimally design the waveform shapes and absolute and relative delays of the key data and clock signals so as to maximize the data flow

through the critical paths and successfully latch the data into the final register.

4) device level - to choose a device and interconnect technology which permits the design of an integrated system which, with the aforementioned approaches, will satisfy the functional, reliability, and performance goals of the total synchronous system.

2) Systematic Design Approach

The design of most systems uses a two-pronged approach: 1) a top-down design flow in which each functional level of abstraction is designed, based on estimates of the characteristics of the lower levels, and then, upon completion, transferred to the next lower level and 2) a bottom-up approach in which the lower levels are implemented, based on a general understanding of the overall goals of the system, in order to satisfy specified density, power, and performance requirements of the total system. Most high performance design efforts utilize both approaches concurrently, constantly feeding back, both up and down, useful constraining information until the total design reaches an optimal implementation. Sophisticated design systems are currently in existence to accelerate the design flow, permitting designers to focus on the optimal design and implementation of their respective design tasks.

The following five subsections describe the design steps appropriate for the implementation of a high performance synchronous digital system. Initially, the system is partitioned into pipelined data paths based on certain assumed systems requirements. This is described in the first subsection. If underlying circuit characteristics must be determined, then a bottom-up approach, discussed in the next three subsections, should be applied to determine the local logic stage, register, and clock skew characteristics of the specific circuit. Further enhancements to the operating speed of the system can be derived by detailed optimization of the clock and data signals. These are discussed in the final two subsections. Thus, this chapter provides a summary of the important design equations developed within this dissertation. More detailed information of each of the design equations has been provided in the previous chapters.

Partitioning the Data Paths

As discussed in detail in Chapter 7, synchronous data paths are commonly partitioned into multiply cascaded data paths in order to increase the clock frequency of the synchronous system. The limiting equations quantifying the maximum clock frequency are described in Chapter 3 and are given by equations (3.2) and (3.3) below. The maximum

frequency of a synchronous digital system is constrained by the data paths with the greatest $T_{PD} + T_{SKEW}$ delay. These data paths are defined to be the critical worst case timing paths of the system.

$$T_{PD} = T_{c-Q} + T_{logic} + T_{int} + T_{set-up} \quad (3.2)$$

$$T_{PD} + T_{SKEW} \leq T_{clock\ period} = 1/f_{clk} \quad (3.3)$$

All data paths and their clock skew which limit the maximum clock frequency of the system should be partitioned into smaller, faster data paths when designing high performance synchronous digital systems. As shown in Figure 7.2, design equations are described in Chapter 7 which determine the clock frequency for a specified data throughput, thereby defining the correct level of pipelining of a performance limited data path. Equations describing this design paradigm are shown below:

Design Space Constrained by Maximum Data Throughput

$$f_{clk} \geq \frac{D_T - N T_{fN}}{T_{eM} D_T} \quad (7.13)$$

$$M \leq \frac{D_T - N T_{fN}}{T_{eM}} \quad (7.14)$$

Design Space Constrained by Maximum Clock Frequency

$$M = \frac{N T_{fN}}{T_{\text{Clock Period}} - T_{eM}} \quad (7.15)$$

$$D_T = N T_{fN} + M T_{eM} \quad (7.4)$$

Optimal Choice of Data Throughput and Clock Frequency

$$N_{\text{opt}} = \frac{1}{T_{fN}} \sqrt{T_{\text{REG}}(T_{\text{REG}} + T_{\text{SKEW}})} \quad (7.23)$$

Additional equations relating the data throughput, clock frequency, and pipelining efficiency are provided in Chapter 7. Implicit to all of these equations is the assumption that the performance characteristics of the registers, logic stages, and clock distribution network are known. If these attributes must be determined, then the following three subsections should be followed.

Determining the Logic Path Delays

As discussed in Chapter 6, the total delay through an N stage data path can be described by equation (6.8) where

$$T_{\text{PD}} = T_{c-Q} + \sum_{i=1}^N T_{fi} + T_{\text{set-up}} \quad (6.8)$$

T_{fi} is the time delay of each individual logic stage composed of a lumped RC interconnect section and a single stage of logic. Each T_{fi} can be determined either

transcendentally by equation (6.4) or approximated by equation (6.6).

$$T_{fi} = \sqrt{T_i V_{l\text{og}i} / k_i} \quad (6.6)$$

Once the delay of each individual logic stage has been determined, the average logic delay per stage per data path, T_{fN} , can be determined.

Determining the Register Delay Characteristics

As discussed in Chapters 5 and 6, the clock-to-Q and set-up characteristics can be determined for each data path. In section 3 of Chapter 6, design equations for $T_{\text{set-up}}$ are shown to depend upon whether the data signal leads or lags the clock signal at the final register. These design equations are given below:

Positive Data-to-Clock Skew

$$T_{\text{set-upD}} \geq |T_{D-C} + |V_{Tp}|/k_c - T_{fN+1} + T_{QVTN}| \quad (6.19)$$

Negative Data-to-Clock Skew

$$T_{\text{set-upC}} \geq |T_{D-C}| + T_{fN+1} - |V_{Tp}|/k_c \quad (6.21)$$

The clock-to-Q delay is determined from the summation of the time delays derived from equations (5.1), (5.2), and (5.9) satisfying the threshold requirements of the first logic stage defined by equation (6.2).

$$V_{log1} = V_{Tn} + \sum_{j=1}^R V_{DSj} \quad (6.2)$$

Determining the Clock Skew Characteristics

As discussed in Chapters 3, 4, 6, and 7, the clock distribution network can either hinder or help the flow of data in a synchronous digital system. Depending upon the nature of the cascaded data paths, one can design-in additional negative clock skew to improve the speed of the critical paths while insuring that 1) no minimum constraints occur ($T_{pd} < \text{negative } T_{SKEW}$) and 2) no new maximum constraints occur (the previous data path doesn't become the critical worst case path with the addition of positive clock skew).

To determine the magnitude and lead/lag behavior of the clock distribution network, one must determine the clock delay of each clock path as discussed in Chapter 4 and summarized by equation (4.5).

$$T_{CDi} = \sum_a T_{Ba} + \sum_b T_{INTb} \quad \text{along path } i \quad (4.5)$$

The clock skew between any two clock paths, i and j , within the same clock distribution network is given by equation (4.6).

$$T_{SKEWij} = T_{CDi} - T_{CDj} \quad (4.6)$$

The clock skew of any data path can be made more negative by adding delay elements to the clock path to the final register, separate from the clock path to the initial register, T_{Efi} [see equations (4.7) and (4.8)]. For a specific data path with data flowing from the initial register to the final register, the magnitude and lead/lag behavior of the clock skew is given by equation (8.1), where T_{SKEW} can be negative or positive as shown in Figure 3.3 and T_{CDi} is the clock delay to the initial register and T_{CDf} is the clock delay to the final register for each local data path.

$$T_{SKEW} = T_{CDi} - T_{CDf} \quad (8.1)$$

Thus, once the delay and clock skew characteristics of the data path have been determined, the data path can be partitioned into multiply cascaded data paths for optimal data throughput. Additional performance can be achieved by shaping the waveforms to minimize the set-up and clock-to-Q times. Finally, the overall throughput can be improved by using clock skew when appropriate as described in the previous subsection. Greater speed enhancements are possible by shaping portions of certain signal waveforms for maximum performance improvement. These are discussed in the following subsection.

Shaping of Register Input Waveforms

As discussed in Chapters 5 and 6, the clock and data input waveforms at the final register can be designed to perform the functional requirements of these two signals as quickly as possible. The primary purpose of the register is to latch the data signal upon arrival of the clock signal. Necessary and sufficient conditions for latching data into a bistable register are repeated below:

$$1: V_{\text{CLK}} < V_{\text{DD}} + V_{\text{Tp}} \quad (5.39)$$

$$2: V_{\text{DATA}} > V_{\text{logRF}} \quad (6.14)$$

$$3: A_1 V_{2\text{out}} + A_2 V_{\text{CLK}} > 0 \quad (5.41)$$

$$4: B_1 V_{1\text{out}} + B_2 V_{\text{DATA}} > 0 \quad (5.42)$$

From this result, the shape of the data signal at the input of the register, k_{N+1} , is given by equation (6.11), where V_{logRf} is given by equations (6.2) and (6.14). Thus, equation (6.11) describes how the output waveform of

$$k_{N+1} = \frac{V_{\text{logRf}}}{T_{N+1}[\exp(-T_{fN+1}/T_{N+1}) + T_{fN+1}/T_{N+1} - 1]} \quad (6.11)$$

the final logic stage should be designed so as to satisfy a specific T_{fN+1} , assuming V_{logRf} and T_{N+1} are known.

The clock signal during region 1, V_{DD} to $V_{DD} + V_{Tp}$, is also of key importance since this time represents wasted time. Therefore, the slope of the waveform in this region of interest, represented by the ramp k_c in equation (5.1), should be as large as possible.

$$T_1 = |V_{Tp}/k_c| \quad (5.1)$$

Analyzing Synchronous Data Paths

The approach used to determine the maximum clock frequency of a pipelined data path is as follows:

1) For each interconnect and logic stage in a data path, calculate T_{fi} from equation (6.6) by using equations (6.2) to determine V_{logi} . T_i is determined from the RC interconnect impedance (and discussed in more depth in Chapter 2), and equations (6.12), (4.21), and (4.22) are used to determine k_i .

2) Calculate T_{set-up} from equation (6.17) by using equation (6.10) for T_{fN+1} and equations (6.18) and (5.1) for T_{latch} .

3) Calculate T_{c-Q} from equations (5.2) and (5.9).

4) Combine the results of 1, 2, and 3 above to give the total data path delay T_{pD} , as defined by equation (6.8).

5) The delay of each of the two clock signal paths are determined by summing the interconnect and buffer delays along each path as described by equation (4.9). The interconnect delay is derived from equations (4.10) and (4.12) to (4.18) and the buffer delay from equations (4.11), (4.20), (4.23), and (4.24). This will provide the minimum and maximum delay of each clock path. If an estimate of the clock path delay is preferred over a bounded range of values, then the average of the minimum and maximum delay can be used to approximate the clock delay.

6) Once each of the clock delays have been determined, the magnitude and lead/lag behavior of the clock skew is given by equation (8.1).

7) With T_{PD} and T_{SKEW} determined, the maximum operating frequency of a data path is derived from equation (3.3).

By adding delay to either the initial or final clock signal path, the characteristics of the clock distribution network can be changed, as described by equations (4.9) and (8.1). Thus, if negative clock skew is desired in order to increase the clock frequency of a particular data path, additional delay elements should be added to the clock signal path driving the final register of the data path. Once T_{CDff} is increased, steps 5 through 7

described above should be followed to determine the increased clock frequency of the data path.

Thus in section 2 of this chapter, a procedure for designing and analyzing high performance synchronous digital systems has been presented. In order to further dramatize the value of these research results, in the following section some example systems are designed and analyzed with this procedure for designing integrated high performance synchronous systems.

3) Illustrative Design Examples

In this section, four different examples are described and solved using the integrated design approach discussed in this chapter. The first example is analyzed to determine its circuit characteristics and validated by comparing the results of the design procedure developed in this chapter to that of SPICE; thereby, quantifying the relative accuracy of this design approach by comparing it to exact numerical solutions of the nonlinear differential equations describing the circuit. The second example describes how our design approach can be used to improve smaller subsolutions as described in the recent literature. In the third example, a systems level specification is defined for data throughput, requiring the maximum latency and minimum clock frequency to be

determined while the fourth example quantifies how negative clock skew can be used to further improve synchronous performance.

Example 1: Use of Design Procedure to Analyze
Synchronous Data Paths

In this example a representative data path, shown in Figure 8.2, is analyzed to determine its performance characteristics, permitting the maximum clock frequency of the data path and its related clock distribution network to be determined. The relative accuracy of these calculations have been validated by comparing the algorithmically derived performance characteristics with that of SPICE using Level 1 Shichman-Hodges device equations [54]. Note that this example is composed of an initial register R_i , four stages of logic and interconnect, and a final register R_f . Typical values of interconnect resistance and capacitance for a 1 to 3 micrometer interconnect technology are used. The data path is composed of four serially cascaded logic stages: a two input NAND gate, an inverter, a three input NOR gate, and a second inverter.

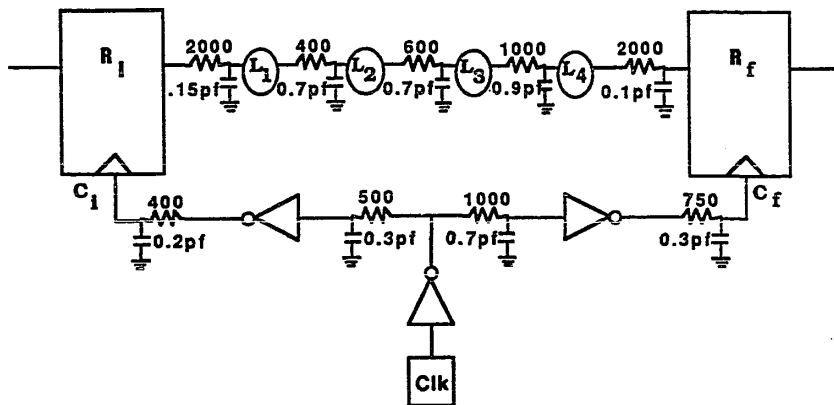


Figure 8.2: Example of an Integrated Synchronous Data Path

The following parameter values were used to characterize the device technology:

$$K' = 2.158 \times 10^{-5} \text{ Amperes/volt}^2$$

$$\lambda = 0.05 \text{ Volts}^{-1}$$

$$U_0 = 500 \text{ cm}^2/\text{volt-second}$$

$$L = 2 \text{ micrometers}$$

$$W = 20 \text{ micrometers}$$

$$\begin{aligned}
 C_O &= 0.005 \text{ picofarads} \\
 C_W &= 0.001 \text{ picofarads} \\
 V_{Tn} &= 1 \text{ Volt} \\
 V_{Tp} &= -1 \text{ Volt} \\
 k_{CLK} &= 2.5 \text{ Volts/ns}
 \end{aligned}$$

The steps described in the subsection on analyzing synchronous data paths were applied in the analysis of the four stage data path depicted in Figure 8.2. The performance characteristics were compared with SPICE simulations of the same circuit and are shown in Table 8.1.

	<u>Equation</u>	<u>Algorithmic</u>	<u>SPICE</u>	<u>Error</u>
T_{PD}	(6.8)	5.37 ns.	5.53 ns.	3.0%
T_{SKEW}	(8.1)	-0.87 ns.	-0.81 ns.	6.9%
f_{clk}	(3.3)	222.1 Mhz.	211.9 Mhz.	4.6%

Table 8.1: Comparison of Algorithmic Results with
SPICE Simulation

Thus, as shown in Table 8.1, the percentage difference between the algorithmically derived maximum clock frequency and the SPICE generated maximum clock frequency is under 5%. In addition, SPICE does not directly provide the total delay of a path since the set-up time of a register cannot be directly determined but

instead must be "backed into" by decreasing the clock period until the register no longer correctly latches the data. Therefore, not only does the algorithmic approach compare favorably with SPICE in analyzing the performance characteristics of a data path but also the set-up time can be directly determined and closed form solutions of all of the key circuit delay characteristics of a synchronous data path are provided in terms of the fundamental material, geometric, and processing characteristics of the device and interconnect technology.

Example 2: Performance Advantages of Integrated Synchronous Design Approach

The approach of investigating how performance becomes limited in synchronous digital systems by emphasizing the interactions between the various subsystems instead of maximizing any one subsystem is novel and therefore difficult to compare with examples published in the open literature. Published efforts to improve synchronous performance tend to focus on one portion of the problem, typically the logic path, and reference how the register and clocking strategy must support their innovative circuit, logic, or technology improvements. Thus, a variety of excellent circuit techniques have been described, too numerous to mention individually, which recommend ways to improve the performance of a logic path.

Recently, in a paper by Yuan and Svensson [71], the importance of the total synchronous system is mentioned; however, emphasis is still placed on their improved circuit design technique for implementing circuit functionality over the optimization of the entire synchronous problem. In order to compare and quantify the value of the research results described in this dissertation to that of the open literature, our results will be applied to the approach described in [71]. In this paper, they develop a circuit technique named TSPC-2 (True Single-Phase-Clock) which statically implements a classical dynamic circuit design approach by inserting fully latching registers into the data path and replacing the multi-phase precharge clock signals with a single-phase clock signal. They compare this circuit design approach to previously published dynamic circuit design techniques [72,73] and show significant improvement in logic delay.

The research results developed in this dissertation can further extend their performance improvements by considering the interactions of the integrated synchronous system. Thus, concepts such as 1) shaping the clock and data signals to minimize the set-up time and clock-to-Q delay by latching the data signal as quickly as possible, 2) using negative clock skew to decrease the required clock period, and 3) applying pipelining techniques to

optimize the system data flow are either not considered or only summarily mentioned. In addition, no quantitative design equations or relationships are provided and all the performance data describing their technique are generated directly from SPICE.

Since no technology parameters are provided in [71], it is difficult to quantitatively extend their results. However, an effort to quantify possible performance improvements has been made and is described below.

In their paper, the maximum extrapolated performance of their 3 micrometer 5 volt CMOS technology is 400-500 Mhz. as derived from their SPICE simulations. Accepting their estimated minimum single stage delay of 0.8 ns. and pipelining every logic stage, the register delay (clock-to-Q and set-up) and clock skew of their circuit, T_e , is in the range of 1.2-1.7 ns.

As described in Chapter 5, all static registers have a minimum time requirement for latching data which is a function of the register and its input waveforms. If we combine this minimum latching time with negative clock skew, then, for the limiting case where $T_e=0$, the maximum performance of their 3 micrometer CMOS technology is a theoretical 1.25 Ghz. (1250 Mhz.). This represents a 250-312.5% improvement over their scaled simulated clock frequency assuming the same technology. It also retains the logic delay of 0.8 ns. as a margin of error for

process and design variation. If we increase T_e to a practical value of 0.5 ns., the maximum possible clock frequency is 769.2 Mhz., representing an improvement in speed of 153.8-192.3% over the authors' ultimate achievable maximum performance.

Thus, this example describes how an integrated synchronous system design approach which considers the interactions between the logic stages, the registers, and the clock distribution network could be used to significantly improve the performance gains achieved solely by optimizing the delay through the logic stages.

Example 3: Derivation of Clock Frequency for a Specified Data Throughput

This example assumes that the delay characteristics of a pipelined data path are known and the focus of the problem is to determine the frequency at which a system should be clocked while not exceeding a specified data throughput goal. This example can be explained in the context of Figure 7.2 where D_{TMAX} cannot be exceeded while providing as high a clock frequency as possible. Thus, the appropriate level of pipelining to maximize f_{CLK} while satisfying the constraint on D_T is determined by the intersection of the D_T and D_{TMAX} curves, defining both M and f_{CLK} .

Equations (7.13) and (7.14) can be used to determine the appropriate clock frequency and latency, respectively, where it is noted that T_{eM} is the average $T_{REG} + T_{SKEW}$ of each local data path along the global pipelined data path.

$$f_{CLK} \geq \frac{D_T - N T_{fN}}{T_{eM} D_T} \quad (7.13)$$

$$M \leq \frac{D_T - N T_{fN}}{T_{eM}} \quad (7.14)$$

Thus, for a 100 stage data path where the average stage delay T_{fN} and average register and skew delay T_{eM} are both 2 ns., f_{CLK} and M can be directly determined for a given target D_T as shown in Figure 8.3. If D_T must be

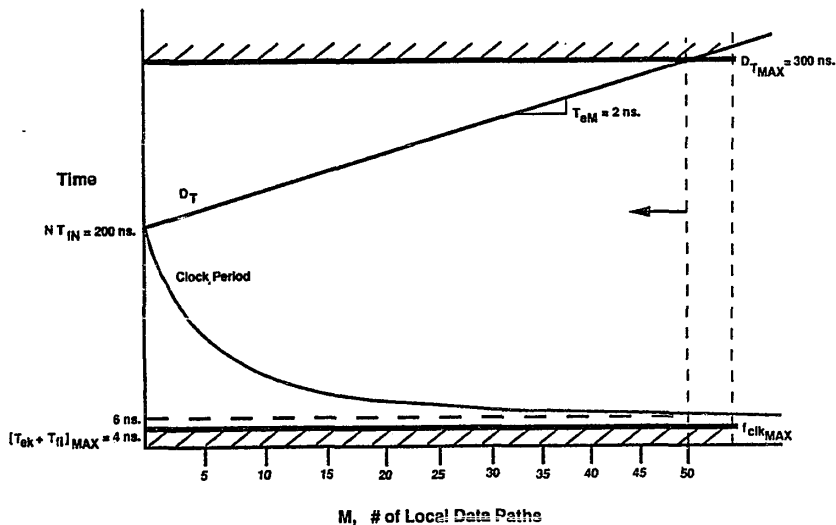


Figure 8.3: Example of Design Paradigm with Constraining Maximum Data Throughput

less than 300 ns., then the maximum latency, from equation (7.14), is 50 local data paths, M (51 pipeline registers). Thus, two logic stages per local data path, $n = 2$, is appropriate for this system. In order not to exceed the target data throughput of 300 ns., the data must flow from register to register a maximum of every 6 ns. ($T_e + 2 T_{fN}$), for a minimum clock frequency of 166.7 Mhz.

Example 4: Derivation of Data Throughput for a Specified Clock Frequency

This example describes how the data throughput is determined for a specified clock frequency (or maximum clock frequency). This example can be explained in the context of Figure 7.2 where in this case the maximum clock frequency, not the maximum data throughput, constrains the design space. The appropriate level of pipelining to minimize D_T while satisfying f_{CLKMAX} is determined by the intersection of the clock period and the f_{CLKMAX} curves, defining both M and D_T .

Equations (7.4) and (7.15) can be used to determine the specific data throughput and latency, respectively, for a given maximum clock frequency.

$$D_T = N T_{fN} + M T_{eM} \quad (7.4)$$

$$M = \frac{NT_{fN}}{T_{\text{Clock Period}} - T_{eM}} \quad (7.15)$$

Thus, for a 100 stage data path where the average stage delay T_{fN} is 2 ns. and the average register and skew delay T_{eM} is 5 ns., D_T and M can be directly determined for a specified clock frequency as shown in Figure 8.4. If the maximum clock frequency is 40 Mhz. (the clock period is 25 ns.), then the latency, given by equation (7.15), is 10 local data paths, M (11 pipeline registers). These ten local data paths add 50 ns. to the data throughput of the global data path, defining a total D_T of 250 ns.

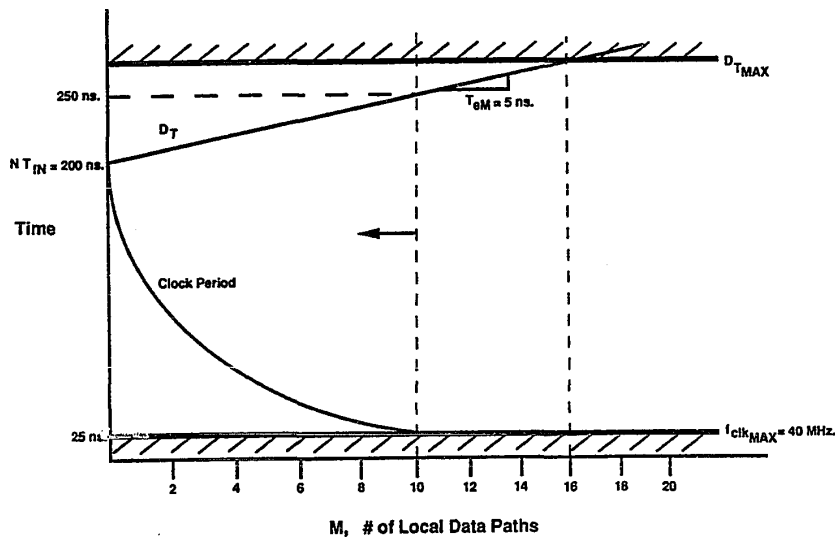


Figure 8.4: Example of Design Paradigm with Constraining Maximum Clock Frequency

**Example 5: Use of Negative Clock Skew to Improve Maximum
Clock Frequency**

This example quantifies how negative clock skew can be used to increase the maximum clock frequency of a synchronous digital system composed of multiple parallel data paths. Figure 8.5 depicts two parallel data paths. The first consists of 25 logic stages with an average stage delay of 2 ns. while the second requires 45 logic stages and has an average stage delay of 2.5 ns. The register delay of these circuits is 5 ns. In order to maximize the speed of these circuits, a negative clock skew of 4 ns. is built into these data paths. Thus from equation (7.23), N_{opt} for path 1 is approximately one logic stage per data path as is also path 2. If we make the assumption that the average stage delay is the actual individual logic and interconnect delay of each stage, then path 1 requires an additional 44 registers. The overall system clock frequency is given by equations (6.8) and (3.3) and is constrained by path 2, since the average stage delay is larger. The clock period is $T_{REG} + T_{fN} + T_{SKEW}$ for a total of 3.5 ns. Thus, the maximum possible clock frequency of this synchronous digital system, using a negative clock skew of 4 ns., is 285.7 Mhz. For an ideal clock distribution system, where T_{SKEW} equals 0 ns., the maximum clock frequency is 133.3 Mhz. If the clock

skew is further increased in the positive direction, the maximum clock frequency will decrease further.

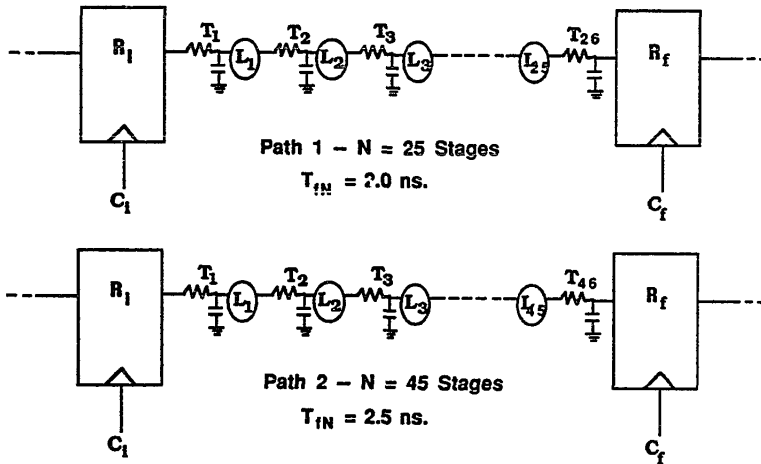


Figure 8.5: Example Circuit with Two Parallel Data Paths

4) Summary

The use of an integrated synchronous system design approach which considers the interactions among the logic path, the registers, and the clock distribution network has been summarized and demonstrated with four examples. Close agreement (less than 5% error) between the design equations developed in this dissertation and SPICE was demonstrated for a representative circuit analysis problem as shown in Table 8.1. Examples depicting how synchronous

performance can be improved over standard or even aggressive circuit design approaches were described. Design equations relating system level requirements in terms of circuit and device characteristics were demonstrated. Thus, this chapter has outlined and exemplified many of the various design principles, equations, and relationships described in this dissertation for applying an integrated synchronous system approach to the design and analysis of high speed synchronous digital systems.

..

CHAPTER 9
DIRECTIONS FOR FUTURE RESEARCH

This research has focused on the underlying principles of high performance synchronous digital systems and their systematic application to engineering problems. Areas for possible extensions of the research described in this dissertation are discussed in this chapter. Improvements to this research are possible by applying these results to a more general class of applicable systems using more sophisticated models with greater accuracy. In addition, the insertion of these design and analysis algorithms into structured design tools would greatly accelerate their use in the engineering community. Specific fertile areas for possible research are described below:

1) Extension of Class of Applicable Systems

The research results presented in this dissertation assume fully synchronous systems with a single phase clock driving a bistable NAND gate register. Extensions to this work are possible by considering more general systems such as described below.

Asynchronous Timed Systems

Asynchronous or self-timed systems [74-78] are commonly used with synchronous systems to create partial synchronous/asynchronous systems in which 1) the global synchronization is asynchronous, communicating with a fully synchronous local system (e.g., parallel processors such as systolic arrays [79-81]) or 2) the global synchronization is fully synchronous, communicating with local asynchronous peripheral modules (e.g., classical Von Neumann architecture computers). In systems which require communication between synchronous and asynchronous subsystems, additional performance constraints occur in which incoming data signals compete with the synchronizing clock pulses in defining the state of a register. This condition of metastability [82-97] can severely degrade the system performance; therefore, special purpose arbitration circuitry [98-104] is commonly used to "arbitrate" between the two incoming signals, thereby significantly decreasing the probability of entering the metastable state. These issues represent an entirely new class of performance limitations caused by the synchronization of a system.

Generalization of Clocking Strategy to Multi-Phase Clocks

The research results described in this dissertation assume a single phase clock, as shown in Figure 3.3. The bistable register, discussed in Chapter 5, requires only a single phase clock to latch the data signals. Both of these types of circuit strategies represent the most common form of synchronization and data storage methodologies used in current industrial applications. However, in certain companies, multi-phase clocking strategies (either two or four independent clock signals) are used [38,72,105,106]. In these designs, typically two non-overlapping clock signals are globally distributed throughout the system with their opposite polarity generated and distributed locally. These clock signals drive special registers, each synchronized by different clock phases, permitting the logical manipulation of portions of the data stream to occur at different times; thereby, potentially improving the overall synchronous performance of the digital system. Multi-phase clocking strategies, however, require a non-activity time since the clock phases are not permitted to overlap. This lost time as well as the additional complexity and control requirements on the clock distribution circuitry due to the added clock signals can potentially compromise the possible performance advantages of the multi-phase

clocking strategy. The research results developed and described in this dissertation have significant applicability to the multi-phase clocking problem and would provide a useful starting place for investigating the problem of how multi-phase clocking interacts with the data path and registers to affect the performance of synchronous digital systems.

Wider Variety of Register Circuits

Chapter 5 describes limitations to latching data into a register in terms of the feedback latch mode (region 3) in a register. In order to analyze these characteristics, a specific implementation of a register was used (a CMOS bistable NAND gate). Other register circuit designs and device technologies are also commonly used in industrial applications. Therefore, extensions to this work would analyze the register latching behavior of other commonly used device technologies (e.g., ECL, TTL, NMOS, E/D GaAs) with different register circuit configurations (e.g., master/slave, JK) or non-latching registers such as dynamic registers [73].

2) More Sophisticated Models

Throughout this dissertation, models are used to represent the behavior of various circuits and devices. Further accuracy can be attained and more general problems can be analyzed by extending these research results to more sophisticated models. Specific examples are provided below.

More Accurate CMOS Device Models

Models representing the device behavior of CMOS transistors were used in both Chapter 4 and in Chapter 5. Classical Shichman-Hodges current-voltage equations [54] were used to model the cascaded buffers in the clock distribution networks as well as the bistable NAND gate register. Additional accuracy can be attained by using more sophisticated current-voltage and capacitance models [107]. These more accurate models would tighten the clock skew bounds as well as provide a more precise small signal representation of the bistable register. Unpublished work by this author indicates that for values of γ from 0 to $0.37 \text{ V} \cdot 5$, the percentage error increases by only 1.2%. However, when the effect of transistor capacitances is analyzed, the percentage error increases from 3.7% to 16.1% (for the range of parameter values used), depending upon the relative magnitude of the device and load capacitances.

Tightening of Clock Skew Bounds

The tightness of the bounds on clock delay, as discussed in Chapter 4, is directly dependent upon the tightness of the individual interconnect and buffer delay bounds making up the clock signal path. Recent work by Wyatt and others [108-118] describe extensions to the Penfield-Rubinstein algorithm [49,50] which both tighten and generalize the interconnect delay bounding problem. Improved algorithms for tightening the delay bounds of a CMOS buffer, as well as generalizing the CMOS buffer to other technologies, are also possible and worthy of further investigation. With tighter bounds on both the interconnect delay and the buffer delay, the overall clock skew bounds are improved, thereby permitting greater accuracy in the analysis of critical worst case paths in synchronous digital systems.

Improved Models for Data Throughput

The model for data throughput used in Chapter 7, equation (7.1), represents a straight forward expression for the data throughput of a synchronous digital system. This equation models the system as a bit-serial architecture in which the data throughput is the time required for data to serially propagate through a pipelined data path. More sophisticated expressions for the data throughput of a system exist which consider the

nature of the specific functional behavior of the system [68,69,119-121]. For example, system characteristics such as floating point vs. fixed point architectures in terms of operations per cycle could be used to express the data throughput of a system and provide more sophisticated relationships between clock frequency and data throughput.

3) Implementation of Algorithms in Design Tools

A practical extension of these research results is the integration of certain design and analysis algorithms described in this dissertation into specific types of CAD tools. As discussed in Chapter 4, the clock skew bounding algorithm would have immediate utility in many general purpose timing analyzers [55-59]. These tools evaluate whether a synchronous digital system satisfies its clock frequency specifications by determining exhaustively the critical paths and comparing the delay of these paths with their specified minimum clock period. These timing analyzers typically don't consider clock skew and, as described throughout this dissertation, clock skew, both negative and positive, can considerably affect system performance.

The data throughput, clock frequency, and latency algorithms developed in Chapter 7 would also be appropriate for integration into certain types of CAD tools, specifically in the emerging area of behavioral

synthesis [122-124]. Minimal research has been done on partitioning high level systems for maximum performance. Much of the current research in this area concentrates on the worthwhile activity of maintaining functional correctness in minimal area. Therefore, the algorithms developed in Chapter 7 could potentially enhance the performance optimization of these behavioral synthesis tools.

CHAPTER 10

CONCLUSIONS

The primary contributions of this research involve the analysis and integration of the limiting factors affecting the performance of synchronous digital systems. Specifically, the underlying principles and relationships constraining the performance of an integrated synchronous system were established. Furthermore, the application of these principles into an integrated design approach focusing on the design and analysis of high performance synchronous digital systems was demonstrated.

These results were developed by partitioning synchronous digital systems into three components: the logic path, the memory elements, and the clock distribution network. The development of an integrated approach to the design and analysis of high speed synchronous digital systems was made possible by investigating how these subsystems interact with each other. The three components were first analyzed individually and their behavior quantitatively described. Then, the interactions of the individual components on each other were investigated in terms of how each component limits data flow.

Relationships were developed which describe how the clock distribution network constrains the flow of data. Two regimes were shown to exist: positive clock skew, which occurs when the clock signal at the final register C_f of a data path leads the clock signal at the initial register C_i and negative clock skew, when C_i leads C_f . Design equations quantifying how positive clock skew can degrade the maximum clock frequency of a data path were described. Negative clock skew was shown to increase the maximum clock frequency as long as specific minimum timing constraints are satisfied.

In order to include the effects of clock skew on the design and analysis of high speed synchronous data paths, an algorithm was developed which bounds the minimum and maximum delay of each clock signal path, thereby bounding the clock skew of each local data path. Delay bounds of the distributed interconnect associated with each clock path were determined by following the pattern of the Penfield-Rubinstein algorithm for RC networks. In order to determine bounds on the delay of the cascaded buffers along each clock signal path (assumed for specificity to be a CMOS inverter), minimum and maximum values of output conductance were determined for each buffer stage. The maximum delay occurs when one of the transistors is in saturation and the other is in cut-off. The minimum delay occurs when both devices operate in the linear region.

These two conditions provide the smallest and largest conductance values, thereby bounding the buffer output conductance. Delay bounds of the individual distributed interconnect sections and cascaded buffers along a clock signal path are summed, providing a bounded range of clock skew for each local data path.

In order to examine how the memory element affects the performance of high speed data paths, the behavior of a bistable register was analyzed. It was shown that the transient response of a bistable NAND gate register can be separated into four different regions. This permits analytical expressions to be developed for each region in terms of the small signal parameters of the register. Once the register enters its regenerative latching mode, necessary and sufficient conditions for latching data into a register can be defined. From this result, the limiting condition for latching data into a bistable register can be analytically expressed.

A general approach for determining the total delay of a data path, including the effects of the interconnect, the logic stages, and the set-up and hold times of the final register was developed in terms of the data and clock signal waveforms. This integrated approach to designing and analyzing synchronous systems merges the latching conditions of the register and the waveform shape and skew characteristics of the clock signal with the data

signal propagating through the interconnect and logic stages. From this perspective, equations for designing and analyzing high speed synchronous data paths were developed which merge the waveform and technology characteristics of the key circuit components into an integrated synchronous system design approach.

The goal of all high performance synchronous digital systems is to move data as quickly as possible from the input of the system to its output. A design paradigm relating data throughput and clock frequency as a function of the level of pipelining was developed for studying the performance behavior of synchronous systems. This perspective permitted the development of design equations relating data throughput, latency, and clock frequency in terms of the logic and register delays, clock skew, the performance efficiency of pipelining, and the number of logic stages within a data path. An algorithm was described for partitioning a global data path into local pipelined data paths and is based on a figure of merit derived from the performance efficiency of pipelining and the clock frequency. This algorithm optimizes the effects of increased latency and increased clock frequency on data throughput and can be used to determine the optimal number of logic stages between pipeline registers as a function of the average stage delay of the global data path, the delays inherent to the register, and the clock skew.

These results have utility to systems level design by permitting high level performance and functional partitioning requirements to be defined in terms of low level circuit details.

A structured design procedure was described which summarizes and integrates the primary design equations developed throughout the dissertation. Specific emphasis was placed on integrating the results into a coherent design and analysis methodology. This design procedure was used to solve a variety of engineering problems for which significant performance improvements were achieved. These equations for the design and analysis of high performance data paths were demonstrated by their application to representative example circuits and exhibited reasonable accuracy while providing significant insight into how the performance of synchronous systems is affected by circuit and material characteristics. Design equations relating system level requirements such as clock frequency and latency for a specified data throughput and set of circuit characteristics were demonstrated and considered in terms of its effect on the design paradigm.

Further extensions to this research effort were also discussed which will permit performance gains to be applied to a greater variety of systems. Recommended future areas of investigation would extend these results

to a wider class of clocking strategies and provide more accurate models describing data throughput and device behavior.

These results provide a perspective for designing high speed synchronous digital systems which combines circuit level details such as waveform shapes and signal skew with systems level information such as latency and data throughput. This perspective was merged into an integrated synchronous system design approach which considers the interactions among the various component subsystems, thereby permitting the design and analysis of high performance synchronous digital systems.

The integrated synchronous system design approach was shown to be instrumental in maximizing the performance of synchronous digital systems. This approach can be used to extend the performance gains achieved by the enhancement of the individual components of a synchronous system and thereby permit the system to operate at its maximum performance capability. This result, coupled with the detailed analyses of clock distribution networks and register latching characteristics, permits the design of synchronous digital systems which can approach their fundamental performance limitations.

REFERENCES

- [1] R. H. Dennard, F. H. Gaensslen, H. N. 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. 256-268, October 1974.
- [2] K. C. Saraswat and F. Mohammadi, "Effect of Scaling of Interconnections on the Time Delay of VLSI Circuits," IEEE Transactions on Electron Devices, Vol. ED-29, No. 4, pp. 645-650, April 1982.
- [3] A. K. Sinha, J. A. Cooper Jr., and H. J. Levinstein, "Speed Limitations due to Interconnect Time Constants in VLSI Integrated Circuits," IEEE Electron Device Letters, Vol. EDL-3, No. 4, pp. 90-92, April 1982.
- [4] R. L. M. Dang and N. Shigyo, "Coupling Capacitances for Two-Dimensional Wires," IEEE Electron Device Letters, Vol. EDL-2, No. 8, pp. 196-197, August 1981.
- [5] E. T. Lewis, "An Analysis of Interconnect Line Capacitance and Coupling for VLSI Circuits" Solid-State Electronics, Vol. 27, Nos. 8/9, pp. 741-749, 1984.
- [6] P. E. Cottrell and E. M. Buturla, "VLSI Wiring Capacitance," IBM Journal of Research and Development, Vol. 29, No. 3, pp. 277-288, May 1985.
- [7] C. D. Taylor, G. N. Elkhouri, and T. E. Wade, "On the Parasitic Capacitances of Multilevel Parallel Metalization Lines", IEEE Transactions on Electron Devices, Vol. ED-32, No. 11, pp. 2408-2414, November 1985.
- [8] W. H. Chang, "Analytical IC Metal-Line Capacitance Formulas," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-24, No. 9, pp. 608-611, September 1976.
- [9] W. H. Chang, "Correction to 'Analytical IC Metal-Line Capacitance Formulas'," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-25, No. 9, pp. 712, August 1977.
- [10] M. I. Elmasry, "Capacitance Calculations in MOSFET VLSI," IEEE Electron Device Letters, Vol. EDL-3, No. 1, pp. 6-7, January 1982.

- [11] T. Sakurai and K. Tamaru, "Simple Formulas for Two- and Three-Dimensional Capacitances," IEEE Transactions on Electron Devices, Vol. ED-30, No. 2, pp. 183-185, February 1983.
- [12] A. E. Ruehli and P. A. Brennan, "Efficient Capacitance Calculations for Three-Dimensional Multiconductor Systems," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-21, No. 2, pp. 76-82, February 1973.
- [13] A. E. Ruehli and P. A. Brennan, "Accurate Metalization Capacitances for Integrated Circuits and Packages," IEEE Journal of Solid-State Circuits, Vol. SC-8, No. 8, pp. 289-290, August 1973.
- [14] A. E. Ruehli and P. A. Brennan, "Capacitance Models for Integrated Circuit Metalization Wires," IEEE Journal of Solid-State Circuits, Vol. SC-10, No. 6, pp. 530-536, December 1975.
- [15] P. E. Cottrell, E. M. Buturla, and D. R. Thomas, "Multi-Dimensional Simulation of VLSI Wiring Capacitance," Proceedings of International Electron Devices Meeting, pp. 548-551, December 1982.
- [16] T. Sakurai, "Approximation of Wiring Delay in MOSFET LSI," IEEE Journal of Solid-State Circuits, Vol. SC-18, No. 4, pp. 418-426, August 1983.
- [17] H. B. Bakoglu and J. D. Meindl, "Optimal Interconnect Circuits for VLSI," Proceedings of IEEE International Solid-State Circuits Conference, pp. 164-165, February 1984.
- [18] H. B. Bakoglu and J. D. Meindl, "Optimal Interconnection Circuits for VLSI," IEEE Transactions on Electron Devices, Vol. ED-32, No. 5, pp. 903-909, May 1985.
- [19] H-T. Yuan, Y-T. Lin, and S-Y. Chiang, "Properties of Interconnection on Silicon, Sapphire, and Semi-Insulating Gallium Arsenide Substrates," IEEE Transactions on Electron Devices, Vol. ED-29, No. 4, pp. 639-644, April 1982.
- [20] M. I. Elmasry, "Interconnection Delays in MOSFET VLSI," IEEE Journal of Solid-State Circuits, Vol. SC-16, No. 5, pp. 585-591, October 1981.
- [21] O. Akcasu, "Modeling of Delay and Crosstalk Interconnects," Proceedings of IEEE International Solid-State Circuits Conference, pp. 80-81, February 1988.

- [22] D. L. Carter and D. F. Guise, "Analysis of Signal Propagation Delays and Chip Level Performance Due to On-Chip Interconnections," Proceedings of International Conference on Computer Design, pp. 218-221, October 1983.
- [23] N. P. van der Meijs and J. T. Fokkema, "VLSI Circuit Reconstruction from Mask Topology," Integration, the VLSI Journal, Vol. 2, pp. 85-119, 1984.
- [24] E. Barke, "Line-to-Ground Capacitance Calculation for VLSI: A Comparison," IEEE Transactions on Computer-Aided Design, Vol. 7, No. 2, pp. 295-298, February 1988.
- [25] C. D. Taylor, G. N. Elkhouri, and T. E. Wade, "On the Parasitic Capacitances of Multilevel Skewed Metalization Lines," IEEE Transactions on Electron Devices, Vol. ED-33, No. 1, pp. 41-46, January 1986.
- [26] A. J. Walton, R. J. Holwill, and J. M. Robertson, "Numerical Simulation of Resistive Interconnects for Integrated Circuits," IEEE Journal of Solid-State Circuits, Vol. SC-20, No. 6, pp. 1252-1258, December 1985.
- [27] H. B. Bakoglu and J. D. Meindl, "New CMOS Driver and Receiver Circuits Reduce Interconnection Propagation Delays," Proceedings of IEEE Symposium on VLSI Technology, pp. 54-55, September 1985.
- [28] F. Anceau, "A Synchronous Approach for Clocking VLSI Systems," IEEE Journal of Solid-State Circuits, SC-17, No. 1, pp. 51-56, February 1982.
- [29] D. Wann and M. Franklin, "Asynchronous and Clocked Control Structures for VLSI Based Interconnection Networks," IEEE Transactions on Computers, Vol. C-32, No. 3, pp. 284-293, March 1983.
- [30] S. Dhar, M. Franklin, and D. Wann, "Reduction of Clock Delays in VLSI Structures," Proceedings of IEEE International Conference on Computer Design, pp. 778-783, October 1984.
- [31] K. Wagner and E. McCluskey, "Tuning, Clock Distribution, and Communication in VLSI High-Speed Chips," Stanford University, Stanford, California, CRC Technical Report 84-5, June 1984.
- [32] N. Park and A. Parker, "Synthesis of Optimal Clocking Schemes," Proceedings of ACM IEEE 22nd Design Automation Conference, pp. 489-495, June 1985.

- [33] S. H. Unger and C-J. Tan, "Clocking Schemes for High-Speed Digital Systems," IEEE Transactions on Computers, Vol. C-35, No. 10, pp. 880-895, October 1986.
- [34] E. G. Friedman and S. Powell, "Design and Analysis of 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.
- [35] H. B. Bakoglu, 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 IEEE International Conference on Computer Design, pp. 118-122, October 1986.
- [36] K. D. Wagner, "A Survey of Clock Distribution Techniques in High-Speed Computer Systems," Stanford University, Stanford, California, CRC Technical Report 86-20, December 1986.
- [37] C. V. Gura, "Analysis of Clock Skew in Distributed Resistive-Capacitive Interconnects," University of Illinois, Urbana, Illinois, SRC Technical Report No. T87053, June 1987.
- [38] D. Noise, R. Mathews, and J. Newkirk, "A Clocking Discipline for Two-Phase Digital Systems," Proceedings of International Conference on Circuits and Computers, pp. 108-111, September 1982.
- [39] M. S. McGregor, P. B. Denyer, and A. F. Murray, "A Single-Phase Clocking Scheme for CMOS VLSI," Proceedings of 1987 Stanford Conference on Advanced Research in VLSI, pp. 257-271, March 1987.
- [40] M. Hatamian and G. L. Cash, "Parallel Bit-Level Pipelined VLSI Designs for High-Speed Signal Processing," Proceedings of the IEEE, Vol. 75, No. 9, pp. 1192-1202, September 1987.
- [41] Randell and Treleaven, VLSI Architecture, Prentice-Hall, 1983, "Clocking of VLSI Circuits," by J. Alves Marques and A. Cunha, pp. 165-178.
- [42] D. Mijuskovic, "Clock Distribution in Application Specific Integrated Circuits," Microelectronics Journal, Vol. 18, pp. 15-27, July/August 1987.

- [43] M. Hatamian, L. A. Hornak, T. E. Little, S. T. Tewksbury, and P. Franzon, "Fundamental Interconnection Issues," AT&T Technical Journal, Volume 66, Issue 4, pp. 13-30, July/August 1987.
- [44] K. D. Wagner, "Clock System Design," IEEE Design & Test of Computers, pp. 9-27, October 1987.
- [45] S. D. Kugelmass and K. Steiglitz, "A Probabilistic Model for Clock Skew," Proceedings of IEEE International Conference on Systolic Arrays, pp. 545-554, 1988.
- [46] M. Afghahi and C. Svensson, "A Scalable Synchronous System," Proceedings of International Symposium on Circuits and Systems, pp. 471-474, May 1988.
- [47] M. R. Dagenais and N. C. Rumin, "Automatic Determination of Optimal Clocking Parameters in Synchronous MOS VLSI Circuits," Proceedings of 1988 Stanford Conference on Advanced Research in VLSI, pp. 19-33, March 1988.
- [48] T. Williams and K. Parker, "Design for Testability-A Survey," Proceedings of the IEEE, Vol. 71, No. 1, pp. 98-112, January 1983.
- [49] P. Penfield, Jr. and J. Rubinstein, "Signal Delay in RC Tree Networks," Proceedings of the Caltech Conference on VLSI, pp. 269-283, January 1981.
- [50] J. Rubinstein, P. Penfield, Jr., and M. A. Horowitz, "Signal Delay in RC Tree Networks," IEEE Transactions on Computer-Aided Design, Vol. CAD-2, No. 3, pp. 202-211, July 1983.
- [51] C. Lee and H. Soukup, "An Algorithm for CMOS Timing and Area Optimization," IEEE Journal of Solid-State Circuits, Vol. SC-19, No. 5, pp. 781-787, October 1984.
- [52] D. A. Hodges and H. G. Jackson, Analysis and Design of Digital Integrated Circuits, New York, New York: McGraw-Hill, 1983.
- [53] L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," ERL Memo ERL-M520, University of California, Berkeley, May 1975.
- [54] H. Shichman and D. A. Hodges, "Modeling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits," IEEE Journal of Solid-State Circuits, Vol. SC-3, pp. 285-289, September 1968.

- [55] J. K. Ousterhout, "A Switch-Level Timing Verifier for Digital MOS VLSI," IEEE Transactions on Computer-Aided Design, Vol. CAD-4, No. 3, pp. 336-349, July 1985.
- [56] Tamura, K. Ogawa, and T. Nakano, "Path Delay Analysis for Hierarchical Building Block Layout System," Proceedings of ACM IEEE 20th Design Automation Conference, pp. 403-410, June 1983.
- [57] A. J. de Geus, J. B. Reed, M. Rekhson, and G. Wikle, "IDA: Interconnect Delay Analysis for Integrated Circuits," Proceedings of ACM IEEE 21st Design Automation Conference, pp. 536-541, June 1984.
- [58] N. P. Jouppi, "Timing Analysis for nMOS VLSI," Proceedings of ACM IEEE 20th Design Automation Conference, pp. 411-418, June 1983.
- [59] R. Putatunda, "AUTODELAY: A Second-Generation Automatic Delay Calculation Program for LSI/VLSI Chips," Proceedings of IEEE International Conference on Computer-Aided Design, pp. 188-190, November 1984.
- [60] J. Beausang and A. Albicki, "A Method to Obtain an Optimal Clock Scheme for a Digital System," Proceedings of IEEE International Conference on Computer Design, pp. 68-72, October 1985.
- [61] S-H. Lu, "A Safe Single-Phase Clocking Scheme for CMOS Circuits," IEEE Journal of Solid-State Circuits, Vol. SC-23, No. 1, pp. 280-283, February 1988.
- [62] L. W. Cotton, "Circuit Implementation of High-Speed Pipeline Systems," Proceedings of the Fall Joint Computer Conference, pp. 489-504, 1965.
- [63] P. R. Cappello and K. Steiglitz, "Bit-Level Fixed-Flow Architectures for Signal Processing," Proceedings of the IEEE International Conference on Circuits and Computers, pp. 570-573, September 1982.
- [64] P. R. Cappello and K. Steiglitz, "Completely-Pipelined Architectures for Digital Signal Processing," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-31, No. 4, pp. 1016-1023, August 1983.
- [65] C. E. Leiserson and J. B. Saxe, "Optimizing Synchronous Systems," Proceedings of 22nd Annual Symposium on Foundations of Computer Science, pp. 23-26, October 1981.

[66] P. R. Cappello, A. LePaugh, and K. Steiglitz, "Optimal Choice of Intermediate Latching to Maximize Throughput in VLSI Circuits," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 935-938, April 1983.

[67] P. R. Cappello, A. LaPaugh, and K. Steiglitz, "Optimal Choice of Intermediate Latching to Maximize Throughput in VLSI Circuits," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-32, No. 1, pp. 28-33, February 1984.

[68] J. R. Jump and S. R. Ahuja, "Effective Pipelining of Digital Systems," IEEE Transactions on Computers, Vol. C-27, No. 9, pp. 855-865, September 1978.

[69] K. O. Siomalas and B. A. Bowen, "Synthesis of Efficient Pipelined Architectures for Implementing DSP Operations," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, No. 6, pp. 1499-1508, December 1985.

[70] M. Hatamian and G. Cash, "A 70-Mhz 8-bit x 8-bit Parallel Pipelined Multiplier in 2.5um CMOS," IEEE Journal of Solid-State Circuits, Vol. SC-21, No. 4, pp. 505-513, August 1986.

[71] J. Yuan and C. Svensson, "High-Speed CMOS Circuit Technique," IEEE Journal of Solid-State Circuits, Vol. SC-24, No. 1, pp. 62-70, February 1989.

[72] N. Goncalves and H. J. DeMan, "NORA: A Racefree Dynamic CMOS Technique for Pipelined Logic Structures," IEEE Journal of Solid-State Circuits, Vol. SC-18, No. 3, pp. 261-266, June 1983.

[73] Y. Ji-Ren, I. Karlsson, and C. Svensson, "A True Single-Phase-Clock Dynamic CMOS Circuit Technique," IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 5, pp. 899-901, October 1987.

[74] C. Seitz, "Self-Timed VLSI Systems," Proceedings of Caltech Conference on VLSI, pp. 345-355, January 1979.

[75] M. J. Stucki and J. R. Cox, Jr., "Synchronization Strategies," Proceedings of Caltech Conference on VLSI, pp. 375-393, January 1979.

[76] T-A. Chu, "On the Models for Designing VLSI Asynchronous Digital Systems," Integration, the VLSI Journal, Vol. 4, pp. 99-113, June 1986.

- [77] D. C. Kirkpatrick and V. M. Powers, "An Asynchronous Design Style to Achieve Ultimate Operating Speed," Proceedings of 5th Phoenix Conference on Computers and Communications, pp. 662-673, March 1986.
- [78] C. E. Leiserson, F. M. Rose, and J. B. Saxe, "Optimizing Synchronous Circuitry by Retiming," Proceedings of 3rd Caltech Conference on VLSI, pp. 87-116, March 1983.
- [79] A. L. Fisher and H. T. Kung, "Synchronizing Large VLSI Processor Arrays," IEEE Transactions on Computers, Vol. C-34, No. 8, pp. 734-740, August 1985.
- [80] M. Hatamian and G. Cash, "High Speed Signal Processing, Pipelining, and VLSI," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1173-1176, April 1986.
- [81] S. Y. Kung and R. J. Gal-Ezer, "Synchronous Versus Asynchronous Computation in Very Large Scale Integrated (VLSI) Array Processors," Proceedings of SPIE, Vol. 341, pp. 53-65, 1982.
- [82] L. Marino, "General Theory of Metastable Operation," IEEE Transactions on Computers, Vol. C-30, No. 2, pp. 107-115, February 1981.
- [83] M. Pechoucek, "Anomalous Response Times of Input Synchronizers," IEEE Transactions on Computers, Vol. C-25, No. 2, pp. 133-139, February 1976.
- [84] B. Liu and N. Gallagher, "On the Metastable Region of Flip-Flop Circuits," Proceedings of the IEEE, Vol. 65, pp. 581-583, April 1977.
- [85] H. J. M. Veendrick, "The Behavior of Flip-Flops Used as Synchronizers and Prediction of their Failure Rate," IEEE Journal of Solid-State Circuits, Vol. SC-15, No. 2, pp. 169-176, April 1980.
- [86] G. Elineau and W. Weisbeck, "A New J-K Flip-Flop for Synchronizers," IEEE Transactions on Computers, Vol. C-26, No. 12, pp. 1277-1279, December 1977.
- [87] T. Kacprzak, A. Albicki, and T. Jackson, "Design of N-Well CMOS Flip-Flops with Minimum Failure Rate Due to Metastability," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 765-767, May 1986.

- [88] T. J. Chaney and E. E. Molnar, "Anomalous Behavior of Synchronizer and Arbiter Circuits," IEEE Transactions on Computers, Vol. C-22, No. 4, pp. 421-422, April 1973.
- [89] G. R. Couranz and D. F. Wann, "Theoretical and Experimental Behavior of Synchronizers in the Metastable Region," IEEE Transactions on Computers, Vol. C-24, pp. 604-616, June 1975.
- [90] T. J. Chaney, "Comments on 'A Note on Synchronizer or Interlock Maloperation'," IEEE Transactions on Computers, Vol. C-28, No. 10, pp. 802-805, October 1979.
- [91] R. Rosenberger and T. J. Chaney, "Flip-Flop Resolving Time Test Circuit," IEEE Journal of Solid-State Circuits, Vol. SC-17, No. 4, pp. 731-738, August 1982.
- [92] S. Flannagan, "Synchronization Reliability in CMOS Technology," IEEE Journal of Solid-State Circuits, Vol. SC-20, No. 4, pp. 880-882, August 1985.
- [93] T. Kacprzak and A. Albicki, "Analysis of Metastable Operation in RS CMOS Flip-Flops," IEEE Journal of Solid-State Circuits, Vol. SC-22, No. 1, pp. 57-64, February 1987.
- [94] L. Kleeman and A. Cantoni, "On the Unavoidability of Metastable Behavior in Digital Systems," IEEE Transactions on Computers, Vol. C-36, No. 1, pp. 109-112, January 1987.
- [95] L. Kleeman and A. Cantoni, "Metastable Behavior in Digital Systems," IEEE Design & Test of Computers, pp. 4-19, December 1987.
- [96] W. K. Stewart and S. A. Ward, "A Solution to a Special Case of the Synchronization Problem," IEEE Transactions on Computers, Vol. C-37, No. 1, pp. 123-125, January 1988.
- [97] T. Kacprzak, "Analysis of Oscillatory Metastable Operation of an RS Flip-Flop," IEEE Journal of Solid-State Circuits, Vol. SC-23, No. 1, pp. 260-266, February 1988.
- [98] W. W. Plummer, "Asynchronous Arbiters," IEEE Transactions on Computers, Vol. C-21, No. 1, pp. 37-42, January 1972.

- [99] R. C. Pearce, J. A. Field, and W. D. Little, "Asynchronous Arbiter Module," IEEE Transactions on Computers, Vol. C-24, pp. 931-932, September 1975.
- [100] D. J. Kinniment and J. V. Woods, "Synchronization and Arbitration Circuits in Digital Systems," Proceedings of the IEE, Vol. 123, pp. 961-966, 1976.
- [101] C. L. Seitz, "Ideas About Arbiters," Lambda, (now VLSI Systems Design), pp. 10-14, First Quarter 1980.
- [102] J. H. Hohl, W. R. Larsen, and L. C. Schooley, "Prediction of Error Probabilities for Integrated Digital Synchronizers," IEEE Journal of Solid-State Circuits, Vol. SC-19, No. 2, pp. 236-244, April 1984.
- [103] K. N. Oikonomon, "Ideal Arbiters: Analysis and Design," AT&T Technical Journal, Volume 66, Issue 2, pp. 78-96, March/April 1987.
- [104] T. Sakurai, "Optimization of CMOS Arbiter and Synchronization Circuits with Submicrometer MOSFET's," IEEE Journal of Solid-State Circuits, Vol. SC-23, No. 4, pp. 901-906, August 1988.
- [105] L. A. Glasser and D. W. Dobberpuhl, The Design and Analysis of VLSI Circuits, Reading, Massachusetts: Addison-Wesley, 1985.
- [106] Y. Suzuki, K. Odagawa, and T. Abe, "Clocked CMOS Calculator Circuitry," IEEE Journal of Solid-State Circuits, Vol. SC-8, No. 6, pp. 462-469, December 1973.
- [107] Y. P. Tsividis, Operation and Modeling of the MOS Transistor, New York, New York: McGraw-Hill, 1987.
- [108] J. L. Wyatt, Jr., "Waveform Bounding for Fast Timing Analysis of MOS VLSI Circuits," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 760-761, May 1983.
- [109] M. Horowitz, "Timing Models for Pass Transistors," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 198-201, May 1983.
- [110] T-M. Lin and C. A. Mead, "Signal Delay in General RC Networks," IEEE Transactions on Computer-Aided Design, Vol. CAD-3, No. 4, pp. 331-348, October 1984.

- [111] J. L. Wyatt, Jr. and Q. Yu, "Signal Delay in RC Meshes, Trees and Lines," Proceedings of IEEE International Conference on Computer-Aided Design, pp. 15-17, November 1984.
- [112] Q. Yu, J. L. Wyatt, Jr., C. Zukowski, H-N. Tan, and P. O'Brien, "Improved Bounds on Signal Delay in Linear RC Models for MOS Interconnect," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 903-906, June 1985.
- [113] J. L. Wyatt, Jr., "Signal Delay in RC Mesh Networks," IEEE Transactions on Circuits and Systems, Vol. CAS-32, No. 5, pp. 507-510, May 1985.
- [114] C. A. Zukowski, "Relaxing Bounds for Linear RC Mesh Circuits," IEEE Transactions on Computer-Aided Design, Vol. CAD-5, No. 2, pp. 305-312, April 1986.
- [115] J. L. Wyatt, Jr., "Signal Propagation Delay in RC Models for Interconnect," Circuit Analysis, Simulation and Design, Part II: VLSI Circuit Analysis and Simulation, A. Ruehli, ed., Vol. 3 in the series Advances in CAD for VLSI, North-Holland, 1987.
- [116] P. R. O'Brien and J. L. Wyatt, Jr., "Signal Delay in ECL Interconnect," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 755-758, May 1986.
- [117] M. A. Cirit, "RC Trees Revisited," Proceedings of Custom Integrated Circuits Conference, pp. 6.7.1-6.7.4, May 1988.
- [118] C. A. Zukowski, The Bounding Approach to VLSI Circuit Simulation, Boston, Massachusetts: Kluwer Academic Publishers, 1986.
- [119] S. G. Smith, M. S. McGregor, and P. B. Denyer, "Techniques to Increase the Computational Throughput of Bit-Serial Architectures," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 543-546, April 1987.
- [120] N. R. Strader II, "VLSI Bit-Sequential Architectures for Digital Signal Processing," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 931-934, April 1983.

[121] E. H. Wold and A. M. Despain, "Pipeline and Parallel-Pipeline FFT Processors for VLSI Implementation," IEEE Transactions on Computers, Vol. C-33, No. 5, pp. 414-426, May 1984.

[122] A. C. Parker, "Automated Synthesis of Digital Systems," IEEE Design & Test of Computers, pp. 75-81, November 1984.

[123] G. De Micheli, A. Sangiovanni-Vincentelli, and P. Antognetti, Design Systems for VLSI Circuits Logic Synthesis and Silicon Compilation, Dordrecht, The Netherlands: Martinus Nijhoff, 1987.

[124] D. D. Gajski, Silicon Compilation, Reading, Massachusetts: Addison-Wesley, 1988.

APPENDIX A

BASIC PROGRAM FOR GENERATING REGISTER OUTPUT WAVEFORM

```

10 REM Linear Model using Laplace Transforms -- V1(t)
20 REM Two Parallel Bistable NAND Gates with a Ramp Input Signal
30 REM Small Signal Analysis-Changing Clock and Data Input Signals
40 REM                                VIOUTCD.BAS          5/16/88
50 READ VTP,VTN,VDD,E,C1,C2,K1,K2,TAU
60 DATA -1,1,5,5,.15e-12,.15e-12,1e+9,1e+9,1e-9
70 REM Region 1: Input clock signal above Vdd + Vtp and no current can flow
80 TD1 = ABS(VTP)/K1
90 REM
100 REM Region 2: Open Loop System (One Time Constant System)
110 REM      Operating Point at 2.0 ns (Velk = 3.00 volts)
120 READ GMN,GN1,GN2,GMP
130 DATA 2.762e-5,3.920e-4,8.510e-4,2.158e-4
140 DELTA2 = 5E-11
150 GDS = GN1*GN2/(GN1 + GN2)
160 GMPRIME = GMN/(1 + GN1/GN2)
170 A = GMPRIME + GMP - GMPRIME*GMN/(GN1 + GN2 + GMN)
180 B = GDS - GMPRIME*GN1/(GN1 + GN2 + GMN)
190 FOR T = 0 TO 2.5E-08 STEP DELTA2
200 VDATA = K2*(T+TD1-TAU)
210 IF VDATA < 0 THEN VDATA = 0
220 E2 = VDD + VTP
230 VOUT1 = (K1*A*C1/B^2)*(EXP((-B/C1)*T) + (B/C1)*T - 1)
240 PRINT "Vout1("T+TD1") = "VOUT1,VDATA
250 TD2 = T
260 VOUT23 = VOUT1
270 IF VDATA < VTN GOTO 290
280 IF VOUT1 >= VTN GOTO 320
290 IF VOUT1 >= VDD + VTP GOTO 800
300 NEXT T
310 REM
320 REM Region 3: Closed Loop System (Two Time Constant System)
330 REM      Operating Point at 3.35 ns (Velk = 1.65 volts, Vout = 2.5 volts)
340 READ GMP1A,GMP2A,GMN1A,GMN2A,GN2A
350 DATA 3.237e-6,1.720e-4,2.158e-7,1.942e-6,6.422e-4
360 READ GMN1B,GMN2B,GN2B,GMP1B,GP1B,GMP2B,GP2B
370 DATA 1.444e-4,2.851e-4,4.061e-4,2.158e-4,0,2.190e-4,1.079e-6
380 A1 = (GMN1A*GMN2A/(GN2A + GMN1A)) - GMP1A
390 A2 = (GMN1A*GN2A/(GN2A + GMN1A)) - GMP2A
400 GA = 0
410 B1 = (GMN2B/(1 + (GN2B/GMN1B))) - GMP1B
420 B2 = GMN1B*GN2B/(GMN1B+GN2B) - GMP2B
430 GB = GP1B + GP2B
440 Q = SQR((GA/C1)^2 + (GB/C2)^2 - 2*GA*GB/(C1*C2) + 4*A1*B1/(C1*C2))
450 ALPHA1 = (-(GA/C1 + GB/C2) + Q)/2
460 ALPHA2 = (-(GA/C1 + GB/C2) - Q)/2
470 D = GB/C2
480 DELTA3 = 5E-11
490 FOR T = 0 TO 2.5E-08 STEP DELTA3
500 VDATA = K2*(T + TD1 + TD2 - TAU)
510 E3 = E2 - TD2*K1
520 UT = 1
530 EPSILON3 = DELTA3/4
540 IF T < E3/K1-EPSILON3 THEN UTK=0 ELSE UTK=1
550 R1A = (ALPHA1-D)*EXP(-ALPHA1*T)/(ALPHA1*(ALPHA2-ALPHA1))
OK

```

```

560 R2A = (ALPHA2-D)*EXP(-ALPHA2*T)/(ALPHA2*(ALPHA1-ALPHA2))
570 R3A = D/(ALPHA1*ALPHA2)
580 VOUT1A = VOUT23*1E+09*(R1A + R2A + R3A)
590 R1C = ((-ALPHA1+D)*EXP(-ALPHA1*T))/(ALPHA1^2*(-ALPHA1+ALPHA2))
600 R2C = ((-ALPHA2+D)*EXP(-ALPHA2*T))/(ALPHA2^2*(-ALPHA2+ALPHA1))
610 R3C = (ALPHA1*ALPHA2*(1+D*T)-D*(ALPHA1+ALPHA2))/(ALPHA1^2*ALPHA2^2)
620 REM R1DC = ((-ALPHA1+D)*EXP(-ALPHA1*(T-E3/K1)))/(ALPHA1^2*(-ALPHA1+ALPHA2))
630 REM R2DC = ((-ALPHA2+D)*EXP(-ALPHA2*(T-E3/K1)))/(ALPHA2^2*(-ALPHA2+ALPHA1))
640 REM R3DC = (ALPHA1*ALPHA2*(1+D*(T-E3/K1))-D*(ALPHA1+ALPHA2))/(ALPHA1^2*ALPHA2^2)
650 VOUT1C = -(A2/C1)*K1*((R1C+R2C+R3C)*UT - (R1DC+R2DC+R3DC)*UTK)
660 R1D = (EXP(-ALPHA1*T))/(ALPHA1^2*(-ALPHA1+ALPHA2))
670 R2D = (EXP(-ALPHA2*T))/(ALPHA2^2*(-ALPHA2+ALPHA1))
680 R3D = (ALPHA1*ALPHA2*(T-(ALPHA1+ALPHA2)))/(ALPHA1^2*ALPHA2^2)
690 REM R1DD = (EXP(-ALPHA1*(T-VDATA/K2)))/(ALPHA1^2*(-ALPHA1+ALPHA2))
700 REM R2DD = (EXP(-ALPHA2*(T-VDATA/K2)))/(ALPHA2^2*(-ALPHA2+ALPHA1))
710 REM R3DD = ((ALPHA1*ALPHA2*(T-VDATA/K2))-D*(ALPHA1+ALPHA2))/(ALPHA1^2*ALPHA2^2)
720 VOUT1D = -(A1*B2/(C1*C2)*K2*((R1D+R2D+R3D)*UT - (R1DD+R2DD+R3DD)*UTK))
730 VOUT1 = VOUT1A + VOUT1C + VOUT1D
740 PRINT "Vout1("T+TD2+TD1") = "VOUT1 + VOUT23,VOUT1A,VOUT1C,VOUT1D,VDATA
750 REM PRINT R1,R2,R3
760 TD3 = T
770 VOUT34 = VOUT1 + VOUT23
780 IF VOUT1 + VOUT23 >= 3.5 GOTO 810
790 NEXT T
800 VOUT34 = VOUT23
810 REM
820 REM Region 4: Closed Loop System (Two Time Constant System)
830 REM      Operating Point at 4.00 ns (Vclk = 1.00 volts, Vout = 4.28 volts)
840 READ G14A,G24A
850 DATA 5.158e-5,4.920e-4
860 GA4 = G14A + G24A
870 DELTA4 = 1E-10
880 FOR T = 0 TO 2.5E-08 STEP DELTA4
890 VOUT1 = VDD - (VDD-VOUT34)*EXP(-T*(GA4/C1))
900 PRINT "Vout1("T+TD3+TD2+TD1") = "VOUT1
910 IF VDD - VOUT1 < .001 GOTO 930
920 NEXT T
930 PRINT
940 REM Region 2 Parameter Values
950 PRINT "GMPRIME is "GMPRIME", Gds is "GDS", A is "A", and B is "B
960 PRINT
970 REM Region 3 Parameter Values
980 PRINT "A1 is "A1", A2 is "A2", and GA is "GA
990 PRINT "B1 is "B1", B2 is "B2", and GB is "GB
1000 PRINT "Q is "Q", ALPHA1 is "ALPHA1", ALPHA2 is "ALPHA2", and D is "D
1010 PRINT
1020 REM Region 4 Parameter Values
1030 PRINT "GA4 is "GA4
1040 PRINT
1050 PRINT E2,E3,E4,TD1,TD2,TD3
1060 END
Ok

```