Design and Optimization of Fast Adder Circuits

Using Mixed CMOS Logic Styles

By
Yuanzhong Wan

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Applied Science

Department of Electronics
Carleton University
Ottawa, Ontario, Canada

September 2004

© 2004 Yuanzhong Wan

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-97445-6

Our file Notre référence

ISBN: 0-612-97445-6

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

The information used in this thesis comes in part from the research program of Dr. Maitham Shams and his associates in the VLSI group. The research results appearing in this thesis represent an integral part of the ongoing research program. All research results in this thesis including tables, graphs and figures but excluding the narrative portions of the thesis are effectively incorporated into the research program and can be used by Dr. Maitham Shams and his associates for educational and research purposes, including publication in open literature with appropriate credits. The matters of intellectual property may be pursued co-operatively with Carleton University and Dr. Maitham Shams, where and when appropriate.

PAGINATION ERROR.

ERREUR DE PAGINATION.

TEXT COMPLETE.

LE TEXTE EST COMPLET.

Abstract

The development of modern, functionally-dense integrated circuits would be quite impossible if engineers were to use the traditional soldering-iron/breadboard techniques. This time consumption issue is the first to be addressed in the thesis, where the description of a closed-form means, replacing the usual SPICE calculations, of predicting gate delay is explored. We go on to describe how this delay-prediction method may be applied to a practical and important problem, namely mixed-logic fast adder design.

Two things have been demonstrated - first: brute-force simulation is not always necessary when designing ICs. Secondly, the use of mixed-logic (in particular, the mixture of C-CMOS and CPL) can produce devices which are substantially faster than those implemented in 'pure' (C-CMOS for example) logic styles, while exhibiting similar power-dissipation statistics and somewhat better power-delay product (PDP) figures. The closed form optimization method is inherently adaptable to mixed logic designs.

The use of the mixed logic design method for making design decisions was in the development of fast serial adders. An eight 8-bit Ripple Carry Adders was fabricated to verify the potential of mixed logic design. The measured results exhibit that the combinations of C-CMOS and CPL in 8-bit RCAs exhibit better performance both in delay, power dissipation, area, and PDP. The success of the closed-form delay formulas was evident in the fact that the 64-bit optimized static adder was faster than other statics reported to date and faster than even some of the dynamics. This experiment also provided evidence that mixed logic design aided by closed form delay optimization and prediction may play a more and more important role in the high performance design.

Acknowledgements

First and foremost, I would like to thank my thesis supervisor Professor Maitham Shams who gave me this great opportunity, provided the ideas, guidance and support to complete this thesis.

Secondly, I would like to thank faculties, staffs of Department of Electronics at Carleton University, especially graduate study chair Dr. Langis Roy, graduate assistant Peggy Piccolo, computer systems manager Jacques Lemieux, and hardware manager Nagui Mihhail for their supports and helps during this thesis work. I am also grateful to Canadian Microelectronics Cooperation (CMC) for their CAD tool support and chip fabrication.

Finally, I would to express my sincere gratitude to my wife, Xiaopeng Jiang, for her support and encouragement, to my son, Jiaye, for all the happiness he has brought to our lives, and to all my family members for their unconditional loves.

Table of Contents

Chapter 1: Introduction	1
1.1 Thesis Motivation	1
1.2 Analytical Delay Models	1
1.3 Thesis Objectives	4
1.4 Thesis Outline	5
Chapter 2: CMOS Logic Styles	6
2.1 Introduction.....	6
2.2 Static CMOS Logic Styles	7
2.2.1 Conventional CMOS Logic	7
2.2.2 Pass-transistor Logic	8
2.2.3 Differential Cascaded Voltage Switch Logic	9
2.3 Dynamic Logic Styles	10
2.3.1 Domino CMOS	10
2.3.2 NORA CMOS Logic	11
2.3.3 True Single-Phase Clock (TSPC) Dynamic Circuit.....	11
2.4 Conclusion	12
Chapter 3: MOSFET Delay Modeling.....	13
3.1 Introduction.....	13
3.2 Short Channel Effects	13
3.3 Delay Definitions	14
3.3.1 Step Delay	14
3.3.2 Propagation Delay	15

3.4	Current Model.....	17
3.5	Overlapping and Opposing Currents	17
3.6	Capacitance Model	19
3.6.1	Gate Capacitance	19
3.6.2	Diffusion Capacitance.....	21
3.7	Modeling of Series Connected Transistors	22
3.8	Effect of Input Slope (dv/dt) on Delay	23
3.9	Conclusion	25
Chapter 4: Delay Parameters Extraction		26
4.1	Introduction.....	26
4.2	Extraction of Current Parameters	26
4.3	Extraction of Input Slope Factors	28
4.4	Extraction of Series-Connection Degradation Factors	29
4.5	Extraction of MOSFET Capacitances per Unit Width.....	31
4.6	Conclusion	34
Chapter 5: Modeling and Optimization of CMOS Gate Delay		35
5.1	Introduction.....	35
5.2	C-CMOS Gate Delay Modeling and Optimization.....	35
5.3	DCVSL Gate-Delay Modeling and Optimization	40
5.4	CPL Gate-Delay Modeling and Optimization	44
5.4.1	Currents in CPL Gates	44
5.4.2	Critical Paths: G-Drive Alone, or Both G- and S-Drives	45
5.4.3	Critical Paths: S-Drive Alone	47
5.5	Conclusion	50

Chapter 6: Delay Optimization of CMOS Circuit in Mixed Logic-Style	51
6.1 Introduction.....	51
6.2 Delay Estimation and Optimization in C-CMOS Blocks	51
6.3 Delay Optimization in Mixed Logic Styles	54
6.3.1 Mixed Logic Delay Optimization	54
6.3.2 Specific Devices: 3-bit Full Adders.....	56
6.4 Delay Estimation and Optimization for CPL Chains.....	61
6.4.1 Delay Modeling of CPL Chains.....	61
6.4.2 Delay Optimization of CPL Chains	66
6.5 Conclusion	68
Chapter 7: Applications to Fast Adder Circuits Design	69
7.1 Introduction.....	69
7.2 Fast Addition.....	69
7.2.1 The Ripple-Carry Adder	70
7.2.2 The Carry-Bypass Adder	70
7.2.3 The Carry-Select Adder	71
7.2.4 The Carry-Lookahead Adder	72
7.3 A Ripple-Carry Adder Test Chip Design.....	72
7.3.1 A RCA Test Chip Design.....	74
7.3.2 Delay Optimization.....	76
7.3.3 Results and Comparison	77
7.4 A 64-bit High-Speed Static Adder Test Chip Design	82
7.4.1 A 64-bit Adder Test Chip Design	82
7.4.2 Critical Path Optimization	87
7.4.3 Results and Comparison	87

7.5	Conclusion	88
Chapter 8: Conclusion.....		90
8.1	Introduction.....	90
8.2	Background Preparation	91
8.2.1	Analytical Delay Model.....	91
8.2.2	Parameter Extraction.....	91
8.2.3	Delay Optimization.....	91
8.3	Thesis Contributions	92
8.4	Applications	92
8.5	Future Work	94
Chapter 9: References		95

List of Tables

TABLE 3.1:	MOSFET Gate Capacitance.....	20
TABLE 4.1:	Current-Modeling Parameters (0.18 μ m VDD=1.8V)	28
TABLE 4.2:	Input Slope Factors (0.18 μ m VDD=1.8V).....	29
TABLE 4.3:	Series-Connection Degradation Parameters (0.18 μ m VDD=1.8V)	31
TABLE 4.4:	Series-Connection Degradation Factors (0.18 μ m VDD=1.8V)	31
TABLE 4.5:	MOSFET Unit Capacitances (0.18 μ m).....	32
TABLE 5.1:	Performance Comparison of CMOS XORs.....	39
TABLE 5.2:	Performance Comparison of DCVSL XOR Gates	43
TABLE 5.3:	Performance Comparison of CPL XORs with G-Drive	47
TABLE 5.4:	Performances Comparison of CPL XORs with S-Drive	50
TABLE 6.1:	C-CMOS Block Delay Modeling (0.13 μ m).....	53
TABLE 6.2:	Optimal Sizes in C-CMOS Blocks (0.18 μ m).....	53
TABLE 6.3:	Performance Comparison of C-CMOS Blocks (0.18 μ m)	54
TABLE 6.4:	Optimal Transistor Sizing Formulas	55
TABLE 6.5:	Optimal Transistor Sizes of 3-bit FAs.....	58
TABLE 6.6:	Optimal Transistor Sizes.....	66
TABLE 7.1:	Summary of eight 8-bit RCAs	77
TABLE 7.2:	Summary of Measurement for 8-bit RCAs.....	78
TABLE 7.3:	Performance Comparison	89

List of Figures

Figure 2.1:	Classification of Digital CMOS Logic Styles.....	6
Figure 2.2:	Schematic of a C-CMOS NAND Gate	7
Figure 2.3:	Schematic of a CPL NAND/AND Gate	8
Figure 2.4:	Schematic of a DCVSL NAND/AND Gate.....	9
Figure 2.5:	Cascaded Domino CMOS Logic Gates [3].....	10
Figure 2.6:	NORA CMOS Logic Circuits [3]	11
Figure 2.7:	A Pipelined TSPC Logic Circuit [3].....	12
Figure 3.1:	Velocity Saturation Effect and Mobility Degradation	14
Figure 3.2:	Definition of Step Delay	15
Figure 3.3:	Step Delays Vs Models (0.13 μ m Technology).....	16
Figure 3.4:	Definition of Propagation Delay.....	16
Figure 3.5:	Comparison of Current Models	18
Figure 3.6:	Step Delays of Overlapping Currents (0.13 μ m Technology).....	19
Figure 3.7:	Step Delays of Opposing Currents (0.13 μ m Technology)	19
Figure 3.8:	Step Delays of Series Connected Transistors (0.13 μ m Technology)	23
Figure 3.9:	Input Slope Effect (0.13m Technology).....	24
Figure 3.10:	Propagate Delays involving Overlapping and Opposing Currents (0.13 μ m Technology)	25
Figure 4.1:	A Test Bench for Delay Parameter Extraction.....	27
Figure 4.2:	A Test Bench for Series-Connected Transistors Parameter Extraction	30
Figure 4.3:	A CMOS Inverter Chain [1]	31
Figure 5.1:	A Conventional CMOS Cell [1]	36
Figure 5.2:	A C-CMOS XOR Design Under Test (DUT).....	37

Figure 5.3:	Model/Simulation predictions for a C-CMOS XOR Gate.....	38
Figure 5.4:	The Iteration of the Optimal Sizing for a C-CMOS XOR Gate	39
Figure 5.5:	A DCVSL XOR/XNOR Gate Under Test	40
Figure 5.6:	DCVSL Delays and Associated Currents	41
Figure 5.7:	Model/Simulation predictions for a DCVSL XOR/XNOR Gate.....	42
Figure 5.8:	The Iteration of Optimal Sizing for a DCVSL XOR/XNOR Gate	43
Figure 5.9:	A CPL Cell with a Drive and Load Gate [1]	44
Figure 5.10:	The Currents in a CPL Gate.....	45
Figure 5.11:	A CPL XOR Gate with G-Drive Gate Under Test.....	46
Figure 5.12:	Model/Simulation predictions for a CPL XOR/XNOR Gate	46
Figure 5.13:	The Input Slope Effect in a CPL gate involving S-Drive	47
Figure 5.14:	Model/Simulation predictions for a CPL XOR/XNOR Gate Involving S- Drive	49
Figure 6.1:	Examples of Critical Path in C-CMOS Block	52
Figure 6.2:	Schematics of a C-CMOS Full Adder and CPL Full Adder.....	56
Figure 6.3:	Schematic of a CPL Full Adder	57
Figure 6.4:	A Test Bench for 3-bit FAs	57
Figure 6.5:	Delay Comparison of Optimized FAs	59
Figure 6.6:	Delay Comparison of FAs in Mixed Styles	59
Figure 6.7:	Comparison of Power Consumption.....	60
Figure 6.8:	Comparison of PDP in Mixed Logic Styles.....	60
Figure 6.9:	Schematic and Currents of Level Restorer	61
Figure 6.10:	Model/Simulation predictions of Rising Delay for Level Restorer	62
Figure 6.11:	Schematic and Currents of Latch Insertion.....	63
Figure 6.12:	Model/Simulation predictions of Rising Delay for Latch Insertion	63

Figure 6.13:	Schematic of Buffer Insertion.....	64
Figure 6.14:	Model/Simulation predictions of Rising Delay for Buffer Insertion.....	65
Figure 6.15:	Schematic of Latch and Buffer Insertion.....	65
Figure 6.16:	Model/Simulation predictions of Rising Delay for Latch and Buffer Insertion	66
Figure 6.17:	Optimal Buffer Insertion.....	67
Figure 7.1:	A 4-bit Ripple-Carry Adder	70
Figure 7.2:	A 4-bit Carry-Bypass Adder	70
Figure 7.3:	A Carry-Select Adder	71
Figure 7.4:	The Actual RCA Test Chip	73
Figure 7.5:	Architecture of a RCA Test Chip	73
Figure 7.6:	Schematics of Tri-State Data In/Out Buffer.....	74
Figure 7.7:	The Critical Path in a 8-b RCA.....	75
Figure 7.8:	Schematics of a CPL Full Adder Involving G/S-Drive	75
Figure 7.9:	Delay Comparison of Eight 8-bit RCAs (Measured/Simulated)	80
Figure 7.10:	Power Comparison of Eight 8-bit RCAs (Measured).....	81
Figure 7.11:	PDP Comparison of Eight 8-bit RCAs (Measured).....	81
Figure 7.12:	The Architectures of a Test Chip and 64-bit Adder	82
Figure 7.13:	Schematic of a 1-b gpk Circuit	83
Figure 7.14:	Schematic of a 4-b GPK Circuit	84
Figure 7.15:	Schematic of a 4-bit Carry Generator Circuit.....	86
Figure 7.16:	Schematic of a 16-bit CLA Circuit	86
Figure 7.17:	Simulation Result of a 64-bit Adder	87
Figure 7.18:	(a) Delay Vs. Supply Voltage (b) Layout of Test Chip.....	88

List of Abbreviations

Γ	Width ratio of PMOS transistor to NMOS
μ	Carrier Mobility
μ_0	Low-field bulk mobility
μ_{eff}	Efficient carrier mobility
γ	The ratio of equivalent MOSFET linear resistance to saturating resistance
δ	An empirical parameter to calculate γ
ζ	An empirical parameter to calculate γ
θ	Mobility degradation factor
ϑ	Velocity saturation
α	Velocity saturation index of α -power law
ν	MOSFET unit resistance per reciprocal transistor width
ν_{pf}	PMOS unit falling resistance per reciprocal transistor width
ν_{pr}	PMOS unit rising resistance per reciprocal transistor width
ν_{nf}	NMOS unit falling resistance per reciprocal transistor width
ν_{nr}	NMOS unit rising resistance per reciprocal transistor width
τ	Transition time of input signal
a	An empirical parameter to calculate input slope factor S
C	Capacitance
C -CMOS	Conventional CMOS

C_D	MOSFET diffusion capacitance
C_{bottom}	MOSFET bottom capacitance
C_{sw}	MOSFET side-wall capacitance
C_j	MOSFET junction unit capacitance per area
C_{jsw}	MOSFET side-wall unit capacitance per length
C_d	MOSFET unit diffusion capacitance per transistor width
C_{dpr}	PMOS unit diffusion rising capacitance per transistor width
C_{dpf}	PMOS unit diffusion falling capacitance per transistor width
C_{dnr}	NMOS unit diffusion rising capacitance per transistor width
C_{dnf}	NMOS unit diffusion falling capacitance per transistor width
C_G	MOSFET gate capacitance
C_{G0}	MOSFET gate overlap capacitance
C_{GC}	MOSFET gate channel capacitance
C_{GCB}	MOSFET gate to body channel capacitance
C_{GCD}	MOSFET gate to drain channel capacitance
C_{GCS}	MOSFET gate to source channel capacitance
C_g	MOSFET unit gate capacitance per transistor width
C_{gpf}	PMOS unit gate falling capacitance per transistor width
C_{gpr}	PMOS unit gate rising capacitance per transistor width
C_{gnf}	NMOS unit gate falling capacitance per transistor width
C_{gnr}	NMOS unit gate rising capacitance per transistor width

<i>CFP</i>	Ceramic flat pack
<i>CI</i>	Input capacitance except itself
<i>CI_{Rise}</i>	Input rising capacitance except itself
<i>CI_{Fall}</i>	Input falling capacitance except itself
<i>CL</i>	Load capacitance
<i>CL_{Rise}</i>	Load rising capacitance
<i>CL_{Fall}</i>	Load falling capacitance
<i>CLA</i>	Carry lookahead adder
<i>CMC</i>	Canadian Microelectronics Cooperation
<i>Cox</i>	MOSFET gate oxide unit capacitance per area
<i>CO</i>	Output capacitance except itself
<i>CO_{Fall}</i>	Output falling capacitance except itself
<i>CO_{Rise}</i>	Output rising capacitance except itself
<i>CPL</i>	Complementary rans-transistor logic
<i>CSA</i>	Carry select adder
<i>D</i>	Propagation delay
<i>D_{Fall}</i>	Falling propagation delay
<i>D_{Rise}</i>	Rising propagation delay
<i>DCVSL</i>	Differential cascaded voltage switch logic
<i>DS</i>	Step delay
<i>DS_{Fall}</i>	Falling step delay

DS_{Rise}	Rising step delay
E	Electric field
FA	Full adder
I	Current
I_D	MOSFET drain current
K	Technology related current parameter
L	MOSFET length
m	Number of transistor gates connected with a single input signal
m_p	Number of PMOS's gates connected with a single input signal
m_n	Number of NMOS's gates connected with a single input signal
n	Number of transistors connected in series
n_p	Number of PMOS connected in series
n_n	Number of NMOS connected in series
$NORA$	None race logic
q_i	Number of transistor sources connected with a single input signal
q_i^p	Number of PMOS sources connected with a single input signal
q_i^n	Number of NMOS sources connected with a single input signal
q_o	Number of transistor drains connected to a output node
q_o^p	Number of PMOS drains connected to a output node
q_o^n	Number of NMOS drains connected to a output node
PCB	Printed circuit board

PDP	Power-Delay-Product
PTL	Pass transisor logic
R	Resistance
RC	Resistance-Capacitance
RCA	Ripple carry adder
S	Signal slope factor
S_{pf}	PMOS falling signal slope factor
S_{pr}	PMOS rising signal slope factor
S_{nf}	NMOS falling signal slope factor
S_{nr}	NMOS rising signal slope factor
$SCMs$	Series connected MOSFETs
$TSMC$	Taiwan Semiconductor Manufacturing Company Ltd
$TSPC$	True single phase clock logic
W	MOSFET width
W_p	PMOS width of cell gate
W_n	NMOS width of cell gate
W_{Dp}	PMOS width of drive gate
W_{Dn}	NMOS width of drive gate
W_{Lp}	PMOS width of load gate
W_{Ln}	NMOS width of load gate
V_{DD}	Power supply voltage

V_{ds}	MOSFET drain to source voltage
V_{gs}	MOSFET gate to drain voltage
V_{sb}	MOSFET body bias voltage
V_t	Threshold voltage
x	Current index which captures velocity saturation
X	Delay degradation factor for internal capacitance of SCMs
y	Captures mobility degradation effect
Y	Delay degradation factor for load capacitance of SCMs

Subscripts

C	Correspondence to cell gate
D	Correspondence to drive gate
L	Correspondence to load gate
n	NMOS transistor
p	PMOS transistor
r	rising transition
f	falling transition

1.1 Thesis Motivation

CMOS technology has evolved as the major VLSI technology. Most contemporary designs are making more and more dense CMOS devices operate at higher and higher speeds - integration and performance demands that have pushed VLSI circuits into the deep sub-micron regime. The overall design problems can be placed into time to market, reliability, testability, time delay, chip area, and power consumption. Of these, it is the accurate modeling of time delay that is very important, since it is the 'worst cases' that determine how fast (and capable) a circuit will be. The modeling of time delay has traditionally involved device-level simulators such as *SPICE*, but these methods, operating from the first principles of physics, exhibit a computation time that scales exponentially with the number of devices and this fact is now mandating a fundamentally different means of delay prediction in efficient and cost-effective design.

1.2 Analytical Delay Models

A number of simulation substitutes have been developed to address the problem of delay prediction, always with the goal of simultaneous speed and accuracy. Among these

are the equivalent inverter method [2], the RC-tree formulas [4], ones which estimate the propagation delay of series-connected MOSFET strings (SCM) [8], logic effort [29], and Shams's analytical delay model [1].

In the equivalent inverter method of [2], the behavior of an inverter 'equivalent' to the gate is used to facilitate calculation of its delay. Two parallel-connected transistors having widths W_1 and W_2 are replaced by a single transistor of $W = W_1 + W_2$, and a single-transistor equivalent to a series connection can similarly be devised: $W = (W_1 \cdot W_2) / (W_1 + W_2)$. The delay of the CMOS circuit is then held to be that of the equivalent inverter. Since all the internal capacitances are ignored, a delay computed in this manner may well be under-estimated. In addition, the equivalent inverter method also disregards the transistors' mode of operation (saturation, linear).

In an RC-tree method, a transistor is regarded as a resistor forming the charge/discharge path of a capacitor. Penfield and Rubenstein [4] calculated the signal delays in generalized RC trees, taking the possible fan-out of CMOS interconnection lines into account. This was improved upon by the Penfield-Rubenstein slope delay model which took into account the input's transition (rise or fall) time, an improvement over an earlier delay model of theirs.

The use of such delay models requires that tables of transistor parameters be available - the resistances are functions of transistor widths and transition times, as are the capacitances. Moreover, the performance of the model is poor in the submicron regime, particular for SCM's such as NAND and NOR gates [8], a result of assuming linear MOSFET operation. The behavior of short channel transistors in this regard is nonlinear. These inaccuracies can easily result in suboptimal transistor size values.

A delay model for submicron MOSFET circuits was proposed by Sakuari and Newton [8]. By means of their α -power law short-channel MOSFET model, the delay of a CMOS inverter can be calculated as a function of its ramp input. The delay of a series-connected MOSFET structure can then be estimated by extrapolation: multiplying by the ratio (delay of NAND/NOR)/(delay of inverter). While this approach provides a method of delay estimation in series-connected submicron MOSFET structures, there would still remain problems due to the input waveform structure (dv/dt) of SCMs.

Logical Effort [29] is a simple way of estimating delay in CMOS gates to design fast circuits. It addresses the gate delay with two terms: *Logical effort* and *Electrical effort*. Logical effort is a theoretical construct that measures the inherent in the circuit topology that implements a logic gate's function. Electrical effort is the ration of a logic gate's output capacitance to it's input capacitance. The product of Logic effort and Electrical effort, plus a term that regrests the parasitic capacitance of the logic gates, give a relative measure of the gate delay. Logical effort provides a simple and broadly applicable method for estimating the delay resulting from factors such as topology, capacitance, and gate sizes. Logic effort is good for optimal logic gate' sizing. But logical effort is inaccurate. This model is based on experimental estimations such that, for example, the ration of PMOS to NMOS transistor size is already assumed 2 for an inverter, and all MOSFET unit capacitances are assumed equal. Also, this model neglects the effects of input slopes.

An delay model for short channel MOSFETs is developed by Shams [1]. Based on the accuracy short channel current model, the delay model take account the delay effectors such as "oppose" and "overlap" current, input waveform effect, and the degradation factor

of SCMs. Also optimal closed-form transistor sizing formulas for C-CMOS, DCVSL, and PTL CMOS logic circuits were developed.

1.3 Thesis Objectives

Over the past two or three decades CMOS has come to its present position as the most important of the VLSI design technologies, a predominance that has come at a price in the form of industry demand for ever-increasing functionality at ever-higher speeds and lower power dissipation- requirements that have seen VLSI designers being forced to explore ever deeper submicron realms. Solving the resulting complicated design problems involves optimization of digital CMOS VLSI circuits with respect to signal delay, chip area and power dissipation, the most important of which involves the accurate modeling of signal delay.

Effective optimization methods aimed at achieving maximal speeds in single-technology logic circuits are widely available but systematic ways suitable for circuits involving mixtures of logic families are not.

The objectives of this research include:

- Verifying the delay models and optimal formulas [1] to estimate and optimize the delay in both $0.13\mu m$ and $0.18\mu m$ technologies.
- Exploring the closed-form transistor-switching delay model developed in [1]. Investigating this model with different factors that affect circuit delay, such as overlapping and opposing current, input time derivative, MOSFETs capacitance, and the series combination of a number of transistors for TSMC $0.18\mu m$ technology.
- Showing how the delay model can be applied to modern mixed-logic circuits (conventional and differential CMOS).

- Finding the optimal number of buffer insertions into a CPL chain in particular.
- Building real silicon chips and verifying predictions of the closed-form method's potential as a design tool.

1.4 Thesis Outline

This thesis is organized as follows: Chapter 2 briefly describes the CMOS logic styles that are of greatest interest in modern VLSI design, including dynamic and static circuits. Chapter 3 introduces an analytical delay model suited to sub-micron MOSFETs. The factors affecting delay, such as short channel effects, overlapping and opposing currents, the input slope effect and the series-connected-transistor effect, are described. Chapter 4 presents a parameter extraction procedure, wherein the BSIM3v3 model is assumed to mimic the behavior of Nature. Chapter 5 shows how the analytical delay model is used to derive optimal transistor sizing formulas pertinent to various conventional and differential CMOS logic styles. Chapter 6 presents a practical example of how one goes about using the new method in mixed-logic design at the block (circuit) level. Chapter 7 describes an actual physical application and hence a confirmation of the optimized mixed logic technique's effectiveness. Chapter 8 is a brief conclusion.

2.1 Introduction

The increasing demand for high-speed very large scale integrated circuit (VLSI) can be addressed at different levels of abstraction ranging from architectural theory through the increasingly concrete levels of circuit design, circuit layout and process technology. The concerns of this thesis are those of circuit design. Figure 2.1 is a simple digital CMOS IC ‘family tree’, the lowest branches of which represent static and dynamic approaches to logic gate design. Further branching occurs on the static circuit side, to Conventional (C-CMOS), Pass-Transistor Logic (PTL) and to Differential Cascade Voltage Switch Logic (DCVSL). On the dynamic circuit side are to be found the important sub-categories Domino logic, None Race logic (NORA) and True Single-Phase Clock (TSPC) circuit.

Figure 2.1: Classification of Digital CMOS Logic Styles

2.2 Static CMOS Logic Styles

A number of desirable characteristics have led to static CMOS logic becoming ubiquitous in electronics design. In this section, we will study the design of the static circuit options shown in the family tree, namely C-CMOS, PTL and DCVSL.

2.2.1 Conventional CMOS Logic

Figure 2.2: Schematic of a C-CMOS NAND Gate

The most widely used of static logic styles is conventional static CMOS (C-CMOS) [3]. A typical logic gate (a NAND gate) in C-CMOS is built from N and P blocks as shown in Figure 2.2. Such gates have their ‘pro’s and ‘con’s:

C-CMOS devices display a robustness where voltage and transistor scaling is concerned. In the static state there is no direct path between V_{DD} and ground, and so the output exhibits a rail-to-rail voltage swing and the static power consumption is very low. An ordinary C-CMOS inverter has a strong current drive capability because there is always a low resistance path between the output and one or the other rail (through either the P-block or the N-block). The complementary nature of C-CMOS gates makes their layout straightforward and efficient.

The necessary P-block is sometimes an obstacle to the employment of the CMOS style. The carrier mobility of PMOS transistors is lower than that seen in NMOS devices,

and this makes it necessary that the P-block be physically larger than the N-block. To the capacitance increase thus caused is added that due to the fact that each input is connected to at least two MOSFET gates (one PMOS, one NMOS).

2.2.2 Pass-transistor Logic

Figure 2.3: Schematic of a CPL NAND/AND Gate

A popular and widely-used alternative to C-CMOS is Pass-Transistor Logic (PTL), the most widely-used version of which is so-called Complementary Pass-transistor Logic.

A CPL [6] gate is fed by complementary pairs of input signals which are processed by two NMOS logic networks. The voltage drops along the CPL chains are compensated by two small pull-up PMOS transistors and the otherwise feeble output current is enhanced by two output inverters (Figure 2.3). As with C-CMOS, the use of CPL is accompanied by advantages and disadvantages:

The operation speed of a CPL function is typically higher than its C-CMOS equivalent because the absence of PMOS transistors significantly reduces a source of parasitic capacitances. Also CPL is mainly used to implement complex functions (XOR and MUX based) which employ smaller and fewer transistors because of the input signals can driving either gate or source terminals.

A serious problem in CPL circuits is voltage drop. Without pull-up PMOS transis-

tors, a CPL gate's output voltage swing is lower than the input swing by the NMOS threshold voltage (V_t). Another problem is limited logic depth (number of 'links' in the CPL chain). The ohmic reduction in current drive capability as stages are added places a limit on attainable logic depth. Finally, the layout of a CPL cell is not so straight forward and efficient as for a CMOS one because of irregular transistor placements and high density wiring.

2.2.3 Differential Cascaded Voltage Switch Logic

Figure 2.4: Schematic of a DCVSL NAND/AND Gate

A DCVSL-based high-performance logic family was first described by Heller et al. [11]. A number of DCVSL variations have been described in the literature, all employing the basic concept of using pairs of differential input signals to form a logic function and simultaneously store its (differential) value in a static output flip-flop. Figure 2.4 shows a basic static DCVSL NAND/AND gate:

The advantages of DCVSL are: (1). easy design of complex logic functions like the XOR, the MUX and the Full Adder, (2). further efficiency is due to the absence of large PMOS transistors in the part of the device which generates the actual logic function. PMOS appear only as pull-up and storage devices, (3). the differential structure of DVSL devices is responsible for their inherent noise immunity.

Unfortunately DCVSL is not free of problems, the principal ones of which are high dynamic power consumption and poor output current capability. Also the PMOS current

in the evaluate branch keeps charging the output node until its voltage drops under V_t . This results in a large dynamic power dissipation.

2.3 Dynamic Logic Styles

Dynamic logic displays significant advantages over static circuitry in high-density and high-performance applications. The basic dynamic circuit styles (Domino, NORA, TSPC) of Figure 2.1 will now be reviewed.

2.3.1 Domino CMOS

Figure 2.5: Cascaded Domino CMOS Logic Gates [3]

A Domino gate consists of an N-Block gate stage, a static CMOS inverter to drive the circuit's output, and a PMOS transistor to restore the logic at the domino output node as shown in Figure 2.5.

Domino logic has a single PMOS transistor, a situation which leads to a substantially lowered transistor count and input capacitance as compared to ordinary CMOS. The enhanced speed resulting is further improved by the logic block's being built only of high-carrier-mobility NMOS transistors.

However, a Domino circuit shows a power dissipation that is higher than the equivalent CMOS device because the output nodes are always precharged to VDD during each clock cycle. Domino logic is also vulnerable to noise - because of subthreshold leakage currents. The necessity of global clock distribution presents timing, power and layout

problems. Domino circuits are inflexible in that only positive-logic functions maybe implemented.

2.3.2 NORA CMOS Logic

Figure 2.6: NORA CMOS Logic Circuits [3]

As Figure 2.6 indicates, NORA logic is very similar to Domino logic. The main difference is that an alternation of N and P logic blocks in the cascades makes possible the direct coupling of these blocks and the inter-stage CMOS inverters unnecessary. NORA may be regarded as both an alternative and complement to the NMOS-only Domino CMOS. Another advantage of NORA CMOS logic is that it allows pipelined system architecture.

Like the Domino circuit described beforehand, the use of NORA circuitry is attended with difficulties related to global clock distribution, namely timing, power and layout problems. Slow clock transitions can lead to conductance overlap between the NMOS and PMOS devices and a consequent high power dissipation. And as is the case with all dynamic CMOS structures, NORA CMOS logic gates suffer from charge sharing and leakage problems.

2.3.3 True Single-Phase Clock (TSPC) Dynamic Circuit

A TSPC circuit diagram is shown in Figure 2.7. The circuit consists of alternating N-blocks and P-blocks, each of which is driven by the same clock signal CLK . A N[P]-block is a dynamic NMOS [PMOS] logic stage, dynamic latch cascade.

Figure 2.7: A Pipelined TSPC Logic Circuit [3]

This TSPC circuit differs from the NORA circuit architecture in that it uses only a single clock polarity (hence the name). Since an inverted clock signal is not used anywhere in the system, no clock skew problem can occur and higher system clock frequencies are possible. The latches appearing in a cascade of TSPC stages offer design opportunities that do not exist in Domino and NORA cascades.

In addition to the global clock-distribution problems of dynamic systems, a TSPC half latch has high noise and leakage sensitivity due to its high output impedance.

2.4 Conclusion

We have discussed six of the most commonly-used CMOS logic styles, each of them having characteristic advantage and disadvantages. The static categories have the advantage of robustness with regard to operating environments (noise, battery voltage), something which makes the design process rather trouble-free and adaptable to automation. Dynamic logic, on the other hand, makes it possible to implement faster, smaller yet more complex circuitry, but this comes at the price of susceptibility to parasitic effects and charge leakage.

The remainder of this thesis will be concentrated on the application of circuit optimization techniques to the three static logic types described in this chapter: C-CMOS, CPL, and DCVSL.

MOSFET Delay Modeling

3.1 Introduction

Designing a digital integrated circuit in a reasonable period of time requires that accurate time-delay estimates may be available at the outset, otherwise a costly iterative process will be necessary. An analytical model which takes into account the most important of the influences on time delay will be studied in this chapter.

The delay model [1] will be presented in a sequential fashion, with device variations such as short channel effects being incorporated first. Two fundamental delay types, step and propagation, will be then be defined and simple equations relating them to physical and dynamic quantities will be given. An estimate of CMOS saturation current is found to be necessary, and an extension of the Sakurai [7] α -current model will be presented to provide this. This is true also of gate capacitance - some helpful capacitance models will be discussed in detail, and their relevance to our model described. The inclusion of current superposition (overlapping and opposing currents), and of the input slope effect will be discussed in detail, as will the effects seen in series-connected transistors.

3.2 Short Channel Effects

Differences of short-channel and long channel device electrical characteristics are

Figure 3.1: Velocity Saturation Effect and Mobility Degradation

mainly due to *velocity saturation* and *mobility degradation*. Channel carrier velocity v is proportional to the strength of the electric field component in the direction of travel, $v = \mu_{eff}E$, where μ_{eff} is the efficient carrier mobility. The μ_{eff} of a long-channel transistor is assumed constant, but in a short channel device it is a function of the electric field E_t transverse to the longitudinal E . The carrier velocities exhibit the behavior shown in Figure 3.1 (a). ‘Velocity saturation’ in the case of short channel devices is evident.

The reduction of oxide thickness in a short-channel device (which causes the transverse field to increase) is accompanied by a degrading of the effective carrier mobility. This behavior is shown in Figure 3.1(b) and described by the empirical Equation (3.1).

$$\mu_{eff} = \frac{\mu_0}{1 + \theta(V_{gs} - V_t)} \quad (3.1)$$

where μ_0 is the low-field bulk mobility and θ is the mobility degradation factor.

3.3 Delay Definitions

Before we study expressions for the most important delay times encountered, we will present the definitions of delay.

3.3.1 Step Delay

The step delay times DS_{Fall} and DS_{Rise} refer to the time lag between the application of a rising or falling step (‘infinitely’ fast) input and the corresponding output change.

Figure 3.2: Definition of Step Delay

Figure 3.2 shows the step pulse input and output voltage waveforms of a typical inverter circuit. The times, measured between the 50% levels of the appropriate voltage levels, are

$$DS_{Fall} = T1 - T0 \quad (3.2)$$

$$DS_{Rise} = T3 - T2 \quad (3.3)$$

These step delays are related to W (transistor width) in a formula given by Shams, which gives either of the times taken to charge [discharge] an output capacitance C [1]:

$$DS = \frac{C \cdot V_{DD}/2}{I_D} = \frac{v \cdot C}{W} \quad (3.4)$$

The '2' comes from the fact that the output capacitance is charged to a level of $0.5V_{DD}$. I_D is the transistor drain saturation current and C is the total output node capacitance. The formula can be written in a form which involves the more physically-accessible parameters W and v , the latter, the resistance per unit reciprocal transistor width, combines within it all supply voltage and technology-dependent parameters. Figure 3.3 illustrates the step-delay characteristics of both PMOS and NMOS transistors in terms of load capacitance C , supply voltage V_{DD} , and transistor size W , both modelled (Equation (3.4)) and simulated.

3.3.2 Propagation Delay

A ramp input rather than a step is used as the stimulus when measuring the propagation delays D_{Fall} and D_{Rise} . These delays are analogous to the step delay measured earlier. The input and output voltage waveforms of a typical inverter circuit are shown in

Figure 3.3: Step Delays Vs Models (0.13 μm Technology)

Figure 3.4: Definition of Propagation Delay

Figure 3.4.

The 'fall' in D_{Fall} [D_{Rise}] refers to the inverter's output, which will be in response to a rising [falling] input. The propagation delays in Figure 3.4 are

$$D_{Fall} = T_1 - T_0 \quad (3.5)$$

$$D_{Rise} = T_3 - T_2 \quad (3.6)$$

Both delays are simply represented by [10]:

$$D = DS + S_{\tau} \tau \quad (3.7)$$

In this expression, D is the propagation delay, DS the step delay, S_{τ} the input slope factor and τ is the input slope. The meaning of these quantities will be discussed in the next

sections.

3.4 Current Model

A current model, an extension of Sakurai's α -power law [7] will now be presented, taking into consideration the short-channel effects discussed in the previous sections. The saturation current in modern transistors often follows

$$I_D = K \cdot W(V_{gs} - V_t)^\varepsilon \quad (3.8)$$

where W is the effective width of the device and K depends both on the technology and the 'effective' channel length (usually the minimum channel length in digital design). The exponent ε is usually taken to be a constant - α in Sakurai's formula and 2 in the square law.

But in this drain current model, ε is more complex, depending on x which accounts for the velocity saturation effect, and $y \cdot V_{gs}$ to describe the effect of increasing transverse field on carrier mobility. The proposed new law is

$$I_D = K \cdot W(V_{gs} - V_t)^{(x - y \cdot V_{gs})} \quad (3.9)$$

Figure 3.5 shows a comparison between the results of SPICE simulation, of the α -power model, and of the new model: As V_{gs} increases, it increases the number of the mobile charges in the channel, which boosts the current, but at the same time, the effective carrier mobility is reduced which has a current-reducing effect. An examination of the graph shows that the α -power law consistently underestimates the drain current, whereas the new model's predictions coincide nicely with the SPICE values.

3.5 Overlapping and Opposing Currents

In some CMOS logic styles, such as DCVSL, CPL, or CPL with level restoration or

Figure 3.5: Comparison of Current Models

latches, there are cases where a current I_p is superposed with another one (I_n), either 'overlapping' or 'opposing' it. Shams modified Equation (3.4) to take account of this situation [1]:

$$DS = \frac{CV_{DD}/2}{I_p \pm I_n} = \frac{v_p v_n}{W_p v_n \pm W_n v_p} C \quad (3.10)$$

Figure 3.6 shows step delays of simulation results and models for overlapping currents in a transmission gate. (a) shows the behaviour of the falling step delay as the supply voltage changes, (b) and (c) shows the effect on falling and rising step delays of transistor size changes ($W_p = 2.5W_n$), and (d) shows the effect of load capacitance on the rising step delay.

Figure 3.7 shows the falling step delays for the case of opposing current as predicted both by simulation and by this model. Part (a) shows how the falling step delay depends on load capacitance, Part (b) the dependence of falling step delay with NMOS transistor size changes.

Figure 3.6: Step Delays of Overlapping Currents (0.13μm Technology)

Figure 3.7: Step Delays of Opposing Currents (0.13μm Technology)

The gaps exist in these figures, but our main purpose of the model is to optimize the delay not to evaluate the delay.

3.6 Capacitance Model

3.6.1 Gate Capacitance

A CMOS transistor gate is isolated from the conducting channel by an oxide layer having a capacitance per unit area equal to $C_{ox} = \epsilon_{ox}/t_{ox}$. When multiplied by the gate

area this becomes C_G , the gate capacitance, which can be decomposed into two contributions: overlap and channel capacitance.

Overlap capacitance, C_{G0} , is a parasitic capacitance which is itself the sum of two others: gate to source and gate to drain. It depends solely on to the structural details of the transistor. It is strictly linear and is given by:

$$C_{G0} = C_{GS0} + C_{GD0} = C_{ox} W(L - L_{eff}) = C_0 W \quad (3.11)$$

Where L is the transistor length and L_{eff} the 'effective' transistor length. The difference between the two is a technology-dependent quantity which can be merged with C_{ox} to form the overlap capacitance per unit transistor width, C_0 , as indicated in the formula.

The second part of C_G is channel capacitance, or C_{GC} . This is perhaps the most significant CMOS parasitic circuit element, and consists itself of three parts: C_{GCS} (gate-to-source capacitance), C_{GCD} (gate-to-drain capacitance) and C_{GCB} (gate-to-body capacitance), all of which depend upon the operating region and upon terminal voltages. Table 3.1 list these three components, along with the total gate capacitance in different operating regions.

TABLE 3.1: MOSFET Gate Capacitance

Operating Region	C_{GCB}	C_{GCS}	C_{GCD}	C_{GC}	C_{G0}	C_G
CutOff	WLC_{ox}	0	0	WLC_{ox}	WC_0	$WLC_{ox} + WC_0$
Linear	0	$2/3WLC_{ox}$	$2/3WLC_{ox}$	WLC_{ox}	WC_0	$WLC_{ox} + WC_0$
Saturation	0	$2/3WLC_{ox}$	0	$2/3WLC_{ox}$	WC_0	$2/3WLC_{ox} + WC_0$

A transistor always operates in its saturation region because delay is measured at $0.5V_{DD}$ [1], and it is necessary to have a formula for gate capacitance appropriate only to that region:

$$C_G = C_{GC} + C_{G0} = \frac{2}{3}WLC_{ox} + WC_o = W\left(\frac{2}{3}LC_{ox} + C_o\right) = W \cdot C_g \quad (3.12)$$

Since the length of transistor is kept to a (constant) minimum in digital design, we can combine the two terms in parenthesis into a gate capacitance per unit width C_g , as shown. Since channel capacitance C_{GC} is voltage dependent, then so is the unit gate capacitance C_g .

For simplicity, we introduce two average gate capacitances per unit width, one pair applicable to NMOS devices ($C_{g_{nr}}$, $C_{g_{nf}}$), and another for PMOS devices ($C_{g_{pr}}$, $C_{g_{pf}}$). Here C_g itself denotes unit gate capacitance, the subscripts n and p indicate NMOS and PMOS, while r and f designate rising and falling delay transitions. These are going to be used in the gate level analytical delay models.

3.6.2 Diffusion Capacitance

Diffusion capacitance (C_D) is that of the reverse-biased source-body and drain-body pn-junctions. This depletion-region capacitance is a nonlinear decreasing function of the reverse bias level and consists of two components: bottom capacitance and side-wall capacitance. The bottom capacitance is given by $C_{bottom} = C_j WL_s$, with C_j the junction capacitance per unit area and L_s the side length of side wall. The side-wall capacitance, which is typically graded from 0.35 to 0.5 with the reverse bias voltage, is given by $C_{sw} = C_{jsw}(W + 2L_s)$, where C_{jsw} is the side-wall junction capacitance per unit length. The total diffusion capacitance is:

$$C_D = C_{bottom} + C_{sw} = C_j WL_s + C_{jsw}(W + 2L_s) \quad (3.13)$$

Since L_s is a technology-determined parameter, we can combine it with the other quantities appearing in this equation and write:

$$C_D = 2L_s C_{jsw} + W(C_{jsw} + L_s C_j) = C_{D0} + W \cdot Cd \cong W \cdot Cd \quad (3.14)$$

Where Cd is the unit diffusion capacitance per unit width; C_{D0} can be ignored because it

is constant and small.

MOSFET diffusion capacitance is a non-linear function of the terminal voltages, and to make a simple first-order analysis possible, it will be necessary to find a linearized approximation for it, one which produces an average capacitance value in the saturation region. Two average diffusion capacitances per unit width for each of NMOS and PMOS-type transistors are introduced, namely Cd_{nr} , Cd_{nf} , Cd_{pr} , and Cd_{pf} , where Cd denotes unit diffusion capacitance and the other subscripts have the same meanings as before (sec. 3.6.1). These also are going to be used in gate level delay models.

3.7 Modeling of Series Connected Transistors

Structures involving series-connected transistors are important because of their use in NAND/NOR gates, PLA's and arithmetic modules. The modeling of these series configurations is more complex than that of single devices because the MOSFETs operate in their nonlinear regions. Shams found that the delay of N series-connected transistors is proportional to the delay of a single transistor with the same load through the delay degradation factors X and Y . The step delay of the series-connection is given by[1]:

$$DS = \frac{v}{W}(X \cdot C + Y \cdot CL) = \frac{v}{W}(X \cdot W \cdot Cd + Y \cdot CL) = v \cdot X \cdot Cd + \frac{v \cdot Y}{W} CL \cong \frac{v \cdot Y}{W} CL \quad (3.15)$$

where delay degradation factors X and Y are:

$$X = Y(N^2 - 1) + 1 \quad (3.16)$$

$$Y = Y(N - 1) + 1 \quad (3.17)$$

$$Y = \delta - \zeta \frac{V_{DD}}{V_T} \quad (3.18)$$

Where Y is the ratio of equivalent MOSFET linear resistance to saturating resistance, δ and ζ are empirical parameters. Note that $v \cdot X \cdot Cd$ is a constant because the diffusion capacitance C is proportional to the transistor width W . Hence, in pull-up and pull-down transistor chains, only the load capacitance determines the optimal transistor sizing for

Figure 3.8: Step Delays of Series Connected Transistors (0.13 μ m Technology)

minimizing the delay, and the internal nodes can be ignored in this regard.

Figure 3.8 shows the rising and falling step delays (Y-axis) of a series connection made up of N (X-axis) transistors. From this figure, we can see that curves of model are approximately matched with the curves of simulation. The gap between them is due to ignored inner capacitances.

3.8 Effect of Input Slope (dv/dt) on Delay

Hedenstierna described a simple way to account for the effect of the input signal slope on the propagation delay [10]:

$$D = DS + S_t \tau \quad (3.19)$$

That is, the propagation delay is given by the step delay plus the product of the input slope and the input rise or fall time. This approximation is only valid if the input slope is less than one-third of the output slope [10], a condition that usually holds in VLSI circuits.

Sakurai has derived the following expression for S_t [8].

Figure 3.9: Input Slope Effect (0.13 μm Technology)

$$S_{\tau} = \frac{1}{2} - \frac{1 - V_T/V_{DD}}{1 + a} \quad (3.20)$$

Each of the four propagation delay categories has associated with it a value of a (an empirical parameter). Figure 3.9 illustrates the dependence of each on the input signal slope, as calculated by simulation and by the above expression. These figures demonstrate that Equation (3.19) and (3.20) can be used for prediction of the effects of input slope on propagation delay.

For cascaded gates, Shams found that a step delay is approximately equivalent to half of the corresponding transition time τ ($\tau = 2DS$), that the propagation delay of a gate D_i can be expressed in terms of both its own step delay DS_i and the driving gate's step delay DS_{i-1} , as follows [1].

$$D_i = DS_i + S \cdot DS_{i-1} \quad (3.21)$$

S is once again the slope factor ($S = 2S_{\tau}$). There will be four categories of this delay.

The input slope factor will not be the same in the overlapping and opposing current

Figure 3.10: Propagate Delays involving Overlapping and Opposing Currents (0.13 μ m Technology)

cases. The overall (current-weighted) slope factors are [1]:

$$S = \frac{S_n I_n \pm S_p I_p}{I_n \pm I_p} = \frac{S_n W_n / v_n \pm S_p W_p / v_p}{W_n / v_n \pm W_p / v_p} \quad (3.22)$$

Figure 3.10 shows the matches achieved between these models and the results of SPICE simulation to determine the dependence of propagation delay on input slope.

3.9 Conclusion

In this chapter, an analytical delay expression [1] was verified both in 0.13 μ m and 0.18 μ m technologies. A current model which takes account of short channel effects was presented and integrated into the delay expression. A linear formula for MOSFET capacitance as a function of transistor width was introduced. The effects of overlapping and opposing currents on delay were added. The effects of connecting transistors in series and input slope factors were explored. The comparison between numerical simulation results and this analytical delay model shows a good match.

Delay Parameters Extraction

4.1 Introduction

The formulas we have presented are parametrized in terms of quantities that must be obtained experimentally. The parameters affect the results of delay measurements and may be ‘extracted’ from sets of such measurements taken from a model circuit and processed with curve-fitting methods (optimization - the BSIM3v3 CMOS model from UC Berkeley was assumed). A similar approach is used in [1] for a $0.5\mu m$ CMOS technology, while we consider $0.13\mu m$ and $0.18\mu m$ CMOS technology.

The goal was to find the parameters used in the prediction of channel current (x , y , K and v of section 3.4), the ones which particularize the input slope effect (α and S_t , section 3.9), a set of series-connection degradation factors (Y , section 3.8) and MOSFET gate and diffusion capacitances (four each, section 3.7). These were extracted from delay measurements taken on test circuits similar to those of Figure 4.1a and Figure 4.1b. Each one of these determinations will be described in brief below.

4.2 Extraction of Current Parameters

Figure 4.1 shows two NMOS transistors and two PMOS transistors connected so that four types of step delay can be characterized (NMOS rising and falling, PMOS ris-

Figure 4.1: A Test Bench for Delay Parameter Extraction

ing and falling). A load capacitance value of $C_L = 100\text{fF}$ was used, a value compared to which the diffusion capacitances are negligible. The transistor width W was set to the reasonable value of $2\mu\text{m}$. The output nodes were (dis)charged to $(V_{SS})V_{DD}$ before the measurements.

Prediction of channel current ultimately requires the parameter

$$v = \frac{V_{DD}}{2I_D/W} = \frac{V_{DD}}{2KV_{gs} - V_t)^{(x-y \cdot V_{gs})}} \quad (4.1)$$

which is in turn dependent upon parameters x , y and K . Their values were obtained by measuring step delays at three different supply voltages (V_{DD}) and applying the appropriate BSIM 3v3 equations. These parameters can be extracted in the following steps.

1. Measure the step delays DS_1 , DS_2 , and DS_3 at three different supply voltage settings

$$V_{DD} = V_1, V_{DD} = V_2, \text{ and } V_{DD} = V_3, \text{ respectively.}$$

2. Calculate current index x and y from the following set of equations.

$$y = \frac{a_2 \cdot c_1 - a_1 \cdot c_2}{b_1 \cdot c_2 - b_2 \cdot c_1} \quad (4.2)$$

$$x = \frac{a_1 + y \cdot b_1}{c_1} \quad (4.3)$$

Where a_1 , a_2 , b_1 , b_2 , c_1 , and c_2 are:

$$a_1 = \log \frac{DS_2 \cdot V_1}{DS_1 \cdot V_2} \quad a_2 = \log \frac{DS_3 \cdot V_1}{DS_1 \cdot V_3} \quad (4.4)$$

$$c_1 = \log \frac{V_1 - V_t}{V_2 - V_t} \quad c_2 = \log \frac{V_1 - V_t}{V_3 - V_t} \quad (4.5)$$

$$b_1 = V_1 \cdot \log V_1 - V_t - V_2 \cdot \log(V_2 - V_t) \quad (4.6)$$

$$b_2 = V_1 \cdot \log(V_1 - V_t) - V_3 \cdot \log(V_3 - V_t) \quad (4.7)$$

3. Calculate K from

$$K = \frac{V_1 C}{2DS_1 W(V_1 - V_t)^{(x-y \cdot V_1)}} \quad (4.8)$$

The parameter values, appropriate to TSMC 0.18 μm technology, were determined by carrying out the steps described above and are listed in Table 4.1. It is apparent that an NMOS transistor exhibits superb conductivity in both the falling and rising modes and also that P-channel conductivity should be high in the case of a rising transmission. It is also evident that a PMOS device is probably not the best design choice when high falling transmission speeds are needed.

TABLE 4.1: Current-Modeling Parameters (0.18 μm VDD=1.8V)

Type/Transition	Kc	x	y	v
NMOS/Falling	401.8	1.24	0.05	1.69E-3
NMOS/Rising	212.2	2.65	0.39	4.77E-3
PMOS/Falling	65.8	2.83	0.36	19.5E-3
PMOS/Rising	152.9	1.62	0.10	4.09E-3

4.3 Extraction of Input Slope Factors

Hedenstiema's [10] linear expression for propagation delay is Equation (3.19) ($D = DS + S_\tau \tau$). The parameters of this equation are the step delay (DS) and the slope. One can find the step-delay (DS) by direct measurement. The result of one with $\tau \rightarrow 0$, and then another measurement with a larger τ value will allow one to compute S_τ , the deriva-

tive of delay time with respect to input rise time. The slope factor S is related to this and the parameter a by $S_\tau = \frac{1}{2} - \frac{1 - V_t/V_{DD}}{1 + a} = 0.5S$. (S is the slope parameter appearing in Equation (3.19) - $D_i = DS_i + S \cdot DS_{i-1}$ - the total delay as the sum of the present gate's step delay and that of the previous one in a cascade.)

The parameter a is calculated by rearranging Sakurai's Equation (3.20) ($a = \left(1 - \frac{V_t}{V_{DD}}\right) / \left(\frac{1}{2} - \frac{D - DS}{\tau}\right) - 1$). Once again, four sets of parameters had to be calculated, one for each of the NMOS and PMOS falling and rising delays. Results appropriate to TSMC $0.18\mu m$ technology are listed in Table 4.2. One can see that between 30% - 40% of the total delay of a two-gate cascade is due to the step delay of the first stage.

TABLE 4.2: Input Slope Factors ($0.18\mu m$ VDD=1.8V)

Type/Transition	a	S
NMOS/Falling	1.02	$S_{nf} = 0.3$
NMOS/Rising	0.76	$S_{nr} = 0.41$
PMOS/Falling	0.44	$S_{pf} = 0.34$
PMOS/Rising	1.36	$S_{pr} = 0.39$

4.4 Extraction of Series-Connection Degradation Factors

A suitable test arrangement is diagrammed in Figure 4.2. A series length of three devices was chosen because this is a typical figure (actual configurations seldom have more than five devices connected in series). A load capacitance of $C_L = 100\text{fF}$ was used so that the diffusion capacitance would be negligible in comparison. All three transistors had identical $2\mu m$ widths. The input/output signal path starting at the gate of the transistor fur-

Figure 4.2: A Test Bench for Series-Connected Transistors Parameter Extraction

test from the output node was chosen as a 'worst case' situation. The internal and output nodes were all initially (dis)charged to (VSS)VDD as shown in Figure 4.2.

The steps used were:

1). At $V_{DD} = V_1$ measure the step delay DS_{11} . Similarly, at $V_{DD} = V_2$ measure the step delay DS_{22} .

2). Calculate δ and ζ from the following set of equations:

$$\zeta = \frac{(d_2 - d_1) \cdot V_t}{V_1 - V_2} \quad (4.9)$$

$$\delta = d_1 + \frac{b \cdot V_1}{V_t} - \frac{C + CL}{(N^2 - 1) \cdot C + (N - 1) \cdot CL} \quad (4.10)$$

Where d_1 , d_2 are:

$$d_1 = \frac{2DS_1 \cdot kW(V_1 - V_t)^{(x-y \cdot V_1)}}{V_1 \cdot ((N^2 - 1) \cdot C + (N - 1) \cdot CL)} \quad (4.11)$$

$$d_2 = \frac{2DS_2 \cdot kW(V_2 - V_t)^{(x-y \cdot V_2)}}{V_2 \cdot ((N^2 - 1) \cdot C + (N - 1) \cdot CL)} \quad (4.12)$$

The values of δ and ζ for four types delays are listed in Table 4.3:.

3). Substitute these values of δ and ζ into Equation (3.18), and thence calculate the degradation factors X and Y (3.16, 3.17).

TABLE 4.3: Series-Connection Degradation Parameters ($0.18\mu m$ VDD=1.8V)

Type/Transition	δ	ζ
NMOS/Falling	0.52	0.02
NMOS/Rising	0.99	0.13
PMOS/Falling	1.29	0.15
PMOS/Rising	1.12	0.06

TABLE 4.4: Series-Connection Degradation Factors ($0.18\mu m$ VDD=1.8V)

Type/Number	2	3	4	5
Y_{nf} (NMOS/Falling)	1.46	1.93	2.39	2.86
Y_{nr} (NMOS/Rising)	1.81	2.62	3.42	4.23
Y_{pf} (PMOS/Falling)	2.11	3.22	4.34	5.47
Y_{pr} (PMOS/Rising)	1.99	2.97	3.96	4.94

The values of the series-connection degradation factor 'Y' corresponding to 2 - 5 transistors and $V_{DD} = 1.8$ V are listed in Table 4.4.

4.5 Extraction of MOSFET Capacitances per Unit Width

Figure 4.3: A CMOS Inverter Chain [1]

The rising and falling propagation delays between the input and the output of the cell gate for a inverter chain of Figure 4.3 are given by [1]:

$$D_{CellFall} = D_{S_{CellFall}} + S_{nf} D_{S_{DriveRise}} \quad (4.13)$$

$$D_{CellRise} = DS_{CellRise} + S_{pr} DS_{DriveFall} \quad (4.14)$$

The step delays appearing in these equations are:

$$DS_{CellFall} = \frac{v_{nf}}{W_n} (W_{Lp} Cg_{pf} + W_{Ln} Cg_{nf} + W_p Cd_{pf} + W_n Cd_{nf}) \quad (4.15)$$

$$DS_{CellRise} = \frac{v_{pr}}{W_p} (W_{Lp} Cg_{pr} + W_{Ln} Cg_{nr} + W_p Cd_{pr} + W_n Cd_{nr}) \quad (4.16)$$

$$DS_{DriveFall} = \frac{v_{nf}}{W_{Dn}} (W_p Cg_{pf} + W_n Cg_{nf} + W_{Dp} Cd_{pf} + W_{Dn} Cd_{nf}) \quad (4.17)$$

$$DS_{DriveRise} = \frac{v_{pr}}{W_{Dp}} (W_p Cg_{pr} + W_n Cg_{nr} + W_{Dp} Cd_{pr} + W_{Dn} Cd_{nr}) \quad (4.18)$$

Where subscript D denotes drive gate, subscript L denotes load gate, subscript p denotes PMOS transistor, subscript n denotes NMOS transistor, subscript r denotes rising transiting, and subscript f denotes falling transiting.

TABLE 4.5: MOSFET Unit Capacitances ($0.18\mu m$)

Type/Transition	Cg (fF/ μm)	Cd (fF/ μm)
NMOS/Falling	2	2.6
NMOS/Rising	1.6	1.7
PMOS/Falling	1.5	2.7
PMOS/Rising	1.9	1.5

The four types of unit gate capacitances, Cg_{nr} , Cg_{nf} , Cg_{pr} , Cg_{pf} , and four types of unit diffusion capacitances Cd_{nr} , Cd_{nf} , Cd_{pr} , Cd_{pf} , for the TSMC $0.18\mu m$ technology are listed in Table 4.5

A straightforward procedure based on simulations of delay times of cells having differing transistor widths was used.

1). Initially, let $w_p = w_{Dp} = w_{1p} = 5\mu m$ and $w_n = w_{Dn} = w_{1n} = 2\mu m$. Measure the rising and falling delays of the cell gate: $D_{1Rise} = D_{RiseCell}$ and $D_{1Fall} = D_{FallCell}$.

2). Let $w_{Lp} = w_{2p} = 7.5\mu m$. Measure the rising and falling delays of the cell gate:

$D_{2Rise} = D_{RiseCell}$ and $D_{2Fall} = D_{FallCell}$. Then calculate the following:

$$Cg_{pr} = \frac{W_{1p}}{v_{pr}} \cdot \frac{D_{1Rise} - D_{2Rise}}{W_{1p} - W_{2p}} \quad (4.19)$$

$$Cg_{pf} = \frac{W_{1n}}{v_{nf}} \cdot \frac{D_{1Fall} - D_{2Fall}}{W_{1p} - W_{2p}} \quad (4.20)$$

3). Reset all transistor sizes back to their Step1 values. Let $w_{Ln} = w_{3n} = 5\mu m$. Measure the cell gate's rising and falling delays, $D_{3Rise} = D_{RiseCell}$ and $D_{3Fall} = D_{FallCell}$, and then calculate the following:

$$Cg_{nr} = \frac{W_{1p}}{v_{pr}} \cdot \frac{D_{1Rise} - D_{3Rise}}{W_{1n} - W_{3n}} \quad (4.21)$$

$$Cg_{nf} = \frac{W_{1n}}{v_{nf}} \cdot \frac{D_{1Fall} - D_{3Fall}}{W_{1n} - W_{3n}} \quad (4.22)$$

4) Reset all transistor sizes to their Step 1 values. Let $w_p = w_{4p} = 7.5\mu m$. Measure the falling delay of the cell gate: $D_{4Fall} = D_{FallCell}$. Calculate the following:

$$Cd_{pf} = \frac{W_{1n}}{v_{nf}} \cdot \left(\frac{D_{1Fall} - D_{4Fall}}{W_{1p} - W_{4p}} - S_{nf} \frac{v_{pr}}{W_{1p}} Cg_{pr} \right) \quad (4.23)$$

5) Reset transistor sizes back to their Step 1 values. Let $w_n = w_{5n} = 5\mu m$. Measure the cell gate's rising delay: $D_{5Rise} = D_{RiseCell}$. Calculate the following:

$$Cd_{nr} = \frac{W_{1p}}{v_{pr}} \cdot \left(\frac{D_{1Rise} - D_{5Rise}}{W_{1n} - W_{5n}} - S_{pr} \frac{v_{nf}}{W_{1n}} Cg_{nf} \right) \quad (4.24)$$

6) The remaining unknowns are cd_{nf} , and cd_{pr} . Reset transistor sizes to their Step1 values. Let $w_p = w_{Lp} = w_{Dp} = w_{6p} = 4\mu m$. Measure the cell gate rising and falling delays, $D_{6Rise} = D_{RiseCell}$ and $D_{6Fall} = D_{FallCell}$. Calculate cd_{nf} and cd_{pr} by substituting the parameters from Step2 to Step5 into Equation (4.13) and (4.14):

$$Cd_{nf} = \frac{D_{6Fall} - a1 - S_{nf} D_{6Rise} + S_{nf} S_{pr} a1}{v_{nf} \cdot (1 - S_{nf} S_{pr})} \quad (4.25)$$

$$Cd_{pr} = \frac{D_{6Rise} - a2 - S_{pr} a1 + S_{pr} v_{nf} Cd_{nf}}{v_{pr}} \quad (4.26)$$

where a_1 and a_2 are:

$$a_1 = \frac{v_{nf}}{W_{1n}} \cdot (W_{6p} Cg_{pf} + W_{1n} Cg_{nf} + W_{6p} Cd_{pf}) \quad (4.27)$$

$$a_2 = \frac{v_{pr}}{W_{6p}} \cdot (W_{6p} Cg_{pr} + W_{1n} Cg_{nr} + W_{1n} Cd_{nr}) \quad (4.28)$$

4.6 Conclusion

Parameters that are used in the estimation of channel-current, for forecasting the influence of input level rate-of-change on delay, for estimating the delay introduced on connecting MOS devices in series, and for estimating transistor gate and diffusion capacitances were extracted using experimental methods for TSMC $0.18\mu m$ technology. Four parameter sets were necessary, one set for building each of NMOS-falling, NMOS-rising, PMOS-falling and PMOS-rising delay predictors.

Modeling and Optimization of CMOS Gate Delay

5.1 Introduction

Over the past two or three decades CMOS has come to its present position as the most important of the VLSI design technologies, and is increasingly the subject of industry demand for ever-increasing functionality at ever-higher speeds - requirements that have seen VLSI designers being forced to explore ever deeper submicron realms. Solving the resulting complicated design problems involves optimization of digital CMOS VLSI circuits with respect to signal delay, chip area and power dissipation, the most important of which involves the accurate modeling of signal delay.

In this chapter, we will apply the analytical delay equations of the previous chapter to CMOS gates which have been constructed in the C-CMOS, DCVSL, and CPL styles, and verify closed-form optimal transistor-width expressions suited to each for TSMC 0.18 μ m technology.

5.2 C-CMOS Gate Delay Modeling and Optimization

Figure 5.1 depicts a generalized C-CMOS cell schematic implemented in the conventional pull-up and pull-down style along with a drive gate and load gate. Capacitance

Figure 5.1: A Conventional CMOS Cell [1]

CI represents the combined capacitances seen at the cell input (*i.e.*, those of the drive gate and the interconnect capacitances), and CO those at the cell output (load gate and interconnect capacitances). Each of CI and CO could well have differing values for rising and falling transitions.

The cell of the diagram is assumed driven by one of identical structure and in turn to be driving yet another of identical structure. Subscripts ' D ' and ' L ' are used to distinguish drive from load gates. The indicated pull-up [pull-down] section consists of a series/parallel arrangement of n_p [n_n] (series), q_o_p [q_o_n] (parallel) transistors, and m_p [m_n] transistors in parallel connected with a input source having transistors of width W_P [W_n], as shown in the 'cell' part of the figure. Each pull-up and pull-down section in the drive gate or the load gate is also characterized by some W , n , m , and q_o .

One may combine all of the **driver capacitances** (that is, all capacitances at the output with the exception of those related to the cell itself) into a single equivalent one [1]:

$$C_{D_{Rise}} = C_{I_{Rise}} + q_{o_{Dp}} \cdot W_{Dp} \cdot C_{d_{pr}} + q_{o_{Dn}} \cdot W_{Dn} \cdot C_{d_{nr}} \quad (5.1)$$

$$C_{D_{Fall}} = C_{I_{Fall}} + q_{o_{Dp}} \cdot W_{Dp} \cdot C_{d_{pf}} + q_{o_{Dn}} \cdot W_{Dn} \cdot C_{d_{nf}} \quad (5.2)$$

The **load capacitances** can be 'lumped' as well:

$$C_{L_{Rise}} = C_{O_{Rise}} + m_{Lp} \cdot W_{Lp} \cdot C_{g_{pr}} + m_{Ln} \cdot W_{Ln} \cdot C_{g_{nr}} \quad (5.3)$$

Figure 5.2: A C-CMOS XOR Design Under Test (DUT)

$$CL_{Fall} = CO_{Fall} + m_{Lp} \cdot W_{Lp} \cdot Cg_{pf} + m_{Ln} \cdot W_{Ln} \cdot Cg_{nf} \quad (5.4)$$

The propagation delays from the input of drive gate to the output of the cell gate are written [1]:

$$D_{Rise} = DS_{CellRise} + (1 + S_{pr}) \cdot DS_{DriveFall} \quad (5.5)$$

$$D_{Fall} = DS_{CellFall} + (1 + S_{nf}) \cdot DS_{DriveRise} \quad (5.6)$$

where $DS_{CellRise}$ and $DS_{CellFall}$ are the cell's rising and falling step delays:

$$DS_{CellRise} = \frac{v_{pr} \cdot Y_{pr}}{W_p} (qo_n \cdot W_n \cdot Cd_{nr} + qo_p \cdot W_p \cdot Cd_{pr} + CL_{Rise}) \quad (5.7)$$

$$DS_{CellFall} = \frac{v_{nf} \cdot Y_{nf}}{W_n} (qo_n \cdot W_n \cdot Cd_{nf} + qo_p \cdot W_p \cdot Cd_{pf} + CL_{Fall}) \quad (5.8)$$

and $DS_{DriveRise}$ and $DS_{DriveFall}$ are the step delays of the drive gate,

$$DS_{DriveRise} = \frac{v_{pr} \cdot Y_{Dpr}}{W_{Dp}} (m_n \cdot W_n \cdot Cg_{nr} + m_p \cdot W_p \cdot Cg_{pr} + CD_{Rise}) \quad (5.9)$$

$$DS_{DriveFall} = \frac{v_{nf} \cdot Y_{Dnf}}{W_{Dn}} (m_n \cdot W_n \cdot Cg_{nf} + m_p \cdot W_p \cdot Cg_{pf} + CD_{Fall}) \quad (5.10)$$

The average delay is:

$$D = \frac{D_{Rise} + D_{Fall}}{2} \quad (5.11)$$

By way of example, consider a 2-input XOR circuit like that of Figure 5.2. The worst case path is that involving the transistor furthest away from the output node - we

Figure 5.3: Model/Simulation predictions for a C-CMOS XOR Gate

consider the delay when it alone is switched while all of the others in the chain have been ‘on’ long enough to ensure that the inner and output node voltages are steady at V_{DD} or V_{SS} (see Figure 5.2, dashed lines). The pull up network parameters are $m_p = 1$, $n_p = 2$ and $qo_p = 2$, and those of the pull down network $m_n = 1$, $n_n = 2$ and $qo_n = 2$. Equation (5 . 11) is used to calculate the propagation delay as a function of transistor size and the result is compared to that of a SPICE simulation in Figure 5.3. A model/simulation difference of only 2 picoseconds or so speaks well of the model’s accuracy.

The method of calculating (not just graphical estimation) the W_p and W_n values corresponding to minimal delay was developed in [1]:

$$W_p = \sqrt{\frac{v_{pr} \cdot Y_{pr}(1 + S_{nf})(qo_n \cdot W_n \cdot Cd_{nr} + CL_{Rise})}{m_p \cdot Cg_{pf} \frac{v_{nf}}{W_{Dn}} Y_{Dnf}(1 + S_{pr}) + m_p \cdot Cg_{pr} \frac{v_{pr}}{W_{Dp}} Y_{Dpr}(1 + S_{nf}) + qo_p \cdot Cd_{pf} \frac{v_{nf}}{W_n} Y_{nf}(1 + S_{pr})}} \quad (5.12)$$

$$W_n = \sqrt{\frac{v_{nf} \cdot Y_{nf}(1 + S_{pr})(qo_p \cdot W_p \cdot Cd_{pf} + CL_{Rise})}{m_n \cdot Cg_{nr} \frac{v_{pr}}{W_{Dp}} Y_{Dpr}(1 + S_{nf}) + m_n \cdot Cg_{nf} \frac{v_{nf}}{W_{Dn}} Y_{Dnf}(1 + S_{pr}) + qo_n \cdot Cd_{nr} \frac{v_{pr}}{W_p} Y_{nf}(1 + S_{nf})}} \quad (5.13)$$

Figure 5.4: The Iteration of the Optimal Sizing for a C-CMOS XOR Gate

Since each of W_p and W_n is a function of the other, this pair of equations must be iterated to a self-consistent solution, about 10 iterates being sufficient (Iteration would not be necessary if non-interdependent approximations were to be used for the two widths.). The progress of the calculation is shown in Figure 5.4:

These calculated transistor sizes are near those observed in the delay-time measurements summarized in Figure 5.3.

Table 5.1 lists the rising, falling and average propagation delays, along with the power dissipation and power-delay product (PDP) expected for differing transistor sizes:

TABLE 5.1: Performance Comparison of CMOS XORs

Size (μ m) W_p / W_n	Rising Delay (ps)	Falling Delay (ps)	Avg. Delay (ps)	Power (μ w)	PDP (psXuw)
Minimum Size (1/0.5)	229.6	124.5	177.1	101.9	18046.5
Larger Size (12/6)	100.8	95.9	98.4	345.2	33967.7
Standard Cell	200.8	176.3	188.5	151.4	28538.9
Optimal Size (6/3.9)	110.7	77.6	94.1	226	21266.6

The table shows that this size formula [1] predicts superior delay behavior overall. Power consumption is somewhat higher than that of the minimum-size and standard cells but greatly lower than that of a larger-size XOR configuration. Since delay-time improvement was the goal of the work, it must be regarded as having been successful, a success which was often accompanied by improvement in other aspects of circuit behavior. The figures, taken together, also provide an illustration of the compromises that must be made between speed, power consumption, area and PDP.

5.3 DCVSL Gate-Delay Modeling and Optimization

Figure 5.5: A DCVSL XOR/XNOR Gate Under Test

Figure 5.5 shows the important parts of a DCVSL XOR/XNOR gate with its drive and load gates. The pull-up network parameters are $m_p = 0$, $n_p = 1$ and $qo_p = 1$. A DCVSL pull-down network has two parts corresponding to the two outputs (these are often interconnected), with $m_n = 1$, $n_n = 2$ and $qo_n = 2$. This circuit will serve as a point of departure for the delay modeling and optimization discussion to follow.

In DCVSL gates, the rising output is actually produced by the falling output and, it can be seen that the worst-case delay will be associated with the rising path indicated by

Figure 5.6: DCVSL Delays and Associated Currents

the dashed line of Figure 5.5. The rising propagation delay from the input of drive gate to the output of cell gate is [1]:

$$D_{Rise} = DS_{CellRise} + (1 + S_{pr})DS_{CellFall} + (1 + S_{nf})DS_{DriveRise}, \quad (5.14)$$

where, as before, $DS_{CellFall}$ and $DS_{CellRise}$ are the cell gate's falling and rising step delays, and $DS_{DriveRise}$ is the drive gate's rising step delay.

During a rising step delay (Figure 5.6 (a)), it is only the PMOS current I_p that is charging the output nodes and so the step delay will be:

$$DS_{CellRise} = \frac{v_{pr}}{W_p} (q o_n W_n C d_{nr} + W_p (C g_{pr} + C d_{pr}) + C L_{Rise}) \quad (5.15)$$

During the falling delay (Figure 5.6 (b)) the current I_n flowing in the pull-down chain tends to discharge the output node capacitance while the PMOS current I_p is flowing into the same point. Shams found that the cell gate's falling step delay for this opposing current is

$$DS_{CellFall} = \frac{v_{pr} v_{nf} Y_{nf}}{W_p v_{pr} - 0.5 W_p v_{nf} Y_{nf}} (q o_n W_n C d_{nf} + W_p (C g_{pf} + C d_{pf}) + C L_{Fall}) \quad (5.16)$$

In Equation (5.14) only the rising step delay of the drive gate $DS_{DriveRise}$ is used (and not $DS_{DriveFall}$):

Figure 5.7: Model/Simulation predictions for a DCVSL XOR/XNOR Gate

$$DS_{DriveRise} = \frac{v_{pr} Y_{Dpr}}{W_{Dp}} (m_n W_n C_{g_{nr}} + C_{D_{Rise}}) \quad (5.17)$$

Figure 5.7 shows the propagation delays predicted by SPICE and by our model. It is evident that by using this model, virtually identical results may be obtained with vastly less effort.

To find the transistor widths that will yield the best delay performance, Shams gives expressions [1]:

$$W_n = \sqrt{\left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right)^{(1+S_{pr})} \frac{\left(W_p (C_{d_{pf}} + C_{g_{pf}}) + CL_{Fall} + 0.5 \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right) W_p C_{d_{nf}}\right)}{\frac{(1+S_{nf}) Y_{Dpr} m_n C_{g_{nr}}}{W_{Dp}} + \frac{q o_n C_{d_{nr}}}{W_p}}} + \frac{1}{2} \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right) W_p \text{ and } (5.18)$$

$$\frac{1}{\Gamma} = \frac{W_n}{W_p} = \sqrt{(1+S_{pr}) \left(\frac{1}{2} \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right)^2 \frac{CL_{Fall} + q o_n W_n C_{d_{nf}}}{CL_{Rise} + q o_n W_n C_{d_{nr}}} + \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right) \frac{W_n (C_{d_{pf}} + C_{g_{pf}})}{CL_{rise} + q o_n W_n C_{d_{nr}}}\right) + \frac{1}{2} \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right)} \quad (5.19)$$

where Γ is the optimum width ratio (PMOS to NMOS) for minimizing the rising delay. To get optimal sizes for a DCVSL XOR/XNOR gate, 15 iterations were required to obtain a solution of these simultaneous equations (see Figure 5.8). As in the C-CMOS case, the computed optimal transistor sizes ($W_N = 8.9 \mu m$, $W_P = 7.2 \mu m$) lie in the vicinity of the

Figure 5.8: The Iteration of Optimal Sizing for a DCVSL XOR/XNOR Gate

minima in the plots of size versus delay of Figure 5.7, a fact that attests to the utility of the model.

And as before, some comparisons between devices made to the specifications of the new model with more conventional ones have been tabulated in Table 5.2.

The results are consistent with those noted for the C-CMOS gate performance. That is, better or much-better delay figures coming at a cost of intermediate power-consumption figures. The delay and power consumption figures may be combined into a single figure-of-merit, the PDP (power-rising-delay product) - the PDP of the optimal device is larger than minimum one, but much better-than larger one. The figures, taken together, also provide an illustration of the compromises that must be made between speed, power consumption, area and PDP

TABLE 5.2: Performance Comparison of DCVSL XOR Gates

Size (um) W_p / W_n	Rising Delay(ps)	Falling Delay(ps)	Ave. Delay(ps)	Power(uw)	PDP(psXuw)
Minimum Size(0.5/1)	377.8	76.7	227.2	188.1	71064.2
Larger Size(10/14)	159.9	67.7	113.8	990.6	158396.9
Optimal Size(7.1/8.9)	156.8	63.6	110.2	712.1	111657.3

5.4 CPL Gate-Delay Modeling and Optimization

Figure 5.9: A CPL Cell with a Drive and Load Gate [1]

The essentials of an XOR gate implemented in the third logic style to which the new delay optimization method will be applied are outlined in Figure 5.9, where a CPL cell gate with its associated drive and load gates is shown. A cross-coupled latch is usually used at the output of CPL circuits in order that short-circuit energy dissipation might be reduced and to pull up the output voltage. The PMOS latch transistors are generally ones of minimum width, to minimize interference in circuit operation (One branch speeds up and the other slows down when the PMOS size is increased). The so-called gate and source drives (Figure 5.9) will most often have differing structural details, and so they are given distinct names to reflect this fact (*S-drive*, *G-drive*). The critical delay path may involve the G-Drive, the S-Drive, or both of them. As we know, a NMOS transistor is slow to assert a logic '1' (see Chapter 4 Table 4.1), so the worst-case delay will always be a rising delay in a CPL gate. The rising delay will therefore be the principal object of our attentions as we seek ways to model and optimize CPL delay.

5.4.1 Currents in CPL Gates

A rising CPL output voltage is actually produced by an overlap of current contributions coming from the S-Drive (or V_{DD}) and the PMOS latch. Figure 5.10 (a) shows the

Figure 5.10: The Currents in a CPL Gate

overlapping currents I_n and I_p during a rising propagation delay. It shows that the current I_p from the PMOS device begins to charge the output capacitance when the current I_n drops from its peak and the output node almost reaches $\frac{1}{2}V_{DD}$. To simplify our delay model, we can ignore the relatively small current I_p in a rising delay model.

During a falling transition, the PMOS transistor is already turned on since it's gate is connected to the other branch which is executing a slow rising transition. Hence, the falling output is produced by the opposing currents as shown in Figure 5.10 (b).

5.4.2 Critical Paths: G-Drive Alone, or Both G- and S-Drives

Because the switching time when a G-Drive is involved is so much longer than that seen with an S-Drive (Table 5.3, Table 5.4), a critical path which incorporates both types can be approximated as one involving a G-Drive alone. In these cases the rising propagation delay from the input of drive gate to the output of cell gate is given by [1]:

$$D_{Rise} = DS_{CellRise} + (1 + S_{nr})DS_{GDriveRise} \quad , \quad (5.20)$$

with $DS_{GDriveRise}$ being the drive gate step-delay, which is

$$DS_{GDriveRise} = \frac{v_{pr}^Y GDpr}{W_{GDp}} (m_n W_n Cg_{nr} + CD_{Rise}) \quad (5.21)$$

and $DS_{CellRise}$ is the cell gate step delay:

Figure 5.11: A CPL XOR Gate with G-Drive Gate Under Test

$$DS_{CellRise} = \frac{v_{nr} Y_{nr}}{W_n} (q_{o_n} W_n C_{d_{nr}} + W_p C_{g_{pr}} + W_p C_{d_{pr}} + CL_{Rise}) \quad (5.22)$$

Figure 5.11 outlines the parts of interest of a CPL XOR gate under test. Both drive and load gates are inverters having PMOS/NMOS widths $5\mu\text{m}/3\mu\text{m}$. The dashed line follows the worst-case path from the drive-gate input up to the output of the cell gate. The rising propagation delays predicted by this model and by SPICE simulation are plotted in Figure 5.12. The close correspondence between the two is apparent.

Shams found the optimal size for a minimum worst-case rising delay is [1]:

$$W_n = \sqrt{\left(\frac{v_{nr} Y_{nr}}{v_{pr} Y_{Dpr} (1 + S_{nr})} \right) \left(\frac{W_p (C_{d_{pr}} + C_{g_{pr}}) + CL_{Rise}}{m_n C_{g_{nr}}} \right) W_{GDp}} \quad (5.23)$$

Figure 5.12: Model/Simulation predictions for a CPL XOR/XNOR Gate

In Table 5.3 are compared the performances of three CPL XOR circuits. It is clear that the rising delay is always the worst of the two - the falling delay is only some 50% of the rising one. Compared to the minimum size circuit, the optimally-sized circuit shows 30% less delay time. Although it dissipates 30% more power, its PDP.(power-rising-delay product) is 10% less. Furthermore, the optimal one is 20% faster than the larger-size circuit, uses 40% less power, has PDP that is 50% less, and also has a smaller chip area. These results favor the model [1] and demonstrate its suitability as a CPL design tool.

TABLE 5.3: Performance Comparison of CPL XORs with G-Drive

Size (μm) W_p/W_n	Rising Delay(ps)	Falling Delay(ps)	Ave. Delay(ps)	Power(μw)	PDP(Rising)(psXuw)
Minimum Size(0.5/1)	100.6	50.5	75.5	138.2	13902.9
Larger Size(0.5/10)	90.2	43.7	66.9	296.7	26762.3
Optimal Size(0.5/3.7)	70.2	36.9	53.6	180.2	12650

5.4.3 Critical Paths: S-Drive Alone

Figure 5.13: The Input Slope Effect in a CPL gate involving S-Drive

In this test case, the source end of the NMOS chain making up the critical path is connected to the output of an S-Drive, with the NMOS gates in the chain having been previously turned on by a G-Drive (see Figure 5.9). This structure differs completely from the ones studied previously, namely C-CMOS, DCVSL, and CPL-with-G-Drive. In the S-

Drive CPL test, the NMOS chains act only to transport signals and never as the originators of logic levels. Figure 5.13 clearly shows that the rising propagation delay is almost unaffected by the input signal slope.

The total input and output node capacitances CD and CL (which do not include capacitances related to the cell gate) are given by:

$$CD_{Rise} = CI + q_o D_n \cdot W_{Dn} \cdot Cd_{nr} + q_o D_p \cdot W_{Dp} \cdot Cd_{pr} \quad (5.24)$$

$$CL_{Rise} = CO + m_{Ln} \cdot W_{Ln} \cdot Cd_{nr} + m_{Lp} \cdot W_{Lp} \cdot Cd_{pr} \quad (5.25)$$

where q_o indicates the number of branches connected to the output node and q_i indicates the number of source terminals that are connected to a single output from an S-Drive gate. In this case, the signals at the gates of the chain NMOS transistors have been turned on with CD and CL already discharged to ground before a rising transition. During a $0 \rightarrow 1$ transition, CD is charged to one-half V_{DD} through the S-Drive, while CL is charged to one-half V_{DD} through the S-Drive and the NMOS chain. The propagation delay from the input of the drive gate to the output of the cell gate is approximately

$$D_{Rise} = DS_{CellRise} + DS_{SDriveRise} \quad , \quad (5.26)$$

where $DS_{CellRise}$ is the cell gate step delay:

$$DS_{CellRise} = \left(\frac{v_{nr} Y_{nr}}{W_n} + \frac{v_{pr} Y_{SDpr}}{W_{SDp}} \right) (q_o n W_n Cd_{nr} + W_p Cg_{pr} + W_p Cd_{pr} + CL_{Rise}) \quad (5.27)$$

$DS_{SDriveRise}$ is the drive gate step delay:

$$DS_{SDriveRise} = \frac{v_{pr} Y_{SDpr}}{W_{SDp}} \cdot ((q_i n + 2(n_n - 1)) \cdot W_n Cd_{nr} + CD_{Rise}) \quad (5.28)$$

The test arrangement of Figure 5.11 was modified for use in verifying our delay models. The input nodes at the NMOS sources were connected to an S-Drive gate instead of to V_{DD} or ground, while the other input nodes (NMOS gates) were connected to V_{DD} or

Figure 5.14: Model/Simulation predictions for a CPL XOR/XNOR Gate Involving S-Drive

ground, as appropriate. The plots of Figure 5.14 show the rising propagation delays predicted by this model and with SPICE. There is a clear similarity.

The optimal delay-minimizing W_n was

$$W_n = \sqrt{\left(\frac{v_{nr} Y_{nr}}{v_{pr} Y_{SDpr}} \right) \left(\frac{W_p (Cd_{pr} + Cg_{pr}) + CL_{Rise}}{(q_i^n + 2(n_n - 1)) Cd_{nr} + qo_n Cd_{nr}} \right) W_{SDp}} \quad (5.29)$$

The performances of three XORs were compared - Table 5.4 lists the results. The optimized circuit compares well with both the minimum- and larger-size circuits; in the former case it proved to be 30% faster while using about 30% more power (but with 10% lower PDP figure). In the second comparison, that with the larger-sized circuit, the optimal one again showed some 30% less delay, but this time it also used 30% less power, had a PDP only half as large, and smaller chip area.

This comparison was the fourth one, involving popular logic styles, to demonstrate the efficacy of our time-saving analytic delay model

TABLE 5.4: Performances Comparison of CPL XORs with S-Drive

Size (μm) W_p/W_n	Rising Delay(ps)	Falling Delay(ps)	Ave. Delay(ps)	Power(uw)	PDP(Rising)(psXuw)
Minimum Size(0.5/1)	95.9	48.6	72.3	138.6	13291.7
Larger Size(0.5/10)	80	48.7	64.3	298.2	23856
Optimal Size(0.5/3.8)	60.2	37.6	48.9	190	11438

5.5 Conclusion

In this chapter, it was demonstrated that some simple and judicious approximations could lead to a powerful design tool having the predictive power of time-consuming and expensive simulation where signal delay times are concerned. The model led to computing transistor sizes which offered time-delay performance improvements very nearly equal to ones resulting from simulation. That the results are optimal or nearly so is evidenced by close agreement with the results of detailed numerical simulation.

Delay Optimization of CMOS Circuit in Mixed Logic-Style

6.1 Introduction

One method of enhancing the performance of digital CMOS circuits is to use combinations of logic styles such as C-CMOS, DCVSL, and CPL [1]. The idea behind combining these logic forms is to realize the best properties of each in the resulting application - the superior current-drive capabilities and noise margins of C-CMOS, complex gate possibilities of DCVSL and the speed and low power consumption of CPL.

This chapter demonstrates how expressions [1] can be used to estimate and optimize the delay in a block circuit. The first section is devoted to delay estimation and optimization of C-CMOS logic blocks. Then one section is devoted to delay estimation and optimization of blocks in mixed logic styles. Another section is devoted to address the problems in a CPL chain. Finally, the chapter concludes with some remarks.

6.2 Delay Estimation and Optimization in C-CMOS Blocks

Consider a number of C-CMOS gates interconnected as shown outlined in Figure 6.1. The worst case paths are traced with a dashed line in each figure. In the previous chapter, we explored expressions for such propagation delays:

Figure 6.1: Examples of Critical Path in C-CMOS Block

$$D_i = DS_i + S \cdot DS_{i-1} \quad (6.1)$$

DS_i is the step delay of the present stage, and DS_{i-1} that of the previous stage. The propagation delays of each of the gate combinations were both simulated and calculated, with results that are in close agreement. In Table 6.1 is presented a list of these results - falling delays, rising delays and average delays for the circuits of Figure 6.1. The simulation/calculation delay differences are listed in the final column. The largest of these is around 7%. The data of this table prove that the method [1] of C-CMOS delay estimation produces results nearly the same as those of simulation. (There is a small insignificant caveat however: the delay modeling of C-CMOS blocks was done under the assumption of TSMC 0.13 μ m technology, while the delay optimization assumed TSMC 0.18 μ m technology. The reason for the difference is that while this research is directed first at advanced design technology (0.13 μ m), the fabrication facilities available through CMC are of the 0.18 μ m type.)

By applying the optimal sizing formulas (Equation (5.12), (5.13)) of the previous chapter, we can find optimal element sizes for the circuits of Figure 6.1. Table 6.2 lists the optimal gate sizes and total areas for TSMC 0.18 μ m technology. Note that the optimal size is the one needed to obtain the average delay.

The performances of minimum, larger and optimal size devices are compared in Table 6.3. The minimum and larger sizes of NAND and NOR gates have been 'normalized' to that of a macro-inverter having a PMOS/NMOS size ratio list in Table 6.3. For example, the minimum PMOS to NMOS size ratio in a 3-input NOR gate is 3 μ m/0.5 μ m, and the larger 30 μ m/5 μ m.

TABLE 6.1: C-CMOS Block Delay Modeling ($0.13\mu\text{m}$)

	Delay(ps)	INV	NAND2	NOR2	NAND3	NOR3	Total	Error(%)
SPICE (a)	Rising	12.66	23.84	40.51			87.83	
	Falling	14.1	34.66	33.95			71.89	
	Ave.	13.38	29.25	37.23			79.86	
MODEL (a)	Rising	14.04	21.96	42.47			92.05	4.8
	Falling	16.24	35.54	32.75			70.95	1.31
	Ave.	15.14	28.75	37.61			81.5	2.05
SPICE (b)	Rising	12.66	23.84	42.7	34.11	69.89	222.88	
	Falling	14.6	34.56	35.6	63.07	53.68	161.83	
	Ave.	13.63	29.2	39.15	48.59	61.79	192.36	
MODEL (b)	Rising	14.04	21.96	43.5	39.13	68.03	221.01	0.84
	Falling	16.24	35.54	35.6	59.9	60.8	173.73	7.35
	Ave.	15.14	28.75	39.55	49.52	64.42	197.37	2.61

TABLE 6.2: Optimal Sizes in C-CMOS Blocks ($0.18\mu\text{m}$)

Opt. (Wp/Wn)	INV (μm)	NAND2 (μm)	NOR2 (μm)	NAND3 (μm)	NOR3 (μm)	Total (WxL)
(a)	5.2/3.7	5.0/5.3	7.3/3.1	N/A	N/A	50.3L
(b)	5.4/3.8	5.4/5.7	8.4/3.7	4.6/6.4	8.8/2.7	123.1L

In the case of the circuit of block (a), the optimal circuit has 32% less delay than the minimal-size implementation, but has 185% more power dissipation, 400% more area, and 92% higher PDP. The optimal circuit has 3% less delay than the larger-size implementation, but only has 56% power dissipation, 50% area and 54% PDP. In block circuit (b), the corresponding figures are: 25% less delay than the minimal-size circuit, 237% more

TABLE 6.3: Performance Comparison of C-CMOS Blocks (0.18 μ m)

Circuit Block	Size (μ m)	Falling (ps)	Rising (ps)	Ave. (ps)	Power (μ W)	Area (wxLmin)	PDP (psX μ W)
(a)	Min. (1/0.5)	159.9	175.2	167.5	85.23	10.5L	14276
	Larg. (10/5)	98.1	133.2	115.6	437.9	105L	50621
	Opt.	109	116.5	112.8	243.3	50.3L	27444
(b)	Min. (1/0.5)	257.3	368.6	312.9	136.4	28.5L	42680
	Larg. (5/3)	192.5	309	250.8	930.7	285L	233420
	Opt.	213.2	260.7	236.9	459.6	123.1L	108879

power dissipation, 332% more area and 155% higher PDP, and 5.5% less delay than the larger-sized circuit, 49.4% of its power dissipation, 43.2% of its area and 47% of its PDP figure. This comparison reveals that this optimization method provides significant gain at lowered delay, but with worsened power, area and PDP figures.

6.3 Delay Optimization in Mixed Logic Styles

Digital CMOS performance enhancement may be achieved with designs that make use of a combination of logic styles, like C-CMOS, DCVSL, and CPL.

6.3.1 Mixed Logic Delay Optimization

Table 6.4 summarizes the optimal transistor sizing formulas that were presented in Chapter 5 for the CMOS logic styles mentioned above. We have written corresponding MATLAB functions; these follow a common format given in [1]:

TABLE 6.4: Optimal Transistor Sizing Formulas

C-CMOS [1] (Av. Delay)	$W_p = \sqrt{\frac{v_{pr} \cdot Y_{pr}(1+S_{nf})(qo_n \cdot W_n \cdot Cd_{nr} + CL_{Rise})}{m_p \cdot Cg_{pf} \frac{v_{nf}}{W_{Dn}} Y_{Dnf}(1+S_{pr}) + m_p \cdot Cg_{pr} \frac{v_{pr}}{W_{Dp}} Y_{Dpr}(1+S_{nf}) + qo_p \cdot Cd_{pf} \frac{v_{nf}}{W_n} Y_{nf}(1+S_{pr})}}$
	$W_n = \sqrt{\frac{v_{nf} \cdot Y_{nf}(1+S_{pr})(qo_p \cdot W_p \cdot Cd_{pf} + CL_{Rise})}{m_n \cdot Cg_{nr} \frac{v_{pr}}{W_{Dp}} Y_{Dpr}(1+S_{nf}) + m_n \cdot Cg_{nf} \frac{v_{nf}}{W_{Dn}} Y_{Dnf}(1+S_{pr}) + qo_n \cdot Cd_{nr} \frac{v_{pr}}{W_p} Y_{nf}(1+S_{nf})}}$
DCVSL [1]	$W_n = \sqrt{\frac{\left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right)^2 (1+S_{pr}) \left(W_p (Cd_{pf} + Cg_{pf}) + CL_{Fall} + 0.5 \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right) W_p Cd_{nf}\right)}{(1+S_{nf}) \frac{Y_{Dpr} m_n Cg_{nr}}{W_{Dp}} + \frac{qo_n Cd_{nr}}{W_p}} + \frac{1}{2} \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right) W_p}$
	$\frac{1}{\Gamma} = \frac{W_n}{W_p} = \sqrt{(1+S_{pr}) \left(\frac{1}{2} \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right)^2 \frac{CL_{Fall} + qo_n W_n Cd_{nf}}{CL_{Rise} + qo_n W_n Cd_{nr}} + \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right) \frac{W_n (Cd_{pf} + Cg_{pf})}{CL_{rise} + qo_n W_n Cd_{nr}}\right) + \frac{1}{2} \left(\frac{v_{nf} Y_{nf}}{v_{pr}}\right)}$
CPL [1] G-Drive	$W_p = W_{min}$
	$W_n = \sqrt{\left(\frac{v_{nr} Y_{nr}}{v_{pr} Y_{Dpr}(1+S_{nr})}\right) \left(\frac{W_p (Cd_{pr} + Cg_{pr}) + CL_{Rise}}{m_n Cg_{nr}}\right) W_{GDP}}$
CPL S-Drive	$W_p = W_{min}$
	$W_n = \sqrt{\left(\frac{v_{nr} Y_{nr}}{v_{pr} Y_{SDpr}}\right) \left(\frac{W_p (Cd_{pr} + Cg_{pr}) + CL_{Rise}}{(q_i n + 2(n-1)) Cd_{nr} + qo_n Cd_{nr}}\right) W_{SDp}}$

$$(W_p, W_n) = LogicStyleOpt \left(VDD, \right. \\
\left. \left\{ \langle DriveInfo \rangle W_{Dp}, m_{Dp}, n_{Dp}, q_i^{Dp}, q_o^{Dp}, W_{Dn}, m_{Dn}, n_{Dn}, q_i^{Dn}, q_o^{Dn} \right\}, \right. \\
\left. \left\{ \langle CellInfo \rangle W_p, m_p, n_p, q_i^p, q_o^p, W_n, m_n, n_n, q_i^n, q_o^n \right\}, \right. \\
\left. \left\{ \langle LoadInfo \rangle W_{Lp}, m_{Lp}, n_{Lp}, q_i^{Lp}, q_o^{Lp}, W_{Ln}, m_{Ln}, n_{Ln}, q_i^{Ln}, q_o^{Ln} \right\} \right) \quad (6.2)$$

where *LogicStyle* is *CMOS* for a C-CMOS implementation, *DCVSL* for a DCVSL implementation, *CPLS* for a CPL implementation with S-Drive, *CPLG* for a CPL

implementation with G-Drive. And V_{DD} must be specified because some of the parameters (like v , Y , S and capacitances) are dependent upon it. The optimization process for transistor sizes is an iterative one, one that starts by assuming minimum values for these.

An example of using these sizing formulas to perform the delay optimizations of three mixed-logic 3-bit Full Adders takes up the remainder of this section.

6.3.2 Specific Devices: 3-bit Full Adders

The C-CMOS, DCVSL and CPL full adders (FAs) of Figure 6.2-6.3 were selected as examples for our study. In each case, the critical path lies in a carry circuit which is characterized by the presence of two series-connected NMOS transistors in a pull-down network (which implements the carry-out logic $C_o = A \cdot B + (A + B)C_i$) and so a realistic comparison is possible. The adders' sum circuits do not affect their speeds and are thereby designed only with minimum size in mind. Twenty-seven 3-bit FA constructs using various logic style combinations are optimized, simulated and compared below.

The C-CMOS FA circuit (Figure 6.2 (a)) is the simplest of the circuits. The DCVSL full adder of Figure 6.2 (b) is a network of NMOS transistors connected as a 3-input XOR gate with dynamic latch. The addition logic of the CPL FA of Figure 6.3 is executed by NMOS pass-transistor chains and level restoration by cross-coupled PMOS devices.

Figure 6.2: Schematics of a C-CMOS Full Adder and CPL Full Adder

Figure 6.3: Schematic of a CPL Full Adder

As in the previous chapter, we will compare delay, power consumption and the power-delay product (PDP) of the adders, using the test arrangement of Figure 6.4, where three 1-bit adders are connected in cascade to make a 3-bit adder. Since the transition times of the input waveform significantly affect delay and short-circuit dissipation, input inverters were inserted to condition the input onset and decay times, to make them constant from one test to the next. Output inverters were inserted to provide a constant load for each device, further facilitating comparison. In order to evaluate the power dissipation in the data paths, such as CPL circuits, a separate power supply was used and counted for the input conditioning inverters and tested FAs. The power dissipation was averaged over a period of 20ns in each of 5 worst-case switching activities. The propagation delay was taken as the delay between the first FA's 'CI' to the appearance of a valid 'CO' from the last FA.

Figure 6.4: A Test Bench for 3-bit FAs

TABLE 6.5: Optimal Transistor Sizes of 3-bit FAs

Logic Styles	1st FA (Wp/Wn)		2nd FA (Wp/Wn)		3rd FA (Wp/Wn)	
	Co (μm)	Inv (μm)	Co (μm)	Inv (μm)	Co (μm)	Inv (μm)
CCC	11.5/7.3	11.4/7.9	18.8/12.2	14.1/10	17.2/11.3	6.8/5
PBPBPB	0.5/9.3	4.9/3.9	0.5/9.2	4.8/3.8	0.5/8.5	3.7/2.7
PPP	0.5/11.2	N/A	0.5/9.6	N/A	0.5/8.4	N/A
DBDBDB	13/15.5	12.4/9.6	20.7/25.8	14.8/11.5	17.6/22.5	8.9/6.6
DDD	12.7/15.5	N/A	16.5/21.3	N/A	11.9/15.7	N/A
CPBC	9/5.8	6.9/5.2	0.5/10	3.9/2.8	5.5/3.6	3.9/2.7
CPC	9.7/6.2	8.9/6.6	0.5/10.2	N/A	4.5/3	3.5/2.5
PBCPB	0.5/10.6	5.4/4	9.1/5.9	5.7/4.3	0.5/10.3	2.8/2

A MATLAB circuit netlist appropriate to this test was made using the optimization functions developed in the previous section. Table 6.5 shows a partial list of optimal transistor sizes. The upper-case letter 'C' stands for C-CMOS, 'P' for CPL, 'D' for DCVSL, and 'B' represents an output inverter. For example, 'PBCPB' represents a combination of CPL, C-CMOS and CPL FAs with two buffers: $FA_{CPL} + \text{Buffer} + FA_{C-CMOS} + FA_{CPL} + \text{Buffer}$. As in past sections, we compared the results of our optimization methods with commonly-used design approaches, ones using minimum-sized and one using larger-sized transistors - the results for five FA configurations are summarized in Figure 6.5. The minimum size ratio is same as the sum circuits ($2\mu\text{m}/1\mu\text{m}$), and the larger size is the largest size in each style of Table 6.5 for three identical FAs. The not-unexpected outcome of the tests is that the circuits employing formula-optimized transistor sizes are faster than the others, for all of the styles considered.

Figure 6.5: Delay Comparison of Optimized FAs

The next step was to find out which combination of styles gave the best results. To this end, some twenty-seven configurations of optimized adders were simulated assuming the same test environment used before (Figure 6.4). Their performances are compared in terms of propagation delay, power dissipation, and power-delay product and the results were graphically summarized in Figure 6.6, 6.7, 6.8.

Figure 6.6 shows the propagation delays. It can be seen that the pure CCC (all C-CMOS) configuration is slower than most of the mixed-logic ones, the difference being as much as 15% in some cases (PCP, DCP, DBCP).

Figure 6.6: Delay Comparison of FAs in Mixed Styles

Figure 6.7 shows the power dissipations in the same formats. A number of the mixed logic arrangements are superior to C-CMOS in this regard, with the CPC ($FA_{C-CMOS} + FA_{CPL} + FA_{C-CMOS}$) device being the best.

Figure 6.7: Comparison of Power Consumption

The power-delay products are depicted in Figure 6.8. Some of the mixed logic styles are better than pure C-CMOS, the ones built with CPL components having the best performance.

Figure 6.8: Comparison of PDP in Mixed Logic Styles

These comparisons indicate that some circuits which combine CPL and C-CMOS components are better than C-CMOS-only ones where delay, power dissipation, and PDP are concerned.

Since the main goal of this work is to identify situations in which propagation delay is minimized (and hence to demonstrate the closed-formula prediction method) we will next narrow our investigation to finding which combination in fact has the best delay. (The physical construction of a test chip to verify our findings will be described in the next chapter.)

6.4 Delay Estimation and Optimization for CPL Chains

An important consideration is the delay associated with the chain of pass-transistors which gives CPL its name. These chains unfortunately suffer from poor noise margins and poor current drive abilities, problems that must be dealt with simultaneously with delay reduction.

6.4.1 Delay Modeling of CPL Chains

We studied four methods for attacking the noise and current-drive problems in the last chapter, namely the use of a level restorer, insertion of a latch alone, insertion of a buffer alone, and finally the use of a latch/buffer combination. The optimal transistor sizes appropriate to each of them were determined by applying the models described in that chapter.

Figure 6.9: Schematic and Currents of Level Restorer

Level Restorer : A commonly-used way of solving the voltage drop and drive problems is to place a *level-restorer* in each branch (see Figure 6.9A) of the differential CPL chain. The gate of the PMOS device is connected to an inverter output, its drain to the inverter's input and its source to V_{DD} . As described before, the worst-case delay here is the rising delay on the path from the input of the S-Drive buffer to the output of the CPL chain. Usually, the current I_p which feeds back from the inverter output through to the PMOS device charges the node capacitance after one inverter delay (Figure 6.9B) after

Figure 6.10: Model/Simulation predictions of Rising Delay for Level Restorer

which it causes the output to be pulled up from $V_{DD} - V_T$ to V_{DD} , so the *level-restorer* doesn't help to reduce the rising delay.

Furthermore, the diffusion capacitance of the PMOS transistor is added to the total load capacitance, also to the detriment of the CPL device's speed. To lessen these effects, the size of the PMOS device should be kept to a minimum. The equations which model the level-restorer's rising propagation time, on the path between the S-Drive buffer's input through to the output of output buffer are:

$$Delay_{Rise} = (1 + S_{np})DS_{CellRise} + DS_{LoadFall} + DS_{DriveRise} \quad (6.3)$$

$$DS_{CellRise} = \left(\frac{v_{pr}}{W_{Dp}} + \frac{v_{nr} Y_{nr}}{W_n} \right) (qo_n W_n C_{d_{nr}} + W_p C_{d_{pr}} + CL_{Rise}) \quad (6.4)$$

$$DS_{DriveRise} = \frac{v_{pr}}{W_{Dp}} (qi_n W_n C_{d_{nr}} + CS_{Rise}) \quad (6.5)$$

$$DS_{LoadFall} = \frac{v_{nf}}{W_{Ln}} CO_{Fall} \quad (6.6)$$

Where $DS_{CellRise}$, and $DS_{DriveRise}$ are the rising step delays of the cell gate and the drive gate and $DS_{LoadFall}$ is the falling step delay of the output buffer. Figure 6.10 shows the rising delay predicted by this model and that predicted using SPICE simulation.

Latch Insertion . Cross-coupled PMOS transistors may be used to get a level restoration effect without introducing inverter delay (Figure 6.11a). The gate of a PMOS device

Figure 6.11: Schematic and Currents of Latch Insertion

is connected to the output of the other branch, its drain to the output of the CPL chain of this branch and its source to V_{DD} . This structure will still suffer from a poor current drive capacity when it is cascade connected. As in the previous case, the worst-case delay is the rising delay. Figure 6.11b shows the ‘overlapping’ currents: I_n from the S-Drive through the CPL chain, and I_p from the PMOS latch which is controlled by the other branch. The current I_p begins to charge the output capacitance soon after I_n begins falling. The latch therefore helps increase the output voltage but does little to improve the rise time.

To simplify the delay model, one could ignore the effect of I_p and set the size of the PMOS transistors to a minimum. The equations used to model the rising propagation time on the path joining S-Drive buffer’s input to the CPL chain’s output were:

$$Delay_{Rise} = DS_{CellRise} + DS_{DriveRise} \quad (6.7)$$

Figure 6.12: Model/Simulation predictions of Rising Delay for Latch Insertion

Figure 6.13: Schematic of Buffer Insertion

$$DS_{CellRise} = \left(\frac{v_{pr}}{W_{Dp}} + \frac{v_{nr} Y_{nr}}{W_n} \right) \quad (6.8)$$

$$\times (q_{o_n} W_n C_{d_{nr}} + W_p (C_{d_{pr}} + C_{g_{pr}}) + CL_{Rise})$$

$$DS_{DriveRise} = \frac{v_{pr}}{W_{Dp}} (q_{i_n} W_n C_{d_{nr}} + CS_{Rise}) \quad (6.9)$$

Figure 6.12 shows the rising delay predictions of SPICE and our model.

Buffer Insertion . The last method, buffer insertion, is the most common way of dealing with both long delay times and poor current drive (Figure 6.13). An inverting buffer is connected directly to each of the differential branches in hopes of lessening output voltage drop while increasing the output current drive. However, this approach suffers from low speed and large power dissipation because of the voltage $(V_{DD} - V_T)$ at the buffer's input is in a range where it partially turns on both the PMOS and NMOS of the buffer simultaneously. As mentioned before, the rising delay is always the worst-case delay in this case. The equations which were used to predict this delay along the path joining the drive buffer's input to the output buffer's output are:

$$Delay_{Rise} = (1 + S_{np}) DS_{CellRise} + DS_{LoadFall} + DS_{DriveRise} \quad (6.10)$$

$$DS_{CellRise} = \left(\frac{v_{pr}}{W_{Dp}} + \frac{v_{nr} Y_{nr}}{W_n} \right) (q_{o_n} W_n C_{d_{nr}} + CL_{Rise}) \quad (6.11)$$

$$DS_{DriveRise} = \frac{v_{pr}}{W_{Dp}} (q_{i_n} W_n C_{d_{nr}} + CS_{Rise}) \quad (6.12)$$

Figure 6.14: Model/Simulation predictions of Rising Delay for Buffer Insertion

$$DS_{LoadFall} = \frac{v_{nf}}{W_{Ln}} CO_{Fall} \quad (6.13)$$

In Figure 6.14 are compared the delays determined by SPICE simulation to those predicted by this model.

Combined Latch and Buffer Insertion . This idea (Figure 6.15) uses both a latch and a buffer to solve the voltage drop and current drive problems. For reasons explained previously, the worst-case delay in this case will be the rising delay. This structure combines the advantages of the latch and buffer insertion ideas, giving higher speed and higher current drive capability. The equations here can be used to model this rising delay along the path from the S-Drive input to the load buffer output:

$$Delay_{Rise} = (1 + S_{np}) DS_{CellRise} + DS_{LoadFall} + DS_{DriveRise} \quad (6.14)$$

$$DS_{CellRise} = \left(\frac{v_{pr}}{W_{Dp}} + \frac{v_{nr} Y_{nr}}{W_n} \right) \times (qo_n W_n C_{d_{nr}} + W_p (C_{d_{pr}} + C_{g_{pr}}) + CL_{Rise}) \quad (6.15)$$

Figure 6.15: Schematic of Latch and Buffer Insertion

Figure 6.16: Model/Simulation predictions of Rising Delay for Latch and Buffer Insertion

$$DS_{DriveRise} = \frac{v_{pr}}{W_{Dp}} (q_i n W_n C_{d_{nr}} + C_{S_{Rise}}) \quad (6.16)$$

$$DS_{LoadFall} = \frac{v_{nf}}{W_{Ln}} C_{O_{Fall}} \quad (6.17)$$

In Figure 6.16 the rising delays predicted by this model are compared to the SPICE simulation results.

6.4.2 Delay Optimization of CPL Chains

Delay Optimization. We used the formulae of previous sections to derive optimal transistor sizes for the four level-restoration/current-drive improvement methods. Our findings are listed in Table 6.6. The calculated values lie in the vicinity of the minima of the previous figures.

TABLE 6.6: Optimal Transistor Sizes

	W_p/W_n (Model)	W_p/W_n (SPICE)
Level Restore	$\frac{0.5\mu m}{4.2\mu m}$	$\frac{0.5\mu m}{5\mu m}$
Latch	$\frac{2.2\mu m}{4.4\mu m}$	$\frac{2.2\mu m}{4.4\mu m}$
Buffer	$\frac{0}{4.1\mu m}$	$\frac{0}{5\mu m}$
L & B	$\frac{0.5\mu m}{4.4\mu m}$	$\frac{0.5\mu m}{4.4\mu m}$

Figure 6.17: Optimal Buffer Insertion

Optimum Buffer Insertion. Long chain CPL circuits are important because they are often used in circuits such as adders or deep multiplexers. The most common approach for dealing with long delays in these devices is to insert into the CPL chain every n pass devices a buffer and/or latch made of transistors having the widths determined above. We used simulation to find the optimum value of n : We studied the effect of inserting a latch, buffer, level-restorer or latch-with-buffer into a 100 transistor CPL chain at every n^{th} point for a series of n values ($n = 1, 2, \dots, 10$). TSMC $0.18\mu\text{m}$ technology was assumed. Each situation was optimized using the formulas from previous sections in a simulation. The results obtained are summarized in Figure 6.17, where ‘p’ indicates a pass-transistor, ‘l’ a latch, ‘b’ a buffer and ‘rb’ indicates a level-restorer. The simulation showed that the optimal number (n) of transistors by inserting buffers in a CPL chain is about 4 as dashed cycles in Figure 6.17- a sequence of 4-bit CPL chains, and the best solution in this long CPL chain example to be a latch with buffer. We will apply these experimental results to a practical application design in the next chapter.

6.5 Conclusion

In this chapter, we presented approximate delay models and optimal transistor sizing formulas for minimizing the delay in circuits block C-CMOS-only and mixed logic styles. Twenty-seven 3-bit ripple carry adder combinations were used to generate the data needed to evaluate the merits of mixed logic design. The results reveal that some of the circuits which combined C-CMOS and CPL methods have better performance than pure C-CMOS ones. Four methods to solve the voltage drop and poor current drive capability in CPL circuit were investigated. An exploration of the effects of inserting buffers into long CPL chains showed the optimal number of CPL pass-transistors between insertions to be about 4.

Applications to Fast Adder Circuits Design

7.1 Introduction

The binary adder is a realistic platform for evaluating a CCMOS/CPL admixture because its importance in high-performance processors and DSP devices - it is often the speed limiting element in these settings. The need for these devices becomes more and more acute as the frequency of 64-bit processors approaches 1 Ghz.

In this chapter, the simulation, optimization and fabrication of two mixed-logic adder chips is described (TSMC 0.18 μm). The design and construction of a 64-bit high-speed static adder was also carried out to provide an illustration of how our closed-form optimization method is used in a practical setting.

7.2 Fast Addition

Addition is the most used operation and speed-limiting element, hence, optimization of the adder is the very important. This optimization can be processed either at logic or circuit level. Typical logic-level optimizations try to rearrange the equations so that a faster circuit is obtained such as the carry lookahead adder (CLA). Circuit optimizations, on the other hand, manipulate transistor sizes and circuit topology to optimize the speed. In this

section, we will briefly review the basic additions in logic level.

7.2.1 The Ripple-Carry Adder

Figure 7.1: A 4-bit Ripple-Carry Adder

An N -bit ripple-carry adder (RCA) [2] is formed by cascading N full adders (FA), feeding the carry-out C_i of FA_i directly into the carry-in input of FA_{i+1} as shown in Figure 7.1. The carry bits “ripple” from left to right. The carry-in C_0 to full adder FA_1 is hardwired to 0. The sum output is the $(N+1)$ -bit number $Sum = (S_n, S_{n-1}, \dots, S_0)$, where S_n equals C_n .

In the worst case operation of the ripple-carry adder, a carry generated at the least significant bit position propagates all the way to the most significant bit position. Hence, the total delay is proportional to the number of bits.

$$t_{RCA} = N \cdot t_{Carry} \quad (7.1)$$

7.2.2 The Carry-Bypass Adder

Figure 7.2: A 4-bit Carry-Bypass Adder

A method to speed up the worst case operation of the adder is to use the carry-bypass adder (CBA) as shown in Figure 7.2 [3]. In an N -bits adder, if the value of A_i and B_i are such that all propagate signal P_i ($P_i = A_i \oplus B_i$) are high, an incoming carry C_0 is forwarded immediately to the next block through a multiplexer, if this is not the case, the

carry is obtained by way of the normal route.

The propagation time for an N -bit adder containing number of equal-length (M -bits like 4-bit in Figure 7.2) adder stages is:

$$t_{CBA} = (M-1) \cdot t_{carry} + \frac{N}{M} t_{mux} + M \cdot t_p \quad (7.2)$$

Where t_{carry} is the delay of the carry through a single full adder cell, and t_{mux} is the gate delay of the multiplexer, and t_p is the delay to generate the propagate signal (P_i).

7.2.3 The Carry-Select Adder

Figure 7.3: A Carry-Select Adder

An approach to increase the speed of a parallel adder that expends area in favor of speed is to use a carry-select adder (CSA) as shown in Figure 7.3 [3]. Usually, two ripple-carry adder structures (any adder structure may be used) are built, one with a zero carry-in and the other with a one carry-in. The previous carry then selects the appropriate sum using a multiplexer or tristate adder gates. The stage carries and the previous carry are gated to form the carry for the succeeding stage.

Consider an N -bit full carry-select adder constructed by chaining a number of equal-length (M -bits) adder stages, the worst case propagation delay is:

$$t_{CSA} = M \cdot t_{carry} + \left(\frac{N}{M} - 1\right) t_{mux} \quad (7.3)$$

Where t_{carry} is the delay of the carry through a single full adder cell, and t_{mux} is the gate delay of the multiplexer.

7.2.4 The Carry-Lookahead Adder

The rippling effect of the carry is still present in CSA and CBA, the carry-lookahead adder (CLA) offers a possible way to overcome it. The carry-lookahead addition avoids this N times delay by accelerating the computation of carries using a treelike circuit. The carry of the i th stage, may be expressed as

$$C_i = G_i + P_i \cdot C_{i-1} \quad (7.4)$$

Where the generate signal " G_i " and the propagate signal " P_i " are given by

$$G_i = A_i \cdot B_i \quad (7.5)$$

$$P_i = A_i + B_i \quad (7.6)$$

Expanding this yields:

$$C_i = G_i + P_i \cdot G_{i-1} + P_i P_{i-1} G_{i-2} + \dots + P_i \dots P_1 C_0 \quad (7.7)$$

The sum S_i is generated by

$$S_i = C_{i-1} \oplus A_i \oplus B_i \quad (7.8)$$

For an N -bit adder, each bit carry and sum output are independent of the previous bit, and the addition time should be independent of the number of bits. But in low-level hardware implementation, some hidden dependences, such as large fan-in and fan-out, slow down the addition with the number of bit increasing.

7.3 A Ripple-Carry Adder Test Chip Design

A unified delay-as-function-of-width model opens the door to effective mixed logic designs - in particular to ones involving a mixture of C-CMOS and CPL (see last chapter). Mixed logic design allows one to realize the best properties of each component in the

Figure 7.4: The Actual RCA Test Chip

Figure 7.5: Architecture of a RCA Test Chip

resulting circuit - in the case of CCMOS/CPL, the superior current-drive capabilities and noise margins of C-CMOS are combined with the speed and low power consumption of CPL.

Ripple-carry adders are the simplest and most compact adders, but their performance is limited by the fact that the carry signal must propagate asynchronously from the carry-in to the carry-out point. To make actual measurements of the time required for this carry propagation, we designed and fabricated a ripple-carry adder (RCA) test chip which embodied eight of the more-promising mixed logic implementations of the device (The architecture of the test chip is shown in Figure 7.5 and an actual photograph in Figure 7.4.)

7.3.1 A RCA Test Chip Design

Architecture. Figure 7.5 shows the architecture of a RCA Test Chip. It consists of input buffers, output buffers, a 3 to 8 decoder, tri-state input/output buffers, and eight 8-bit adders. A 3-bit binary code selects the particular RCA to be tested and routes its signals to/from the chip I/O pins via tri-state buffers (Figure 7.2). Adder-generated output signals are buffered so that the a typically-large off-chip loads may be driven without affecting adder performance. The details of the adders and the buffer circuitry (in the dashed box of Figure 7.2) will be described in the next sections.

Tri-State In/Out Buffers. The tri-state input and output buffers are shown in Figures 7.3A and 7.3B. In the input buffer, the assertion of the enable signal essentially disconnects the buffer itself from the supply rails, leaving its output floating (in a high-impedance state). This buffer's only purpose is to route input signals to individual adders, and it doesn't need a capability of supplying large load currents. Note that the buffer acts as an inverter.

Figure 7.6: Schematics of Tri-State Data In/Out Buffer

The non-inverting output buffer on the other hand must supply a large load at the [DataOutBus] point, and so a structure having a strong drive capability was used in our design, something that is ensured by there being only one transistor between the output node and each of the rails.

Figure 7.7: The Critical Path in a 8-b RCA

8-bit RCAs Design. The experimental results of Chapter 6 indicate that C-CMOS/CPL combinations display the greatest advantages in delay, power and PDP figures. We concentrated on the 8-bit RCA designs found (chapter-6) most likely (out of a dauntingly-large number of combinations). We fitted eight of them, all having the same general structure shown in Figure 7.7, onto the test chip. A C-CMOS-only device was included to supply a reference point for comparison. Three of the seven mixed-logic adders were combinations of G-Drive CPL and C-CMOS, and four were S-Drive CPL and C-CMOS combinations. Diagrams of 1-bit G-Drive and S-Drive CPL full adders (FAs), and their signal connections, are shown in Figure 7.8.

Figure 7.8: Schematics of a CPL Full Adder Involving G/S-Drive

In an RCA structure, the critical path is that from the first FA's carry-in to the eighth one's carry-out (shown by the dashed line in Figure 7.7). Since the bit signals A and B have settled before the carry-in signal arrives, the number of serially-connected transistors in a G-Drive FA (Figure 7.8A) will be effectively just one, but in an S-Drive FA (Figure 7.8B) the effective (and actual) number is two. This means that a G-Drive FA's

carry-out generation will be faster than an S-Drive FA's. Each of the mixed logic 8-bit adders was made up of a serial connection of 1-bit FAs:

S-Drive devices:

- 1). "S_PLBPLBPLBPLBPLBPLBPLB"--- latch/buffer combination inserted between each pair of FAs (8 LBs in all).
- 2). "S_PPLBPPLBPPLBPPLB"--- latch/buffer inserted every two FAs (4 LBs).
- 3). "S_PPPPLBPPPPPLB"--- latch/buffer inserted every four FAs.
- 4). "S_PPPPPPPPLB"--- latch/buffer inserted every eight FAs.

Reference Design:

- 5). "CCCCCCCC"--- the reference design: eight sequential pure C-CMOS FAs.

G-Drive devices:

- 6). "G_PLPLBPLPLBPLPLBPLB"--- latch inserted after every FA and a buffer after every other latch.
- 7). "G_PLBPLBPLBPLBPLBPLBPLB"--- a latch/buffer after each FA.
- 8). "G_PLPLPLPLPLPLPLPL"--- latch inserted after each FA.

Table 7.1 lists the layout area and the number of transistors for each of the eight 8-bit RCAs:

7.3.2 Delay Optimization

The method used to size the transistors is based on the methods of Chapter 6. These same methods were used to optimize the entire carry-in/carry-out critical path. The calculation for each logic style may be done with the MATLAB functions described earlier.

TABLE 7.1: Summary of eight 8-bit RCAs

Design	Transistor N.	Layout Area (μm^2)
1	256	17640
2	232	17040
3	220	16740
4	214	16740
5	272	17190
6	240	17040
7	256	18390
8	224	15750

7.3.3 Results and Comparison

This test chip was packaged in a 24-lead CFP (ceramic flat pack) and soldered to a Printed Circuit Board Test Fixture (PCB-TF2 from CMC). The measurement equipment used included:

- Tektronix TDS 684B Four Channel Digital Real-Time Oscilloscope (1GHz 5Gb/s)
- HP 80000 Data Generator System (1GHz)
- Keithley 2400-LV Source Meter
- Cables

The propagation delay and average power dissipation of the RCAs (including I/O buffers) were determined at five supply voltages. The propagation delay was measured between 50% levels at the input and output, as usual. The average power was measured for the whole chip with only one RCA activated. The worst-case delay and power dissipation values observed at the listed V_{DD} levels are shown in Table 7.2.

TABLE 7.2: Summary of Measurement for 8-bit RCAs

Design		1.2V	1.5V	1.8V	2.5V	3.3V
1	Delay(ns)	4.802	3.170	2.790	2.274	2.102
	Power(mw)	20.7	26.1	69.6	147.2	416.8
2	Delay(ns)	4.016	2.714	2.270	2.034	1.766
	Power(mw)	17	23.8	56.7	133.2	408.9
3	Delay(ns)	3.910	2.742	2.368	2.022	1.862
	Power(mw)	18.6	24.9	72.4	131	411.2
4	Delay(ns)	4.190	3.000	2.486	2.064	1.864
	Power(mw)	20.6	24.5	69.4	137.7	398.2
5	Delay(ns)	3.764	2.616	2.284	1.840	1.644
	Power(mw)	20.9	36.9	69.1	152.0	441.0
6	Delay(ns)	3.612	2.464	2.036	1.582	1.390
	Power(mw)	20.5	36.6	68.6	129.1	432.4
7	Delay(ns)	3.670	2.564	2.150	1.726	1.538
	Power(mw)	20.7	36.6	69.6	128.4	409.6
8	Delay(ns)	3.822	2.564	2.102	1.638	1.514
	Power(mw)	21.0	37.0	70.5	131.2	415.6

The results of simulation and measurement for each supply voltages (1.2V, 1.5V, 1.8V, 2.5V, and 3.3V) are plotted in Figure 7.9. For the most part, the measured results agree with the simulated ones - what little delay discrepancy there is may be attributed to routing capacitance. From these diagrams, a significant increase in propagation delay can be observed near $V_{DD}=1.2$ volts, a level that happens to equal the sum of the NMOS and PMOS threshold voltages. This area is generally known as the practical lower limit of logic gate operation - lowering the supply voltage any further will cause the transistors to

operate in their weak inversion (subthreshold) regions where process variations have fatal effects on logical gate functionality and speed. For supply voltages above 1.2v, the G-Drive RCAs seem to be faster than C-CMOS and S-Drive RCAs, which agree with our prediction in previous section. Design 6# (G_PLPLBPLPLBPLPLBPLB) seems to be fastest of the logic styles and design 2# (S_PPLBPPLBPPLBPPLB) seems to be fastest of the S-Drive ones. (Note there, each “P” has two series connected transistors see Figure 7.8--that means a 4-bit structure offers supper speed). The results make it clear that a C-CMOS/CPL combination is superior to pure C-CMOS..

The measured average power dissipations when operating at each of the five supply voltages are plotted in Figure 7.9. The results show that power dissipation increases more rapidly than the expected V_{DD}^2 proportionality. (This is because while the dynamic power contribution scales as V_{DD}^2 , the short circuit contribution scales faster than this). The C-CMOS RCA is seen to be the most power-hungry of the logic styles studied above, at all supply voltages. This supports the original idea that the logic mixture will inherit the advantages of its components - in this case, the low power demands of CPL.

The PDP values calculated from Table 7.2 are plotted in Figure 7.11. The PDP figure of merit is a quantitative measure of the effects of trading off power for speed, and vise-versa. This figure shows that the PDPs of all logic styles are seen to increase gradually as the supply voltage is raised.

This test chip not only verifies the advantages of mixed logic design successfully , but also demonstrates the correctness of our Chapter 6 findings on the best buffer insertion frequency for CPL chains.

Figure 7.9: Delay Comparison of Eight 8-bit RCAs (Measured/Simulated)

Figure 7.10: Power Comparison of Eight 8-bit RCAs (Measured)

Figure 7.11: PDP Comparison of Eight 8-bit RCAs (Measured)

7.4 A 64-bit High-Speed Static Adder Test Chip Design

Confidence in the new method would result if it stood up well in the face of a 'real' and significant problem, a problem like the minimization of the critical-path propagation time in a 64-bit static adder. The architecture of an adder of our design is described fully in this section so that the details of applying the methods of Chapter 6 to the delay-path optimization procedure can be better appreciated by the reader.

7.4.1 A 64-bit Adder Test Chip Design

Test Chip Architecture . A test chip using TSMC $0.18\mu\text{m}$ 6-metal CMOS technology was designed and built (Figure 7.12 (a)). The test chip contains two branches, one of which includes the 64-bit adder with its input drivers and output buffers. The other one is identical, but without the adder. This structure allows one to make realistic timing measurements appropriate to the adder alone by subtracting the input-output delay of the second branch from the analogous one of the first. A worst-case delay test pattern was applied to the input pads of both branches and the indicated subtractions performed to get the adder-only delay values.

Figure 7.12: The Architectures of a Test Chip and 64-bit Adder

Adder Architecture . Figure 7.12 (b) shows the overall structure of a 64-bit adder in which a high-speed, efficient Carry-Select block (CSA) is combined with a Carry Looka-

head (CLA) function. The decision to design it as a combination of 4-bit units was strongly influenced by the simulation results of Chapter 6.

The adder has a five level tree-like structure (Figure 7.12): at the lowest (input) level is generated the 1-bit 'gpk' (generate, propagate, kill) signals needed in the next part of the hierarchy. Here, groups of four are combined into 'GPK' signals which in turn are passed on to stage 3 where they are used to form 16-bit GPK groups. These in turn are used in the 4-bit Carry Generation operation of stage-4, and then a final selection is made from each of two 16-bit sum groups to assemble the adder's output word. The first 4-bit stages use the CLA technique to generate the group carry signals 'C[16]', 'C[32]', 'C[48]', and 'C[64]'. The last stage selects 16-bit SUM blocks for output by means of a CPL multiplexor.

Figure 7.13: Schematic of a 1-b gpk Circuit

1-bit gpk generator. The carry propagate ('p'), carry generate ('g'), and carry kill ('k') signals are the basic ones needed in a CLA adder. The appropriate logic equations are:

$$g_i = A_i \cdot B_i \quad \bar{g}_i = \bar{A}_i + \bar{B}_i \quad (7.9)$$

$$p_i = A_i \oplus B_i \quad \bar{p}_i = \overline{A_i \oplus B_i} \quad (7.10)$$

$$k_i = \overline{A_i + B_i} \quad \bar{k}_i = A_i + B_i \quad (7.11)$$

Figure 7.13 shows the circuitry used for fast gpk generation. The 'p' signal is produced by a CPL XOR/XNOR gate because these are simpler and faster than other XOR

Figure 7.14: Schematic of a 4-b GPK Circuit

configurations. CMOS buffers with latches are inserted into the CPL chains to enhance speed and drive characteristics.

4-b GPK generation . The ‘GP’, ‘GG’, and ‘GK’ signals, which determine whether the carries will propagate, generate or kill in grouped blocks, are required for the group carry (GC) calculation (described in the next subsection). These are generated as follows:

$$GG_i = \bar{k}_i \cdot (g_i + \bar{k}_{i-1} \cdot p_i \cdot GG_{i-1}) \quad (7.12)$$

$$\overline{GG}_i = \bar{g}_i \cdot (k_i + \bar{g}_{i-1} \cdot p_i \cdot \overline{GG}_{i-1}) \quad (7.13)$$

$$GP_i = p_i \cdot GP_{i-1} \quad (7.14)$$

$$GK_i = \bar{g}_i \cdot (k_i + \bar{g}_{i-1} \cdot p_i \cdot GK_{i-1}) \quad (7.15)$$

$$\overline{GK}_i = \bar{k}_i \cdot (g_i + \bar{k}_{i-1} \cdot p_i \cdot \overline{GK}_{i-1}) \quad (7.16)$$

GG_0 and GK_0 are hardwired to logic ‘0’, $\overline{GG}_0$ and $\overline{GK}_0$ to logic “1”, and GP_0 to logic “0”. Note that only a single ‘GP’ signal is needed by the following stages. Figure 7.14 shows the mixed-logic implementation.

The ‘GP’ signal is generated by a 4-input C-CMOS NAND gate because of its structure: the gate is good at asserting a logic “1” because it has only one PMOS transistor in series with the input signal. NMOS transistors were used in the ‘stacked’ structure of the figure because of their rapid turn-on speed. The result is a high speed multi-input complex

gate. The signal '*PO*' is applied to the transistors nearest to the NAND output so as to lessen worst-case delay.

The '*GG*' and '*GK*' signals (Figure 7.14) are developed by CPL circuits in order to take advantage of CPL's high operation speed and of the noise immunity inherent in differential structures. To take easy advantage of differential signal availability, vertical PMOS transistors rather than NMOS ones are used to generate logic "1" levels in the inner nodes because of their superior ability to assert these levels - this leads to an improved inner node drive capability. Vertical NMOS transistors are used to convert ('kill' - logic $1 \rightarrow 0$) in the inner node because of their speed at asserting logic "0"s. Pass-transistors with latch/buffer combinations are used to transfer either a logic "1" or "0" from the inner node to the output node. In the worst-case, only one of either "1" or "0" is propagated through the pass-transistor network.

The adder calculates the '*GPK*' signals in two stages (Figure 7.12). The '*gpk*' signals are assembled into four-bit groupings, '*GP*', '*GG*', and '*GK*', and these in turn are combined into 16-b group signals.

4-bit Group Carry Circuit (GC). '*GG*', '*GP*' and '*GK*' coming from the third level are used by the Group Carry circuit (GC) to calculate

$$CO_i = \overline{GK}_i \cdot (GG_i + \overline{GK}_{i-1} \cdot GP_i \cdot CO_{i-1}) \quad (7.17)$$

$$\overline{CO}_i = \overline{GG}_i \cdot (GK_i + \overline{GG}_{i-1} \cdot GP_i \cdot \overline{CO}_{i-1}) \quad (7.18)$$

where $CO_0 = CI$ and $\overline{CO}_0 = \overline{CI}$. The mixed-logic circuit of the 4-bit Carry Generator is shown in Figure 7.15. As described in the previous section, a vertical PMOS/NMOS transistor combination with pass transistors having latches and buffers is used to obtain high output currents and high operating speed. The Carry Output signals '*CO*' (=

Figure 7.15: Schematic of a 4-bit Carry Generator Circuit

'CO₁', 'CO₂', 'CO₃', and 'CO₄') were generated separately to reduce the load on the pass-transistor network and get a higher speed. The longest delay will occur on the path over which the Carry-In ('CI') signal travels to the Carry-Out ('CO₄') through the pass-transistor network, instead of the one where "1" or "0" is propagated in the GPK block.

The fourth stage, the 4-bit carry generator, produces simultaneously four carry signals 'CO[16]', 'CO[32]', 'CO[48]', and 'CO[64]'. These group carry signals are used to select 16-bit output SUM blocks.

16-bit CLA Circuit . This SUM stage is a 16-bit CLA SUM circuit based on four 4-bit CLA blocks as shown in Figure 7.16. The carry output signals 'C4', 'C8', and 'C12' are generated simultaneously in the third stage and then fed into the 4-bit CLA circuits

Figure 7.16: Schematic of a 16-bit CLA Circuit

which control the sum-selection multiplexor.

7.4.2 Critical Path Optimization

The identification of worst-case critical paths in CLA/CSA structures is important because any existing simplicity-performance trade-offs should, if possible, be made in favour of simplicity (that is, reduced size) in the non-critical paths. The worst-case delay path of our adder, the object of our optimization efforts, is indicated by the line starting at input bit- A_0 and ending at the 16-bit multiplexor output as shown in Figure 7.12.

The method used to size the transistors was developed in Chapter 6; using it, we were able to optimize the adder's entire critical path. A regression method was used to size the widths of the transistors in the non-critical signal paths.

7.4.3 Results and Comparison

Figure 7.17: Simulation Result of a 64-bit Adder

Figure 7.17 shows the result of a SPICE simulation of the longest delay path. The longest simulated delay ($A_0 \rightarrow SUM_{63}$) was 650ps. The effect of varying the supply voltage was also considered (Figure 7.18 (a)).

The adder is composed of 9642 transistors and occupies an area of $450\mu\text{m} \times 1000\mu\text{m}$ (Figure 7.18 (b)).

Figure 7.18: (a) Delay Vs. Supply Voltage

(b) Layout of Test Chip

The delay times and chip areas of a number of reported adders are summarized in Table 7.3. A direct and “fair” comparison of differently-structured adders implemented with different technologies is difficult. We found that the designers of conventional high-speed adders often made use of special hardware techniques and even of special manufacturing technologies in an effort to improve the operating speed of their circuits - things like dynamic circuits and SOI. (Dynamic circuits require more complex clock signals and/or transistor structures and SOI technique is costly).

For example, the design reported in [19] uses some 12,800 transistors and an advanced $0.225\ \mu\text{m}$ Cu CMOS technology to achieve sub-half nanosecond performance. The design of [18] uses a yet more advanced and costly $0.18\ \mu\text{m}$ Cu-SOI CMOS technology.

Our static adder is the fastest design (in simulations) of the static ones reported, having a worst-case (critical path) delay time of 650ps which is fully 35% less than the also static (and complicated) CMOS design described in [28]. Our adder performed better than even some of the dynamic devices.

7.5 Conclusion

Two delay-optimized test chips were simulated and then fabricated (assuming

TABLE 7.3: Performance Comparison

Des.	Ref.	Log.	Vdd(v)	Tech.(μ m)	Tr. #	Area(μ m)	Delay(ps)
[17]	2002 JSSC	Dyn.	2.5	0.25	N/A	800X150 (0.12mm ²)	900 (meas.)
[18]	2001 JSSC	Dyn.	1.5	0.18 (Cu-SOI)	N/A	0.357mm ²	440 (simu.)
[19]	2000 VLSI Sym.	Dyn.	1.8	0.225 (Cu)	12846	190X806 (0.153mm ²)	470 (simu.)
[20]	1996 JSSC	Dyn.	3.3	0.5	6924	96X2560 (0.246mm ²)	930 (simu.)
[23]	2004 JSSC	Dyn.	2.5	0.25	2502	600X65 (0.04mm ²)	780 (simu.)
[24]	2001 ARVLSI	Dyn.	2.5	0.25	N/A	N/A	493 (simu.)
[25]	2002 APC	Dyn.	2.5	0.25	N/A	815X196 (0.16mm ²)	700 (simu.)
[26]	2002 VCDT	Dyn.	1.2	0.1 (SOI)	N/A	N/A	346 (simu.)
[27]	1999 ISCS	Dyn.	2.5	0.25	5460	1284X130 (0.16mm ²)	670 (simu.)
[28]	2001 SSC	Sat.	1.8	0.18	N/A	N/A	1000 (simu.)
[21]	1995 CICC	Sat.	3.3	0.5	3044	1305X207 (0.27mm ²)	2600 (meas.)
Ours		Sat.	1.8	0.18	9642	1000X450 (0.45mm ²)	650 (simu.)

TSMC 0.18 μ m technology in both cases). One of them, an array of eight 8-bit RCAs, was successfully used to verify the advantages of mixed logic design. The performance of the other, a practical 64-bit high-speed static adder, verified that closed-formula optimization is a realistic circuit-design tool. On the practical side, the adder's 650ps critical-path delay makes it the fastest of a number of recently-reported static designs, and faster than even some dynamic ones.

We therefore have good reason to believe that the C-CMOS/CPL mixed-logic style will come to play an ever more important role in high speed circuit design.

8.1 Introduction

The accurate modeling of signal delays is obviously a very important part of the high-speed digital IC design. The traditional tools to do this are simulators which work from the first principles of Physics, the package *SPICE* being probably the most well-known representative of these. But the accurate and detailed delay information which they yield comes at the cost of extensive, time-consuming calculation.

The work described herein was the applications of some closed-form optimization and delay prediction expressions elaborated on this thesis. These expressions demonstrate close correlations with circuit simulation results. In other words, we have demonstrated an accurate analytical delay model [1] able to predict circuit performance at the device-, gate- and circuit-block levels for both $0.13\mu\text{m}$ and $0.18\mu\text{m}$ technologies. Practical transistor sizing formulas for delay optimization applicable to CMOS gates were implemented in *MATLAB*, ones that may be combined to effect delay minimization in the worst-case digital signal paths of mixed logic circuits. On the practical side, two test chips were made for the purpose of evaluating the methods suggested by theoretical work and for determining their potential as a design optimization method.

8.2 Background Preparation

8.2.1 Analytical Delay Model

Closed-form short-channel transistor delay models were presented based on [1], ones which took into account a number of factors affecting circuit-delay and among which was a MOSFET saturation current model which took account of superposed current sources and sinks ('overlapping' and 'opposing' currents). Formulas giving MOSFET capacitances and channel currents as linear functions of transistor width were also presented in [1]. The effects of input signal derivative (dv/dt) and series-connected transistors were quantified in these delay models.

System delay models specific to C-CMOS, DCVSL and CPL were presented [1] and parametrized in terms of drive gate, cell gate and load gate characteristics (topologies, sizes). These models can be applied to estimate the overall delay though both single-mode and mixed-mode device sequences [1].

8.2.2 Parameter Extraction

The closed-formula delay estimators naturally involved a number of experimentally-determined parameters. The 'experiments' used to determine these were simulation runs involving an accepted model of circuit behaviour - BSIM3V3 in particular. Four sets of parameters were required - one each for NMOS/PMOS rising/falling delays. The standard TSMC $0.18\mu\text{m}$ technology was assumed in this part of the work.

8.2.3 Delay Optimization

Investigations and modifications applicable to CCMOS, DCVSL and CPL CMOS logic styles were made to the general gate-level delay models of Shams [1]. The formulas

for the minimum-delay widths of the transistors in the pull-up and pull-down CMOS networks are calculated. The ‘exact’ formulae are interdependent and must be iterated to self-consistency, but reasonable approximations can be solved directly. But even the iteration is a very simple one, and produces results almost immediately. All delay functions have been expressed in *MATLAB*.

8.3 Thesis Contributions

- Extraction of delay parameters for TSMC 0.18 μ m CMOS technology based on [1].
- Using and verifying accuracy of a variation of the expressions introduced in [1] for modeling delay in MOSFETs.
- Modifying the closed-form transistor sizing formula presented in [1] for PTL circuits.
- Verifying delay expressions and optimization formulas for TSMC 0.18 μ m technology using MATLAB and SPICE simulation.
- Finding the optimum number of serially connected NMOS transistors and the optimum places for insertion buffers and latches in PTL circuits.
- Application of above to designing fast adder circuits and comparing with previous works in this field.

8.4 Applications

Mixed-logic digital CMOS circuits (combinations of C-CMOS, DCVSL, CPL for example) can be used to combine the merits of each type into a whole having superior performance to any one of them individually. The closed-form time delay predictors in this thesis allow a quick comparison between what would otherwise be prohibitively large numbers of candidates for a given application. We demonstrated this by predicting the

behavior of twenty-seven 3-bit adders made up all possible combinations of the three logic styles studied and then choosing the best one, all in short order (in the space of hours, compared to the days that traditional circuit simulation would have taken).

Long chain CPL circuits are often found in circuits such as adders or deep multiplexers. The most common approach for dealing with long delays in these devices is to insert into the CPL chain every n switches transistors. We used simulation to find the optimum value of n is about 4 - a sequence of 4-bit CPL chains, and the best solution in this long CPL chain example to be a latch with buffer. These experimental results were used in our two test chip design.

One of our test chip, eight 8-bit RCAs, was fabricated to verify the potential of mixed logic design. The measured results exhibit that the combinations of C-CMOS and CPL in 8-bit RCAs exhibit better performance both in delay, power dissipation, area, and PDP.

The next application of our delay predictors came in the design of a 64-bit static adder. The 'critical path', the route from C_{in} to C_{out} , was identified and the intervening devices were designed to minimize the propagation time along it. The success of the closed-form delay formulas was evident in the fact that the optimized static adder was faster than other statics reported to date and faster than even some of the dynamics. (SPICE simulation indicated that a delay of 650 pSec could be expected on the $C_i \rightarrow C_o$ path). The results of circuit simulation were confirmed when an actual physical adder was built (the second test chip).

This experiment with a practical circuit element of great importance in high-speed DSP applications provided convincing evidence that mixed logic design (especially C-

CMOS/CPL) aided by closed form delay optimization and prediction may play a more and more important role in the high performance design.

8.5 *Future Work*

A number of problems, for which there was insufficient time, suggested themselves during this work:

- **Interconnect optimization**

Interconnect details should be a part of a comprehensive optimization procedure, but this is an issue that is entirely different from the ones of this thesis.

- **Topology extensions**

There is no reason that the optimization methods can't be applied to dynamic structures as they have been to static ones. Doing this is mainly a programming problem.

- **CAD integration**

This work has proved the value of (for the most part) non-iterative methods of complex circuit design, methods that should be made more widely available by integrating them into a chip design package (CAD) as a standard feature. Doing this would be a challenging and worthwhile engineering software project.

References

- [1] M. Shams, "Modeling and optimization of CMOS logic circuits with application to asynchronous design," PhD thesis, University of Waterloo, 1999.
- [2] N. Weste, K. Eshraghian, "Principle of CMOS VLSI design: a system perspective," Addison-Wesley, pp.216, 1993.
- [3] J.M. Rabaey, A. Chandrakasan, B. Nikolic, "Digital integrated circuits: a design perspective," 2nd Ed., Prentice-Hall, 2002.
- [4] J. Rubinstein, P. Penfield, A. Horowitz, "Signal delay in RC tree networks," IEEE Trans. on CAD, vol. 2, No.3, Jul. 1983.
- [5] M. Anis, M. Allam, M. Elmasry, "Impact of technology scaling on CMOS logic styles," Circuits and Systems II: Analog and Digital Signal Processing, vol. 49, pp.577 - 588, Aug. 2002.
- [6] K. Yano, Y. Sasaki, K. Rihino, and K. Seki, "Top-down pass-transistor logic design," IEEE J. Solid-State Circuit, vol. 31, pp.792-803, Jun. 1996.
- [7] T. Sakurai, and A.R. Newton, "Alpha-power law mosfet model and its applications to CMOS inverter delay and other formulas," IEEE Journal of Solid-state Circuits, vol.25, pp.584-594, Apr. 1990.

- [8] T. Sakurai, and A.R. Newton, "Delay analysis of series-connected MOSFET circuit," IEEE Journal of Solid-state Circuits, vol.26, pp.122-131, Feb. 1991.
- [9] W.C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," Journal of Applied Physics, vol. 19. pp.55-63, Jan.1948.
- [10] N. Hedenstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization," IEEE Transactions on Computer-Aided Design, vol. CAD-6, pp.270-281, Mar. 1987.
- [11] L. G. Heller and W.R.Griffin, "Cascode voltage switch logic: A differential CMOS logic family," Proc. Dig. Tech., pp.16-17, 1984.
- [12] P. Srivastava, A. Pua. and L. Welch, "Issues in the design of Domino logic circuits," Proc. IEEE Great Lakes Symp. on VLSI, pp. 108-112, 1998.
- [13] I.Abu-Khater, A. Bellaouar, M.I. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers," IEEE Journal of Solid-State Circuits, vol. 31, No.10, pp. 1535-1546, Oct. 1996.
- [14] M. Alioto, and G. Palumbo, "Analysis and comparison on full adder block in submicron technology," IEEE Transactions on VLSI System, vol.10, No.6, pp. 806-823, Dec. 2002.
- [15] M. Sayed, and W. Badawy, "Performance analysis of single-bit full adder cells using 0.18, 0.25, and 0.35 μm CMOS technologies," IEEE International Symposium on Circuits and Systems, pp. III-559 -III-562, May 2002.
- [16] A. Shams, T. Darwish, M. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," IEEE Transaction on VLSI System, vol. 10, No.1, pp.20-29, Feb. 2002.
- [17] S. J., Lee, and H. J., Yoo, "Race Logic architecture (RALA): A novel logic concept

- using the race scheme of input variables," IEEE J. Solid-State Circuits, vol. 37, pp. 191-201, Feb. 2002.
- [18] D. L., Stasiak, F. Mounes-Toussi, and S. N. Storino, "A 440ps 64-bit adder in 1.5-v/0.18 μ m partially depleted SOI technology," IEEE J. Solid-State Circuits, vol. 36, pp.1559-1564, Oct. 2001.
- [19] J. Park, H. C. Ngo, J.A. Silberman, and S. H. Dhong, "470ps 64-bit parallel binary adder," Symp. VLSI Circuit Dig. Tech., pp.192-193, 2000.
- [20] S. Naffziger, "A sub-nanosecond 0.5 μ m 64-b adder design," IEEE Int. Solid-State Circuit Conf., pp.362-363, 1996.
- [21] H. Morinaka, H. Makino, Y. Nakase, H. Suzuki, and K. Mashiko, "A 64 bit carry lookahead CMOS adder using modified carry select," IEEE Custom Integrated Circuits Conf., pp. 585-588, 1995.
- [22] W. Hwang, G. Gristede, P. Sanda, S.Y. Wang, and D. F. Heidel, "Implementation of a self-resetting CMOS 64-bit parallel adder with enhanced testability," IEEE J. Solid-State Circuits, vol. 34, pp.1108-1117, Aug. 1999.
- [23] C.H. Huang, J.S. Wang, C. Yeh, and C.J. Fang, "The CMOS Carry-Forward Adders," IEEE J. Solid-State Circuits, vol. 39, No.2, Feb. 2004.
- [24] S. Sheng, L. McMurchie, and C. Sechen, "A high-performance 64-bit adder implemented in output prediction logic," Advanced Research in VLSI Conf., pp. 213-222, Mar. 2001.
- [25] X. Sun, Z. Mao, and F. Lai, "A 64 bit parallel CMOS adder for high performance processors," IEEE Asia-Pacific Conf., pp. 205-208, Aug. 2002.
- [26] J.J. Kim, R. Joshi, C.T. Chuang, K. Roy, "SOI-optimized 64-bit high-speed CMOS

- adder design," Symposium on VLSI Circuits Digest, pp. 122-125, June 2002.
- [27] R. Woo, S.J. Lee; H. J. Yoo, "A 670 ps, 64 bit dynamic low-power adder design," IEEE international Symposium on Circuits and System, vol.1, pp. 28-31, May 2000.
- [28] H.Q. Dao, V.G. Oklobdzija, "Application of logical effort on delay analysis of 64-bit static carry-lookahead adder," Signals, Systems and Computers, Conference Record of the Thirty-Fifth Asilomar, vol. 2, pp.1322-1324, Nov. 2001.
- [29] I. Sutherland, B. Sproull, D. Harris, "Logical effort: designing fast CMOS circuits," Morgan Kaufmann Publishers, San Francisco, California, 1999.