New Methodologies for Low-Power High-Performance Digital VLSI Design

by

Mohamed W. Allam

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2000

©Mohamed W. Allam 2000

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisitions et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre reference

Our file Notre reference

The author has granted a non-exclusive licence allowing the National Library of 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 nationale du 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.

0-612-53483-9

Canada

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Historically, VLSI designers have focused on increasing the speed and reducing the area of digital systems. However, the evolution of portable systems and advanced Deep Sub-Micron (DSM) fabrication technologies, has brought power dissipation as another critical design factor. Low-power design reduces cooling cost and increases reliability especially for high-density systems. Moreover, it reduces the weight and size of portable devices. Yet, high-performance is still the main criterion for most digital systems, which may not be sacrificed to achieve lower power dissipation. This thesis presents new low-power high-performance digital VLSI design methodologies for process, circuit, and algorithm level design.

On the process level, future challenges in device scaling such as short channel effects, subthreshold leakage currents, and hot carrier effects are discussed. The influence of technology scaling on the performance, power, and area of different CMOS logic styles is then analyzed and simulated. This study covers five logic families; namely, CMOS, CPL, Domino, DCVS and CML. The scalability of each logic style and its potential in future technology generations are explored.

On the circuit level, a new logic family for low-power high-performance applications is presented. This logic family combines the speed, low supply voltage, and noise immunity of CML circuits with the low standby current and design simplicity of dynamic circuits. The new logic style reduces the power by 7% and the delay by 73% compared to conventional CMOS logic. A 16-bit CLA adder is designed, simulated, fabricated, and tested using $0.6\mu m$ CMOS technology. Test results have confirmed the functionality of the new logic family at various supply voltages.

Also, a new Domino logic style, called High-Speed Domino (HS-Domino), has been developed. HS-Domino resolves the trade-off between noise margin and speed associated with the conventional Domino logic. Simulation results show that HS-Domino circuits are superior to conventional Domino ones in terms of power, speed, and tolerance to the leakage currents in DSM technologies.

This study also presents new Multiple Threshold CMOS (MTCMOS) scheme for dynamic circuits. This scheme is applied to Domino and DDCVS logic styles. The new implementations reduce the leakage power by orders of magnitude keeping the noise margin intact, and maintain the high performance and low dynamic power of low V_T circuits. Unlike other MTCMOS Domino logic implementations, the new scheme does not require additional hardware.

At the algorithm level, a new algorithm for high radix division is presented. The algorithm uses a look-up table to estimate the quotient digit at each iteration. The look-up table is optimized to reduce power dissipation and delay of the divider. Simulation results show that the new algorithm reduces the power dissipation by 22% and 12% for radix 8 and radix 16 division, respectively, compared to other division algorithms. The algorithm also increases the speed by a factor of 13% and 10%, for radix 8 and radix 16 division, respectively.

Acknowledgements

Thanks to God Almighty for all of his blessings, one of which is this thesis.

I would like to thank my supervisor, Professor M.I. Elmasry for all his technical, financial and personal support during my study years at Waterloo.

I would like also to thank all my colleagues in the VLSI Research Group for their comments and helpful discussions. In particular, I would like to thank Mohab Anis for his help and encouragement. I would like also to thank Muhammad Khellah, Ayman Elsayed, Amr Hafez, Amr Wassal, Alaa Elraey, Muhammed Nummer, Mohamed Elgebaly, Amr Fahim and Nayer Wanas for making the last four years some of the best years of my life.

I am extending my sincere gratitude to my family. To my loving wife, Eman, who sacrificed her comfort and time to support me and ultimately helped drive me to the completion of this work. To our parents who have been always a tremendous source of encouragement, confidence and love. This dissertation is dedicated to them and to my wonderful daughter, Lobna.

Contents

1 Introduction **1**

2 Low-Power Digital CMOS Design **6**

2.1 Power Dissipation in CMOS Digital Circuits 6

2.2 Dynamic Power Dissipation 7

2.2.1 Switching Power Dissipation 7

2.2.2 Short Circuit Power Dissipation 10

2.2.3 Glitching Power Dissipation 12

2.3 Static Power Dissipation 13

2.3.1 Diode Leakage Current 13

2.3.2 Subthreshold Leakage Current 14

2.3.3 Biasing Current 15

2.4 Low-Power CMOS Logic Design 16

2.5 Low-Power VLSI Technologies 17

2.5.1 Threshold Voltage Reduction 17

2.5.2 Technology Scaling 18

2.5.3 Increasing Number of Metal Layers 19

2.5.4 Alternative Technologies 20
2.6 Low-Power Packaging 22
2.7 Low-Power Layout techniques 24
2.8 Low-Power Circuit Techniques 26
2.8.1 Supply Voltage Reduction 26
2.8.2 Reduced Output Voltage Swing 27
2.8.3 Reducing The Physical Capacitance 28
2.8.4 Reducing The Switching Frequency 29
2.9 Low-Power Gate Level Design 31
2.9.1 Low-Power Synthesis 31
2.9.2 Clock Gating 33
2.10 Low-Power Behavioral Design 35
2.11 Low-Power System Design 38
2.12 Summary 40

3 Effect of Technology Scaling on CMOS Logic Styles 41
3.1 Trends and Limitations of CMOS technology Scaling 42
3.1.1 MOSFET Scaling Trends 44
3.2 Challenges of MOSFET Scaling 48
3.2.1 Short channel effects 48
3.2.2 Subthreshold Leakage Currents 50
3.2.3 Dielectric Breakdown (DB) 53
3.2.4 Hot Carrier Effects (HCE) 54
3.2.5 Soft errors 54

3.2.6 Scaling the Interconnects 55
3.3 CMOS Logic Styles 56
3.3.1 Conventional CMOS 57
3.3.2 Complementary Pass Logic (CPL) 59
3.3.3 Domino Logic 62
3.3.4 Differential Cascode Voltage Switch (DCVS) 65
3.3.5 MOS Current Mode Logic (MCML) 68
3.4 Impact of Technology Scaling on Logic Styles 72
3.4.1 Velocity Saturation and Mobility degradation 72
3.4.2 Leakage currents 77
3.4.3 Hot Carrier Effect (HCE) 79
3.4.4 The Drain Induced Barrier Lowering (DIBL) 80
3.4.5 Scaling down V_{dd}/V_{th} ratio 80
3.4.6 Scaling of Interconnects 82
3.5 Area Considerations 82
3.5.1 Technology Scaling and Area 82
3.5.2 Logic Style and Area 84
3.6 Simulation Setup 85
3.6.1 Gate Simulation Setup 86
3.6.2 CLA Adder Setup 88
3.7 Results and Analysis 89
3.7.1 Gate Results 89
3.7.2 CLA results 97

3.8 Summary 98

4 Dynamic Current Mode Logic, A New Low-Power High-Performance Logic Family 100

4.1 Introduction 101

4.2 MOS Current Mode Logic (MCML) 103

4.3 DyCML Circuit Architecture and Operation 105

4.3.1 Operation of the Dynamic Current Source 109

4.3.2 Cascading DyCML Gates 111

4.3.3 DyCML-CMOS Interfacing 114

4.4 Circuit Implementation and Simulation Results 116

4.4.1 Gate Simulation and Comparison 117

4.4.2 Block Level Comparison 121

4.5 Experimental Results 123

4.6 Summary 126

5 New High-Speed Dynamic Logic Styles for CMOS and MTCMOS Technologies 127

5.1 Introduction 128

5.2 Domino Logic 129

5.2.1 Noise Margin and Delay of CD-Domino Circuits 131

5.3 High Speed Domino (HS-Domino) 133

5.3.1 Speed Comparison 135

5.3.2 Power Dissipation of HS-Domino 137

5.4 MTCMOS Implementations for Domino Logic 139

5.4.1 MTCMOS CD-Domino Logic 141
5.5 MTCMOS High Speed Domino Logic (MHS-Domino) 142
5.5.1 Speed Comparison 144
5.5.2 Dynamic Power Comparison 145
5.5.3 Leakage Comparison 146
5.6 MTCMOS Implementation for DDCVS Logic (MDDCVS) 148
5.6.1 MDDCVS Architecture and Operation 149
5.7 MDDCVS Simulation and Comparison 152
5.7.1 Dynamic Power Comparison 153
5.7.2 Leakage Power 154
5.8 Summary 155

6 Low-Power High-Radix Floating Point Division Algorithm Using Quotient Approximation and Error Correction 157
6.1 Introduction 158
6.1.1 Types of Division Algorithms 159
6.1.2 SRT Division Algorithm 159
6.2 FAST Division Algorithms 161
6.3 Modified Division Algorithm 164
6.3.1 Algorithm Description 164
6.4 Algorithm Implementation and Simulation Results 166
6.4.1 Hardware Implementation 166
6.4.2 Simulation Results 171
6.5 Summary 172

7 Conclusion **174**

7.1 Thesis Contributions 175

7.2 Future Work 177

7.3 Publications 179

A CMOS Logic Gates **181**

B Convergence of The Division Algorithm **191**

Bibliography **196**

List of Tables

3.1 Historical trends and SIA road map of LSI's 43

3.2 A Generalized scaling scheme 45

3.3 Influence of scaling on MOS device characteristics 46

3.4 Parameters used for technologies 87

3.5 Logic gates comparison 90

3.6 CLA comparison 98

4.1 Logic Gates Comparison 120

4.2 CLA comparison 123

5.1 Type of Transistors in the Domino Logic 140

5.2 Standby @ Precharge : Rejected 140

5.3 Standby @ Evaluation : Correct 141

6.1 Look up table dimensions for different radices 169

6.2 Simulation results 172

List of Figures

2.1 Sources of load capacitance in a CMOS inverter 8

2.2 Output of an inverter 10

2.3 Leakage current in a CMOS logic gate 14

2.4 Pseudo-NMOS logic gate 16

2.5 The MCM for the Sun Viking, consisting of CPU, cache controller, and eight SRAM mounted on a 1.9"x1.9" copper-polyimide substrate 23

2.6 Solder bumps of a flip chip 24

2.7 A Reduced Swing CMOS Inverter 27

2.8 True Single Phase Clocking Scheme 29

2.9 Power dissipation of an Intel740 graphics chip 33

2.10 Clock gating 34

2.11 Power vs delay 36

3.1 Historical trends of LSI's 42

3.2 Trends of scaling V_{dd}, V_{th} and T_{ox} 48

3.3 Short channel effects 49

3.4 Effect of V_{dd}/V_{th} on the delay 52

3.5 Number of metal layers over the generations 56
3.6 Trend of the ratio of the interconnect RC delay and the clock cycle 57
3.7 AOI implemented in Conventional CMOS 58
3.8 AOI implemented in CPL 60
3.9 MUX implemented in CPL 61
3.10 AO implemented in Domino 62
3.11 XOR implemented in NP-Domino 65
3.12 Full adder implemented in NP-Domino 65
3.13 AOI implemented in original DCVS 66
3.14 AOI implemented in DDCVS 67
3.15 Inverter implemented in MCML 69
3.16 Modes of operation on I-V characteristics during discharging 72
3.17 Effect of velocity saturation on MOS I-V characteristics 74
3.18 Velocity saturation 75
3.19 Mobility degradation 75
3.20 Optimal threshold voltage for static and dynamic circuits versus technology 78
3.21 Section of a gate implemented using CPL 81
3.22 Number of metal layers and average metal pitch trend 84
3.23 Logic gates' setup for simulation 86
3.24 Architecture of a 16 bit CLA adder 88
3.25 Average normalized delay for Group I (AND, OR, AOI) 91
3.26 Average normalized power/MHz for Group I (AND, OR, AOI) . . . 92

3.27 Average normalized EDP for Group I (AND, OR, AOI) 94
3.28 Average normalized delay for Group II (XOR, MUX, FA) 95
3.29 Average normalized power/MHz for Group II (XOR, MUX, FA) . . 96
3.30 Average Normalized EDP for Group II (XOR, MUX, FA) 97

4.1 MCML logic gate 104
4.2 Architecture of a DyCML gate 106
4.3 Voltages at different nodes in the DyCML gate 107
4.4 Dynamic current source voltages and current 110
4.5 Clock delay scheme 112
4.6 Self timing buffer 113
4.7 Voltages at different nodes in the self timing buffer 114
4.8 Differential-single ended buffer 115
4.9 Voltages at different nodes in the reduced-full swing buffer 116
4.10 Divide by 2 DyCML circuit 117
4.11 Maximum operating frequency vs. supply voltage 118
4.12 Simulation setup for logic gates 119
4.13 Block diagram of a 16 bit CLA adder 121
4.14 Generate gate of a 4Bit carry look ahead adder 122
4.15 Microphotograph of the 16-bit CLA adder 124
4.16 Measured Delay 125

5.1 An 8-input Clock-Delayed OR Domino gate 129
5.2 Noise margin of Domino logic versus V_{th} 131
5.3 Normalized delay of CD-Domino versus V_{th} 132

5.4 An 8-input HS-Domino OR Gate 134
5.5 W_{keeper}/W_n ratio versus V_{th} 135
5.6 Speed of the different dynamic styles 136
5.7 Normalized dynamic power versus V_{th} 137
5.8 Normalized leakage power versus V_{th} 138
5.9 An 8-input MTCMOS CD-Domino OR gate 141
5.10 An 8-input MHS-Domino OR Gate 143
5.11 Normalized delay of DVT Domino versus V_{th} 145
5.12 Normalized dynamic power in MTCMOS Domino versus V_{th} 146
5.13 Normalized leakage power in MTCMOS Domino versus V_{th} 147
5.14 An XOR DDSVSL Gate 149
5.15 A two input MDDCVS XOR logic gate 150
5.16 Normalized delay for DDCVS logic styles versus V_{th} 153
5.17 Normalized dynamic power in DDCVS logic styles 154
5.18 Normalized leakage power in MDDCVS versus V_{th} 155

6.1 SRT algorithm with QST approach 162
6.2 Modified SRT algorithm with QST approach 163
6.3 New division algorithm 167
6.4 Modified algorithm with positive quotient digits only 168

A.1 NAND implemented in CMOS 181
A.2 NOR implemented in CMOS 182
A.3 MUX implemented in CMOS 182
A.4 XOR implemented in Conventional CMOS 183

A.5 Full Adder implemented in Conventional CMOS 183
A.6 NAND implemented in CPL 184
A.7 NOR implemented in CPL 184
A.8 XOR implemented in CPL 185
A.9 Full Adder implemented in CPL 185
A.10 AND implemented in Domino 186
A.11 OR implemented in Domino 186
A.12 MUX implemented in MCML 187
A.13 XOR implemented in MCML 187
A.14 Full Adder implemented in MCML 188
A.15 AND/NAND/OR/NOR implemented in DCVS 189
A.16 MUX implemented in DCVS 189
A.17 XOR implemented in DCVS 190
A.18 Full Adder implemented in DCVS 190

Chapter 1

Introduction

Since the invention of the first Integrated Circuit (IC) three decades ago, designers have been looking for methods to speed up digital circuits and to reduce the area of their designs. Recently, advances in VLSI fabrication technology have made it possible to put a complete System On a Chip (SOC) which facilitates the development of Personal Digital Assistants (PDA's), cellular phones, labtops, hand-held computers, and mobile multimedia systems. The evolution of these applications profiles power dissipation as a critical parameter in digital VLSI design.

Power dissipation is defined as the rate of energy delivered from the source to the system/device. In battery operated systems, the amount of energy stored within the battery is limited. Therefore, power dissipation is important for portable systems, as it defines the average lifetime of the battery. Unfortunately, battery technology is not expected to improve the battery storage capacity by more than 30% every five years [1]. This is not sufficient to handle the increasing power requirements of portable systems. low-power devices are expected to have smaller battery size, less

weight, and longer battery lifetime.

Power dissipation is also crucial for Deep Sub Micron (DSM) technologies. Advances in CMOS fabrication technology double the number of transistors per chip every two years and double the operating frequency every three years. Consequently, the power dissipation per unit area grows, increasing the chip temperature. This excessive temperature reduces the reliability and lifetime of the circuit. Hence, large cooling devices and expensive packaging are required to dissipate the extra heat. Also, systems with high power dissipation require a special Printed Circuit Board (PCB) technology to deliver large currents from the power supply to the various devices in the system. For example, a Pentium III®[1] processor requires a 18A power supply [2], which can not be handled by a conventional PCB process. Therefore, the price and area of the digital system increase and the transistor density decreases obliterating the advantages of smaller transistor sizes [3]. Furthermore, large current densities cause serious problems. Electromigration caused by large currents flowing through narrow wires, may produce gaps or bridges in the power rails of the chip with a subsequent permanent damage to the system.

Another reason for low-power design arises from modern automated offices. In 1993, the American Council for Energy Efficient Economy reported that office equipment accounted for 5% of the total commercial energy consumed and it should be 10% by the year 2000 [4]. The generation of this energy costs two billion dollars annually and generates pollution equivalent to five million cars. This problem resulted in the development of the "Green Computers" concept to reduce the

[1] Pentium and Pentium III are trademarks of Intel Corporation

amount of energy consumed by office equipment.

Though power dissipation is important for portable devices, performance (speed) continues to be the main target for digital designers. Consumers expect higher speed, more functionality, and higher levels of integration, from their cellular phones and hand-helds. To emphasis the importance of speed, researchers use Energy Delay Product (EDP) as an evaluation figure for digital systems. Consequently, reducing power dissipation should not come at the expense of performance. In the mean time, increasing performance while keeping power dissipation constant is also considered to be a low-power design problem.

The objective of this dissertation is to develop new design methodologies to reduce power and enhance performance simultaneously. The thesis covers low-power design on different design stages, beginning from the process level up to the algorithm level. On the process level, the impact of CMOS technology scaling on power, delay, and area of various logic styles is presented along with predictions for future scalability of each logic style. On the circuit level, two new circuit techniques to enhance performance and reduce power dissipation are presented. On the algorithm level, a low-power algorithm for high radix division is proposed.

This thesis is organized as follows:

Chapter 2 presents the various power dissipation mechanisms in CMOS digital circuits. The second half of the chapter introduces a review of present low-power design methodologies on various design stages, beginning with the process level up to the system level. For each low-power design approach, the impact on speed and area is analyzed. This chapter serves as a background and motivates the need for

the work presented in later chapters.

Chapter 3 analyzes the impact of technology scaling on CMOS logic styles. The first part of this chapter reviews the historical trends and limitations in CMOS technology scaling. The different phenomena associated with smaller transistor sizes are also explored. The second half of the chapter presents five of the most famous logic styles; namely, CMOS, CPL, Domino, DCVS and MCML. The effect of technology scaling on the performance, power, and area of the five logic styles is then presented along with predictions for future trends. To verify the qualitative analysis, simulation results for large number of gates and circuits are discussed. In those simulations, each logic family is implemented in four different fabrication technologies (0.8, 0.6, 0.35, 0.25 μm CMOS technologies). The results are then analyzed and compared.

In Chapter 4, a new logic style for high-speed low-voltage and low-power design is given. The logic style, called Dynamic Current Mode Logic (DyCML), uses reduced voltage swing logic to reduce dynamic power and delay. DyCML employs dynamic logic design to cancel the static power dissipation of CML circuits. The chapter begins with an explanation of reduced voltage swing design concepts and CML logic circuits. Then, the new circuit architecture and the theory of operation is introduced. A 16-bit DyCML CLA adder is fabricated using $0.6\mu m$ CMOS technology. Both simulation and testing results are given to confirm the advantages of the new logic family over existing ones.

Chapter 5 presents new high speed dynamic logic styles for CMOS and MTC-MOS technologies. The first part of this chapter introduces a new dynamic logic

style, called High Speed Domino (HS-Domino). HS-Domino resolves the trade-off between performance and noise margins in Domino circuits. It eliminates the contention currents during evaluation, which leads to faster transitions and less power dissipation. Simulation results for the new logic style are compared with the conventional Domino style. The second half of this chapter presents new MTCMOS implementation for Domino and DCVS logic styles. The new implementations reduce the leakage power by orders of magnitude keeping the noise margin intact. Simulation results and comparisons with other MTCMOS dynamic circuits are given.

Chapter 6 profiles a new high radix division algorithm for floating point operations. The algorithm utilizes redundancy in the quotient digit to reduce the size of the look-up table. The convergence of the algorithm is studied and proven. Implementation details and simulation results are also given.

Chapter 7 concludes this work with a summary and list of contributions.

Chapter 2

Low-Power Digital CMOS Design

The purpose of this study is to develop new design strategies to reduce power and delay simultaneously on various levels of abstraction. To achieve that objective, an understanding of the power dissipation mechanisms in digital CMOS circuits is critical.

The first part of this chapter presents different kinds of power dissipation in CMOS logic circuits. The rest of the chapter reviews some low-power design techniques on different design stages.

2.1 Power Dissipation in CMOS Digital Circuits

Power dissipation in CMOS digital circuits is categorized into two types: peak power and average power. Peak power affects both circuit lifetime and performance. Excessive instantaneous current drawn from the power supply results in a voltage drop over the supply rails (G_{ND}, V_{dd}). This large current causes a large power dissipa-

tion inside the supply wires because of their impedance. Consequently, this large power consumption causes overheating of the device which reduces the reliability and lifetime of the circuit. Also, a voltage drop along the supply lines hinders the performance of the circuit and causes erroneous digital outputs and digital glitches.

Average power dissipation is significant for calculating the battery weight and lifetime. Average power is categorized into: dynamic power and static power dissipation. Dynamic power is the component proportional to the operating frequency of the circuit or the frequency of node switching. Static power is independent of the operating frequency and is constant most of the time. Dynamic power is important during normal operation especially at high operating frequencies. On the other hand, static power is more important during standby especially for battery powered devices.

2.2 Dynamic Power Dissipation

Dynamic power consists of three components: switching power, short circuit power and glitching power. The value of each of these components is a function of the logic style used and the topology of the circuit. These components are discussed in detail in this section.

2.2.1 Switching Power Dissipation

Switching power is defined as the power consumed by the logic gate to charge the output load from the low voltage level "0" to the high voltage output "1". In a well designed digital circuit, the switching power is the dominant component in power

dissipation. Usually, it is independent of the logic function of the circuit. Switching power is expressed as:

$$P_{switching} = F_{switching} \cdot V_{dd}^2 \cdot C_L \tag{2.1}$$

where $F_{switching}$ is the switching frequency, V_{dd} is the supply voltage and C_L is the net load capacitance. The net loading capacitance C_L consists of: the gate capacitance of subsequent gate(s) input(s) connected to the inverter's output, interconnect capacitance, and the diffusion capacitance of the drains of the inverter transistors. Test chips have shown that the total capacitance is split almost equally between these three types. As the minimum gate length scales down, though, interconnect capacitance becomes dominant. Figure 2.1 shows the basic capacitive elements of a CMOS inverter.

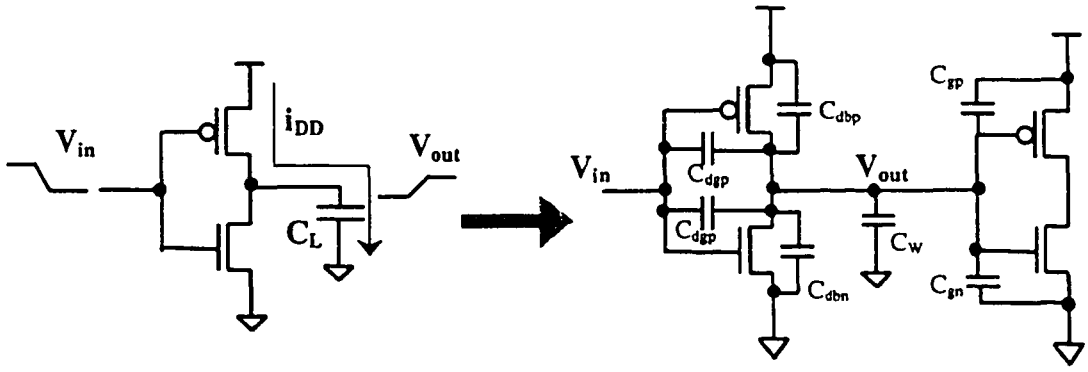


Figure 2.1: Sources of load capacitance in a CMOS inverter

Equation 2.1 indicates that the supply voltage is the dominant factor in the switching power dissipation. Thus, reducing the supply voltage is the most effective techniques to reduce the power dissipation. This equation is valid only for logic

families with rail to rail output swing. Some logic families, like CPL have a reduced voltage swing like CPL (complementary Pass Logic). In these cases, the switching power may be expressed a follows:

$$P_{switching} = F_{switching} \cdot V_{dd} \cdot V_{swing} \cdot C_L \tag{2.2}$$

where V_{swing} is the logic swing of the digital output.

The switching frequency is defined as follows:

$$F_{switching} = F_{Operating} \cdot \alpha \tag{2.3}$$

where α is the switching activity factor of the gate. This factor determines the probability of the gate having "0"→"1" transition at the output[1] [5]. The switching activity is a function of several factors including: the logic function, the logic style, the circuit topology, and the sequencing of operations. For example, an XNOR gate has a higher transition probability than a NAND gate because the XNOR has 50% "1"'s and 50% "0"'s in its truth table whereas the NAND gate has only one "1". This is true for both circuits regardless of the number of inputs. Due to the need to precharge, dynamic circuits usually have higher switching activities compared to static circuits [6].

[1] Assuming that the gate does not experience glitching

2.2.2 Short Circuit Power Dissipation

Short circuit power is the power passing from the supply to the ground during the transitions from logic "0" to logic "1" and from logic "1" to logic "0". Unlike the switching power, which is a function of the number of "0"→"1" transitions, the short circuit power is a function of the toggling frequency. Figure 2.2 illustrates the output voltage of a standard CMOS inverter with the short circuit current. During transitions, both the PMOS transistor and the NMOS transistor of the inverter are ON creating a short circuit path between V_{dd} and G_{ND}.

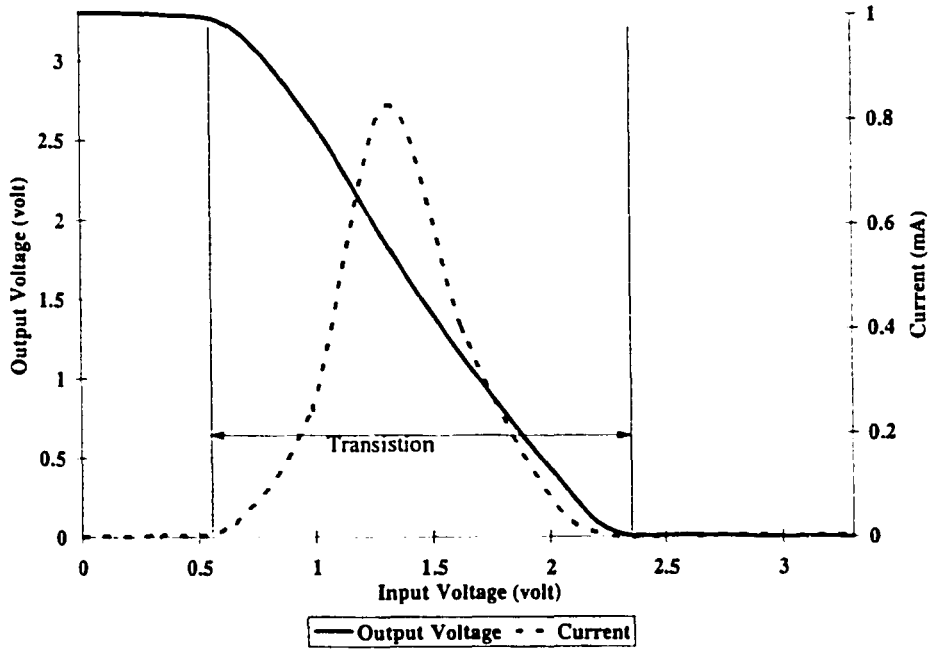


Figure 2.2: Output of an inverter

The peak magnitude of the short circuit current is dependent on device size. However, the average value of the short circuit current is roughly independent of the

device size for a fixed load capacitance. While the peak magnitude of the current increases, the rise/fall time decreases so that the average current is the same. If all devices are sized up so that the load capacitance scales up proportionally, then the rise/fall time remains constant, and the average current (and power) scales up linearly [7].

Short circuit power is either linearly or quadratically proportional to the supply voltage. This power component depends on the transistor channel length and whether or not the transistor is velocity saturated. Although the reduction of the supply voltage increases the duration of the current linearly due to the increased rise/fall times, the peak magnitude of the current is reduced linearly (velocity saturation). Therefore, the average current is approximately constant, and the average power is just a linear function of the supply voltage (P=IV). For larger devices that are not velocity saturated, the average current is linearly proportional to the supply voltage so that the average power is a quadratic function of the supply voltage.

The short circuit power is calculated as follows:

$$P_{SC} = I_{SC}.V_{dd} \tag{2.4}$$

where I_{SC} is the average short circuit current. Short circuit power dissipation may be reduced to about 5 to 10% of the total power consumption of the logic circuit by proper sizing of the CMOS transistors to obtain equal rise and fall times [8].

2.2.3 Glitching Power Dissipation

Glitching power is the power dissipated in intermediate transitions during the evaluation of the logic function of the circuit. When the inputs to the logic gate are not synchronized, erroneous results occurs at the output node until all the inputs settle down to their final values [9]. These intermediate erroneous outputs lead to a power loss in charging and discharging the output load capacitance. It is difficult to calculate the glitching power because it is a function of the topology of the circuit, layout, delay, previous inputs, new inputs, and the gate type. Glitching power dissipation is expressed as follows:

$$P_{Glitch} = V_{dd}{}^2.C_L.F_{Glitch} \tag{2.5}$$

where F_{Glitch} is the average frequency of glitches. Glitching power may consume up to 40% of the total power dissipation of the circuit, especially architectures with large logic depth.

To avoid such power loss, designers may use synchronous circuits in which all the outputs are either latched or gated to synchronize the inputs to the next stage. Dynamic circuits also avoid the problem of glitching power by synchronizing the output with the clock signal. Finally, a careful layout may reduce the skew among the input signals to each logic gate leading to lower glitching activity.

2.3 Static Power Dissipation

Static power is usually a small fraction of the total power dissipation. Unfortunately, as the threshold voltage V_{th} decreases and the number of transistors per chip increases, the static power dissipation becomes more important. In digital circuits, there are three main sources of static power dissipation: diode leakage current of the transistor (in the OFF state), subthreshold current, and biasing current of some logic families.

2.3.1 Diode Leakage Current

Diode leakage occurs when a transistor is turned OFF, and another active transistor charges up/down the drain with respect to the former's bulk potential. In the case of an inverter with a high input voltage, the output voltage becomes "0" because the NMOS transistor is "ON". The PMOS transistor is turned OFF, but its drain to bulk voltage is equal to the supply voltage ($-V_{dd}$). The resulting diode leakage current is approximately

$$I_L = A_D.J_S \tag{2.6}$$

where A_D is the area of the drain diffusion and J_S is the leakage current density set by the technology. Since the diode reaches the maximum reverse bias current for a relatively small reverse bias potential, the leakage current is roughly independent of the supply voltage. The leakage current is proportional to the diffusion area and the perimeter of the drain. Therefore, it is preferred to minimize the diffusion

area and the perimeter in the layout. The leakage current density is exponentially proportional to temperature as well, so that J_S increases dramatically at higher temperatures [10].

2.3.2 Subthreshold Leakage Current

Subthreshold leakage occurs under circumstances similar to the diode leakage current. In the inverter described above, the PMOS is turned OFF. Even for $V_{gs} = 0V$, there is still current flowing in the channel because V_{ds} of the PMOS transistor $\approx -V_{dd}$. The I_d vs. V_{ds} characteristics has an exponential relation in the subthreshold region ($V_{gs} < |V_{th}|$). Figure 2.3 shows the subthreshold current magnitude at $V_{gs} = 0V$.

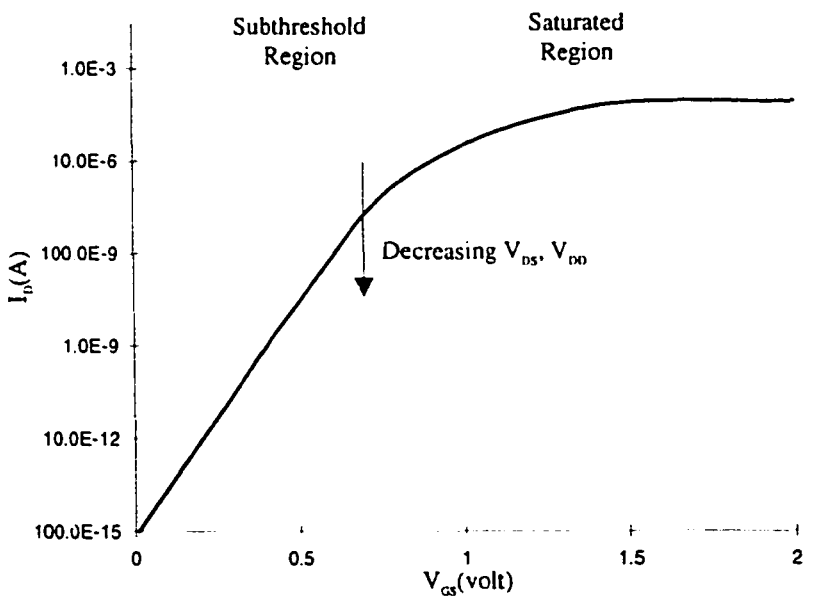


Figure 2.3: Leakage current in a CMOS logic gate

The magnitude of the subthreshold current is a function of the process, device size, and supply voltage [11]. Also, the threshold voltage V_{th} predominantly affects the value of the current, because reducing V_{th} increases the subthreshold current exponentially. Moreover, the subthreshold current is proportional to the transistor size W/L, and it is an exponential function of the supply voltage. Thus, the current can be minimized by reducing transistors sizes, and by decreasing the supply voltage.

2.3.3 Biasing Current

Although one of the main advantages of CMOS circuits is the absence of static biasing currents, which exists in the older NMOS circuits, some CMOS logic circuits still exhibit biasing currents for special purposes. For example, MCML (MOS Current Mode Logic) [2] circuits use biasing current to speed up the evaluation of the logic function by avoiding operation in the cut-off region [12]. Another example is pseudo-NMOS circuits , shown in Figure 2.4, which use this power to replace the whole PMOS block with only one PMOS transistor. This reduces the total number of transistors, hence capacitance which in turn lowers dynamic power dissipation. However, the sizes of the different transistors have to be calculated carefully to guarantee valid logic levels at the output node.

In other cases, where different t_{phl} (high to low transition time) and t_{plh} (low to high transition time) are required, pseudo-CMOS is used because of the simplicity of controlling the t_{plh} through the size of the PMOS transistor [13].

[2]Refer to chapter 3 for more details on CML circuits.

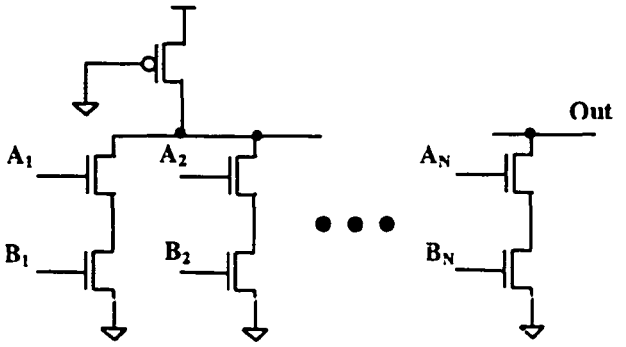


Figure 2.4: Pseudo-NMOS logic gate

2.4 Low-Power CMOS Logic Design

VLSI designers have different options to reduce the power dissipation in the various design stages. For example, the supply voltage may be reduced through fabrication technology, circuit design or dynamically through the system level. Switched load capacitance may be reduced through technology scaling, efficient layout, circuit design, gate level optimization, and/or system level.

Over the last decade, researchers have developed many techniques to reduce power dissipation in CMOS circuits. The gain obtained from each of these techniques depends solely on the application. Some of these techniques may degrade the performance or increase the area to reduce power. Other techniques reduce power, delay and area simultaneously. The rest of this chapter will describe some of these techniques and detail the pros and cons of each of them. This review will cover low-power design techniques on the following design stages: the process (device), the packaging, the layout, the circuit, the gate, the algorithm, the behavioral, and the system levels. Some of the low-power design techniques related to the thesis

will be explained in more detail.

2.5 Low-Power VLSI Technologies

Many modifications may be applied to the process technology in order to reduce power dissipation. These modifications include reducing the threshold voltage, reducing minimum gate length, and increasing the number of metal layers. Power dissipation may also be reduced by using alternative fabrication technology other than the CMOS process.

This section reviews low-power CMOS technologies and presents some of the alternative fabrication technologies for low-power design.

2.5.1 Threshold Voltage Reduction

Until recently, the V_{th} in most CMOS processes has been set to a fairly high potential: 0.7V to 1.0V. For 5V circuit operation, this has little impact on circuit delay, which is inversely proportional to $(V_{dd} - V_{th})^2$. The main benefit of such a large threshold is that the subthreshold leakage is reduced exponentially. While the total leakage current of a chip is still well below the average supply current under operation, the reduced subthreshold current prolongs the duration of the stored charge in dynamic circuits, providing more robust operation (due to longer leakage times). Thus, there has been less tendency to reduce the thresholds until recently, with the decrease of supply voltages to 3.3V, and the emphasis on low-power design.

The reduction of V_{th} enables VLSI designers to lower the supply voltage. This maintains circuit speed and results in a power reduction. However, the limitation

of this technique is that at low thresholds, the subthreshold currents become significant, if not dominant, portion of the average current drawn from the supply. Previous work has shown the optimal V_{th} to range from 0.3V down to below 0.1V depending on the conditions of the circuit operation [1], [14].

2.5.2 Technology Scaling

With every new process generation, all of the lateral and some of the vertical dimensions of the transistor are scaled down. This has an immediate impact on reducing power dissipation, as well as increasing circuit speed. The primary effect of process scaling is the reduction of all the capacitances, which provides a proportional decrease in power and circuit delays. Device sizes may be reduced to keep the delay constant over process scaling, which yields an even larger power reduction [15].

Both gate capacitance and interconnect capacitance may be expressed as $C = W.L(1/t_{ox})$. The width W, length L, and oxide thickness t_{ox} all scale almost equally by a factor s, so that the total capacitance scales down by the same factor s. Diffusion capacitance is a more complex function of process scaling; however it is reduced by a factor between s and $s^{3/2}$. For a constant supply voltage, both the power and circuit delays scale down approximately by the factor s. Thus, power reduction is accomplished with no alterations in the circuit design.

As mentioned earlier, not all the vertical dimensions scale down. In particular, the thickness of the interconnect metal is roughly the same across the processes, due to fundamental processing requirements [16]. This increases the fringing capacitance from the side of the metal to the substrate, and increases the capacitance be-

tween adjacent interconnect segments. With these secondary effects considered, the overall capacitance scaling is somewhere below the factor s, and is difficult to accurately characterize without using a three-dimensional simulation model. New technologies from IBM and Motorola use copper instead of aluminum for interconnects, because copper has better conductivity and scales down better than aluminum[17]. Technology scaling trends and their effect on logic circuits are examined in more detail in chapter 3.

2.5.3 Increasing Number of Metal Layers

Further power reduction may be achieved by using some features in today's more advanced processes; namely, an increased number of metal layers and a trend towards allowing stacked vias. If these are used wisely, not only can the power be reduced, but also the circuit area, and delay times. However, utilizing these advancements requires special circuit redesign methodologies.

In old fabrication technologies with two metal layer, polysilicon has been used extensively in intercell signal routing, as second-level metal has been reserved for intercell routing to allow the CAD tools to perform global routing. In present technologies with more metal layers, the second-level, and perhaps higher metal layers, can be used for intercell routing. Since the capacitance per unit area decreases with each higher metal level, using the higher metal layers helps reduce the interconnect capacitance, which already contributes around 30% to the overall capacitance. That percentage is expected to increase with future VLSI generations [16].

Also, by allowing stacked vias, the areas of the different gates can be compressed. Moreover, this reduces both the intercell and global routing because the terminal connections will be closer. Consequently, most interconnect routes will be reduced in length. However, condensed routing increases the coupling capacitance between interconnects, and cancels part of the power savings previously achieved.

2.5.4 Alternative Technologies

If the current rate of scaling MOSFET's were to continue, devices with lengths of 1nm would be in use in the year 2040. A nano-meter of oxide consists of only a few layers of atoms which approaches fundamental limits.

Therefore, some new device structure will eventually replace the devices being used today. Predictions of when this will happen have consistently underestimated the ingenuity of fabrication engineers, and the conclusion is that the end of CMOS scaling is still too far away to accurately predict [18]. However, the limits which are driving current device scaling can give some insight into what new technology might eventually replace today's devices. The following technologies seek to address the limitations of CMOS MOSFET's by allowing further reductions in the supply voltage and therefore further scaling of the device dimensions, by providing improved performance at the same device dimensions, or both.

Silicon on Insulator (SOI)

The elimination of junction capacitance gives SOI improved performance at the same device dimensions, and improved subthreshold slope allows for further device

scaling [19]. The floating body of SOI devices is a concern for reliability and circuit simulation. It also causes these devices to have low breakdown voltages which has been a major roadblock to their use in the past, but as supply voltages continue to scale down this may present less of a problem.

Multi Threshold Voltage (MTCMOS) Devices

By using low threshold devices for circuits on the critical path and high threshold devices elsewhere, it may be possible to achieve high performance, while maintaining reasonable leakage currents [20]. However, very low threshold devices may not be suitable for the dynamic circuits, which are usually used in high speed applications. This will be explained in detail in Chapter 5.

Low Temperature CMOS (LTCMOS)

Reducing the chip's operating temperature can enhance the performance due to improved carrier mobility and reduced wire resistance [21]. It also reduces leakage current which increases exponentially with temperature. This would allow technologies to be scaled to smaller dimensions. The disadvantage is the increased size and cost of the system.

Dynamic Substrate Biasing

Designing an on-chip circuit which dynamically varies the substrate voltage in order to compensate for threshold variations [22] helps reduce the leakage current. Dealing with temperature and process variations across a single die would require

several of these circuits to control isolated substrate regions. In such chips substrate noise would be a major concern.

New Gate Oxide Materials

Replacing SiO_2 with a higher permittivity material would allow thicker gate oxide films to provide the same control of the channel. Fields in the oxide layer would be reduced, making breakdown less of a concern, and tunneling currents would be reduced. The difficulties with this approach are depositing a very thin very high quality oxide layer rather than simply growing a thermal oxide. The $Si - SiO_2$ interface has been the subject of intensive study for decades and moving to a different material would be a very significant change in the fabrication process.

2.6 Low-Power Packaging

Wire bond packaging has been widely used in chip fabrication. Unfortunately, the wiring bonds and the package pins have large parasitics (2 to 3 orders of magnitude compared to the chip parasitics). These parasitics increase the power and the delay of the chip, leading to slower and less dense I/Os. The package by itself is expensive, heavy and large (4 to 100 times the area of the die). Because the pads in wire bond packages are limited to the circumference of the die, a limited number of pins are possible and the chip becomes pad-limited. Recently, the demand for compact designs with less weight, power dissipation, and delay has contributed to the development of advanced chip scale package (CSP) technologies [23].

MultiChip Module (MCM) is a new packaging technology for attaching and con-

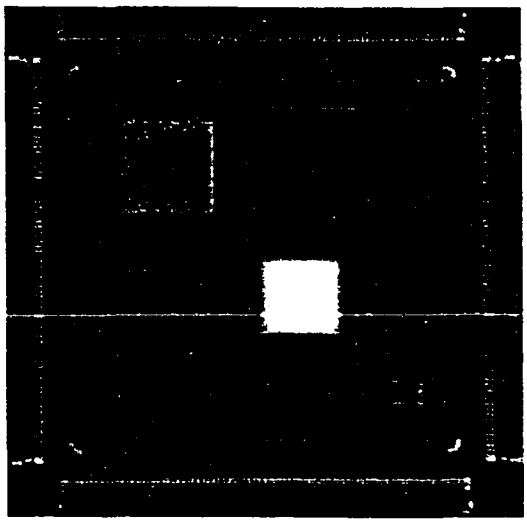


Figure 2.5: The MCM for the Sun Viking, consisting of CPU, cache controller, and eight SRAM mounted on a 1.9"x1.9" copper-polyimide substrate

necting chips onto a silicon membrane [24]. MCM utilizes conventional IC processing techniques, and offers a high density (over 500 per chip) of small, low resistance and low parasitic interconnects between the chip and the substrate. Figure 2.5 is a photo of an MCM package for a SuperSparc CPU from SUN microsystems.

MCM packaging is similar to a printed circuit board (PCB). The difference is that a PCB connects packaged chips whereas MCM connects the silicon dies without packaging. When MCMs are used, the chip interconnect parasitics becomes of the same order as the internal chip parasitics. Therefore, more complex routing is possible, leading to less power dissipation and higher operating frequencies. One of the problems encountered in MCM packaging is the fine placement of the dies on the substrate. Therefore, new placement technologies with very high resolutions have been developed specially for MCM placement.

Flip chip technology is another packaging technique that avoids both the bond-

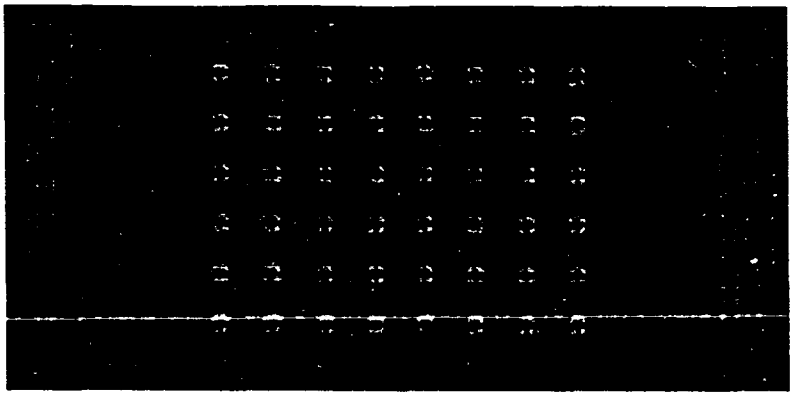


Figure 2.6: Solder bumps of a flip chip

ing wires and the extra parasitics associated with the package [25]. Flip chip is a form of semiconductor packaging that uses solder bumps as the interconnections between the integrated circuit (IC pads) and the substrate as shown in Figure 2.6. Unlike wire bond packaging where the pads have to be placed on the circumference of the chip, flip chip pads may be placed anywhere on the chip, leading to shorter on chip routing. Also, the whole chip/package area may be covered with pads leading to larger number of I/O pins. Examples of flip chip technologies are Chip On Board (COB) and Ball Grid Array (BGA) [26].

2.7 Low-Power Layout techniques

The layout process controls the interconnect parasitics of the design. Usually, layout techniques optimize the area rather than power dissipation. Some modifications have been suggested to reduce power by manipulating the layout.

Floorplanning

Floorplanning is the process of allocating chip area for the placement of the top level blocks, placing the I/O pads and power rails. Traditional floorplanning has reduced the overall chip area. Low-power floorplanning have been suggested to reduce the length of the interconnect wires carrying signals with high switching activity [27]. Careful floorplanning may result in up to 20% power savings.

Placement

Placement is the process of assigning the gates to locations inside each block. Placement is either area driven or timing driven. A switching activity driven placement algorithm produces a power gain of 8% [28].

Clock tree generation

In older systems, a large clock buffer has been used to distribute the clock signal all over the chip, and to avoid large clock skews. This scheme has lead to a large power consumption in the clock buffer and wide interconnects for the clock distribution network. In [29], a balanced buffer insertion algorithm is suggested to reduce the clock skew. The new algorithm partitions the top clock tree into sub-trees to achieve a power reduction up to 60%.

Routing

Most routing algorithms attempt to reduce the total wire length. The length of high switching activity signals may be reduced by assigning a higher priority for

those wires before the routing process. Experimental results of the modified routing algorithm have obtained a slight power reduction compared to the original algorithm [30]. This has occurred because the change in the wire length inside each block is small, which leads to small variation in parasitic capacitance.

2.8 Low-Power Circuit Techniques

Since switching power is the dominant power sink in CMOS circuits, several techniques have been proposed to reduce this power dissipation component. Equation 2.1 shows that power may be reduced by lowering the voltage component V_{dd}^2 or by reducing the average switched capacitance $C_L.f_{switching}$.

2.8.1 Supply Voltage Reduction

Clearly, reducing the voltage supply yields the largest power reduction, due to the squared term (if V_{dd} is scaled down, V_{swing} is usually scaled down too) [31]. A change from a 5V supply to a 1.5V supply yields a 90% reduction in power dissipation. However, the trade-off is an increased circuit delay. Equation 2.7 gives an expression for the delay t_d as a function of V_{dd} (assuming $V_{swing} = V_{dd}$). The V_{th} term causes the delay to increase rapidly for supply voltages near the threshold voltage. Even at a supply voltage of 3.3V, the delay is 23% greater than the ideal case ($V_{th} = 0$V).

$$t_d = \frac{C_L}{I_{Avg}} \frac{V_{dd}}{2} = \frac{C_L.V_{dd}}{K.(V_{dd} - V_{th})^2} \tag{2.7}$$

2.8.2 Reduced Output Voltage Swing

For a further power reduction, the output signal swing can be reduced to a value less than the supply voltage. Since the delay is proportional to signal swing (V_{swing}), reducing the signal swing linearly decreases the delay, as well, for constant I_{Avg} [32].

Some logic circuits have a natural reduced swing like CML and NMOS gates. To limit the swing of any static or dynamic CMOS circuit that has a rail to rail swing, extra circuitry is required as shown in Figure 2.7. This extra circuitry adds parasitic capacitances that add to the total effective capacitance being switched. However, the total energy is reduced because the voltage swing has been reduced. As long as the reduction in voltage swing is greater than the increase in capacitance, the energy and power will be reduced [33].

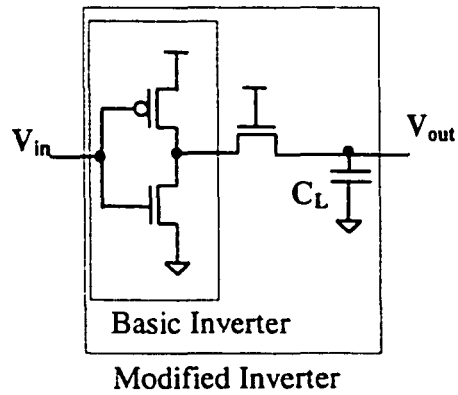


Figure 2.7: A Reduced Swing CMOS Inverter

2.8.3 Reducing The Physical Capacitance

Digital circuits have three types of capacitance: gate capacitance, diffusion capacitance, and interconnect capacitance. If all three components are scaled down by the same factor, then the net power dissipation is scaled down as well. Gate and diffusion capacitances are fixed during cell design, whereas intercell and global interconnect capacitances are controlled by the CAD tools performing the global routing.

If all cells are scaled down by a certain factor, then the total area will be scaled down by the same factor, as well as the interconnects, because of the shorter distance between terminals. However, if the final layout is interconnect limited, the total area will be dominated by the interconnects and the cell area reduction will not reduce the chip area. Usually, the layout is cell limited [33].

Physical capacitance is reduced mainly by transistor sizing. By reducing the width of the transistor, the gate capacitance will decrease and the current passing through the transistor will decrease. Since this current is used to drive the load capacitance, the overall throughput will be the same. However, if the interconnect capacitance is comparable to that of the gate and diffusion, more delay will be added to the gate, and the power has to be compromised with the delay [34].

For large synchronous systems like microprocessors, clock signals consume a significant portion of the chip power [35]. It is important to minimize the number of global clock nets, as well as all the gate capacitances connected to the clock net. To accomplish this, the TSPC (True Single Phase Clocking Scheme) is used as shown in Figure 2.8 [36]. TSPC reduces the number of clocking trees to only

one instead of the two non overlapping clocks phases used before. This provides a 50% reduction in the clock power.

However, TSPC has some drawbacks such as minimum operating frequency, because of the dynamic registers used. Unlike regular CMOS circuits, the TSPC circuits require a minimum clocking frequency during standby which means more standby power dissipation. A modified version of the register exists with two extra transistors, that provide static feedback while the clock is low. Another side-effect of TSPC circuits is that internal glitching occurs at the input of the inverter on the rising edge of the clock [37].

Efficient layout of the gate itself is very important to reduce the size of the gate and interconnect length, which yields smaller capacitance.

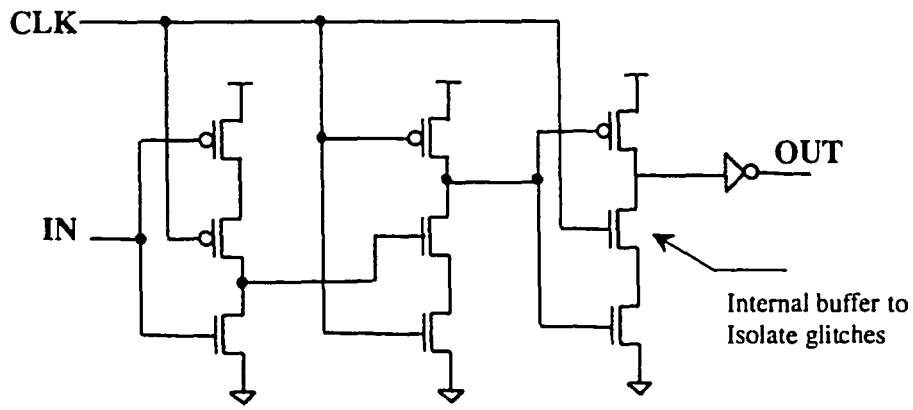


Figure 2.8: True Single Phase Clocking Scheme

2.8.4 Reducing The Switching Frequency

Reducing the number of "0" $\rightarrow$ 1 power dissipating transitions minimizes the switching power dissipation of the gate. Switching frequency may be reduced on several

levels in the design process, beginning from the circuit design up to architectural and algorithmic design [1].

There are several logic styles to design with. Some of these styles are: static CMOS, CPL, MCML, and a variety of dynamic logic styles. Generally, most logic styles perform delay-power tradeoffs, but not always in proportional amounts. The best style is that which minimizes power dissipation given a constant throughput (delay).

Another important factor that has to be considered is that the switching activity varies greatly between logic styles. Static CMOS outputs transition only upon an input transition, whereas dynamic logic styles incur output transitions during input transitions, as well as during the precharge phase of every clock cycle. In addition, the clock nodes on dynamic circuits have a power-dissipating transition every cycle, too. Also, circuits that have a large logic depth, such as multipliers, suffer glitching transitions, because the signals arrive at the gate inputs at different instances in time.

To avoid high glitching activity, self-timing scheme is used to latch the output of each cell when its the data is ready and signal the following logic level(s) to start evaluation. This is only useful on large cells/modules, and is used for the FIFO and memory designs. Extra hardware is required to implement self-timing, which requires extra power dissipation. However, for a large module, the overhead is a small fraction of the total power dissipation. The full benefit of self-timing is achieved, if the module output goes off-chip (very high capacitance), or sits on a high fan-out internal bus.

2.9 Low-Power Gate Level Design

Gate level design is the process of transforming the RTL (Register Transfer Language) code into a gate level netlist. The following is a description of some algorithms used to reduce the power dissipation at the gate level.

2.9.1 Low-Power Synthesis

During the synthesis of the behavioral code, many techniques may be used to reduce power dissipation. The main objective of these techniques is to reduce the effective switched capacitance by reducing either the load capacitance or the switching activity. The following are some examples of low-power synthesis.

Don't-care-sets

Don't-care-sets are the input combinations that would never exist during normal operation. These sets are mainly used to reduce the logic expression, hence, the number of logic gates. In [38], a technique to utilize the Don't-care-sets to reduce switching activity is presented. The new algorithm reduces the over all power by an average of 10%.

Common sub-expression elimination

This is a technique used to reduce the area of multi-level gate netlists. In [39], a modified scheme to minimize power dissipation using sub-expression elimination is presented. Based on an evaluation of power saving for each expression, the

algorithm decides which expressions to be eliminated. Researchers have reported a 12% power reduction using the modified scheme.

State decoding

State decoding is the assignment of a binary code to each logic state in the finite state machine (FSM). In [40], a new approach to reduce power by optimizing the state assignment is presented. The approach relies on reducing the distance between the states that occur inside loops. This approach may obtain up to 17% power reduction.

Gate sizing

Gate sizing is the process of choosing the ideal gate size to reduce power, delay and area. A cost function for power, delay, and area is considered. Depending on the design criterion, different weights are assigned to each of the the three parameters. Different optimization algorithms have been suggested to optimize this cost equation, which has lead to a power reduction of up to 25% [41] .

Rescheduling

Rescheduling is used to reduce glitching and to allow higher clock frequency by reducing the logic depth inside each pipeline stage. In [42], extra stages are added to the pipeline to reduce the overall power. This approach has reduced the power by 8%.

2.9.2 Clock Gating

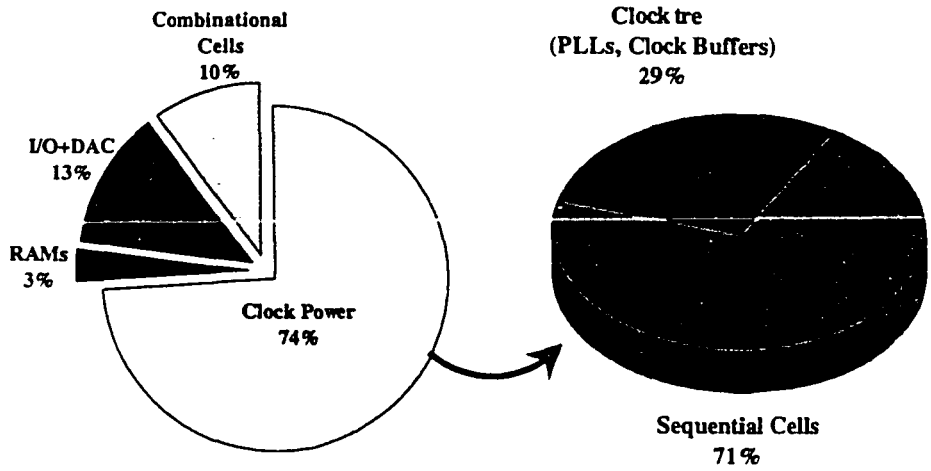
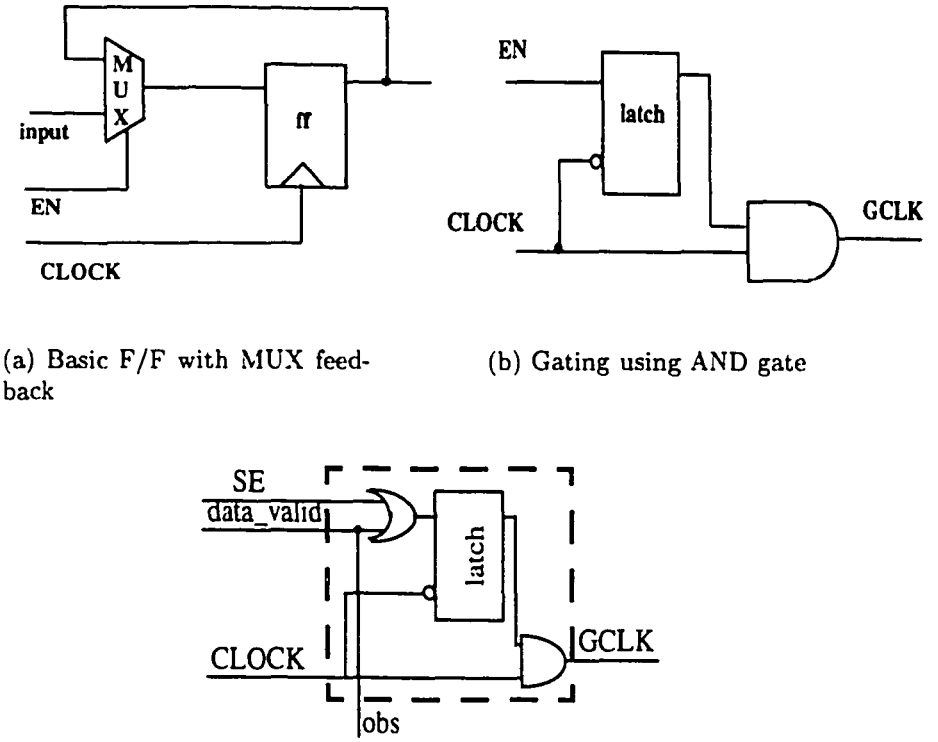


Figure 2.9: Power dissipation of an Intel740 graphics chip

For most digital systems, the clock tree consumes the largest fraction of power. Figure 2.9 shows power consumption of various sections of an Intel740 graphics chip, where the sequential cells are either flip flops or latches connected directly to the system clock. Figure 2.10.a shows how the flip flop is usually connected using a MUX and an enable signal (EN). If the enable signal is "1", the input data is stored in the F/F. Otherwise, the enable signal is low and the F/F stores the last output again. In the last case, although the output value of the F/F does not change, a considerable amount of energy has been lost inside the F/F to restore the same value.

Clock gating activates the F/F only if a new input should be stored in the F/F. Figure 2.10.b illustrates a clock gating cell that uses an AND gate to disable the clock input to the F/F (GCLK) when the enable signal is low. The latch is used to avoid glitches on the clock input of the F/F. This gating cell increases



(a) Basic F/F with MUX feedback

(b) Gating using AND gate

(c) Testing of gated clock F/Fs

Figure 2.10: Clock gating

the area and power of the F/F. Usually, a set of flip flops are grouped together to construct a register with one enable signal. For such case, the clock gating cell may be used for the whole register leading to less overhead in terms of power and area. Therefore, clock gating is preferred for registers rather than separate F/Fs. The minimum width of a register to be clock gated is a fixed number for each standard cell library, and is determined through simulations.

However, clock gating adds more complexity to the testing process of the system because the gating block is not transparent; i.e., it is not easy to scan the data in

and out of the chip. In order to simplify the testing process, the clock gating cell is modified as shown in Figure 2.10.c. The new cell has a Scan Enable (SE) signal that is activated during the testing phase to make the clock gating cell transparent.

Another problem with clock gating is the clock skew. Clock tree generation software balances the clock up to the input of the clock gating block. However, the delays from the gating block to the clock inputs of the different F/Fs are not equal, leading to hold time violations. Careful layout may reduce such problems by assigning high priority to the gated clock signal to reduce the skew between the different F/Fs. Up to 40% power reduction is reported using clock gating.

2.10 Low-Power Behavioral Design

Behavioral design is the mapping of system block levels into RTL code using an HDL language (hardware descriptive language). At this design stage, power dissipation may be reduced by using the following techniques: parallelism and pipelining, glitch reduction, multiple supply voltages, non overlapping clocks, and reducing the memory accesses.

Parallelism and Pipelining

Power dissipation does not scale linearly with delay. Figure 2.11 shows the power dissipation vs. the delay for a 32 bit CLA adder. While optimizing the circuit for speed reduces the delay by 27%, it also increases power dissipation by 280%. However, if two separate adders were implemented with a delay of 1.5ns each, a better throughput with less power dissipation may be obtained. This is called

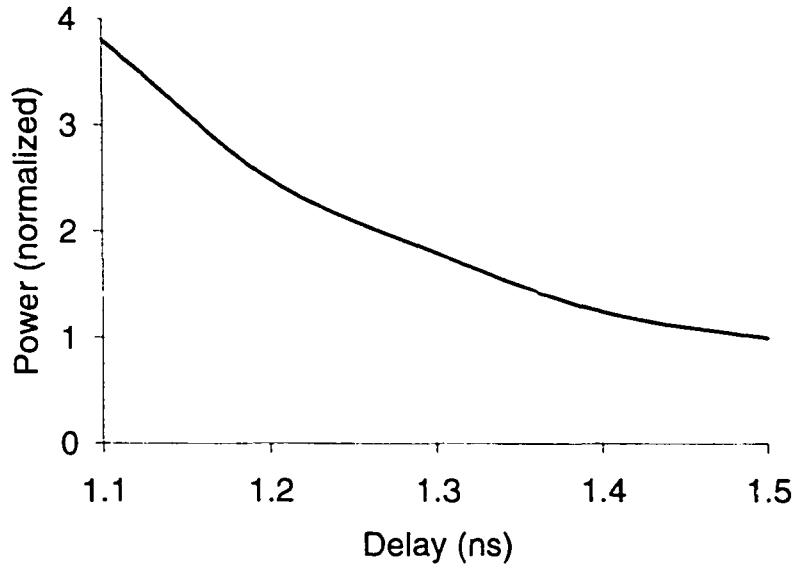


Figure 2.11: Power vs delay

parallelism, where critical modules are replicated with relaxed delay restrictions, and connected in parallel. Parallelism trades area for power to achieve better delay characteristics, and to simplify the design of critical path blocks because of the relaxed delay requirements. Parallelism usually requires a special instruction set and a special programming style to benefit from the parallel hardware [43].

Pipelining is similar to parallelism in concept. The main difference is that pipelining partitions the logic function into cascaded series of events. For example, a MAC (Multiplier- ACcumulator) may be partitioned into a multiplier stage followed by an adder stage. Pipeline stages are connected using registers that synchronize the data transfer from one stage to the other based on the clock signal. The effective throughput of the pipeline is equal to the clock frequency. However, the pipeline has a latency (time lapsed from first input to first output) equal to the

number of stages in the pipeline. Most modern processors and microcontrollers utilize pipelining especially in instruction fetching and decoding.

Glitch reduction

This is an automated algorithm to reduce glitches in the control signals [44]. The algorithm relies on balancing the delay between data signals to reduce the glitches. This is possible by adding an extra delay to fast signals. On the average, power savings of 17% is obtained.

Multiple supply voltage

Lower supply voltages may be used in the non-critical paths to reduce power [45]. A dynamic programming approach is used to solve the scheduling of multiple supply voltages problem. Power savings of up to 50% is reported.

Non overlapping clocks

This scheme creates multiple non overlapping clock signals from the original clock signal. Each of the new clocks has the same frequency of the original clock, but with a different phase shift. Each of the new clock signals is used to trigger a different block. Therefore, at each instant of time, only one block is operating while the others are inactive leading to a power reduction up to 20% [46]

Reducing memory accesses

System on chip (SOC) usually utilizes large memory blocks which contribute to the overall power dissipation. Memory accesses are power consuming, because of

the address decoding process and the bit line drivers. Therefore, it is important to reduce the number of memory accesses in order to reduce power dissipation. In [47], a simulated annealing technique is used to allocate variables to the memory blocks. This algorithm has reduced the power dissipation by 47%.

2.11 Low-Power System Design

System level is the highest level of abstraction in the digital design cycle. Modifications on the system level have the most impact on the quality of implementation [1]. Low-power system design techniques are critical because of their effect on design analysis, verification, synthesis, automated layout and testing. Typical system level design involves decisions on power management techniques, clock strategy, parallelism, pipelining, memory sizing, etc.

Sleep and power down modes

The most effective power management approach is to power down some blocks in the system when they are not used. A power management logic block is used to determine which blocks should be disabled during each clock cycle [4]. Motorola's PowerPC 603 microprocessor utilizes three different power saving modes. The first mode turns most of the blocks OFF except the cache memory to maintain coherency, yielding a power reduction of 85%. In the second mode, the cache is turned OFF, increasing the power savings to 94%. In the last mode, the whole clock is turned OFF and the system requires an external signal to start over again. This mode reduces power dissipation by about 96%.

Performance management

Since the system does not work at full load all the time, some techniques are used to reduce power when the system is not working at full throttle [48]. A workload detection circuit is used determine the required throughput. Based on the required workload, the clock frequency or the power supply may be reduced.

Adaptive filtering

Normally, a digital system is designed to work properly at the worst operating conditions. For example a communication system would be designed for the lowest signal to noise ratio (SNR) possible. However, most of the time the system has higher SNR ratio. Therefore, the resolution of the computations may be reduced without affecting the performance of the system. This is done by designing the data path in a bit-slice fashion ,and depending on the available SNR, some of the LSB bits are turned OFF [49].

Dynamically varying the threshold voltage

This is used to reduce the supply voltage and leakage power, simultaneously. This approach requires multi-tub fabrication technology. During a full system load, the nwell(s) and pwell(s) are biased in such a way as to reduce the effective V_{th} to allow quicker performance [50]. During standby, the biasing is changed to increase the V_{th} and to reduce leakage power. Up to 97% power reduction may be obtained using dynamic V_{th} variation.

2.12 Summary

The first part of this chapter introduced the need for low-power design. A description of the main power dissipation mechanisms in CMOS logic circuits has been given. The effects of technology scaling, operating conditions and circuit implementation on power dissipation have been discussed also.

The rest of the chapter has presented some low-power design techniques to emphasize the different degrees of freedom available to digital CMOS designers to reduce power dissipation. The advantages and disadvantages of each low-power design technique have been pointed out. The most effective approaches reduce supply voltage V_{dd}, and use another technique (like parallelism or dynamic variation of V_{th}) to substitute for performance degradation. Other techniques have targeted the switched capacitance either by reducing the physical capacitance or the switching activity. The use of low-power design methodologies at different design stages is preferred in order to achieve a better overall power reduction.

This thesis deals with low-power design on the technology, circuit, and algorithmic level. In each chapter, a review of the state of the art approaches related to each proposed scheme will be given.

Chapter 3

Effect of Technology Scaling on CMOS Logic Styles

Since the invention of the first Integrated Circuit (IC), CMOS technology has continued to scale down at a dramatic rate. Fueled by the search for faster, cheaper and more compact electronic systems, manufacturers has increased the number of transistors per chip more than 100,000 times over the last twenty five years.

The first part of this chapter outlines the history of technology scaling, the trends and limitations of technology scaling, and the various phenomena associated with smaller transistor sizes. The second part of the chapter reviews five of the most renowned logic styles; namely, CMOS, CPL, Domino, DCVS and MCML. The effect of technology scaling on the performance, power, and area of the five logic styles is then presented along with predictions for future trends. To verify the qualitative analysis, the results from simulating a large number of gates and circuits are examined. In these simulations, each logic family is implemented using

four different fabrication technologies (0.8, 0.6, 0.35, 0.25 μm CMOS technologies). The results are then analyzed, compared and summarized.

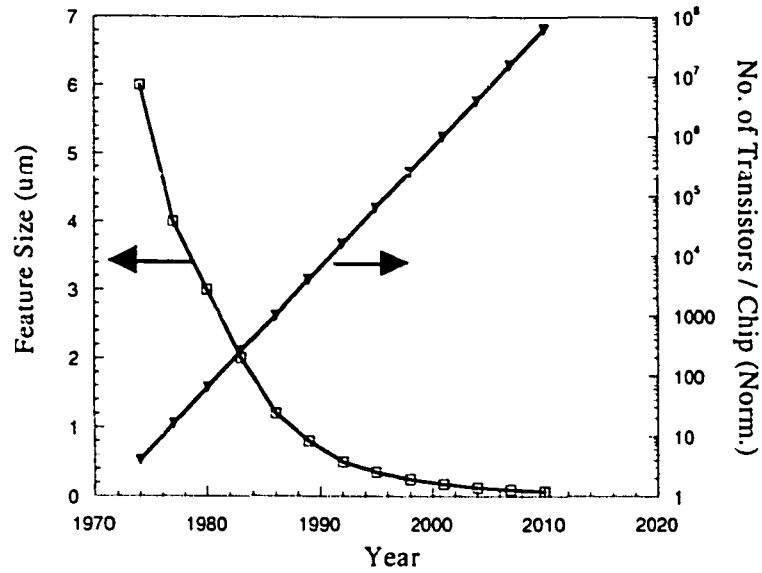


Figure 3.1: Historical trends of LSI's

3.1 Trends and Limitations of CMOS technology Scaling

Device scaling has governed the evolution of CMOS technology. For the past twenty five years, every three years, the minimum lithographic feature size has been decreased by 0.7 times; the chip size has increased by 1.5 times, and DRAM capacity (bits/chip) has increased by 4 times [51]. In 1975 , Moore [52] predicted that the number of transistors that could be integrated on a single die would increase exponentially. This observation has held true for more than 20 years as shown in Figure

Table 3.1: Historical trends and SIA road map of LSI's

Year	Size(μm)	Shrink rate	Normalized IC Complexity (Transistors/chip) 1998 roadmap	Year expected in 1994 roadmap
1974	6.0	-	1	-
1977	4.0	0.67	4	-
1980	3.0	0.75	16	-
1983	2.0	0.67	64	-
1986	1.2	0.60	256	-
1989	0.8	0.67	1K	-
1992	0.5	0.63	4K	-
1995	0.35	0.70	16K	-
1997	0.25	0.71	64K	1998
1999	0.18	0.72	256K	2001
2002	0.13	0.72	1M	2004
2005	0.10	0.77	4M	2007
2008	0.07	0.7	16M	2010
2011	0.05	0.71	64M	-
2014	0.035	0.7	256M	-

3.1, and it is expected to remain valid for another 20 years.

In 1994, the Semiconductor Industry Association (SIA) published a technology roadmap of semiconductors [53]. The projections to the dimensions of each future technology generation have been given until the year 2010. Because of the faster than expected advances in process technology, SIA has modified the roadmap in 1998. Table 3.1 shows the historical trend of CMOS technology and the 1994 and 1998 roadmaps.

This section explains the various trends and limitations in CMOS technology scaling.

3.1.1 MOSFET Scaling Trends

For digital circuit design, the ideal MOSFET transistor is a perfect switch. It should conduct an infinite current in the ON state, and zero current in the OFF state. Scaling the device dimensions has effectively increased the ON current of the device by decreasing its threshold voltage, but at the same time it has caused an increase in the OFF current. For example, an NMOS transistor with its drain connected to (V_{dd}) and its source, gate, and bulk connected to (GND), should not have any current flow. However, for a submicron device, there are significant drain currents, to the source as subthreshold leakage current, to the gate as tunneling current, and to the bulk as gate induced drain leakage. The need to minimize these leakage currents and maximize the ON current, simultaneously, has governed the scaling of MOSFET technologies.

Another characteristic of an ideal switch is an infinite lifetime. Unfortunately, MOSFET devices tend to degrade when exposed to high electric fields in either the gate oxide, or the channel. High field phenomena such as dielectric breakdown, electron migration, and hot carrier effects, have gained interest because they can cause a chip to suddenly fail after operating correctly for months, or even years. Therefore, reliability concerns have further limited practical device designs.

Table 3.1.1 illustrates a generalized scaling scheme, where S_L , S_T and S_V present the scaling factors for the channel length, gate oxide thickness, and power supply, respectively.

The gate capacitance is proportional to WL/T_{ox}, which corresponds to a scaling factor of $S_T/{S_L}^2$. The current may be expressed as $= (W/L)(1/T_{ox})V^2$, which

Table 3.2: A Generalized scaling scheme

Parameter	General scaling
Device dimension	$1/S_L$
Supply voltage	$1/S_V$
Oxide thickness	$1/S_T$
$C_{ox} = (\epsilon Area)/T_{ox}$	S_T/S_L^2
Gate capacitance $= WL/T_{ox}$	S_T/S_L^2
Current $= (W/L)(1/T_{ox})V^2$	S_T/S_V^2
Delay $= CV/I$	S_V/S_L^2
Power dissipation $= CV^2F$	S_T/S_V^3
Power Delay Product	$S_T/S_L^2S_V^2$
Energy Delay Product	$S_T/S_L^4S_V$

decreases by S_T/S_V^2. The delay, realized as CV/I, is scaled by a factor of S_V/S_L^2. Since the delay is the inverse of the operating frequency, then the operating frequency increases by a factor of S_L^2/S_V, while the Energy Delay product (EDP) is reduced by $S_T/S_L^4S_V$. EDP is used as a metric to describe the performance and power dissipation, simultaneously.

It is obvious that the most effective method to reduce delay and thus the EDP, is by scaling down the device dimensions.

In practice, there are two main scaling schemes for MOSFET devices. The first scheme, proposed by Dennard et al. in 1974 [54], is called constant field scaling. In Constant Electric field (CE) scaling, all of the horizontal and vertical dimensions are scaled with the power supply to maintain constant electric fields throughout the device. The second proposed scaling scheme, called Constant Voltage (CV) scaling, has been proposed by Chatterjee et al. [55]. Constant Voltage scaling maintains a constant power supply and scales down the gate oxide thickness more gradually to slow the growth of fields in the oxide. Table 3.1.1 summerizes these two scaling

schemes with first order approximations.

Table 3.3: Influence of scaling on MOS device characteristics

Parameter	Constant Electric field (CE)	Constant Voltage (CV)
Device dimension	$1/S$	$1/S$
Supply voltage	$1/S$	1
Oxide thickness	$1/S$	$1/S$
Electric field across gate oxide	1	S
$C_{ox} = (\epsilon Area)/T_{ox}$	$1/S$	$1/S$
Gate capacitance $= WL/T_{ox}$	$1/S$	$1/S$
Current $= (W/L)(1/T_{ox})V^2$	$1/S$	S
Delay $= CV/I$	$1/S$	$1/S^2$
Power dissipation $= CV^2F$	$1/S^2$	S
Power Delay Product	$1/S^3$	$1/S$
Energy Delay Product	$1/S^4$	$1/S^3$

For CE scaling, Table 3.1.1 shows that if the device dimensions, supply voltage and gate oxide thickness are scaled by a constant parameter S, I_{ds} (the drain to source current) decreases by S. On the other hand, for CV scaling, current increases by S. Another characteristic illustrated in Table 3.1.1 is the scaling of power and EDP. For CE scaling, the switching power dissipation decreases by $1/S^2$, while it increases by S in the CV case.

CMOS technologies are usually optimized to reduce the EDP value [56]. With CV scaling, the EDP metric is scaled by $1/S^3$ while CE scaling scales the EDP metric by $1/S^4$. Clearly, constant electric field scaling provides a better EDP reduction. However, if the V_{th} is scaled as well, it increases the subthreshold leakage power. Comparing EDP for a large number of published devices [57], shows that each new technology generation demonstrates about 10x reduction in the EDP value compared to the generation before.

Intel has used CV scaling in their microprocessors until the 0.8μm technology, where a 5V supply voltage has been used in order to maintain compatibility with the supply voltage of conventional systems, and also in order to obtain higher operation speed. The CE field scaling has been used since the 0.5μm technology evolved.

The main reason for the supply voltage reduction that started in the 0.5μm generation, is that the electric field across the gate oxide would have exceeded the maximum limitation for dielectric breakdown due to the scaling down of the oxide thickness; as well as injecting hot carriers into the gate oxide and the existence of electromigration. All the previous effects have a deleterious impact on chip reliability, which will be illustrated later on. Another reason for reducing the supply voltage has been to decrease the power consumption of the chip.

However, it is not easy to reduce the supply voltage now, because of difficulties in scaling down the threshold voltage of MOSFETs. A too small threshold voltage leads to significantly large subthreshold leakage currents even at a gate voltage of 0V. The lowering of the supply voltage also deteriorates the speed of the circuit. For these reason, aggressive reduction of gate oxide thickness is desirable. Figure 3.2 shows the trends in the scaling of the supply voltage, threshold voltage and the oxide thickness.

It is important also to note that reducing the supply voltage does not guarantee to decrease the chip power dissipation. The main reason is that chip designers leverage VLSI in high performance products by using higher densities to fill the chip with more wires, more fan out logic, and greater memory capacity. Thus, chip power dissipation may increase.

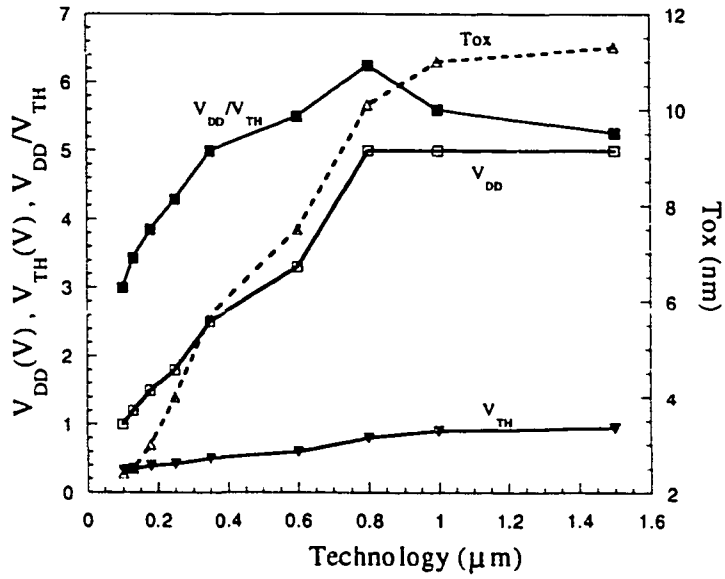


Figure 3.2: Trends of scaling V_{dd}, V_{th} and T_{ox}

3.2 Challenges of MOSFET Scaling

This section describes the various factors that limit the scaling of the MOSFET devices. These factors are: short channel effects, subthreshold leakage currents, gate induced drain leakage, gate tunneling, dielectric breakdown, hot carrier effects, and interconnect scaling.

3.2.1 Short channel effects

In digital circuit applications, a MOSFET transistor functions as a switch. Thus, it is preferred to have a complete cutoff in the OFF state, and a low resistance or high current drive in the ON state. When scaling down the gate length, even in the OFF state, the space charge region around the reverse biased drain depletes significant portion of the channel. This reduces the charge in the channel which is

controlled by the gate. Therefore, the threshold voltage is reduced, resulting in a high leakage current from the drain to the source via the space charge region, as shown in Figure 3.3.

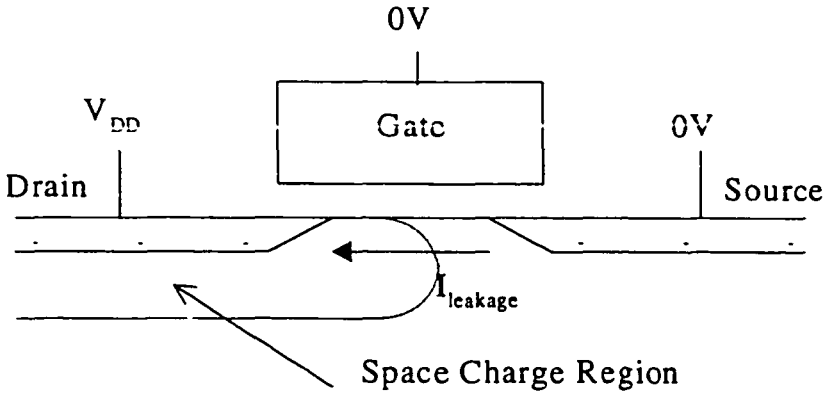


Figure 3.3: Short channel effects

The device threshold voltage is dependent on the effective channel length L_{EFF} and the drain to source voltage V_{ds}, and thus limits the scaling of MOSFETs. These effects are referred to as the short channel effects of MOSFET transistors. At very high V_{ds}, the depletion regions of the source and drain may overlap causing large amounts of current to flow uncontrolled by the gate. This phenomenon is known as punch through.

Therefore, the suppression of the short channel effects has become one of the main priorities in the design of CMOS technologies. In the ON state, reduction of the gate length decreases the channel resistance of MOSFETs. But, in the short channel MOSFET design, the source and drain resistances are increased to suppress the short channel effects. Thus, it is important to consider ways of reducing the overall resistance of MOSFETs while suppressing the short channel effects.

To suppress the short channel effects and, thus, secure good switching off char-

acteristics of MOSFETs, the parameters of MOSFETs are scaled by the same factor S . This results in the reduction of the space charge region by the same factor S and suppression of the short channel effects. Even with the drain current reduced to $1/S$, the propagation delay time of the circuit is also reduced to $1/S$ because the gate charge (CV) scales down by a factor of $1/S^2$. This is the result of scaling down the oxide thickness by S. Thus, the short channel effects are reduced by decreasing the gate oxide thickness to allow the gate more control over the channel region.

Designers have to be also aware that in any CMOS technology, process tolerances cause the channel length to vary statistically, from chip to chip, and from wafer to wafer. The designer, therefore, has to ensure that the threshold voltage remains high enough for the device with the shortest channel length on the chip.

3.2.2 Subthreshold Leakage Currents

As explained in the previous section, short channel effects reduce the threshold voltage of MOSFET transistors, which in turn increases the leakage current. Therefore, subthreshold leakage currents influence the scaling down limits of MOSFET devices.

Simple MOSFET models assume that the drain to source current is zero, when the gate to source voltage is less than the device threshold voltage ($V_{gs}<V_{th}$). In reality the current does not drop immediately to zero. Rather, it decreases exponentially as the gate voltage drops below the threshold voltage [58]. The reason is that some of the thermally distributed electrons at the transistor source have enough energy to overcome the potential barrier controlled by the gate voltage,

and flow to the drain. Thus, leakage current follows the formula:

$$I_{leakage} = \beta(n-1)V_{th}^2 e^{(V_{gs}-V_{th})/nV_T} \tag{3.1}$$

where β is a constant function of the technology, n is the subthreshold swing coefficient, and V_T is the thermal voltage taken as $\approx$ 26mV at T=300K. This equation neglects the effect of V_{ds} on $I_{leakage}$. The simple model for the saturation current when $V_{gs}>V_{th}$ is:

$$I_{ds} = \frac{1}{2}\beta(V_{gs} - V_{th})^2 \tag{3.2}$$

Therefore the ratio between the ON and the OFF currents is:

$$\frac{(V_{dd} - V_{th})^2}{2(n-1)V_T^2} exp(\frac{V_{th}}{nV_T}) \tag{3.3}$$

It is clear that this ratio is dominated by the threshold voltage. This ratio may vary from 10^2 to 10^5, depending, on the application [59]. It is assumed that the minimum value for the ratio is 2x10^4 [51], which gives the threshold voltage a lower limit in order to achieve such a ratio . To determine what minimum value of V_{th} satisfies the requirement, a minimum ratio of V_{dd}/V_{th} must be chosen. As the ratio V_{dd}/V_{th} decreases, circuits performance deteriorates as shown in Figure 3.4 which shows the delay as a function of V_{dd}/V_{th} using Newton Model [60].

Therefore, there is a trade off between low leakage power and performance. The power supply voltage in high performance microprocessors must be significantly greater than V_{th} so that the devices operate in the active regime away from the

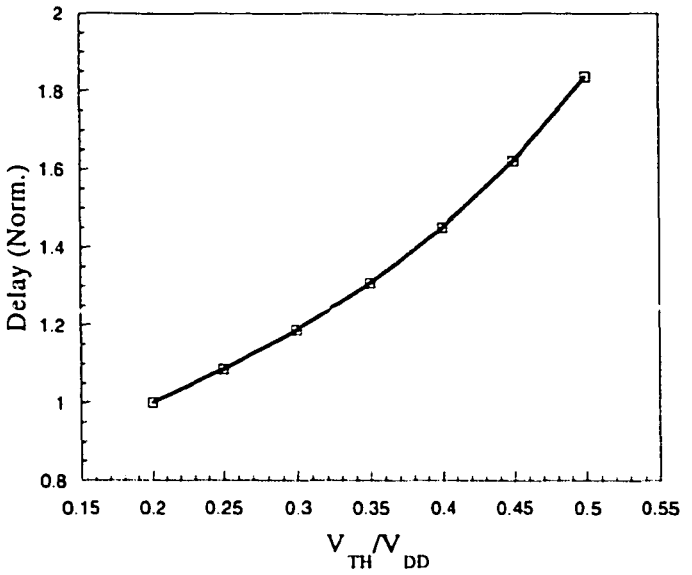


Figure 3.4: Effect of V_{dd}/V_{th} on the delay

subthreshold currents. A good requirement for high performance application is that [61] $V_{dd}>3V_{th}$. This gives the threshold voltage an upper limit. For a chip with an integration level of 100 million transistors, the leakage current of a transistor in the OFF state should not exceed 1 nA/μm [62]. This constraint restricts the threshold voltage to a minimum of 0.2V at the room temperature. Therefore, taking the leakage current as a lower bound, V_{th} ranges as follows: $V_{dd}>3V_{th}>0.6$V [51].

Thus, subthreshold currents and the need for high performance limit the extent to which the supply and threshold voltages can be scaled.

Gate Induced Drain Leakage (GIDL)

In addition to subthreshold leakage from the drain to the source, MOSFETs with high oxide fields may also show significant leakage from the drain to the bulk.

This phenomenon is known as gate induced drain leakage [63]. High fields in the oxide produce high fields in the surface of the silicon. GIDL limits to the gate oxide thickness. In the heavily doped gate to drain overlap region, this produces band bending greater than the silicon band gap over a very short vertical distance. Within this depleted region at the surface of the drain, electrons may tunnel from the valence band into the conduction band producing a drain to bulk current.

Gate Tunneling Current (GTC)

One of the basic principles of quantum mechanics is that particles do not have well defined locations, but exist only with a certain probability within a given region. Consequently, for particles with very little mass and isolated by very thin energy barriers, there is a finite probability that the particle appears on the far side of the barrier even though it does not have sufficient energy to surmount the barrier. This phenomenon is known as tunneling. As gate oxides are scaled down, significant tunneling current may flow from the drain to the gate in an OFF device or from the gate to the source in an ON device. Gate tunneling is considered the fundamental limit on the gate oxide thickness.

3.2.3 Dielectric Breakdown (DB)

Another phenomenon that sets limits to the gate oxide thickness is the dielectric breakdown. Vertical fields within the gate oxide increases steadily as power supply voltage is scaled more gradually than oxide thickness. Eventually, very high fields damage the oxide layer and cause breakdown [64]. The time of breakdown

is dependent on the breakdown acceleration factor, and the field intensity in the oxide.

3.2.4 Hot Carrier Effects (HCE)

In addition to increasing fields in the gate oxide, lateral fields within the channel also increases steadily. At sufficiently high fields, electrons may gain enough energy to cause impact ionization in the channel. The energetic carriers produced lead to gate and substrate current. Interface traps may also be formed in the gate oxide, and hot electrons may become trapped within the oxide. This build up of traps and negative charge in the oxide causes the threshold voltage to increase and the transconductance to decrease. Eventually, reduced current drive causes the circuit containing the degraded device to fail. Both NMOS and PMOS devices experience hot carrier effects, but the lower mobility of holes and their reduced ability to cause impact ionization makes hot carrier effects much less significant in PMOS devices [65].

3.2.5 Soft errors

Soft errors phenomena occurs when the data stored in the memory are changed because of radiation. They are caused by alpha particles [66] in the chip material and by cosmic rays from the space. Since capacitance and supply voltage will decrease in future technologies, a smaller charge (Q=CV) will be needed to flip a memory bit. Therefore, the soft error rate will increase. Attempting to reduce the soft error rate by increasing capacitance on the node results in reduced performance.

3.2.6 Scaling the Interconnects

Computer performance has improved dramatically because of the increased levels of integration. This benefit has been gained not only by scaling devices alone, but also by scaling the interconnects which provide communication between the devices. Interconnect scaling improves interconnect density, but generally at the expense of a degraded interconnect delay. In older technologies, interconnect RC delay represented a very small fraction of the microprocessor clock cycle time and has been only a small factor in the overall chip performance. Now, interconnect RC delay is a significant fraction of clock cycle time, which has a significant impact on the overall chip performance [67]. The interconnect RC delay increases due to the scaling of the interconnect pitch to keep pace with the density requirements. This increases the line resistance. As a result, more interconnect metal layers are needed to be able to meet both density and performance requirements. Figures 3.5 and 3.6 illustrate the number of interconnect metal layers and the trend of microprocessor clock cycle time and RC delays of the interconnects over the past generations, respectively [68] .

The total number of metal layers has increased from 2 to 5 over the past 6 generations, while the ratio of the microprocessor clock cycle added by RC delays has increased from 1% to 32%.

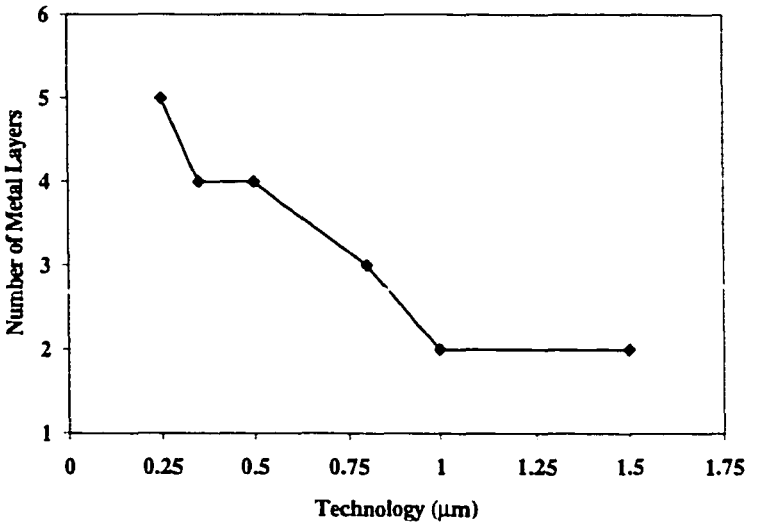


Figure 3.5: Number of metal layers over the generations

3.3 CMOS Logic Styles

The first part of this chapter outlined the different technology scaling trends and the various factors that control the scaling of different technology parameters. However, the impact of technology scaling is not the same for different logic styles. therefore, choosing the appropriate logic style for a certain application is a complex task. It is a function of the system architecture, technology and design objectives. Usually, logic families are characterized by their speed, power dissipation, area, robustness and ease of use.

In this section, five of the most famous CMOS logic styles are presented, namely, conventional CMOS, Complementary Pass Logic (CPL), Domino, Differential Cascode Voltage Switch (DCVS), and MOS Current Mode Logic (MCML). For each logic style, the advantages and disadvantages are discussed with emphasis on the

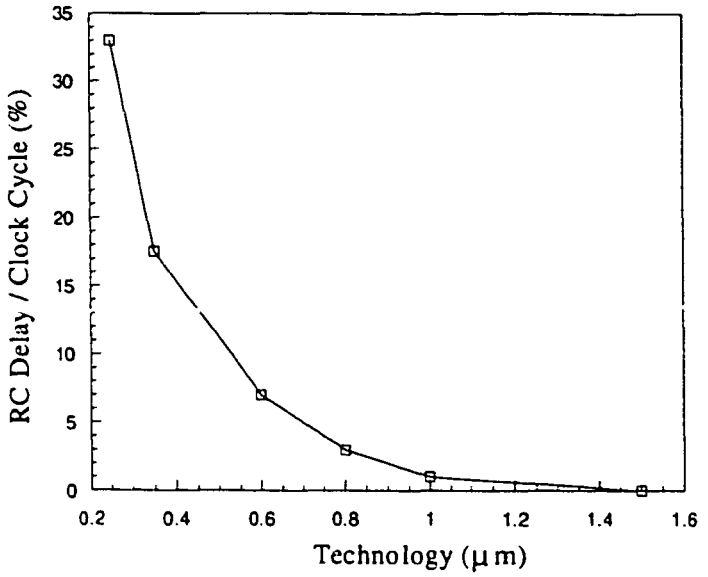


Figure 3.6: Trend of the ratio of the interconnect RC delay and the clock cycle

previously mentioned characteristics.

3.3.1 Conventional CMOS

Conventional CMOS have been known as the logic style of choice over the last fifteen years. It is widely used in industry because of its simplicity and stability.

Logic gates in conventional CMOS are constructed from an N and a P block. An AND-OR-Invert (AOI) CMOS gate is shown in Figure 3.7 .

The N block implements a sum of products function to evaluate the "0" state. The P block evaluates the "1" state of the output by implementing a product of sums function to create a path from V_{dd} to the output node. This is equivalent to stating that the output node is always a low impedance node in steady state. Usually, each CMOS gate has 2N transistors where N is the number of gate inputs.

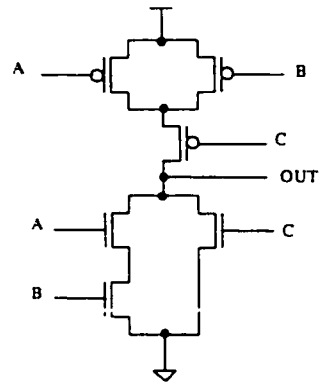


Figure 3.7: AOI implemented in Conventional CMOS

The N and P networks are designed so that whatever the value of the inputs, one and only one of the networks is conducting at steady state.

The main drawback of CMOS circuits is the existence of the P block. Because low mobility PMOS (μ_p) devices are slower than the equivalent NMOS devices (μ_n), the PMOS devices are sized up. Furthermore, the input capacitance of a CMOS gate is large, because each input is connected to the gates of at least one PMOS transistor and one NMOS transistor. This also degrades the gate's speed. Usually, the best gate performance is achieved with a PMOS/NMOS width ratio of $\sqrt{\mu_n/\mu_p}$ [7]. This ratio eventually approaches one in deep submicron technologies, where the carrier drift velocities in NMOS and PMOS transistors become almost equal due to velocity saturation [7].

Another drawback of CMOS has been its relatively weak output driving capability due to series transistors in the output stage. This problem is solved by using output buffers/inverters. Fan out has a larger impact on the gate delay in static CMOS than any other logic style, because it is connected to both an NMOS and a PMOS device and presents a load to the driving gate equal to the sum of their gate

capacitances. This is not the case for other logic styles where an input is connected only to the gate (or drain) of a single NMOS or PMOS transistor. Another impact of the large input capacitance is high power dissipation.

However, static CMOS circuits have a smaller switching activity compared to other logic styles. The fact that input signals are connected to transistor gates only in CMOS circuits, facilitates the usage and characterization of logic cells. Another important advantage of CMOS is its robustness against voltage and transistor scaling, and its reliable operation at low voltages and minimal transistor sizes. The percentage increase in power due to noise is low compared to other logic styles.

CMOS circuits are also known to have the best noise margin among all logic styles. This is attributed to the presence of a static path that restores the correct logic state in the case of noise. It has been shown that the highest noise margin for static CMOS is achieved with a PMOS/NMOS width ratio of μ_n/μ_p [6]. This provides identical current driving capabilities for the NMOS and PMOS network, which limits the short circuit current [7]. Finally, the layout of CMOS gates is straightforward and efficient, due to the complementary transistor pairs.

3.3.2 Complementary Pass Logic (CPL)

A CPL gate [69] consists of two NMOS logic networks (one for each signal rail), two small pull up PMOS transistors for swing restoration, and two output inverters for the complementary output signals. Figure 3.8 shows an AOI circuit implemented using CPL.

Unlike CMOS logic, the CPL gate creates a path from the output node to

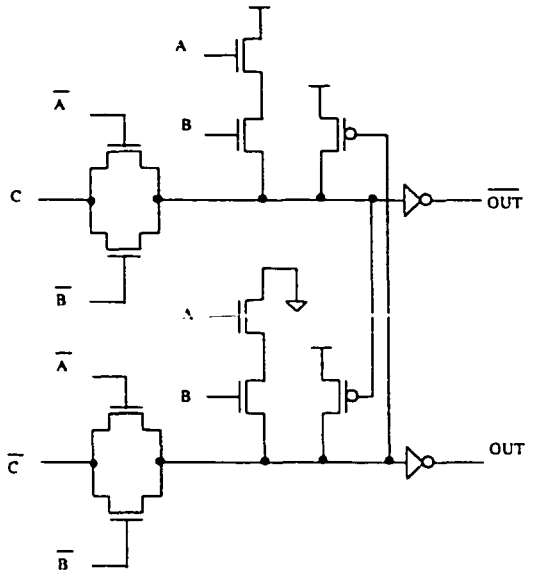


Figure 3.8: AOI implemented in CPL

one of the input nodes of the gate instead, of the power grids. Because the MOS networks are connected to variable gate inputs rather than constant power lines, only one signal path through each network must be active at a time, in order to avoid shorting different inputs together. Therefore, each pass transistor network realizes a multiplexer (MUX) structure. Figure 3.9 presents a MUX implemented using CPL.

All two input functions AND, OR and XOR can therefore, be implemented by this basic gate structure. The overhead of this structure is relatively high for simple monotonic gates such as AND and OR. Therefore, CPL is not recommended for simple monotonic gates. However, CPL uses smaller and fewer transistors to implement XOR and MUX based functions. Usually, CPL gates have small input load and good output driving capability due to the output inverters, and the fast differential stage due to the cross coupled PMOS pull up transistors.

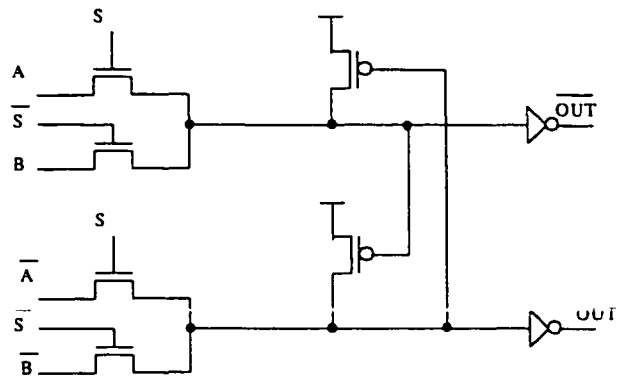


Figure 3.9: MUX implemented in CPL

Most CPL gates require all the inputs and their complements, which increases the routing complexity and overhead. Furthermore, the wiring capacitance (interconnects) increases, which causes the power and delay to increase. The output voltage swing of CPL gate is lower than the input swing by the NMOS threshold voltage V_{th_N}, because CPL gate is constructed from NMOS transistors only. The voltage swing of CPL deteriorates with additional cascaded logic levels. If CPL output is used to drive an inverter, it may cause a DC current, because the PMOS transistor of the inverter is not completely OFF. Therefore, a swing restoring circuit should therefore, be added after each two or three cascaded gates to restore the full output swing. However, the restoring circuit increases the power consumption of the gate.

The layout of pass transistor cells is not as straightforward and efficient as CMOS, due to the irregular transistor arrangements and high density wiring.

Another problem associated with CPL, has been its sensitivity to voltage scaling [70]. This implies that the gate's efficiency is degraded in terms of performance and power consumption, as the supply voltage scales down. Furthermore, CPL

is sensitive to transistor sizing which leads to lower circuit robustness and noise margin. Consequently, CPL circuits are more susceptible to noise.

3.3.3 Domino Logic

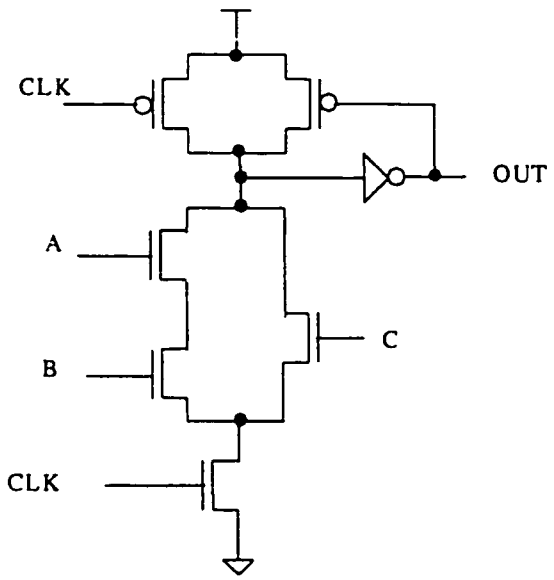


Figure 3.10: AO implemented in Domino

Figure 3.10 illustrates an AND-OR (AO) gate implemented using Domino logic. A Domino gate consists of a dynamic gate stage, a static CMOS inverter to drive the circuit's output, and a PMOS keeper transistor to restore the logic at the Domino output node. The dynamic gate stage consists of an NMOS transistor network to implement the required logic function, and two transistors (NMOS and PMOS) connected to the clock signal to synchronize the operation of the circuit. The CMOS inverter is included for the proper operation of a chain of Domino gates, and to increase the driving capability of the gate. The keeper transistor enhances

the Domino gate immunity against charge sharing and charge loss [66].

The conventional Domino operates as follows: during the precharge phase (when the clock is low), the Domino output node is precharged to V_{dd} and the keeper transistor is turned ON. When the clock goes high (evaluation phase), depending on the inputs; the Domino output either discharges to GND or remains high at V_{dd}. If all inputs are low, the keeper maintains the domino node high and the gate output low. In a cascaded set of logic gates, each stage evaluates, causing the following stage(s) to evaluate, in the same way a line of Dominos fall. Any number of logic stages may be cascaded, provided that the sequence can evaluate within the evaluate clock phase. The Domino input signal to a Domino gate must, therefore, satisfy some setup and hold timing constraints for correct operation of the gate [71].

Domino logic has only one PMOS transistor instead of a complete block like CMOS gates. This substantially lowers the transistor count compared to CMOS. The input capacitance is also smaller than that of CMOS due to the absence of the PMOS transistors. This enhances the speed of the Domino Logic. Furthermore, the evaluation is fast, because the logic block is constructed from only high mobility NMOS transistors. However, Domino logic has higher switching activity than the equivalent CMOS gate because all the output nodes are precharged to V_{dd} during each clock cycle. Consequently, power is consumed during the precharge operation every time that the output capacitor is discharged in the preceding cycle. Also, a large portion of power for Domino circuits is dissipated in driving the capacitance of the clock lines, which are being switched at full rate.

All these factors contribute to the Domino's high power consumption. As previously mentioned, usually there is no short circuit current in dynamic circuits, as the precharge and evaluate transistors are never ON simultaneously, unless the rising or falling edge of the clock is long. Beside its high power consumption, Domino logic is vulnerable to noise. A voltage at the input as low as V_{th} could turn the pull down NMOS transistor ON, and the output eventually reaches GND. This is translated to a noise margin of V_{th}, which is quite low compared to static logic styles. Due to subthreshold leakage currents, the pull down transistor starts to conduct at voltages just below V_{th}. Some subthreshold leakage current can flow through the NMOS even when the input is "0". This effect becomes more pronounced when the input is not completely "0", but approaches V_{th} in the presence of noise. To compensate for the low noise margin, the size of the PMOS keeper must increase, which in turn increases the contention current during evaluation and consequently degrades the gate's performance.

Another problem of Domino circuits is that only noninverting logic functions may be implemented. This is a problem in the implementation of XOR gates, multiplexers, and full adders. A Domino style which overcomes this problem is the NP-Domino [72]. NP-Domino is a further refinement to the traditional Domino, where the Domino buffer is removed, and the cascaded logic blocks are alternately composed of P and N transistors. NP-Domino circuits generate inverting logic, but have the disadvantage of low mobility PMOS transistors and the overhead of producing the inverse of the clock. Figures 3.11 and 3.12 show an NP-Domino implementation of the XOR and Full Adder gates respectively.

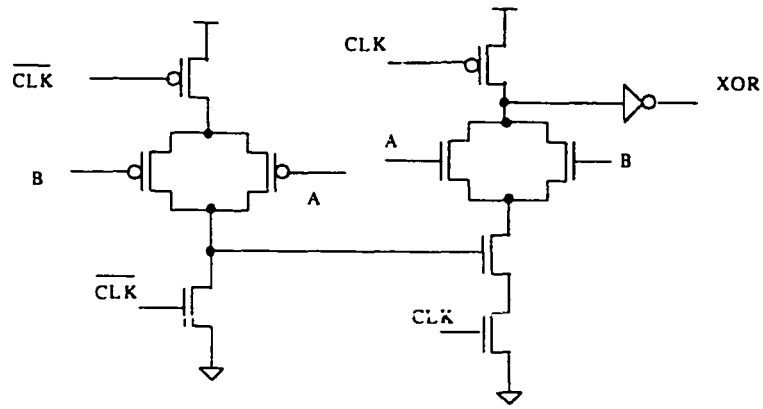


Figure 3.11: XOR implemented in NP-Domino

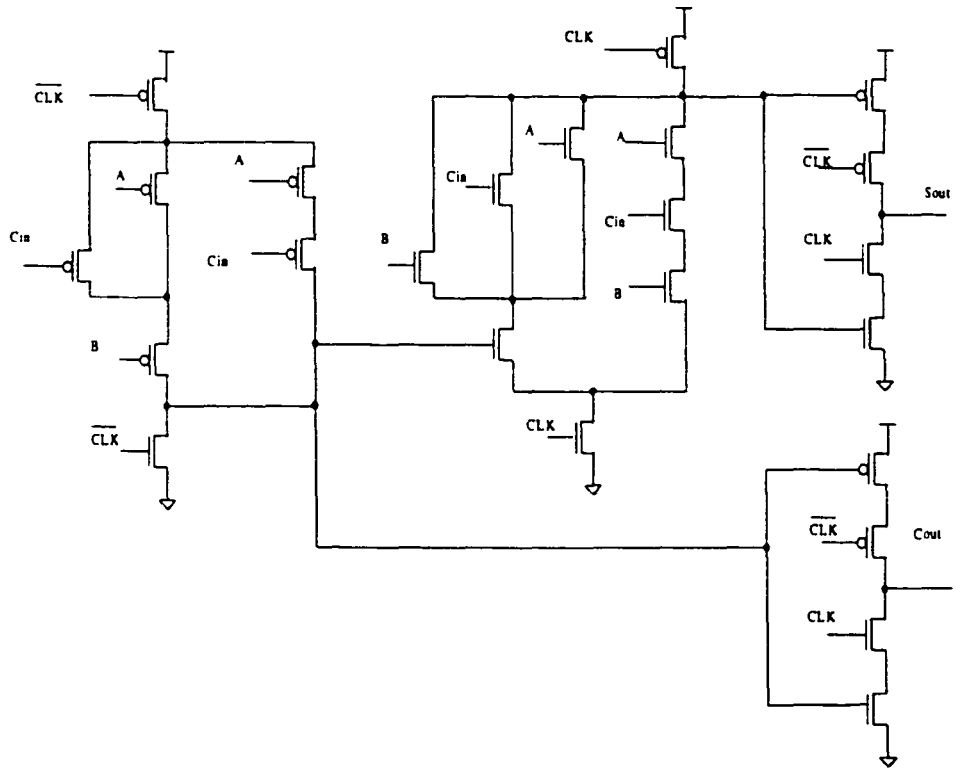


Figure 3.12: Full adder implemented in NP-Domino

3.3.4 Differential Cascode Voltage Switch (DCVS)

In early 1980's Heler et al. proposed DCVS as a new high performance logic family [73] for complex logic architectures. The original static DCVS logic gate is shown

in Figure 3.13.

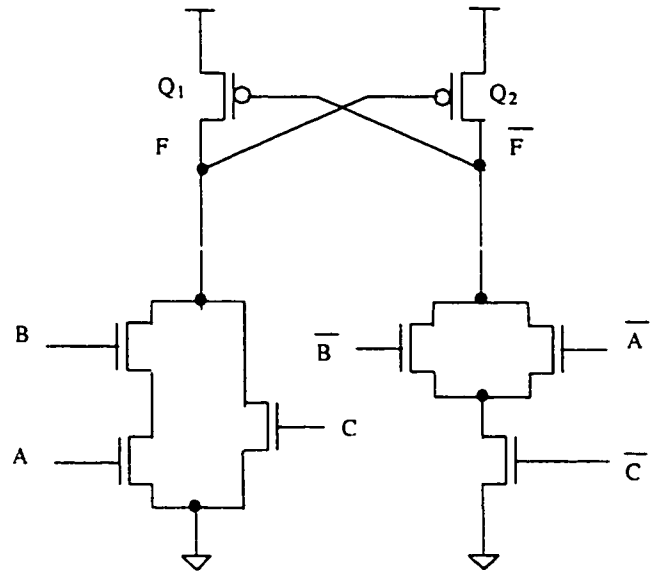


Figure 3.13: AOI implemented in original DCVS

A DCVS gate consists of two logic branches connected to a cross coupled PMOS transistor pair, which act as a static latch. One of the branches implements the logic function F, while the other branch implements the complementary function $\overline{F}$. Each branch may be optimized separately, by using Karnaugh map or any alternative technique. To illustrate the functionality of the DCVS circuit, assume an initial output value of "1" ($F = 1$, $\overline{F} = 0$) and a new set of inputs that changes the output value F to "0". Initially, the logic block connected to F is OFF while the block connected to $\overline{F}$ is ON, discharging the $\overline{F}$ node to "0". When the inputs change value, the branch connected to F turns ON trying to discharge the F node while the $\overline{F}$ branch turns OFF, causing the $\overline{F}$ to float. However, since $\overline{F}$ node is floating, the PMOS transistor Q_1 is still ON trying to charge the output node F to V_{dd}. So, there is a contention between the transistor Q_1 and the logic block F, and

a DC current path exists from V_{dd} to GND. This DC current exists until the F node voltage drops lower than $V_{dd} - V_{TH_P}$, when transistor Q_2 turns ON charging $\overline{F}$ node to V_{dd}. To reduce the contention period, Q_1 and Q_2 should be small enough to reduce the DC path from V_{dd} to GND, and to speed up the transition from "1" to "0". Unfortunately, this increases the time for the output transition from "0" to "1" at node $\overline{F}$, because Q_2 has limited driving capability. Therefore, the original DCVS is considered a ratioed logic style which leads to a high power dissipation and complex design process. Moreover, DCVS has large current spikes during logic transitions which, causes unstability in the supply rails.

Domino DCVS (DDCVS) is a dynamic version of DCVS. DDCVS reduces the contention by splitting the time of the precharge ($0 \rightarrow 1$ transition) from the evaluation phase. Figure 3.14 presents the architecture of an AOI gate implemented in DDCVS logic.

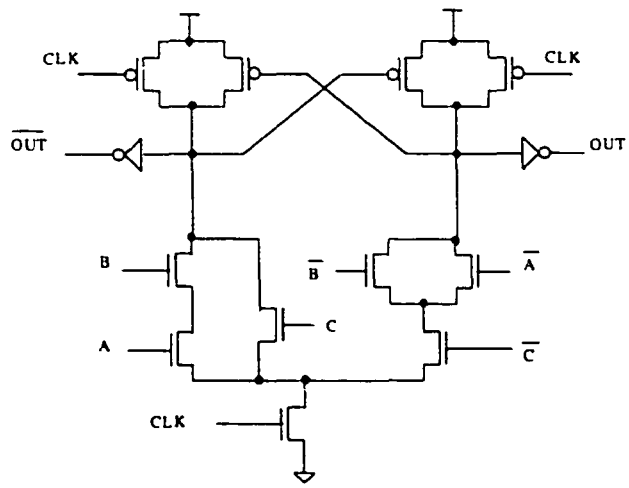


Figure 3.14: AOI implemented in DDCVS

Unlike Domino logic, where the keeper transistor is ON at the start of the eval-

uation phase, DCVS keeper keeper transistors Q_1, Q_2 are OFF at the beginning of evaluation. This reduces the power and delay caused by the contention. DDCVS is considered a general purpose logic style, because it may be used to implement inverting and non inverting logic circuits. DCVS is more area efficient in implementing complex logic gates. Most of the complex logic functions may be implemented using one gate only, which makes DDCVS logic much faster than CMOS circuits. It is also suitable for implementing gates with XOR functionality, such as arithmetic circuits, and MUX style logic gates. Because of its differential nature, DCVS is suitable for self testing techniques which covers stuck-at-faults [74].

Over the past fifteen years, many flavors of Cascode Voltage Switch Logic (CVSL) has been introduced to avoid the problems associated with the DCVS ratioed logic operation and the slow PMOS loads. Differential Cascode Voltage Switch with Pass Logic family (DCVSPG) uses pass logic to implement the logic function of each branch[75]. It avoids the problem of the floating output node that exists in DCVS logic. Switched Output Differential Structure (SODS) replaces the PMOS latch with a clocked latch to avoid the contention [76]. Charge Recycling Differential Logic (CRDL) reduces power dissipation by shorting the output nodes before each evaluation phase [77].

3.3.5 MOS Current Mode Logic (MCML)

MCML is well known as a high performance logic family for mixed signal systems [78], [79]. Figure 3.15 shows the architecture of an MCML inverter/buffer.

Transistor Q_1 acts as a DC current source controlled by V_{ref}. Resistors R_1

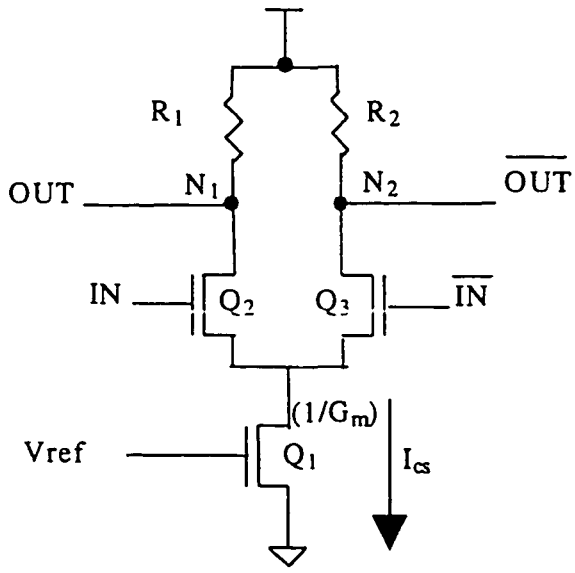


Figure 3.15: Inverter implemented in MCML

and R_2 are pull up resistors. The logic function is implemented by the logic block connected between the resistors and the current source. For an inverter/buffer, the logic block is the differential pair constructed by transistors Q_2 and Q_3.

The operation of the MCML logic is based on the differential pair circuit. Each input variable is connected to a differential pair circuit. The value of the input variable controls the flow of current through the two branches of the differential pair. For example , if $V_{gs}(Q_2)$ is higher than $V_{gs}(Q_3)$, the current passing through Q_2 exceeds the current passing through Q_3. Therefore, the voltage of node N_1 begins to drop, until it reaches a steady state, where the current going through the resistor R_1 matches the current going through transistor Q_2. In the mean time, node N_2 is charged to V_{dd} through resistor R_2.

The output voltage swing V_S is defined as the voltage difference between N_1 and N_2 at steady state. The amount of current passing through the ON branch (Q_2 in

the previous case) controls the discharge delay of the logic gate ($1 \rightarrow 0$ transition), while the load resistor controls the charging of the output nodes($0 \rightarrow 1$ transition).

To achieve the best performance, all of the current needs to pass through the ON branch only, and the load resistors should be small in order to reduce the RC delay. This guarantees that the voltage at one of the output nodes is V_{dd}, while the other nodal voltage is $V_{dd} - I_{cs}.R_L$, where I_{cs} is the value of the current flowing through the current source, and R_L is the load resistance (R_1, R_2). MCML circuits are faster than other logic families, because it uses NMOS transistors only and these transistors operate only in the saturation or linear regions.

If properly designed, MCML transistors must not operate in the cut off mode independent of the input combination. However, a large voltage swing V_S may push one of the input transistors into cut off region, which increases the delay to pull it back to ON mode. Also, a small V_S is not recommended, because it makes the gate more susceptible to noise and it reduces the current difference between the differential branches, which means smaller discharge current. The swing is controlled by the product of R_L and I_{cs}. A larger R_L leads to a larger voltage swing and smaller charge current. Increasing I_{cs}, though, speeds up load discharging, and increases power dissipation. Thus, choosing the appropriate V_S is a compromise between power dissipation and delay. Normally a swing 20% of V_{dd} is used [79].

The small output swing of MCML circuits reduces the cross talk between adjacent signals. The constant current source reduces the switching noise and supply fluctuations. For these reasons, MCML is recommended for mixed signal design to reduce the interference between the digital and analog blocks. The reduced out-

put swing also reduces the dynamic power dissipation in the case of long busses. Therefore, MCML may be used in the implementation of bus transceivers to reduce power and noise. Another important feature of MCML circuits is its noise immunity, due to the differential nature which is required at high frequencies and low supply voltage [80].

However, MCML has some major drawbacks which limit its use in digital systems. First is the static power dissipation due to the constant current source. By using Power/MHz as a measure for power dissipation, MCML power dissipation is reasonable at high operating frequency, but the Power/MHz becomes much higher at lower operating frequencies because the current source is fixed. Compared with CMOS circuits, MCML consumes more power at low frequencies. Therefore, MCML is preferred in high frequency applications only, in order to reduce the overhead of its static biasing power. Secondly, MCML is not suitable for power-down modes because of the DC current source. Hence, it is inappropriate for large systems, where power down techniques are used to reduce the system power. MCML circuits also require special fabrication technologies to implement the large load resistors in a reasonable area. This increases the cost and area of the chip. MCML designs need to include a reference voltage distribution tree to control the current source of each gate, leading to larger chip area and more complex routing. Finally, the matching of the rise and fall delays is not an easy task, because it is a function of the load of each gate.

As a solution for some of MCML problems in digital design, Mizuno *et.al.* [78] have suggested an adaptive pipelined technique for MCML circuits. The new

implementation changes the current source value of the gates in the critical path. Gates that are not in the critical path may use smaller current source. This scheme reduces the overall power dissipation.

3.4 Impact of Technology Scaling on Logic Styles

3.4.1 Velocity Saturation and Mobility degradation

In order to evaluate the output logic of a certain gate implemented by some logic style, a series of charging and discharging processes occur to the output node (at which the logic is determined). As the input of a logic gate changes, it causes the output node(s) to either charge or discharge. This is true for logic styles consisting of an N logic block. A static CMOS inverter is a simple example. The delay of which is the time taken for the output node to fully charge or discharge.

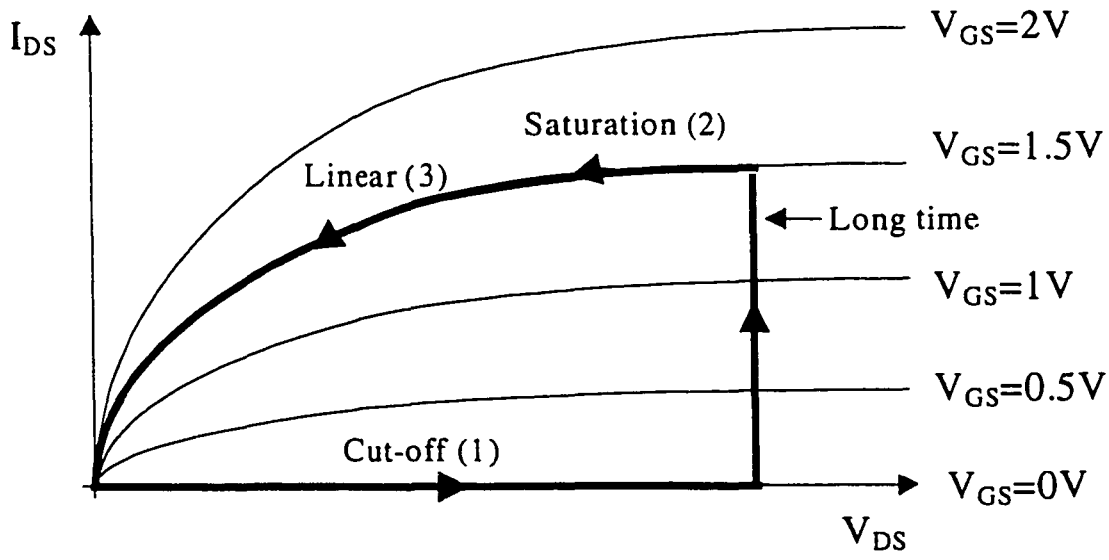


Figure 3.16: Modes of operation on I-V characteristics during discharging

For full swing logic styles, this NMOS will go through all the operating phases (cut-off, saturation and linear modes) while discharging the output node. Figure 3.16 shows the different modes a discharging NMOS device would undergo. The transistor is initially in the cut-off mode (1), when the input is "0". As the input increases, the NMOS operates in 2 regions; Saturation (2) and Linear (3). The NMOS will first operate in saturation where the drain current I_{DS} is large ($I_{DS} \propto (V_{GS} - V_{TH})^{\alpha}$, which discharges the O/P node quickly. α is the velocity saturation index [60], which takes a value of 2 for long-channel devices, and around 1.3 for short-channel devices. The NMOS will operate along a constant V_{GS} curve in the saturation region in the typical I_{DS}/V_{DS} characteristics plot. When the output node reaches $V_{DD} - V_{TH_N}$,the NMOS moves from the saturation to the linear region. I_{DS} in the linear region is less than in the saturation region for the same V_{GS} [60], which causes the discharge to slow down.

The slowest transition however is from cut-off $\rightarrow$ saturation because all the charge stored in the depletion region of the NMOS device has to sink before the channel is constructed between the drain and the source. MCML is therefore, faster than other logic styles (refer to Figure 3.15) This is because Q_2 and Q_3 are never totally OFF, and experience a transition from the saturation $\rightarrow$ linear region and vice versa which take a short time.

Figure 3.17 shows the I-V characteristics of a MOSFET transistor for a long and short channel case. Saturation currents are reduced in magnitude compared to linear currents and no longer follow the long channel behavior (because α approaches 1). This will also cause a decrease in V_{DS-SAT} as technology scales down. In CML

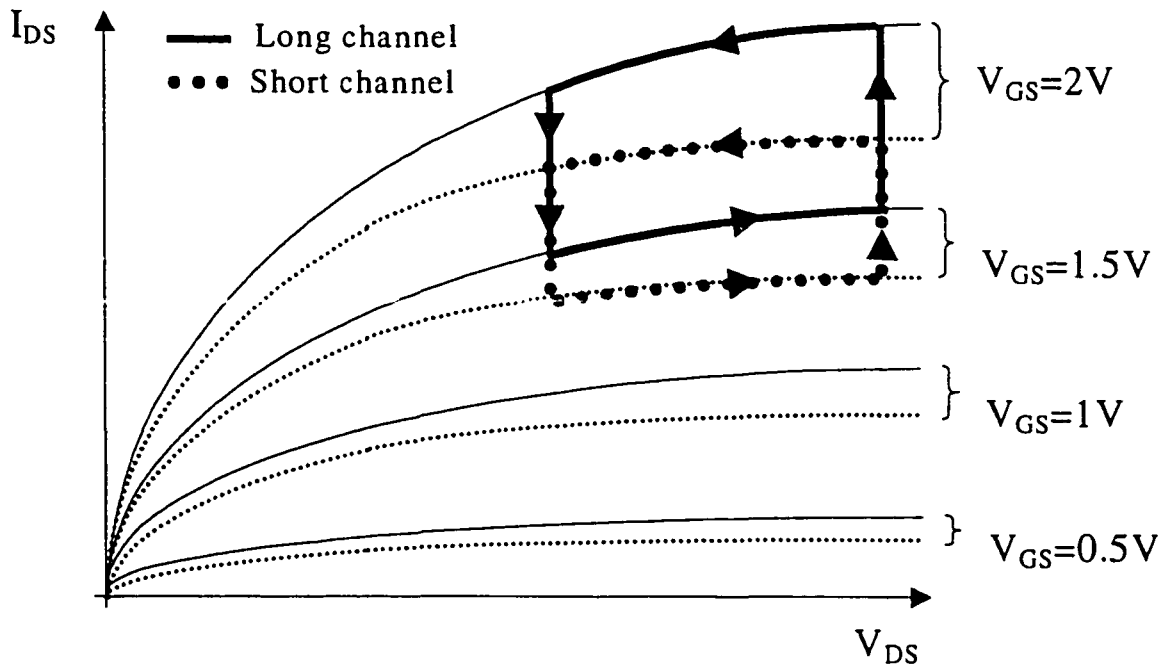


Figure 3.17: Effect of velocity saturation on MOS I-V characteristics

logic, the difference in current in the two branches indicates how fast the gate will evaluate. Due to short channel effects, the saturation velocity will decrease. The CML will operate along the dotted-line loop (due to short channel effects in the DSM regime) instead of the solid-lined loop (due to the long channel effect). The operating current is thus reduced, and consequently the evaluation time drops. The speed advantage of CML over other logic styles will thus start to fade as we move deeper in the DSM regime.

Not only will the carrier velocity tend to saturate as the channel length is scaled down, but the device's mobility will start to degrade as well. Figures 3.18 and 3.19 show the saturation velocity and mobility degradation of the electron respectively [7].

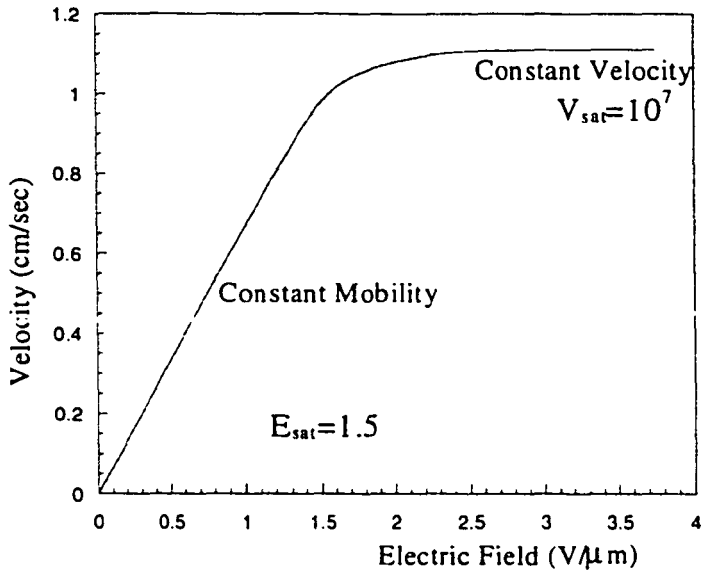


Figure 3.18: Velocity saturation

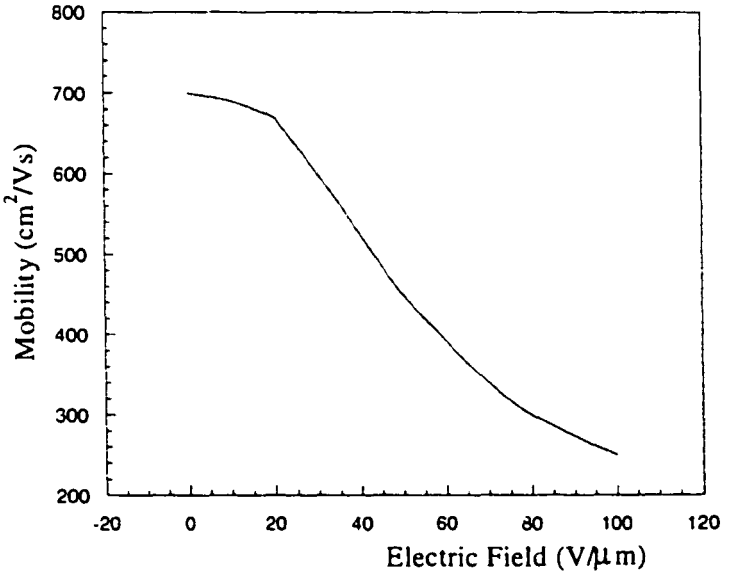


Figure 3.19: Mobility degradation

In NMOS devices, the saturation velocity is reached at a lower critical electric field compared to PMOS. This indicates that μ_n is degraded at a much faster rate

than μ_p [65]. Eventually, a point is reached where both NMOS and PMOS have comparable mobilities and switching speeds. This is particularly important for the implementation of CMOS structures, for two reasons. Firstly, CMOS suffers from degraded performance because of the low mobility PMOS transistors. This speed disadvantage will gradually decrease as the technology scales down when μ_p approaches μ_n. Therefore, the sizing up of the PMOS device is no longer required to attain speed. This will reduce the input capacitance of the gate. Therefore, not sizing the PMOS device would enhance the performance of CMOS in terms of delay, power and area. Secondly, the optimum noise margin in CMOS is achieved when μ_p equals μ_n [6]. With $\mu_p \approx \mu_n$, the CMOS noise margin is enhanced, and equal driving capability is also achieved, which keeps the short-circuit current within bounds [7]. Thus CMOS performance and robustness are both enhanced relative to other styles as technology scales down. On the other hand, as μ_p approaches μ_n, the speed advantage that dynamic circuits (Domino and DCVS) originally had over static logic will start to fade. The dynamic and static circuits will have similar operation, and will evaluate the logic at comparable speeds. However, implementing wide fan-in gates like NOR gates using dynamic logic is still advisable in order to avoid stacking of PMOS devices, which will ultimately degrade the gate's speed when implemented using CMOS logic. Furthermore, as $\mu_n \approx \mu_p$, the high-speed logic evaluation through the high-mobility NMOS (μ_n) pass devices in a circuit implemented in CPL will also decrease compared to the other logic styles.

3.4.2 Leakage currents

The effects of leakage current is more pronounced as CMOS technology moves down in the DSM regime, because leakage current increases exponentially as V_{TH} decreases. As explained in Section 3.3.3, Domino logic is particularly susceptible to noise, due to the effect of leakage currents which could cause the pull-down devices to falsely switch easier. This deteriorates the gate's noise margin. To compensate for the low noise margins, the size of the PMOS keeper must increase, in turn increasing the contention current during evaluation, as well as the loading of the gate's O/P node. This ultimately reduces the gate's performance. A trade-off therefore exists between speed and robustness of Domino circuits in the DSM regime. The rate of improvement in the Domino's performance will therefore gradually decrease as we go deeper in DSM technologies. This is another reason that the performance of CMOS circuits is expected to approach the dynamic logic gates while controlling the gate's noise margin. Figure 3.20 plots the optimal V_{TH} versus process technology for the static and dynamic cases [62]. It is clear that the optimal V_{TH} used in static and dynamic circuits diverge in DSM technologies (less than 0.25μm). Static circuits need lower V_{TH} to maintain gate drive with lower V_{DD}, while in dynamic circuits it becomes difficult to scale V_{TH} due to noise limits.

Leakage currents have minimal effect on CMOS logic, due to the self-restoring logic mechanism CMOS employs.

DCVS logic does not suffer from contention currents during evaluation as domino logic. Leakage currents therefore, do not influence the circuit's evaluation time. However, during precharge, if one of the two branches is OFF, the OFF pull-down

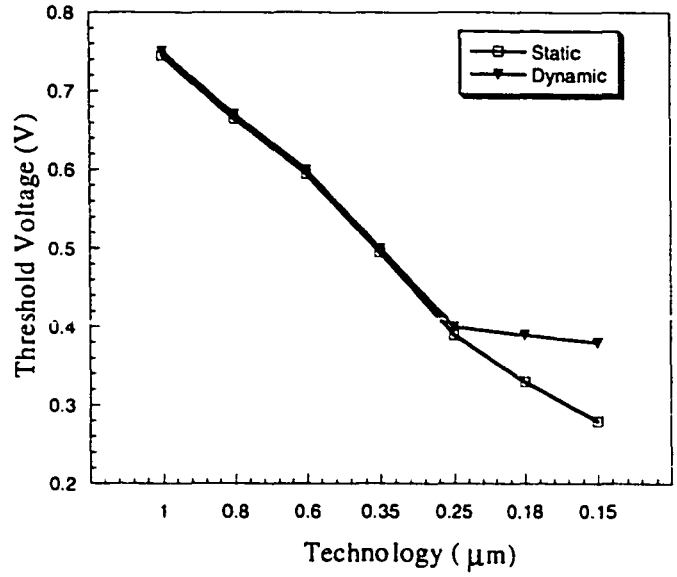


Figure 3.20: Optimal threshold voltage for static and dynamic circuits versus technology

transistors will start to leak. This might deteriorate the logic in the internal node (like domino logic), and consequently corrupt the gate's noise margin. Therefore, the cross-coupled PMOS keeper transistors shown in Figure 3.14 must be sized up enough to compensate for these leakage currents. However, sizing up the keepers too much will increase the loading at the internal nodes, and consequently degrading the circuit's performance. The keepers, therefore, should be sized properly to achieve the desired speed while attaining sufficient noise margins.

Circuits implemented in CPL also experience leakage currents. Referring to the upper branch in Figure 3.9, if A="1" while B="0" and S="0", leakage current flows through the two NMOS devices of the upper branch and is dissipated. A similar behavior takes place if A="0" while B="1" and S="1". Leakage however, has minor effect on CPL's functionality.

The concept of leakage currents does not exist in CML circuits. Referring to Figure 3.15, Q_2 and Q_3 do not operate in the cut-off region. The leakage currents are therefore negligible compared to the bias current produced by the current source Q_1.

3.4.3 Hot Carrier Effect (HCE)

HCE is the phenomenon where carriers get trapped in the oxide, increasing the threshold of NMOS devices, and decreasing the threshold of PMOS devices. MCML may be vulnerable to V_{th} variations caused by the hot carrier effects, because the devices must be matched for correct functionality. The hot carrier phenomenon is another reason to make low voltage operation more favorable. Logic families that can work at a lower supply voltage such as MCML are more preferable, because this will reduce the hot carrier effect and the punch through phenomenon, leading to better reliability and longer lifetime.

Logic styles that can tolerate minor changes in V_{th} become more reliable, because the hot carrier effects and electromigration tend to increase V_{th} over time. For CMOS, Domino, and DCVS, this is translated into a small variation in delay and a better noise margin. The performance of circuits implemented using CPL degrades also, because of the hot carrier phenomenon. This is attributed to the V_{th} drop across the pass transistor. V_{th} increases due to the HCE, producing a greater drop across the transistor. Therefore, the pass transistor switching speeds is reduced, and the transistor current is reduced. The same occurs at CPL's output inverter, which increases the gate's delay.

However, a higher V_{th} may cause MCML to cease functionality. This is attributed to the fact that increasing V_{th} decreases the discharge current, and limits the output voltage swing V_S. When V_S is reduced, the the following MCML stages may not function properly.

3.4.4 The Drain Induced Barrier Lowering (DIBL)

DIBL causes V_{th} to be a function of the operating voltage. As DIBL effect increases in smaller feature size devices, the V_{th} dependence on L_{eff} and V_{dd}, becomes a problem especially for dynamic circuits. This causes a reduction in the noise margin, which is particularly a problem in Domino logic implementations. As mentioned earlier, increasing the noise margin comes at the expense of reduced performance.

3.4.5 Scaling down V_{dd}/V_{th} ratio

V_{dd} scales down at a relatively slower rate than that of V_{th} (see Figure 3.2). This is mainly attributed to reliability restrictions that limit the value of the electric field applied to the gate and across the channel.

Hence, the ratio V_{dd}/V_{th} drops with technology scaling, until it reaches a minimum value of 3 in 0.07μm technology. This small ratio degrades the performance and power dissipation of CPL logic styles. Figure 3.21 illustrates a section of the CPL circuit.

The voltage at the output node of the driver circuit is V_{dd}, while the pass transistor is initially OFF. When transistor Q_1 turns ON, it operates in the saturation mode, where its current I_1 is proportional to $(V_{GS_1} - V_{TH_N})^{\alpha}$. α is the velocity satu-

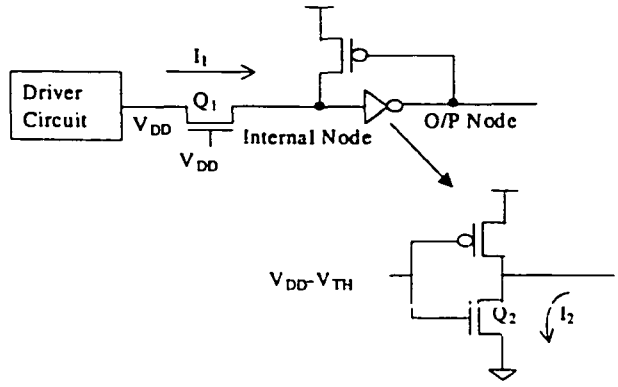


Figure 3.21: Section of a gate implemented using CPL

ration index [60] which takes a value of 1.3 for short channel devices. In the case of CPL, $V_{GS_1} = V_{dd} - V_{TH_N}$ (due to the V_{TH_N} drop), thus $I_1 \propto (V_{dd} - 2V_{TH_N})^{\alpha}$. Worst case delay occurs when $V_{dd} = 3V_{TH_N}$ [81], where the current I_1 is proportional to ${V_{TH_N}}^{\alpha}$. Thus, I_1 is significantly reduced, and the switching speed of Q_1 is reduced. Furthermore, this increases the short circuit current flowing from V_{dd} to GND in the inverter.

A further speed degradation, occurs as Q_2 passes through the saturation phase and then the linear phase, while discharging the output node. Q_2 begins discharging the output node while in the saturation mode when $V_{gs_2} = V_{th_N}$. Thus $I_2 \propto (V_{gs_2} - V_{th_N})^{\alpha}$, V_{gs} is initially at V_{th_N} to discharge the output node. This produces a very small discharging current, hence, a large delay time. This situation continues until the output node voltage drops to $V_{dd} - V_{TH_P}$. At this instant, the keeper turns ON, pulling up the internal node to V_{dd}, and accelerating the discharging process. This provides an additional delay for CPL. Another problem associated with decreasing the V_{dd}/V_{th} ratio is the reduction in gate robustness because the noise margins dwindle significantly.

3.4.6 Scaling of Interconnects

Differential logic circuits such as CPL, DCVS and MCML have complex structures, and a high wiring density due to the dual rail signals. Therefore, the wiring/interconnect capacitance increases, which increases the power and delay. This problem becomes even worse in the deep submicron regime, where the RC delay of the interconnects occupies a large ratio of the clock cycle time. Elmansy has showed that interconnects contribute about 30% to the total delay in the 0.25μm technology [68], as previously shown in Figure 3.6. This is another reason for the degradation in the differential logic styles' performances.

3.5 Area Considerations

Area of CMOS logic circuits is affected by the technology scaling and by the logic style itself. This section discusses both effects.

3.5.1 Technology Scaling and Area

Metal interconnects are needed to connect transistors, route signals and supply rails across the chip. As technology scales down, transistor feature sizes scale down linearly. However, metal wire interconnects do not scale down at the same rate due to the physical limitations of the metal deposition. Therefore, more demands are placed on the interconnects as technology moves to smaller feature sizes. The interconnect pitch (line width+space) decreases to increase integration density. However, the average interconnect length is kept constant because of the use of more

transistors per circuit which increases parasitic capacitance and line resistance. This degrades the chip's performance, and more power is dissipated per unit area, which consequently augments the chip's temperature. To avoid reliability problems and failure of electronic circuits associated with high operating temperatures, the power dissipation should be reduced.

In older technologies, poly layers have been used for routing, because the large width and low resistance of the poly in these technologies. This is not the case in deep submicron (DSM) technologies, where the impedance of the poly layer grows and is not suitable for long wires or interconnects. Such limitations lead to the use of extra vias and metal wires in routing, thus adding additional overhead.

Copper interconnects are used to reduce the interconnect area since the physical limitations on copper size are more relaxed. Copper also has a low resistivity, which allows wires to have small widths, and thus more wiring density. However, many problems are associated with the use of copper wiring such as contamination and cost, which makes it an expensive alternative [82]. Another alternative is to decrease the interconnect capacitance by using insulators with a dielectric constant lower than SiO_2. Yet, Bohr has shown that the use of low dielectric constant insulators causes many problems [83].

These two methods help reduce the RC interconnect delay, but will not suffice as feature size continues to shrink. The use of a larger number of metal layers and stacked vias is a technique for improving interconnect density without reducing pitch . For DSM devices, six levels of metal or more are used. For older technologies, two or three levels of metal were common. Figure 3.22 shows trends for number of

metal l

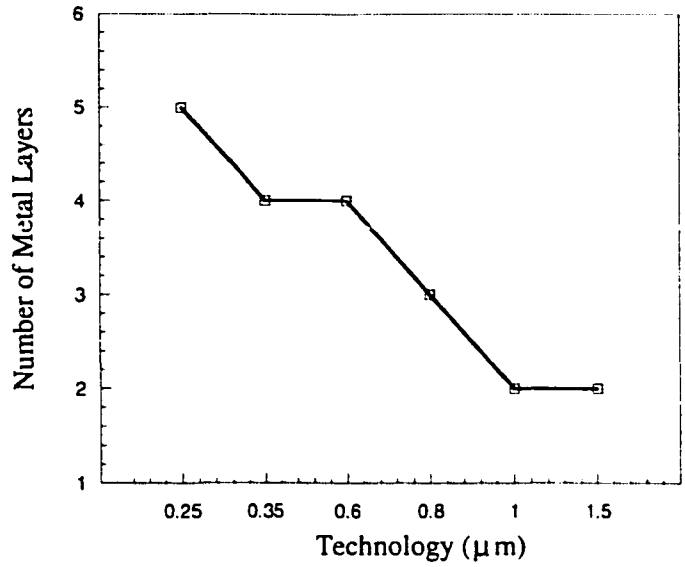


Figure 3.22: Number of metal layers and average metal pitch trend

Finally, the interconnect height is scaled at a slower rate than its width. This increases the wire's aspect ratio, and consequently reduces the wire resistance. This, however, evokes line coupling, which causes crosstalk, increased power dissipation, and degradation in performance.

3.5.2 Logic Style and Area

The choice of logic style affects the area in two ways: cell area and routing area. Cell area is a function of the number and size of the devices. It is also dependent on the complexity of the logic cell, since complex gates require more area for connecting the devices of the gate. Generally, differential logic styles such as CPL, DCVS, and MCML are area efficient in implementing arithmetic circuits and XOR-based logic

systems. For simple gates such as AND and OR, single ended logic styles such as CMOS and Domino are preferred. Input signals are connected to transistor gates only, which facilitates the usage and characterization of logic cells. The layout of CMOS gates is straight forward and efficient due to the complementary transistor pairs.

Routing area is the wire interconnect area for connecting the gates together. Differential logic styles have twice the number of inputs and outputs compared to single ended logic families. This obviously leads to larger interconnect areas. As previously explained, the routing overhead is more severe in DSM technologies because of the relatively large interconnects, which degrades the circuit's performance.

As a rule of thumb, differential logic should be used only for complex gates, especially XOR logic style, where differential logic reduces the total number of logic gates. Generally, implementations that use more devices in critical paths offer less wire congestion and shorter interconnect delays.

3.6 Simulation Setup

To verify the previous qualitative analysis, six gates (AND, OR, XOR, MUX, AOI, Full Adder) are implemented using five logic families; CMOS, CPL, Domino, DCVS, CML. On the block level, a 16-bit CLA adder is also used as a test vehicle to evaluate the performance of each logic family.

3.6.1 Gate Simulation Setup

Zimmerman explained how the the input waveforms affect the performance of the logic families, specially CPL [69]. The rise and fall times control short circuit currents, charge sharing, power dissipation and speed of evaluation. For such reasons, it is preferred to drive each logic gate by a logic gate from the same logic style. Figure 3.23 shows the setup used to compare the different logic gates. For each simulation, ten logic gates of the same type were cascaded to decrease the percentage of error in the measurements. Each logic gate has a fan out of two gates of the same kind. The output of each gate is alternately connected to one of the inputs in the following stage. This ensures that the gate is properly loaded. This is particularly important for CPL implementations, where the output of the CPL gate is alternately connected to the "gate" or "drain" of the device in the next stage . A driving gate is added to drive the input of the first gate. The power and delay of the driving gate are not taken into account in the measurements.

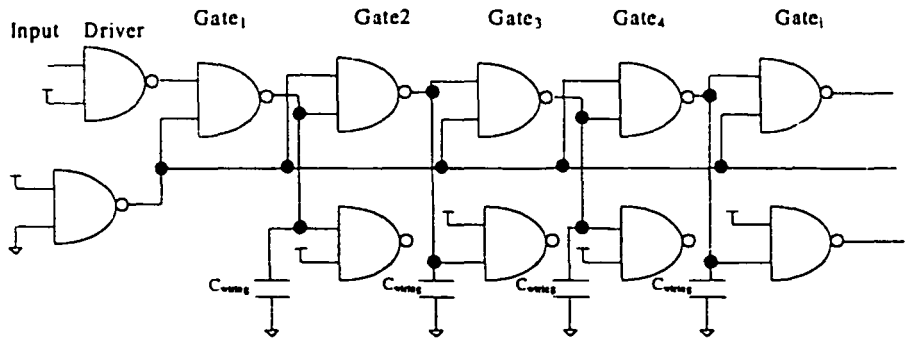


Figure 3.23: Logic gates' setup for simulation

All gates for all the logic styles were sized in order to achieve minimum Energy Delay Product (EDP). Delay was calculated as the worst case delay, whereas

the Power/MHz was used as a measure for power consumption. As previously explained, EDP was chosen as being the metric for comparison, as it best describes the efficiency of a circuit in terms of performance and power. The effect of interconnect (wiring) capacitances is included in the simulation as C_{wiring}. C_{wiring} is the estimated wiring capacitance between two logic gates, which are five logic gates apart both in the vertical and horizontal directions. Clock tree power was included in the power measurement for Domino and DCVS circuits. For MCML gates, the reference voltage is a static signal. Therefore, it's power dissipation was neglected during simulation since it charged the current source only once during startup.

Table 3.4: Parameters used for technologies

Technology	Supply Voltage (V)	Frequency (MHz)	V_{TH_N} (V)	V_{TH_P} (V)
0.8μm	5	50	0.812	0.902
0.6μm	3.3	100	0.657	0.9
0.35μm	3.3	200	0.55	0.75
0.25μm	2.5	400	0.45	0.6

Simulations were conducted in 0.25μm, 0.35μm, 0.6μm and 0.8μm CMOS technologies. Table 3.6.1 shows the operating supply voltage, frequency and threshold voltage used for every technology. An arbitrary frequency was used for each technology. To isolate the effect of the frequency in the final results, all the measurements were done per MHz.

For each case, the gate size was optimized for minimum EDP for each technology used. Because of the non inverting nature of pure Domino logic, the XOR and full adder were implemented using NP-Domino, while the MUX was not implemented in Domino because it is impractical. All the simulations were conducting using

HSPICE. The device models used should have 5 - 10% accuracy compared to the physical measurements.

3.6.2 CLA Adder Setup

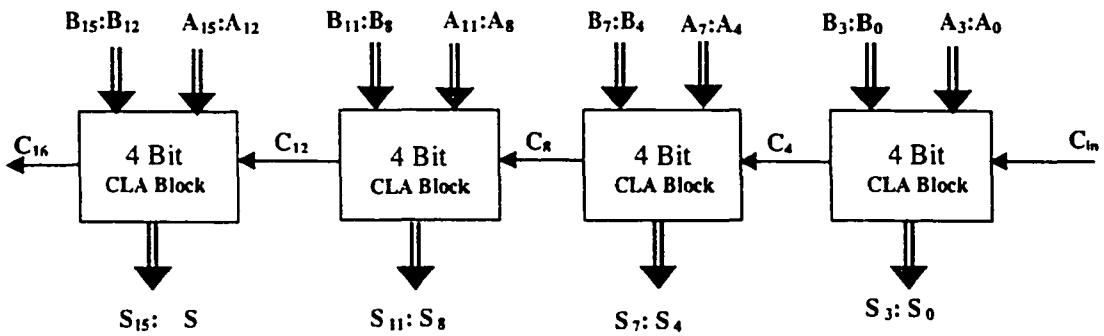


Figure 3.24: Architecture of a 16 bit CLA adder

On the block level, a 16 bit CLA adder [84] was used as a test figure for each of the five logic styles. Figure 3.24 illustrates the architecture of the 16 bit CLA adder. CLA adder was particularly selected to compare the different logic families because of the following reasons

- CLA adder utilizes a variety of logic gates with small and large fan-in and fan-out gates. It also includes simple gates (AND and XOR) as well as complex gates (block propagate and block generate).
- The 16 bit CLA adder contains nine cascaded logic levels which are suitable for testing the performance of cascaded logic gates in each case.
- CLA adder is not a regular architecture. Therefore, it is a good example for an arbitrary logic circuit.

The critical path delay of the CLA was calculated, where the carry propagated through the four "4 bit CLA adder"

3.7 Results and Analysis

3.7.1 Gate Results

To simplify the analysis of the simulation results, the logic gates were partitioned into two groups. The first includes the NAND, NOR and AOI gates (Group I). The second group includes the MUX, XOR, and the FA (Group II). Group I gates are usually implemented using single ended structures. The simulation results for the individual gates[1] are illustrated in Table 3.7.1.

Group I gates

Figure 3.25 shows the average normalized delay of Group I gates. So far DCVS logic is considered the fastest logic style to implement Group I gates. This is attributed to it's dynamic nature, while suffering from no contention currents that might reduce it's evaluation time. Group I gates implemented in CML also possess high speeds. As mentioned in Section 3.4.1, this is because CML's logic devices do not undergo a slow transition from cut-off to saturation mode. On the other hand, gates implemented in CPL are slow, because of the large overhead the keeper and inverter produce on each output node. Domino and CMOS implement gates

[1] Schematics of the various logic gates implemented using each logic style are shown in appendix (A)

Table 3.5: Logic gates comparison

Gate Type	Logic Style	Power (Norm.)				Delay (Norm.)				EDP (Norm.)			
		0.25	0.35	0.6	0.8	0.25	0.35	0.6	0.8	0.25	0.35	0.6	0.8
NAND	CMOS	1.0	2.28	3.54	10.9	1.0	1.29	1.65	02.88	1.0	3.79	9.63	90.8
	CPL	4.93	9.41	12.8	32.8	1.28	1.52	2.03	3.66	8.02	22	53	438
	Domino	6.8	15	17	44.1	0.77	0.91	1.34	2.35	4.06	12.5	30	244
	DCVS	4.9	11.1	17.9	41.2	0.8	0.95	1.19	2.45	3.13	10.1	25.4	248
	MCML	2.67	7.57	17.1	99	0.84	0.92	1.48	2.32	1.89	6.37	37.6	533
NOR	CMOS	1.0	2.3	3.69	11.2	1.0	1.23	1.53	2.74	1.0	3.49	8.62	83.8
	CPL	5.36	8.57	12.3	31.3	1.19	1.42	1.89	3.4	7.57	17.32	43.6	360
	Domino	7.14	12.9	13.4	40	0.59	0.75	1.13	1.93	2.51	7.32	17	149
	DCVS	4.75	10.8	17.4	40	0.73	0.87	1.09	2.24	2.53	8.15	20.5	200
	MCML	2.59	7.36	16.6	96.4	0.77	0.84	1.35	2.11	1.52	5.15	30.4	431
XOR	CMOS	1.0	2.2	2.72	8.91	1.0	1.55	2.06	3.65	1.0	5.31	11.5	119
	CPL	1.67	2.5	4.15	7.93	0.95	1.14	1.48	2.56	1.51	3.27	9.13	52
	Domino	3.48	7.41	10.6	49.1	0.66	0.85	1.16	2.03	1.51	5.31	14.2	202
	DCVS	1.62	4.57	6.67	22.2	0.84	0.92	1.21	2.12	1.15	3.85	9.75	99.8
	MCML	0.88	2.51	5.67	41	0.7	0.77	1.42	2	0.46	1.46	11.4	162
MUX	CMOS	1.0	2.02	3.75	9.55	1.0	1.19	1.49	2.75	1.0	2.87	8.34	72.4
	CPL	1.66	2.96	4.16	10.4	0.47	0.59	0.75	1.5	0.37	1.03	2.37	23.3
	Domino	-	-	-	-	-	-	-	-	-	-	-	-
	DCVS	2.14	5.51	7.27	46.2	0.48	0.6	0.85	1.26	0.49	1.98	5.3	73
	MCML	1.84	3.75	10	65.1	0.63	0.73	0.93	1.38	0.73	2	8.8	125
AOI	CMOS	1.0	1.49	2.35	5.72	1.0	1.4	1.83	3	1.0	2.91	7.87	51.5
	CPL	1.68	2.67	3.95	11.1	0.87	1.09	1.45	2.62	1.26	3.14	8.24	76
	Domino	1.84	3.47	5.37	18.7	0.95	1.23	1.65	2.7	1.65	5.27	14.6	136
	DCVS	2.54	4.74	7.72	42.5	0.52	0.6	0.8	1.4	0.68	1.73	4.94	83.6
	MCML	1.89	3.56	7.58	32.3	0.6	0.7	1.19	1.92	0.69	1.75	10.7	119
FA	CMOS	1.0	1.6	2.54	8.97	1.0	1.27	1.49	2.73	1.0	2.57	5.62	67
	CPL	1.37	3.33	4.52	15.6	0.89	1.03	1.28	2.24	1.1	3.56	7.46	78.3
	Domino	2.49	4.85	5.23	33.3	0.52	0.64	0.98	1.61	0.67	2	5	86
	DCVS	2	3.67	4.82	17.4	0.55	0.67	0.89	1.5	0.59	1.63	3.82	39.3
	MCML	1.32	2.61	7.85	37.4	0.53	0.58	0.92	1.1	0.37	0.89	6.6	44.7

with moderate speeds. It should be noted however, that the fastest NOR gates are implemented in domino logic.

Figure 3.25 shows also that the enhancement in speed over the generations is reduced for all logic styles except CMOS. The speed of CMOS circuits continue to systematically improve, while little speed enhancement (small slope) is noticed for the other logic styles. This is consistent with the qualitative analysis in Section 3.4 which indicated a speed reduction to the logic styles in the DSM regime. Therefore, technology scaling has the little effect on the speed of circuits implemented in CMOS.

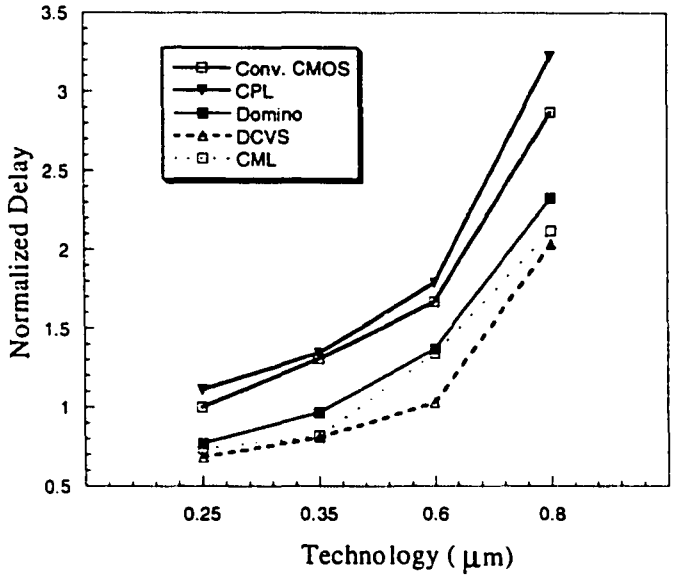


Figure 3.25: Average normalized delay for Group I (AND, OR, AOI)

Power comparison results for Group I are shown in Figure 3.26. CMOS power dissipation is the by far the lowest for all technologies. This is attributed to its low short-circuit currents and switching activity especially for monotonic gates. Dynamic logic styles (Domino and DCVS) in general are power inefficient, and are thus not recommended in low-power systems. This is mainly attributed to

their high switching activity, and high clock tree power. CPL also has high power dissipation because of the extra hardware used to restore the output voltage to V_{dd}.

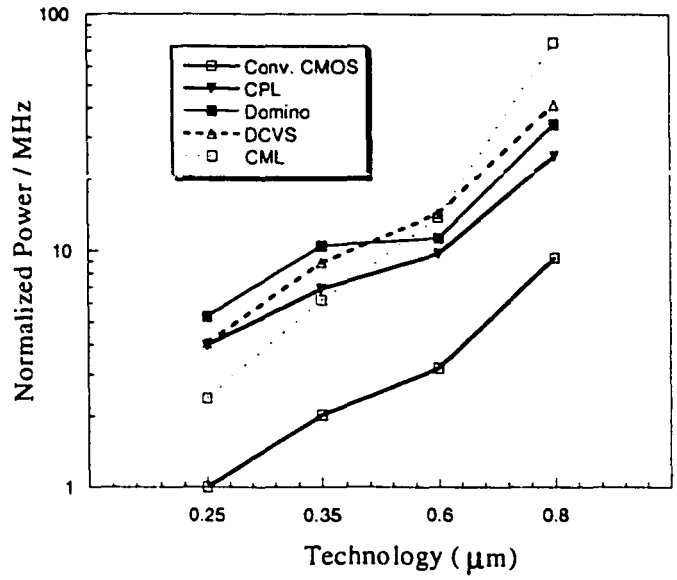


Figure 3.26: Average normalized power/MHz for Group I (AND, OR, AOI)

Figure 3.26 shows that logic styles continue to dissipate less power over the generations at different rates. The quantitative results in Figure 3.26 clearly show that:

- CPL has the least improvement in power dissipation. This is attributed to the large short-circuit currents produced in the DSM regime (Refer to Section 3.4.5).
- The power efficiency of CMOS continues to improve over the years.
- CML has the highest improvement in power over the years. As mentioned in Section 3.3.5, this is because CML is preferred in high frequency applications

in order to reduce the overhead of its static biasing power. High frequency applications mostly take place in small feature-size technologies.

Taking into consideration the delay and power values in Figures 3.25 and 3.26, a plot for the Energy-Delay product for Group I gates can now be plotted. This plot is shown in Figure 3.27. CMOS has the lowest EDP values because of its low power dissipation. MCML has a continually decreasing EDP over the generations which is attributed to its low delay and its power-efficient behavior in the DSM regime. Dynamic logic styles; Domino and DCVS, have almost the same EDP at long channel (0.8μm) technology. However, for smaller feature sizes, DCVS possesses a lower EDP because it does not suffer from contention (unlike Domino). Circuits implemented in CPL experience the highest EDP. This is due the large overhead of the restoration buffers, increased delay and growing short-circuit power dissipation in DSM technologies.

Group II gates

Figure 3.28 shows the average delay of Group II gates. MCML logic is the fastest logic style to implement Group II gates. This is because CML's differential nature best suits the implementation of XOR and MUX circuits. Group II gates implemented in DCVS also possess high speeds. This is again attributed to dynamic nature, while suffering from no contention currents that might reduce it's evaluation time. Group II gates are implemented in NP-Domino, which employs a complex structure (Figures 3.11 and 3.12), instead of conventional domino. However, NP-Domino still provides high speeds compared to static styles. This is mainly

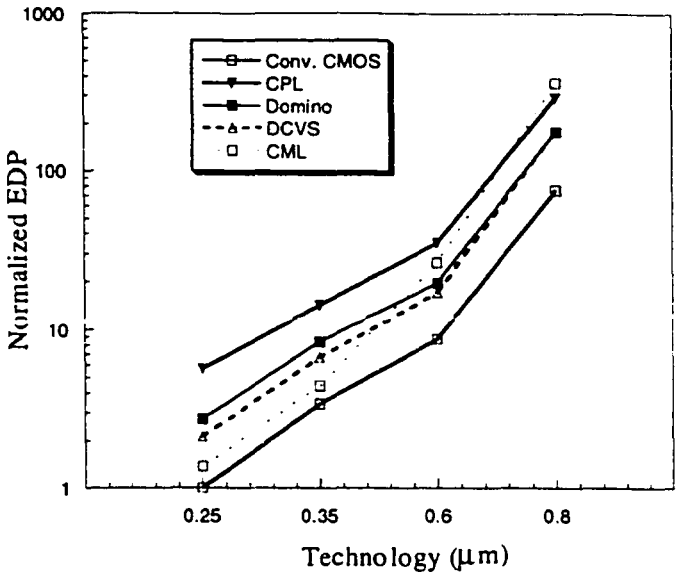


Figure 3.27: Average normalized EDP for Group I (AND, OR, AOI)

attributed to its dynamic nature. Gates implemented in static styles; CMOS and CPL, are slow. However, CPL implements the complex structures of Group II much more efficiently than CMOS. Although CPL's performance is strongly degraded in DSM technologies, it still has an upper hand over CMOS in implementing MUX and XOR circuits. Therefore, XOR and MUX are considered the least efficient gates to be realized using the CMOS implementation, because these gates require inverted signals as inputs.

Similar to Figure 3.25, Figure 3.28 shows that the enhancement in speed over the generations is reduced for all logic styles except CMOS. The speed of CMOS circuits continue to systematically improve, while little speed enhancement (small slope) is noticed for the other logic styles. This is again consistent with the qualitative analysis in Section 3.4 which stated a speed reduction to the logic styles in the DSM

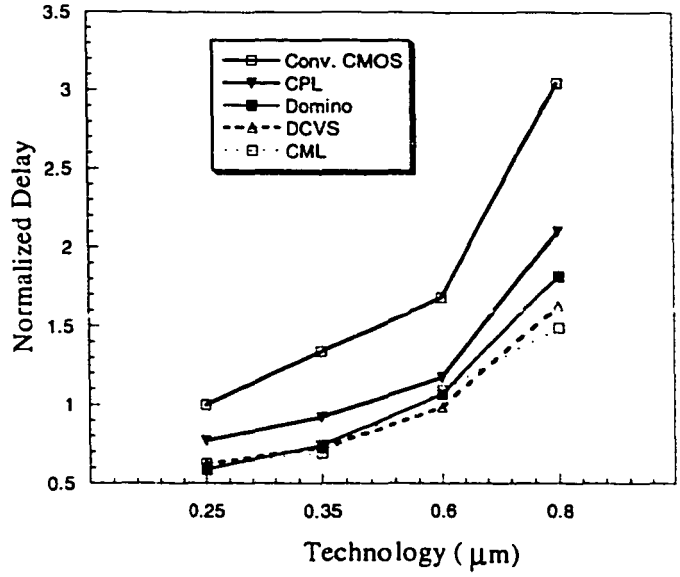


Figure 3.28: Average normalized delay for Group II (XOR, MUX, FA)

regime. Therefore, technology scaling has the least effect on circuits implemented in CMOS. CPL suffers from a sharp speed drop, but is still faster than CMOS in MUX and XOR gates.

Power comparison results for Group II are shown in Figure 3.29. Although CMOS experiences a large delay, its power dissipation is the lowest of all technologies. This is mainly attributed to its low short-circuit currents and switching activity. Again, dynamic logic styles (Domino and DCVS) are power inefficient, and are thus not recommended in low-power systems. This is mainly attributed to their high switching activity, and clock. On the other hand, CPL is relatively power-efficient in implementing XOR and MUX circuits.

The EDP values (see Figure 3.30) show that MCML has better EDP values in small feature sizes, because of the small delay and power. MCML is followed by

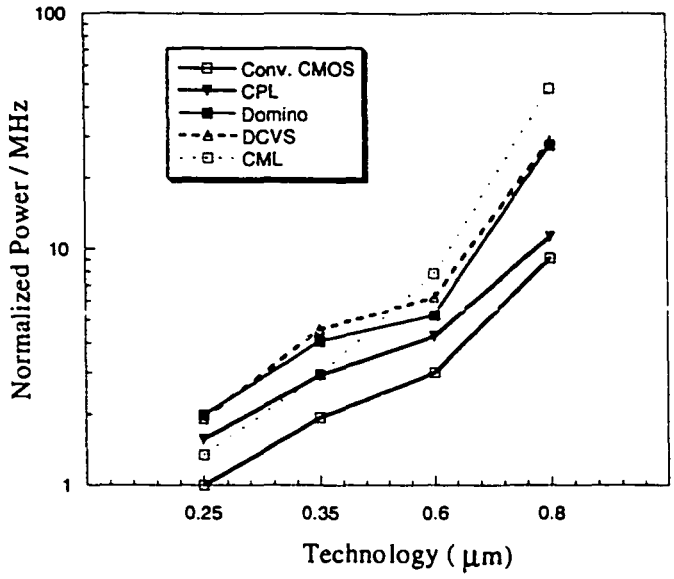


Figure 3.29: Average normalized power/MHz for Group II (XOR, MUX, FA)

DCVS, CPL, which proves that the differential logic styles are more suitable for complex logic gates. Performance of NP-Domino is not high enough because of the P logic blocks. Therefore NP-Domino's EDP is almost equal that of CMOS.

Therefore, XOR and MUX are considered the least efficient gates to be realized using the CMOS implementation, because these gates require inverted signals as inputs. CMOS is more efficient for implementing simple monotonic gates, such as NAND, NOR and AOI. Differential logic styles implement non monotonic gates (XOR, MUX and FA) more efficiently. The high dynamic power associated with dynamic circuits is partly attributed to its high switching activity.

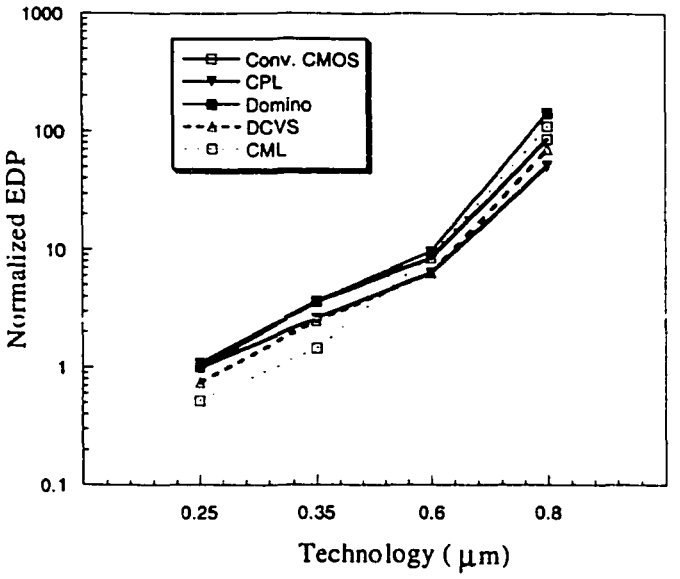


Figure 3.30: Average Normalized EDP for Group II (XOR, MUX, FA)

3.7.2 CLA results

Table 3.6 shows results of the the CLA adder. The normalized power, delay and EDP are indicated. It is clear from the table that Conventional CMOS implementations have the worst delay for all technology generations, and attains average power dissipation. CMOS delay is high because the critical path in the CLA adder has complex gates which are not suitable for CMOS. The power dissipation is high because of the high glitching activity, due to the large logic depth of the adder. Conventional CMOS is therefore, the least efficient way to implement the CLA adder. Domino logic comes as the second worst implementation because of using NP-Domino to implement the XOR gates.

The differential logic structures have better EDP value. This occurs because of the many AOI and XOR structures that are used to build the CLA adder. It

should be noted that in the implementation of the CLA using CPL logic, each gate was built using only one logic branch to reduce the power dissipation.

Table 3.6: CLA comparison

Logic Style	Power (Norm.)				Delay (Norm.)				EDP (Norm.)			
	0.25	0.35	0.6	0.8	0.25	0.35	0.6	0.8	0.25	0.35	0.6	0.8
CMOS	1.0	2.12	5.82	26.6	1.0	1.65	3.1	3.62	1.0	5.78	56	348
CPL	1.23	1.43	2.49	14.8	0.62	1.58	1.95	3.16	0.48	3.57	9.4	148
Domino	1.33	7.86	11.6	50.5	0.67	0.81	1.62	1.75	0.59	5.2	30.7	154
DCVS	1.57	2.96	3.9	14.6	0.74	0.91	1.17	1.54	0.85	2.46	5.3	34.2
MCML	1.96	3.31	4.22	21.5	0.6	0.81	1.15	1.88	0.71	2.17	5.58	75.4

3.8 Summary

Many factors are important in optimizing MOS transistor performance. Gate oxide thickness should be as thin as possible to improve short channel characteristics, maximize the drain current, and facilitate supply voltage scaling for reduced power. However, the minimum gate oxide thickness is dedicated by reliability, defect density, and gate capacitance considerations. Threshold voltage should also be set low to maximize the drain current and to facilitate the reduced supply voltage, but not low as to increase the OFF current and standby power to unacceptable levels or to result in functional failure of dynamic circuits.

As technology scales down, Conventional CMOS logic is the least affected logic style. Its device sizes become relatively smaller, because the PMOS will have almost the same mobility as the NMOS device. Thus, PMOS devices will not need any sizing up to achieve a desirable performance. The performance enhancement for

each new technology will generally decrease with each new generation of technology.

CPL performance degrades much faster than other logic styles because of the reduction of the ratio V_{dd}/V_{th} in technology scaling. Hot carrier effect makes it even worse by increasing V_{th} over the long term. CPL area tends to grow up, increasing power dissipation and area.

Domino's performance and power deteriorates, because of leakage and contention caused by the keeper transistor. DCVS is also affected by leakage power, but it does not have any contention problems during evaluation. Because interconnects are not scaled linearly with technology, the percentage of power consumed in the clock tree increases.

Although MCML tops the logic styles in many circuit implementations in terms of minimum EDP, it is not yet widely used. This occurs because of the complexity of MCML design cycle and the constant current source which is not suitable for digital systems especially during standby. Because of the hot carrier effects, MCML may have some trouble with V_{th} variations. But if MCML is used at a lower supply voltage, the effect of the hot carrier becomes less important.

Chapter 4

Dynamic Current Mode Logic, A New Low-Power High-Performance Logic Family

Reduced-swing logic styles have been known for long time. However, they have not been widely used in digital design because of their high static power dissipation and complex design. This chapter introduces a new reduced swing logic family called Dynamic Current Mode Logic (DyCML). Combining the advantages of MCML (MOS Current Mode Logic) circuits with those of dynamic logic families, DyCML logic circuits achieve high-performance at low supply voltage, and low-power dissipation. Unlike CML circuits, DyCML does not have a static current source, which makes it a good candidate for portable devices and battery powered systems.

The first section of this chapter, explains the concept of reduced swing logic circuits. The following section outlines the history of MCML circuits, as an example

of reduced swing logic, followed by a qualitative analysis of its advantages and disadvantages. Then, the DyCML circuit architecture, and theory of operation is introduced. Following section describes the details of the implementation, and the simulation results. The last section presents the experimental results followed by a summary.

4.1 Introduction

As technology scales down, the transistor capacitance and gate parasitics are reduced, leading to smaller gate switching power. However, interconnect capacitance does not scale at the same ratio. This occurs because of the physical limitations on the minimum width of aluminum interconnects that can be deposited on the chip. Copper interconnect widths can be reduced more than the equivalent aluminum interconnects [17]. Consequently, Copper interconnects seem to solve this problem for two or three more process generations. However, this will not be enough for future process generations. Therefore, it is important to develop new logic styles that reduce power dissipation in the interconnects.

The effect of the interconnects on dynamic power dissipation can be explained using the following equation:

$$P_{Switching} = F_{Switching} \cdot V_{dd} \cdot V_{swing} \cdot C_{interconnect} \tag{4.1}$$

where $F_{Switching}$ is the switching rate of the logic gate driving the interconnect wire, V_{dd} is the supply voltage, V_{swing} is the output voltage swing, which is normally equal

to V_{dd} and $C_{interconnect}$ is the interconnect capacitance.

The interconnect power dissipation may be reduced by reducing the switching activity, supply voltage, voltage swing, or the interconnect capacitance itself. Reducing the supply voltage is not recommended because it degrades the performance. Lowering the output voltage swing reduces the interconnect power dissipation as long as the voltage swing can tolerate the noise.

Reduced output swing may be obtained in two schemes as follows:

1. The logic levels are V_{dd} and $V_{dd} - V_{swing}$. In this case, the logic evaluation relies on NMOS transistors such as CML logic. PMOS transistors are either completely OFF, or partially OFF. Therefore, they are not likely to be used for logic evaluation, because their driving capabilities will be limited.

2. The logic levels are 0 and V_{swing}. In this case, the logic relies on PMOS transistors leading to slower gates. The NMOS transistors are either OFF or partially OFF. Thus, the NMOS driving is limited, and should not be used for logic function evaluation.

Because PMOS driving capability is less than the equivalent NMOS, the second style is not used because it is slower.

CMOS and other full swing logic circuits may be defined as Voltage Mode Logic (VML) circuits. These circuits rely on the value of the input voltage(s) to switch the different gate transistors to either the ON or the OFF state. This differs from the reduced voltage swing circuits, where some transistors are completely ON while the other transistors are partially ON. Such transistors do not work properly in

voltage mode logic. Current mode logic, where all the transistors are ON (partially or fully), is preferred for such cases. The value of the output logic of the gate is based on the the difference between currents passing through the two branches of the circuit, as in the case of MOS current mode logic MCML.

4.2 MOS Current Mode Logic (MCML)

Designers have used Mode Logic (CML) for a long time in high speed logic circuits. Unlike traditional logic styles, CML power is fixed and not a function of operating frequency. For mixed signal designs, CML has been the logic style of choice, because of its low switching noise, stable supply rails and high noise immunity at high operating frequencies. Unfortunately, CML has not received much interest in digital VLSI design. The complex design cycle, the large area and most of all, the static power dissipation have prohibited digital VLSI designers from using it on large scale designs.

CML was originally developed for bipolar technology, as an extension for differential pair circuit, which is used widely in analog designs. When MOSFET technology evolved later, the differential implementation pair was not feasible because of the large variation in V_{th}. Therefore, MOS CML was not possible until Elmasry introduced an implementation based on depletion transistors [12]. However, depletion technologies became obsolete in the 1980's, while CMOS technologies became the de-facto of digital design. CMOS technology has made MOS CML possible [79].

An MCML inverter is shown in Figure 4.1. MCML's architecture and theory of operation are explained in detail in chapter 3. In [78], an analysis for energy and

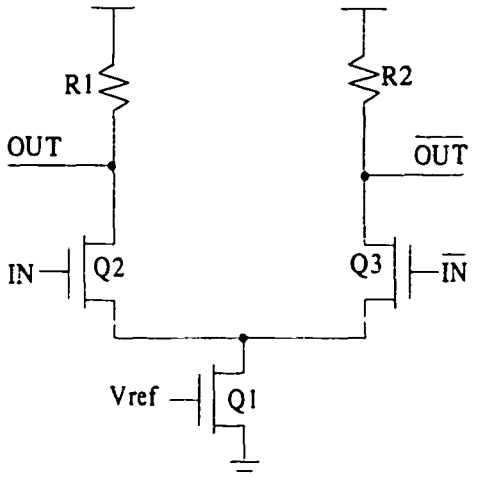


Figure 4.1: MCML logic gate

delay of MCML circuits is given. To summarize, the advantages of MCML are:

- High speed since it does not operate in the cut-off region.
- Low switching noise because of the constant current source.
- High noise immunity because of its differential nature.
- Capability to operate at low supply voltage.
- Efficient implementation of arithmetic circuits (XOR styles).
- Controllable output voltage swing.
- Small output swing efficiency in driving data busses, and high load signals, which reduces dynamic power dissipation and delay.

MCML's disadvantages are:

- DC current source wastes power during standby which is not suitable for digital circuits.
- The number of series transistors is equal to the number of input variables except for the case when some variables are exclusive (not "1" simultaneously).
- The large load resistors require special fabrication technology.
- A special tree for reference voltage V_{ref} distribution needs to be added, which increases the complexity of the layout stage.

It is clear that most of the problems associated with MCML are caused by the static current source and the balancing of loads. As a solution Mizumo *et.al.* have suggested an adaptive pipeline technique to reduce the power in non critical paths in an MCML system [78].

4.3 DyCML Circuit Architecture and Operation

To achieve the high speed characteristics of MCML, but exclude its drawbacks, the current source and load resistors of the MCML gate should be redesigned. Dynamic Current Mode Logic (DyCML) employs a dynamic current source with a virtual ground to eliminate the static power and other side effects associated with the conventional static current source. The new architecture also utilizes active loads, instead of the traditional load resistors to reduce power dissipation.

Figure 4.2 shows the basic architecture of a DyCML logic gate. It consists of the following: an MCML block for logic function evaluation, precharge circuit(Q_2,

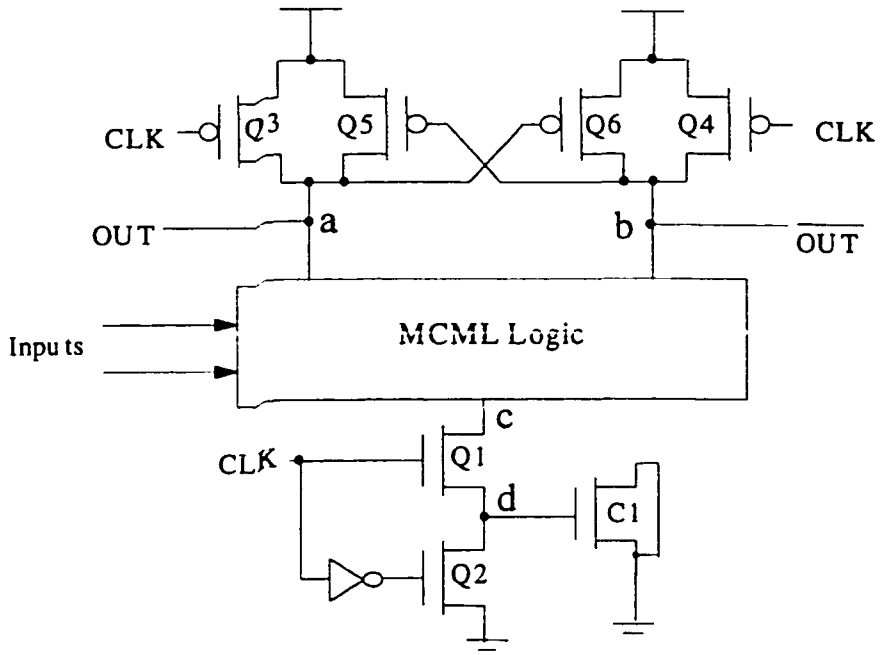


Figure 4.2: Architecture of a DyCML gate

Q_3, Q_4), dynamic current source (Q_1, C_1), and a latch to preserve logic value after evaluation (Q_5, Q_6). The operation of the DyCML is described as follows: during the low phase of the clock, the precharge transistors Q_3, Q_4 turn ON to charge the output nodes to V_{dd}, while transistor Q_2 turns ON to discharge capacitor C_1 to G_{ND}. Meanwhile, transistor Q_1 is OFF, eliminating the DC path from V_{dd} to GND.

During the high clock phase, the precharge transistors Q_2, Q_3 and Q_4 turn OFF, while transistor Q_1 switches ON creating a current path from the two precharged output nodes to the capacitor C_1. The latter acts as a virtual ground. These two paths have different impedances depending on the logic function and inputs; therefore, one of the output nodes drops faster than the other node. The cross

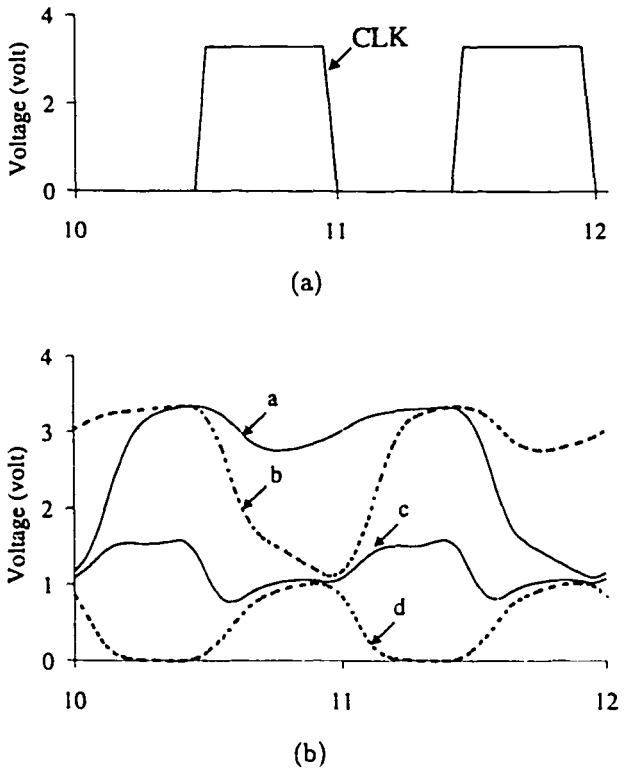


Figure 4.3: Voltages at different nodes in the DyCML gate

connected transistors Q_5, Q_6 speed up the evaluation, and maintain the logic levels after evaluation. During the evaluation phase, when one of the output nodes voltage drops less than $V_{dd} - |V_{TP}|$, the transistor whose gate is connected to this node turns ON, charging the other output node back to V_{dd}. Figure 4.3(b) shows the voltages at different nodes in a DyCML gate.

Transistor C_1 is used as a capacitor. It acts as a virtual ground to limit the amount of charge transferred to the output node(s). The value of this capacitor is dependent on the value of the load capacitance (fan out), and the required output

voltage swing. From Figure 4.3(b), it is clear that V_{ds} of transistor $Q_1 \approx 0$ after the evaluation, and the voltage of node d is $\approx V_{dd} - V_{swing}$. Since the charge stored on transistor C_1 equals the charge drained from the output nodes, the following equations are used to calculate the size of transistor C_1.

$$V_{swing} * C_L = W_{C1} * L_{C1} * C_{ox} * (V_{dd} - V_{swing}) \tag{4.2}$$

$$W_{C1} * L_{C1} = \frac{V_{swing} * C_L}{C_{ox} * (V_{dd} - V_{swing})} \tag{4.3}$$

where V_{swing} is the output voltage swing, W_{C1} and L_{C1} are the width and length of transistor C_1, respectively, C_{ox} is the gate oxide capacitance per unit area, and C_L is the load capacitance per output node. The parasitic capacitances of the MCML block are included in C_L, as well as the gate capacitance of transistors Q_5, Q_6, and the parasitic capacitances of the precharge transistors Q_3, Q_4. Although the voltage of only one output node drops, a small current (charge) flows from the other output node to C_1, until the latch switches ON. Thus, transistor C_1 should be sized up to accommodate this extra charge. From simulation results, the required increase in C_1 size was found to be $\approx 20\%$.

The area of transistor C_1 is a minor fraction of the total gate area. For example, for a DyCML gate with a fan out of 8, implemented in $0.6\mu m$, transistor C_1 area should be $10\mu m^2$, while the logic gate area is around $250\mu^2$; i.e., the capacitor size is about 4% of the gate area. The size of transistor C_1 is small because of the

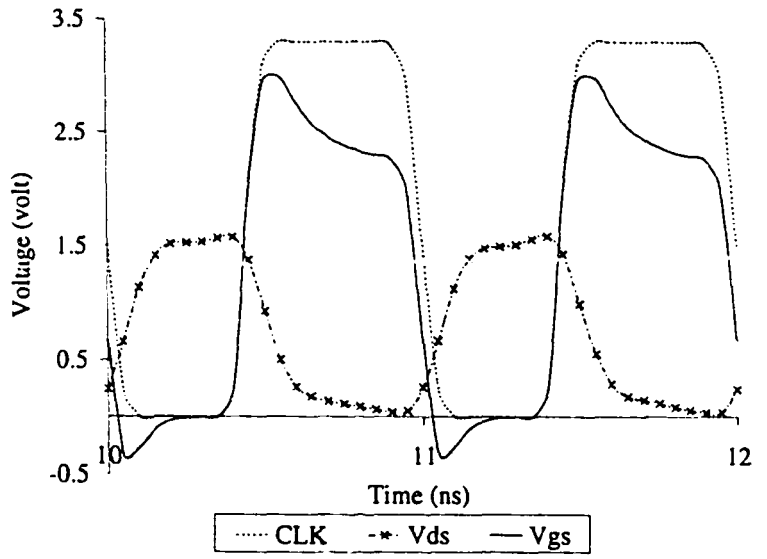
following:

- V_{swing} is around 20% of V_{dd}.
- C_{ox} is large especially for new fabrication technologies where the thickness of the gate oxide is reduced.
- The input transistors (C_L) of the DyCML logic gate are small because these transistors are responsible for steering the current only. The logic transistors are not supposed to completely charge or discharge the output load.

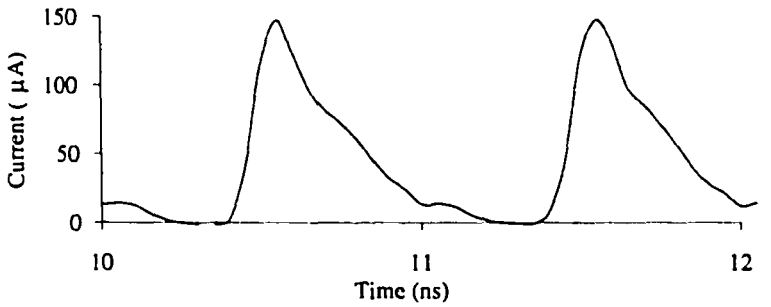
4.3.1 Operation of the Dynamic Current Source

Transistor Q_1 and C_1 construct a dynamic current source, which enhances the performance of the DyCML gates dramatically. The operation of the current source is as follows: at the beginning of the evaluation phase, transistor Q_1 acts as a current source with its gate biased with V_{dd}, driving a large current from the MCML block. As the current charges the capacitor, node d voltage starts to rise, limiting the current flowing through Q_1 until it eventually turns OFF when its V_{ds} becomes zero. This large instantaneous current speeds up evaluation leading to a smaller delay. The instantaneous current is shown in Figure 4.4(b) whereas Figure 4.4(a) shows V_{gs} and V_{ds} of transistor Q_1.

Dynamic power dissipation of DyCML gates is small compared with other dynamic differential logic styless because of the reduced output swing and small input transistors. The latch connected transistors Q_5, Q_6 eliminate the subthreshold leakage problem that degrades the stability of dynamic logic circuits.



(a) Voltages on transistor Q_1



(b) Current passing through transistor Q_1

Figure 4.4: Dynamic current source voltages and current

DyCML does not suffer a static power dissipation because transistors Q_1 and Q_2 would never turn ON simultaneously. Only dynamic power exists, and it is independent on the input combinations. This occurs because the voltage at one of the output nodes is V_{dd}, whereas the other drops to $V_{dd} - V_{swing}$ after each

precharge/evaluation cycle.

The proposed DyCML gate operates properly at a supply voltages as low as $V_{TN} + |V_{TP}|$. This value guarantees that during the evaluation phase, the latch (Q_5, Q_6) switches ON to avoid any charge leakage.

4.3.2 Cascading DyCML Gates

DyCML gates may be cascaded in two different fashions by using.

- a Clock Delay (CD) mechanism where the clock signal is buffered from one gate to another.
- a self timing scheme where a gate generates the clock signal for the gates in the following logic level.

Clock Delay (CD)

Clock delay is a well known scheme in dynamic circuits. The clock signal is delayed between cascaded gates by adding a buffer to reduce the short circuit current occuring at the start of the evaluation phase. Figure 4.5 shows the delayed scheme structure.

A single clock buffer may be used to generate the clock signal feeding more than one gate. This is possible as long as the gates have equal logic depths. This scheme is the simplest and gives the best power and delay results. However, it must be clear that the clock delay should be larger than the gate delay. From the simulation results, it was found that even complex DyCML logic gates with large

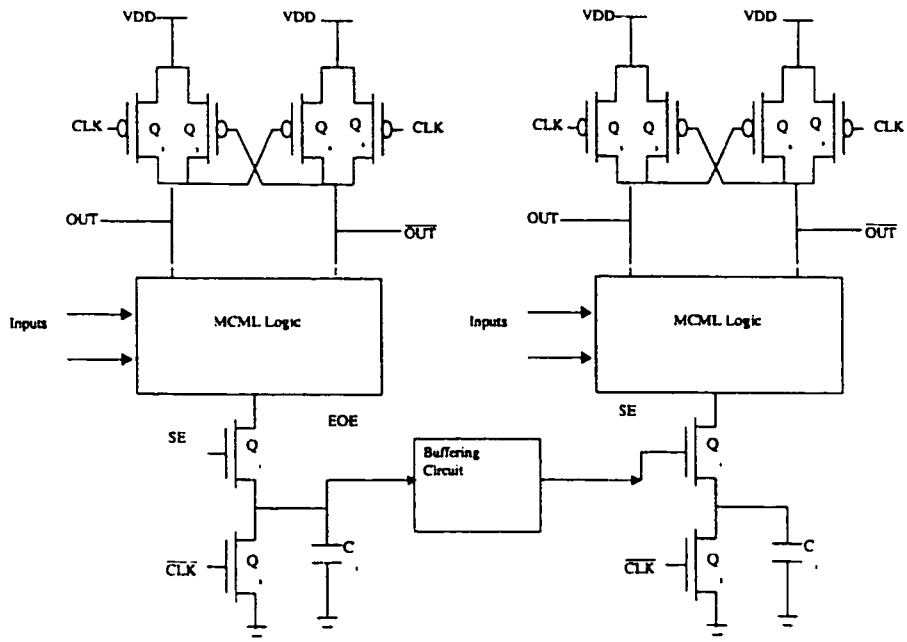


Figure 4.5: Clock delay scheme

fan outs would have half the delay of the clock buffer. Therefore this condition is satisfied because of the speed of the DyCML gate.

Self Timed Scheme (ST)

Self timing requires each gate to generate a completion signal for the following logic level. In DyCML, this signal may be the voltage on the transistor/capacitor C_1 (node d in the DyCML gate schematic). A special buffer is used to convert this signal to a full swing signal to be used as the clock signal for the next block. Figure 4.6 shows the architecture of this buffer. It consists of a cascade of two clocked inverters. The PMOS transistor of the second buffer is removed to reduce the delay

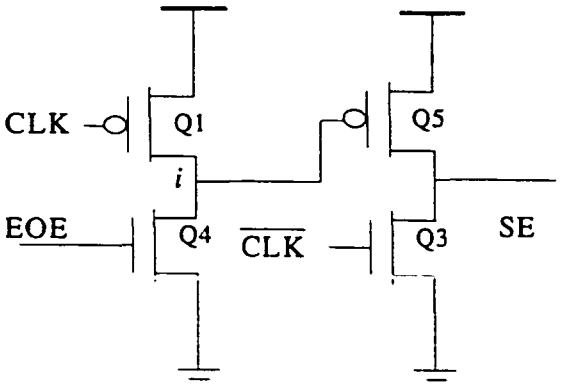


Figure 4.6: Self timing buffer

of the generated clock signal. The input to the first inverter EOE (the End Of the Evaluation) is the voltage on the transistor C_1 from the previous logic level.

The buffer operates a follows: when the clock (CLK) is low, transistor Q_1 turns ON charging node i to V_{dd} which turns transistor Q_5 OFF. Transistor Q_3 turns ON and discharges the output node to "0". Since the transistor C_1's gate is discharged to "0" and the clock is low, transistor Q_4 turns OFF during this clock phase.

When the clock signal becomes high, transistor Q_1 turns OFF while transistor Q_3 turns OFF. Until EOE input starts to rise, no current will pass from node i to the ground, keeping transistor Q_5 OFF. When the input starts to rise, transistor Q_4 switches on, discharging the node i to "0". Consequently, transistor Q_5 turns ON to charge the output node to V_{dd}. Figure 4.3.2 shows the voltages of various nodes in the buffer.

When the supply voltage drops, the delay of each gate will vary depending on the complexity and sizing of the gate's transistors. This clocking scheme is more appropriate for circuits with large variations in the supply voltage. The reason is

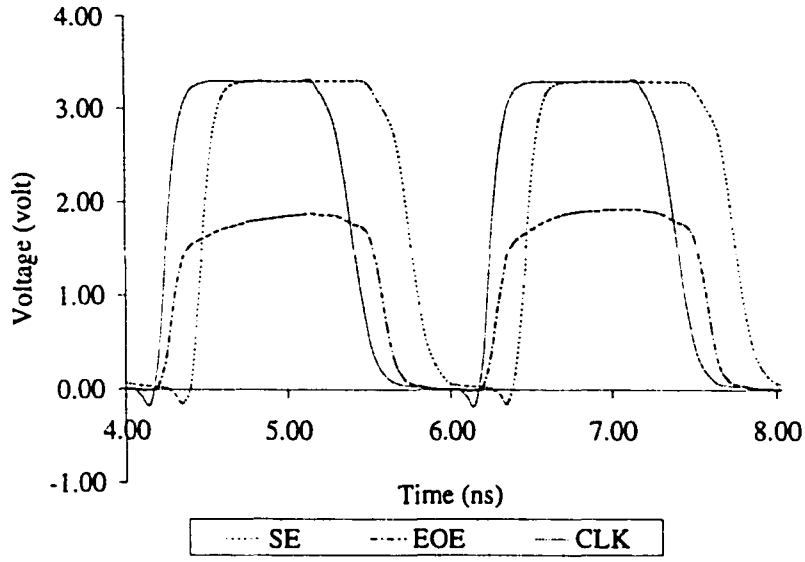


Figure 4.7: Voltages at different nodes in the self timing buffer

that each logic level will not start evaluation until the previous level has already evaluated, unlike the CD technique where the gate will start evaluation as soon as the delayed clock signal arrives. The price for the increased stability is higher delay, and power dissipation because of the buffers.

4.3.3 DyCML-CMOS Interfacing

DyCML gates may be used in conjunction with CMOS gates in the same design. Inputs of DyCML logic may be connected directly to CMOS gates' outputs. No buffering or interfacing circuits are required. To connect the output DyCML gates to the input of CMOS gates, a special buffer is required to convert the reduced swing signal to a full swing signal. Many differential-single ended buffers exist.

Unfortunately, most of them are complex and they rely on a DC current bias. To

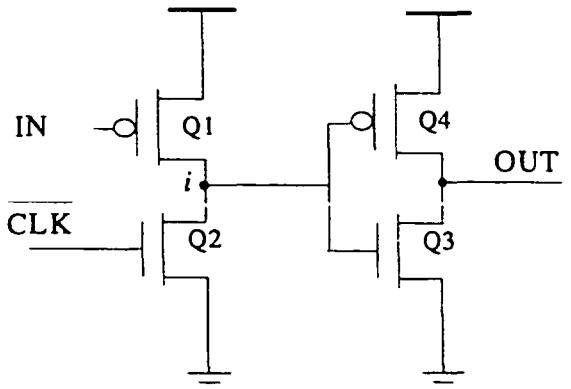


Figure 4.8: Differential-single ended buffer

take advantage of the presence of a clock signal in DyCML logic, a new conversion circuit is designed as shown in Figure 4.8. It consists of a clocked inverter followed by a regular CMOS inverter. The operation of the buffer is as follows: when the clock is "0", transistor Q_2 is ON discharging node i to ground, and therefore, the output node becomes high. Since the DyCML gate outputs are precharged to V_{dd} when the clock is low, transistor Q_1 is OFF. When the clock becomes high, transistor Q_2 turns OFF. Depending on the input signal (the output of the DyCML gate), transistor Q_1 will either turn ON leading to a "0" output, or stay OFF keeping the output at "1". To speed up the interfacing circuit, the voltage swing of the DyCML needs to be increased. This increase is required only at the gates driving the CMOS logic gates. The voltages on the interfacing circuit are shown in Figure 4.3.3.

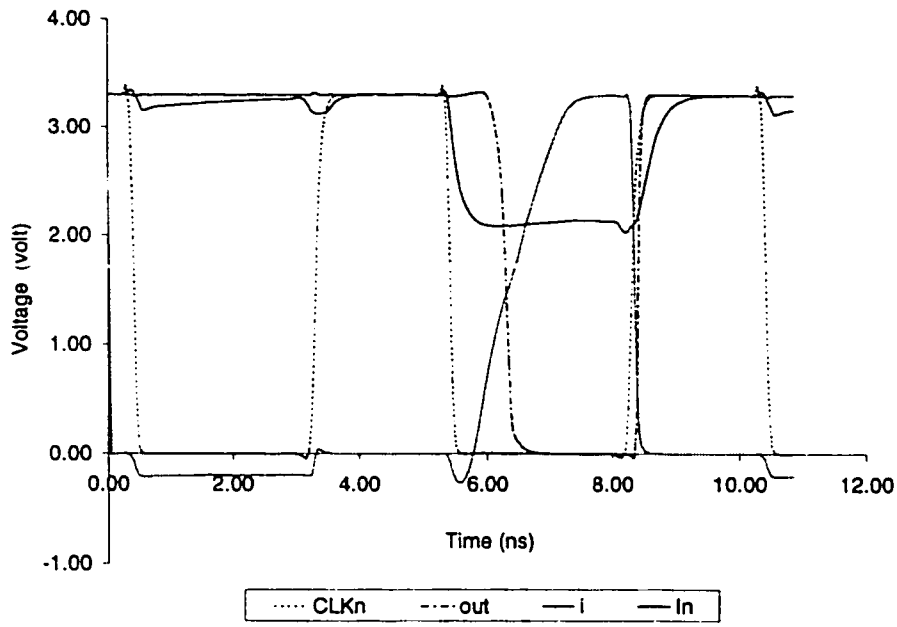


Figure 4.9: Voltages at different nodes in the reduced-full swing buffer

4.4 Circuit Implementation and Simulation Results

To evaluate the performance of the DyCML logic style, a set of logic gates were designed, and simulated using 0.6μm CMOS (HP/MOSIS/CMC) technology. This technology has an effective channel length of 0.5μm, and threshold voltages of about 0.7 and 0.9 volts for the NMOS and PMOS transistors, respectively. Then, DyCML logic is compared to five well known logic styles; namely: standard CMOS, Complementary Pass Logic (CPL), Domino, Dynamic Differential Cascode Voltage Switch (DDCVS), and MCML. To verify the visibility of DyCML logic on the block

level, a 16 bit carry lock ahead adder is designed, and compared with other logic styles.

4.4.1 Gate Simulation and Comparison

Stability

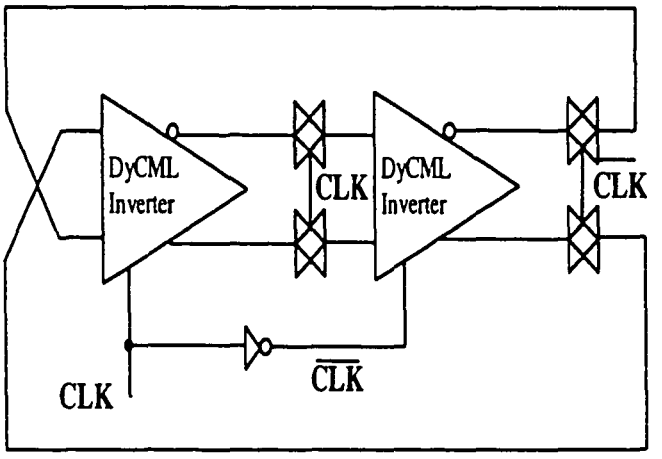


Figure 4.10: Divide by 2 DyCML circuit

A divide-by-2 circuit (T FF) is designed to determine the effect of supply voltage scaling on the performance of the DyCML logic. The schematic of the circuit is shown in Figure 4.10. The maximum operating frequency of the circuit is defined as the maximum frequency at which the divide-by-2 circuit is able to generate a voltage swing of 20%V_{dd} at the output nodes. Figure 4.11 shows the maximum frequency versus supply voltage for the divide by 2 circuit.

The simulation results show that the maximum toggle frequency of DyCML exceeds 3 GHz, which is 60% higher than the equivalent CMOS circuit. The high

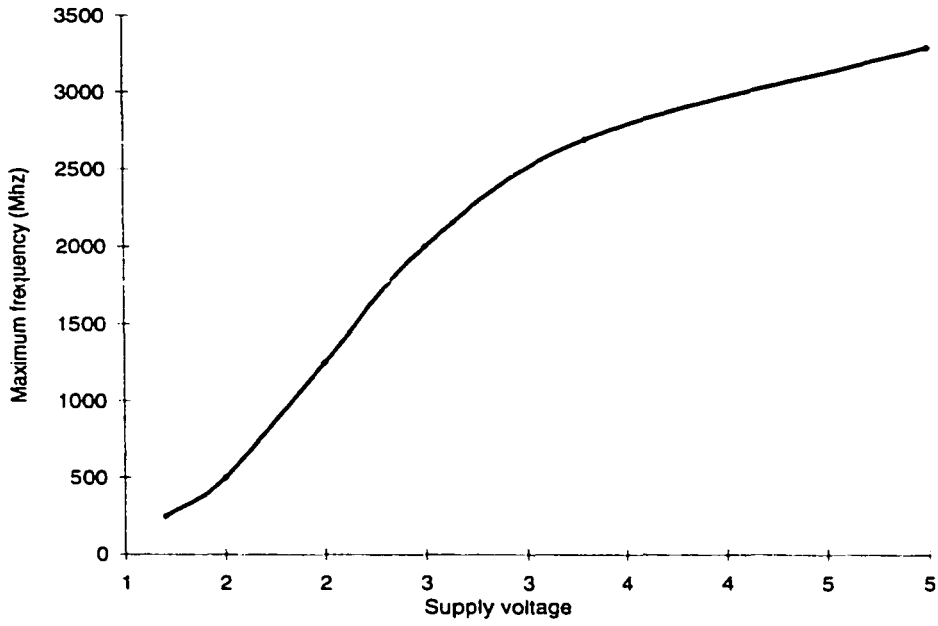


Figure 4.11: Maximum operating frequency vs. supply voltage

speed is achieved mainly, because of the reduced output swing, which requires a smaller amount of charge to be transferred through the logic block. Also, because the logic transistors are ON at the start of evaluation phase unlike CMOS logic where some transistors are OFF, they should be turned ON before the output nodes toggle states.

Gate Level Comparison

As explained earlier in chapter 3, the comparison of different logic styles should be generic. Realistic test circuit architecture, and test conditions are mandatory for a reliable comparison between different logic styles. The test scheme shown in Figure 4.12 is used to compare the different logic styles. For a complete description and

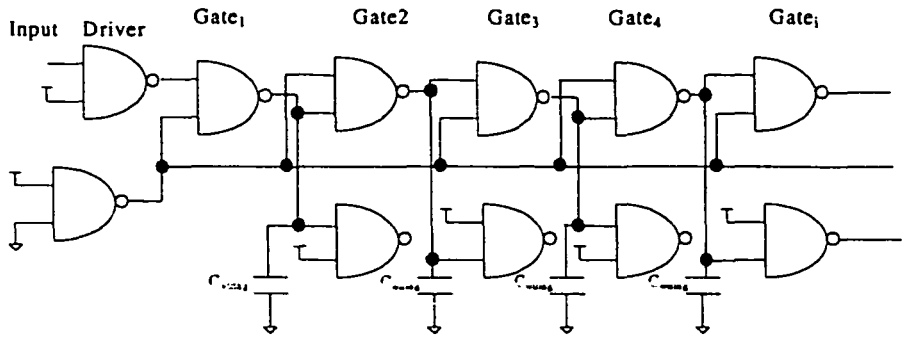


Figure 4.12: Simulation setup for logic gates

explanation of the simulation setup, refer to section 3.6.1.

For comparison purposes , five of the most frequently used logic gates were chosen, namely, NAND/AND, NOR/OR, XOR, MUX, and AOI. The full adder is also included, as it has been used historically to evaluate logic families. The two different cascading techniques for DyCML (CD, ST) were compared to CMOS, CPL, Domino, DDCVS and MCML logic styles. The simulations were executed at 100 MHz frequency while the voltage supply is 3.3 volts. Because of the non-inverting nature of conventional Domino logic, the XOR and full adder were implemented using NP-Domino, whereas the MUX was not implemented in Domino because it is impractical.

Table 4.4.1 shows the simulation results for the different logic styles. All the results are normalized with respect to CMOS results to simplify the comparison.

For simple logic gates (AND, OR, AOI), CMOS has the lowest power dissipation because of the overhead of the extra circuitry in DyCML gates. For more complex gates, (XOR, MUX, FA), DyCML had the smallest power dissipation. The self timed (ST) DyCML had 20 to 30% more power compared to the clock delayed

Table 4.1: Logic Gates Comparison

Gate Type		CMOS	CPL	Domino	DDCVS	MCML	DyCML (CD)	DyCML (ST)
NAND	power	1.00	3.63	4.25	5.04	4.83	1.02	1.32
	delay	1.00	1.23	0.81	0.72	0.90	0.57	0.68
	EDP	1.00	5.53	2.80	2.64	3.91	0.33	0.61
NOR	power	1.00	3.32	3.64	4.70	4.50	0.95	1.23
	delay	1.00	1.23	0.74	0.71	0.88	0.56	0.67
	EDP	1.00	5.06	1.98	2.38	3.53	0.30	0.55
XOR	power	1.00	1.52	3.88	2.45	2.09	0.48	0.63
	delay	1.00	0.72	0.56	0.59	0.69	0.41	0.48
	EDP	1.00	0.79	1.23	0.84	0.99	0.08	0.14
MUX	power	1.00	1.11	-	1.94	2.69	0.51	0.66
	delay	1.00	0.51	-	0.58	0.63	0.36	0.43
	EDP	1.00	0.28	-	0.65	1.05	0.07	0.12
AOI	power	1.00	1.68	2.28	3.28	3.22	1.01	1.21
	delay	1.00	0.79	0.90	0.74	0.65	0.41	0.49
	EDP	1.00	1.05	1.85	1.80	1.36	0.17	0.29
FA	power	1.00	1.78	2.06	1.90	3.10	0.49	0.61
	delay	1.00	0.86	0.66	0.60	0.62	0.41	0.48
	EDP	1.00	1.33	0.89	0.68	1.17	0.08	0.14

(CD) DyCML version. Other logic styles have higher power dissipation for both simple and complex logic gates.

The delay of DyCML gates is the lowest among all the logic styles. As mentioned earlier, this is a result of the high evaluation current, reduced output voltage swing, and the logic transistors that do not operate in the cut off region. The second best delay is obtained using DDCVS followed by Domino, MCML, CPL, and CMOS, respectively. The DyCML (ST) had 17 to 30% higher delay compared to the DyCML (CD) because of the self timing circuitry.

Since energy-delay is equal to $delay^2.power$, DyML has the best EDP product among all the logic styles.

From the simulation results, it is clear that DyCML circuits achieve high speed at a reasonable power dissipation. DyCML is more suitable for complex logic gates, where the power overhead of the extra circuitry vanishes with respect to the power of the logic evaluation block.

4.4.2 Block Level Comparison

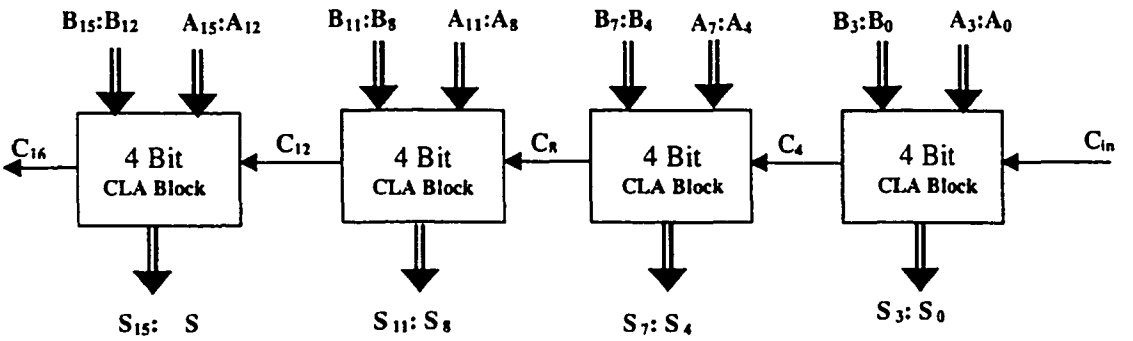


Figure 4.13: Block diagram of a 16 bit CLA adder

In order to verify the functionality of DyCML on the block level, a 16 bit Carry Look Ahead (CLA) adder [84] is used. A simple CLA block diagram is shown in Figure 4.13. Figure 4.14 illustrates a generate gate implemented in DyCML logic as an example for complex logic gates. The gate utilizes the exclusive relationship between propagate and generate signals to reduce the number of transistors, and the gate complexity. The logic expression for this gate is

$$G^* = G_3 + P_3.G_2 + P_3.P_2.G_1 + P_3.P_2.P_1.G_0 \tag{4.4}$$

The reasons for choosing CLA adder as a comparison Figure are explained in

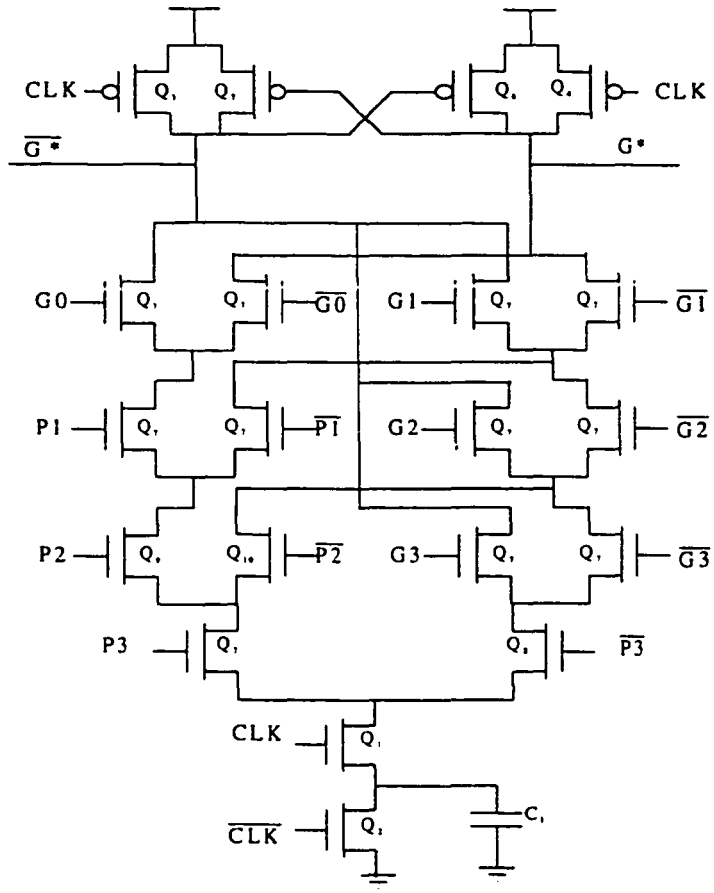


Figure 4.14: Generate gate of a 4Bit carry look ahead adder

section 3.6.2. Table 4.2 presents the power (including clock power if applicable), delay, power-delay product, and energy-delay product results for the seven logic styles. All the results are normalized to CMOS logic results. Both versions of DyCML proved to have the smallest delay among all the logic styles, because of the large evaluation current and reduced output swing. DyCML also had limited power dissipation due to its reduced output swing. DyCML tops all logic styles for

minimum EDP, which proves that the CLA adder is efficiently implemented using DyCML. The clock power in DyCML is only $\approx$ 35% of the total power dissipation, because all the evaluation and precharge transistors are minimum size transistors.

Logic style	Power (norm)	Delay (norm)	PDP (norm)	EDP (norm)
CMOS	1.0	1.0	1.0	1.0
CPL	0.42	0.63	0.26	0.16
Domino	2.0	0.52	1.04	0.54
DCVS	0.67	0.38	0.25	0.09
MCML	0.72	0.37	0.27	0.1
DyCML (CD)	0.47	0.27	0.13	0.034
DyCML (ST)	0.57	0.34	0.19	0.065

Table 4.2: CLA comparison

4.5 Experimental Results

The 16-bit DyCML(CD) CLA adder test chip was fabricated in a $0.6\mu m$ CMOS process. The chip has a built in clock recovery circuit, and a scan chain to force the CLA inputs into the chip. To monitor the outputs, reduced to full swing buffers are used to convert the outputs of the CLA to full swing signals before the output pads. The adder and buffers are designed on $690 * 160\mu m^2$ compared to $870 * 150\mu m^2$, for the equivalent CMOS implementation. The microphotograph of the chip is shown in Figure 4.15.

The chip was tested on frequencies ranging from 0 to $400MHz$. The testing was done in two phases. The first phase checks for the functionality. In this test, a set of random input vectors is shifted into the chip, and the output is compared to

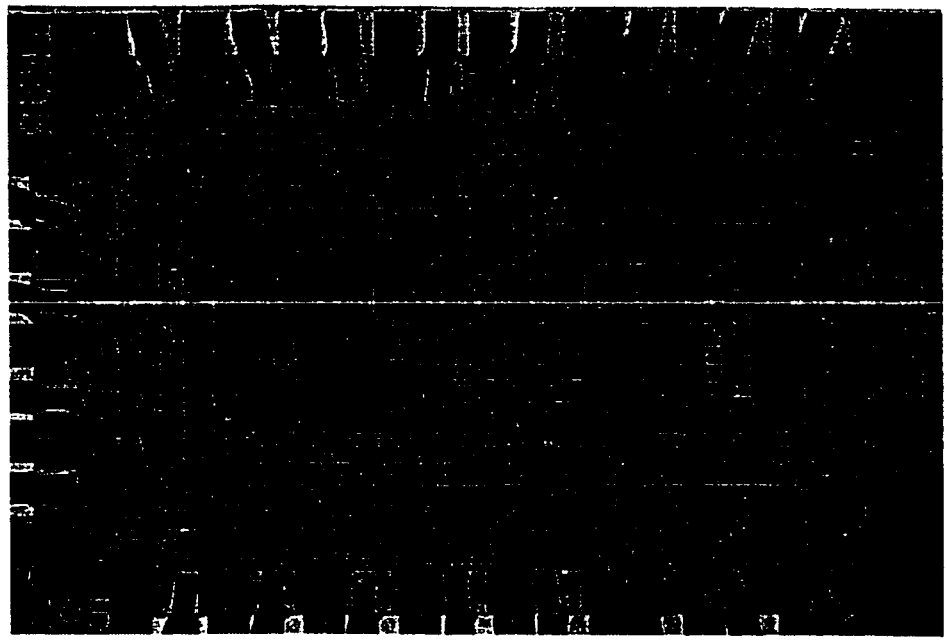


Figure 4.15: Microphotograph of the 16-bit CLA adder

the expected results. The test vector generation and validation is carried out using IMS XL100 logic tester running at 50 MHz. The CLA chip operates successfully down to a supply voltage of 2.2 at 50 MHz. Because of the limitation on the clock frequency of the IMS tester, another test setup had to be implemented to measure maximum operating frequency, power, and delay.

The second test scheme forces the test vectors that yield the worst case delay. These vectors are A=0's, B=1's, and Cin toggling between "1" and "0" at half the clock frequency. Because the power of DyCML is not a function of the input combinations, this test can be used to measure the power. The output signals were recorded using a 50 GHz Tektronix 11801C digital sampling scope. Figure 4.16 shows the measured waveforms. The signals from left to right are the clock signal,

carry out of the first 4-bit CLA block, carry in to the last CLA block, and the last output sum bit S_{15}. The measured delay was 7% less than the simulated delay, while the measured power was 4.5% less than simulated layout, which falls within process variation limits. The measured delay of the CLA adder is 1.24 nSec. At 400 MHz and 3.3V supply, the chip consumes $19.2mW$. The measurement results include the power and delay of the conversion buffers.

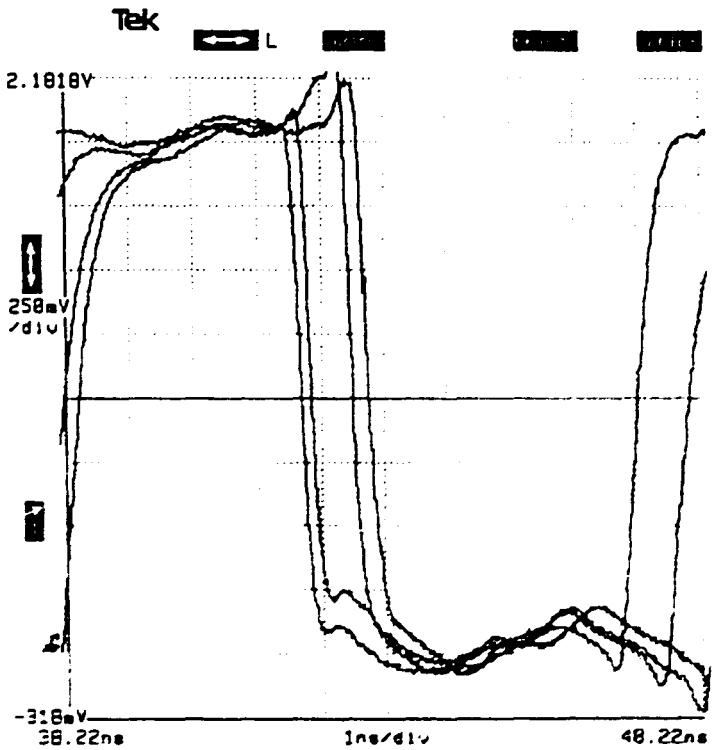


Figure 4.16: Measured Delay

4.6 Summary

In this chapter, a new logic style called DyCML was introduced. DyCML family combines the advantages of both MCML logic circuits and dynamic logic styles. A major advantage of the DyCML is the dynamic current source, which achieves smaller delays compared to the basic MCML circuits. Other advantages inherited from MCML are high-performance, noise immunity, and robustness to supply voltage scaling. DyCML gates reduce power dissipation by reducing the output voltage swing.

Simulation results show that DyCML circuits have better delay, power-delay and energy delay products compared to five of the most famous logic styles. A 16-bit CLA adder is fabricated in 0.6 μm CMOS technology to validate the simulation results. Experimental results show that DyCML circuits achieve high-speed with low power dissipation. The area of DyCML circuits is comparable to the area of the equivalent CMOS circuits.

Chapter 5

New High-Speed Dynamic Logic Styles for CMOS and MTCMOS Technologies

As CMOS technology scales down in deep submicron regime, performance of Domino circuits starts to degrade. Dynamic power dissipation increases also because of contention currents. This chapter introduces a new Domino logic style, referred to as High Speed-Domino (HS-Domino). HS-Domino resolves the speed-Noise Margin (speed-NM) trade-off in Domino circuits and extends the Domino's operation into the deep submicron regime, with no degradation in the gate's noise margin. The second part of the chapter presents a new MTCMOS scheme for Dynamic logic styles. This scheme has been applied to HS-Domino and Domino Differential Cascode Voltage Switch logic (DDCVS). The MTCMOS scheme reduces the subthreshold leakage current during standby, while attaining high performance and sufficient

noise margins. The new scheme does not require extra gating hardware.

5.1 Introduction

In a digital CMOS circuit, switching power dominates the total power dissipation. As explained in chapter 2, reducing the supply voltage is the most efficient approach to reduce switching power dissipation. Lowering the supply voltage is also important in Deep SubMicron (DSM) technologies to avoid several reliability problems. However, reducing the supply voltage alone causes serious degradation in the circuit's performance.

One way to maintain performance is to scale down both the supply voltage and the threshold voltage V_{th}. Again, reducing the threshold voltage, increases the subthreshold leakage current exponentially. At very low V_{th} and with the new System On Chip (SOC) trend, the leakage becomes a large fraction of the total power. Furthermore, dynamic logic circuits such as Domino and DCVS have significantly worse tolerance to device subthreshold leakage compared to static CMOS, because they use precharge logic [62]. Therefore, it is considered risky to utilize the low threshold voltage (LVT) devices to improve the critical path delay in dynamic circuits [85].

In the next section, the operation of Domino logic is presented and the speed-NM problem in Domino logic is explained in detail.

5.2 Domino Logic

Figure 5.1 shows a Clock Delayed Domino (CD-Domino)[1] logic gate. It consists of: a PMOS precharge transistor (Q_1), an NMOS block for logic function evaluation, a static CMOS inverter I_1 to drive the gate output, a PMOS keeper transistor (Q_2) to restore the logic at the Domino node, and two clock drivers (I_2 and I_3).

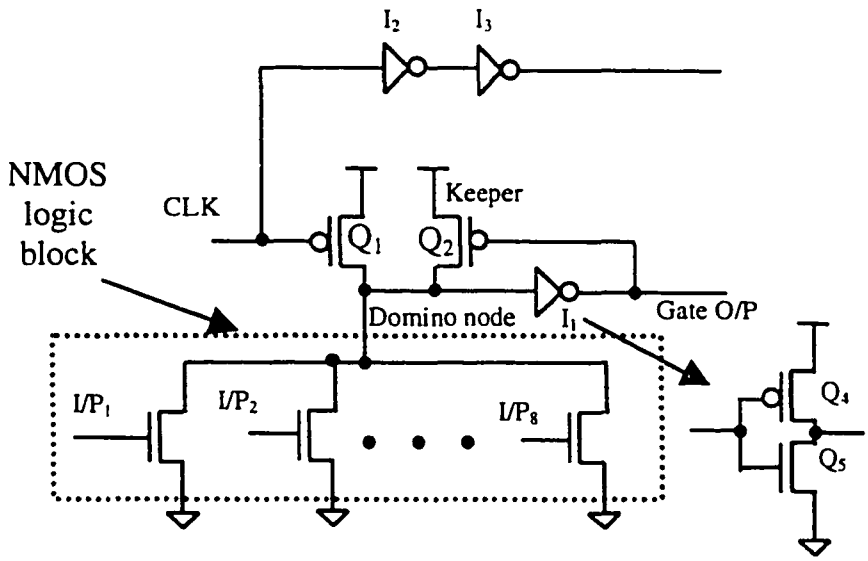


Figure 5.1: An 8-input Clock-Delayed OR Domino gate

CD-Domino is similar to the conventional Domino, except that the NMOS evaluation transistor is removed to speed up evaluation. To reduce the short circuit currents and avoid racing condition, a buffer is added to delay the clock signal from each logic level to the following one. In Figure 5.1, inverters I_1 and I_2 act as buffers to delay the clock signal.

[1]CD-Domino is widely known in the industry as D2 Domino, while conventional Domino is known as D1 Domino

The CD-Domino operates as follows: during the precharge phase (when the clock is LOW), transistor Q_1 turns ON to charge the Domino node to "1", the inverter I_1 output becomes "0", and the keeper transistor Q_2 turns ON to maintain the voltage of the Domino node.

When the clock signal becomes HIGH, the CD-Domino gate operates in evaluation mode. During evaluation, the CD-Domino gate has two possibilities. First, the input data creates a path from the Domino node to G_{ND} to discharge the Domino node to "0". Consequently, the inverter I_1 output becomes "1" and the keeper transistor Q_2 turns OFF. At the beginning of evaluation, the keeper transistor is ON. Therefore, when a path is created from the Domino node to G_{ND}, transistor Q_2 tries to keep this node at V_{dd} while the NMOS block is trying to discharge the same node to G_{ND}. This is called contention, where one device is trying to change a node to "1" while another device is trying to pull it down to "0". Contention slows down evaluation because the NMOS block has to pass current equivalent to that of the keeper in addition to the Domino node discharge current. Contention increases dynamic power dissipation also because of the large current passing from V_{dd} to G_{ND} during evaluation. Therefore, it is preferred to reduce the keeper size to enhance evaluation speed and reduce the contention power.

Second evaluation scenario occurs when the input combination does not create a path from the Domino node to G_{ND}. In this case, the Domino node remains "1" and the gate output remains "0". The keeper transistor stays ON to maintain the Domino node voltage at V_{dd} and to compensate for any leakage currents or charge sharing. Therefore, a larger keeper transistor is useful to increase the stability of

the domino gate. Unfortunately, larger keeper leads to slower evaluation and more power dissipation.

5.2.1 Noise Margin and Delay of CD-Domino Circuits

The NM is defined as the input voltage change that causes a 10% drop of V_{dd} at the Domino node [62]. In this work, the NM is set to 10% of V_{dd}.

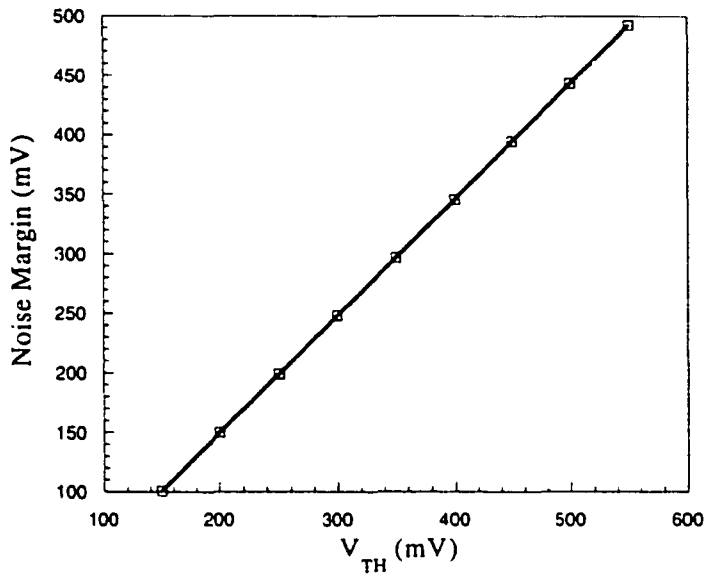


Figure 5.2: Noise margin of Domino logic versus V_{th}

Figure 5.2 shows the NM of an 8-input conventional Domino OR gate for different V_{th} values, while keeping the ratio W_{keeper}/W_n constant and equal to 1/10, where W_{keeper} and W_n are the widths of the PMOS keeper and NMOS pull down devices respectively. The 8-input NOR gate has been used, because Domino logic is usually used for wide fan-in OR gates. OR gate have the worst case leakage current when all the inputs are "0". In this case, all the input transistors turn OFF, and

leakage current flows from the Domino node to G_{ND} through each input transistor. Figure 5.2 shows that the noise margin drops by 1mV for every 1mV decrease in V_{th}

Figure 5.3 illustrates the normalized delay of a 3-stage chain of 8-input conventional Domino NOR gates with a fan-out of 3 versus V_{th} for two cases. In the first case, a constant W_{keeper}/W_n ratio is used (i.e. ignoring the noise margin). In the second case, the W_{keeper}/W_n ratio is increased to keep the noise margin at least 10% of V_{dd}. The second case is referred to as controlled or constant NM.

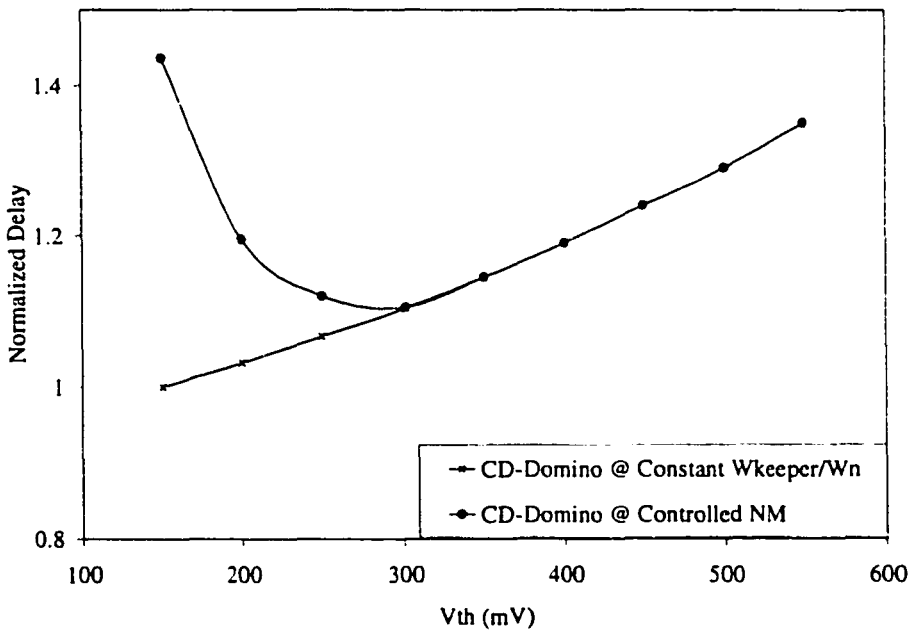


Figure 5.3: Normalized delay of CD-Domino versus V_{th}

The constant W_{keeper}/W_n curve shows that the performance of CD-Domino increases as V_{th} decreases. However, this curve neglects noise margin, which leads to impractical design. The constant (controlled) noise margin curve shows that as

the threshold voltage is reduced, delay increases. This occurs becuase when the threshold voltage drops below certain value (0.3V in this case) the noise margin becomes less than 10% of V_{dd}. Therefore, the keeper has to be sized up to keep the NM $\geq$ 10% of V_{dd}. Larger keeper size leads to more contention current and increases the evaluation time. This conflict between performance and noise margin is called speed-NM trade-off, where the designer has to compromise one of the two parameters to increase the other.

Therefore, Domino circuits are not suitable for DSM technologies because of the high leakage currents and degraded performance. In the next section, a new Domino logic circuit is presented to solve this problem and extend the Domino performance into future fabrication technologies.

5.3 High Speed Domino (HS-Domino)

In this section, a new logic style called HS-Domino is introduced. HS-Domino solves the contention problem by turning the keeper transistor OFF at the start of the evaluation cycle. The architecture of an HS-Domino gate is illustrated in Figure 5.4. The gate output is connected to the keeper through an NMOS (N_1) and a PMOS (P_1). The gates of both transistors are connected to the delayed clock signal.

This HS-Domino gate operates as follows: when the clock is LOW during precharge, transistor N_1 is OFF, P_1 is ON charging the gate of the keeper transistor Q_2 to V_{dd}. Therefore, the keeper transistor turns OFF, and the Domino node is precharged to V_{dd}.

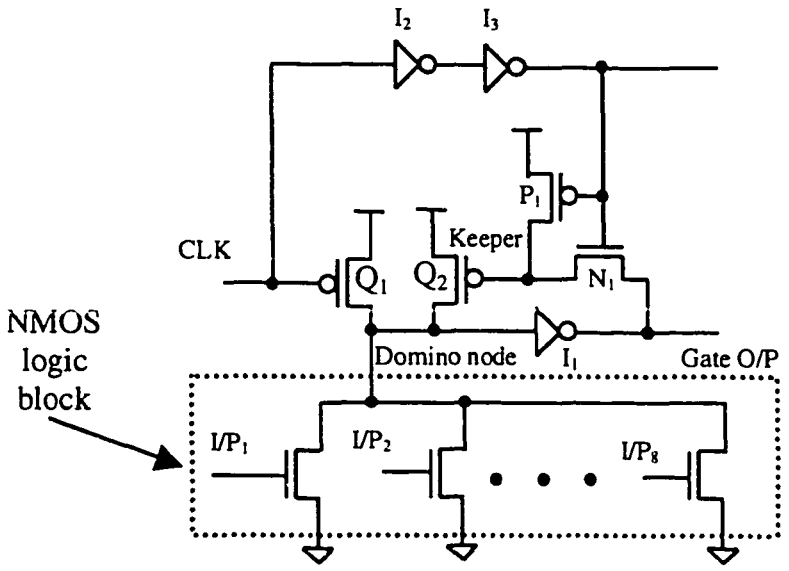


Figure 5.4: An 8-input HS-Domino OR Gate

During the evaluation phase, the operation of the HS-Domino is similar to that of CD-Domino except for the keeper transistor. In HS-Domino, the keeper transistor is OFF at the beginning of evaluation phase. This eliminates the contention between the keeper and the pull-down devices during evaluation. Therefore, the Domino gate evaluates faster and no contention current exists. When the delayed clock signal becomes "1", if the gate has already evaluated and the output of the inverter I_1 is "1", transistor N_1 will be OFF and the keeper transistor remains OFF. Alternatively, if the gate has evaluated to "0", transistor N_1 turns ON to discharge the gate of the keeper transistor. Therefore the keeper turns ON to maintain the voltage of the Domino node at V_{dd} and to compensate for leakage currents.

Usually the delay between the clock signal and the delayed clock signal is larger than the evaluation time of the gate. Therefore, the Domino gate may finish evaluation before the keeper transistor begins to turn ON. Transistors N_1 and P_1 are

minimum size transistors to slow down the switching of the keeper and allow more time for contention-free evaluation.

In HS-Domino, the keeper width may be sized up as V_{th} scales down to maintain a constant (controlled) NM, without worrying about increasing the contention, and speed degradation. Figure 5.5 shows how the keeper is sized up to maintain the noise margin $\geq 10\%$ in HS-Domino.

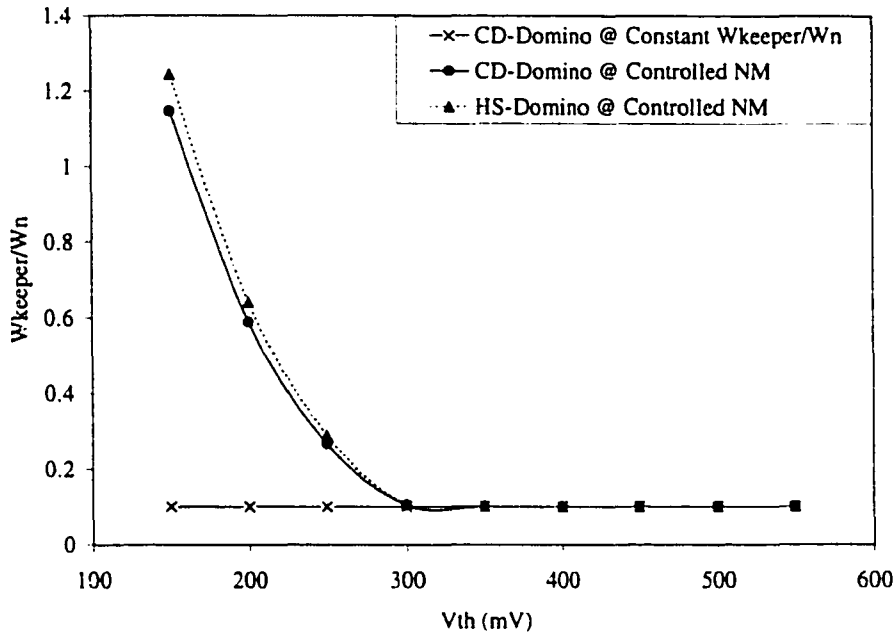


Figure 5.5: W_{keeper}/W_n ratio versus V_{th}

5.3.1 Speed Comparison

In order to compare the performance of HS-Domino with that of CD-Domino, a 3-stage chain of 8-input OR gates was simulated in 0.25 μm CMOS technology.

In the simulation, each gate had a fan-out of 3. The delay versus V_{th} is shown in Figure 5.6.

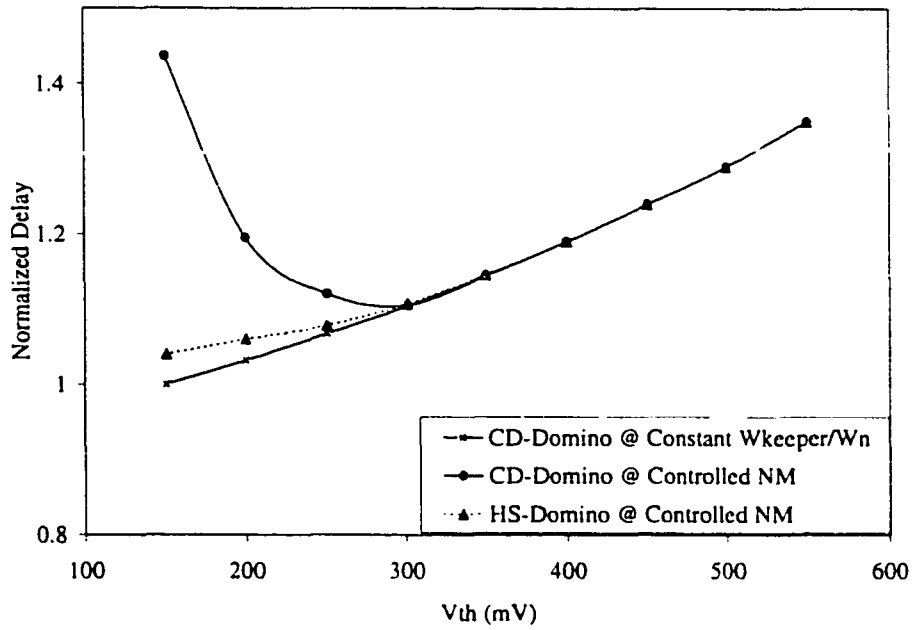


Figure 5.6: Speed of the different dynamic styles

The delay curves of the CD-Domino with constant NM and constant W_{keeper}/W_n ratio are plotted to illustrate the speed advantage of HS-Domino circuits. Figure 5.6 shows how the delay of HS-Domino circuit continues to decrease as V_{th} is scaled down, without tampering the NM. Hence, HS-Domino resolves the speed-NM trade-off. A slight speed difference starts to develop between the modified circuit and conventional Domino with constant W_{keeper}/W_n ratio as V_{th} decreases.This is due to the increases loading at the Domino node as the keeper is sized up to keep the NM intact.

5.3.2 Power Dissipation of HS-Domino

Figure 5.7 compares HS-Domino circuit with the CD-Domino in terms of dynamic power at 500MHz.

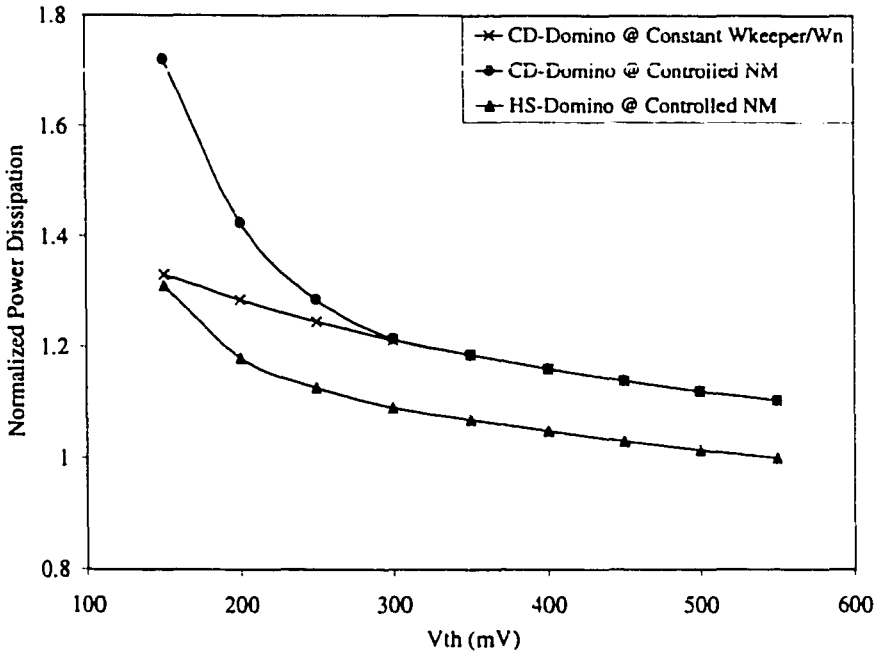


Figure 5.7: Normalized dynamic power versus V_{th}

The power dissipation of CD-Domino circuit is plotted for two cases; constant NM and constant W_{keeper}/W_n ratio. Although the HS-Domino with constant NM introduces slightly higher clock loading, it reduces power dissipation by about 15% compared to either cases of CD-Domino. At low V_{th} values, the power of the HS-Domino approaches that of CD-Domino with fixed keeper size, because the keeper size is increased in HS-Domino to maintain the noise margin which increased the loading at the Domino node. However, the increase in power dissipation in HS-

Domino is much less than that of constant NM CD-Domino. Consequently, HS-Domino consumes less power for a wide range of V_{th} values because the contention is eliminated. The added hardware, transistors N_1 and P_1, are minimum size devices which have a minor effect on the total power dissipation.

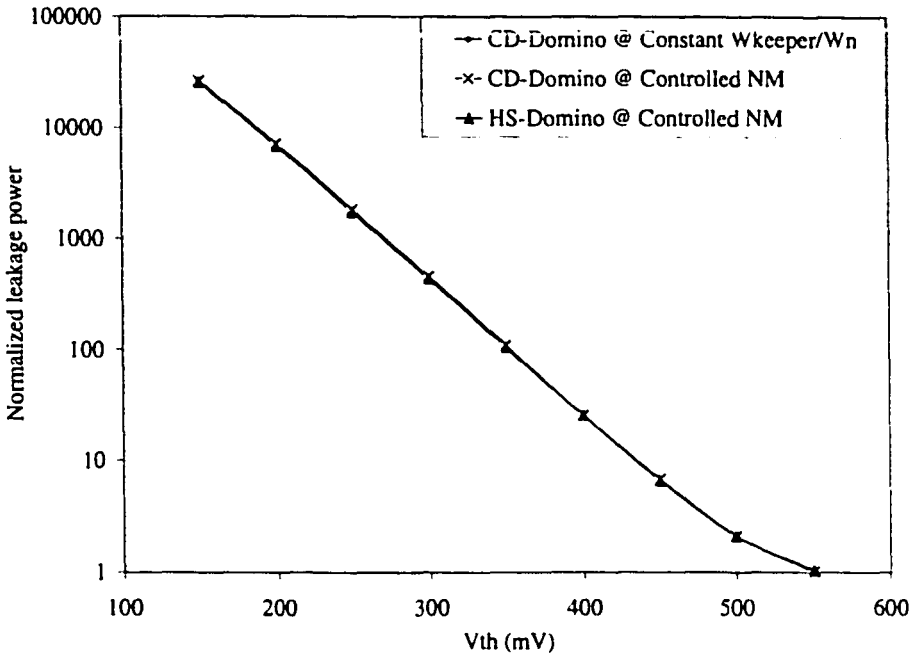


Figure 5.8: Normalized leakage power versus V_{th}

Figure 5.8 illustrates the normalized leakage power of HS-Domino and the two types of CD-Domino. The leakage current is almost the same for the three cases and it increases by an order of magnitude for every 85mV reduction in V_{th}.

Therefore, HS-Domino solves the speed-NM trade-off, but it does not affect leakage power. For low V_{th} devices, the leakage current is increasing rapidly which requires larger keeper size and consequently, dynamic power dissipation increases.

In 1995, MTCMOS technology was proposed as a solution to reduce leakage

current in deep submicron processes. MTCMOS uses low V_{th} (LVT) devices to reduce delay during normal operation and uses high V_{th} (HVT) devices to reduce leakage current during standby [86].

Next section explains the MTCMOS implementation of CD-Domino. The following section introduces MTCMOS implementation for HS-Domino and compares the new implementation with MTCMOS CD-Domino logic.

5.4 MTCMOS Implementations for Domino Logic

Usually, leakage power dissipation is much smaller than dynamic power. During normal operation, leakage power may be neglected. However, leakage power is more important during the inactive or standby modes especially for battery powered systems. Such systems operate in the active mode for a short amount of time while more than 90% they are in standby mode. Therefore, MTCMOS implementation reduces the leakage power during standby to increase battery lifetime.

Domino circuits may be switched to standby mode either during precharge or evaluation. In order to evaluate these two options and determine which transistors to become LVT or HVT, four operating modes have to be considered as follows:

1. Evaluation phase
2. Precharge phase
3. Standby @ Evaluation
4. Standby @ Precharge

Devices in the evaluation path are preferred to be LVT devices to avoid speed degradation. Devices that are used only during precharge may be either LVT or HVT, because they do not affect performance. Table 5.1 summarizes the preference of each device in a CD-Domino gate in each operating mode to increase speed and reduce leakage power. The schematic of the CD-Domino gate is shown in Figure 5.1.

Table 5.1: Type of Transistors in the Domino Logic

Mode	Q1	Q2	Q4	Q5
Precharge	X	X	X	X
Standby @ Precharge	X	X	H	X
Standby @ Evaluation	H	H	X	H
Evaluation	X	X	L	X

where "H" symbolizes a HVT device, "L" is a LVT device, and "X" is a Don't care state (ie. may be either HVT or LVT). Dividing Table 5.1 into two tables , either standby @ precharge (Table 5.2) or standby @ evaluation (Table 5.3)

Table 5.2: Standby @ Precharge : Rejected

Mode	Q1	Q2	Q4	Q5
Precharge	X	X	X	X
Standby @ Precharge	X	X	H	X
Evaluation	X	X	L	X

It is obvious from Table 5.2 that transistors Q2 and Q4 should be HVT and LVT during the standby mode and evaluation mode respectively. This discrepancy indicates that the standby mode cannot take place as precharge.

On the other hand, Table 5.3 shows no contradiction in the kind of device during any mode of operation. Thus, the standby mode is chosen to be during evaluation.

Table 5.3: Standby @ Evaluation : Correct

Mode	Q1	Q2	Q4	Q5
Precharge	X	X	X	X
Standby @ Evaluation	H	H	X	H
Evaluation	X	X	L	X

5.4.1 MTCMOS CD-Domino Logic

In a recent report, an MTCMOS Domino implementation has been devised [87]. This implementation focuses on reducing the leakage while achieving high performance. However, no attention has been paid to noise margins, which is impractical in Dynamic logic. Figure 5.9 illustrates the schematic of the MTCMOS Domino gate, where LVT devices are highlighted. In this implementation, sleep mode is used during evaluation which complies with table 5.3.

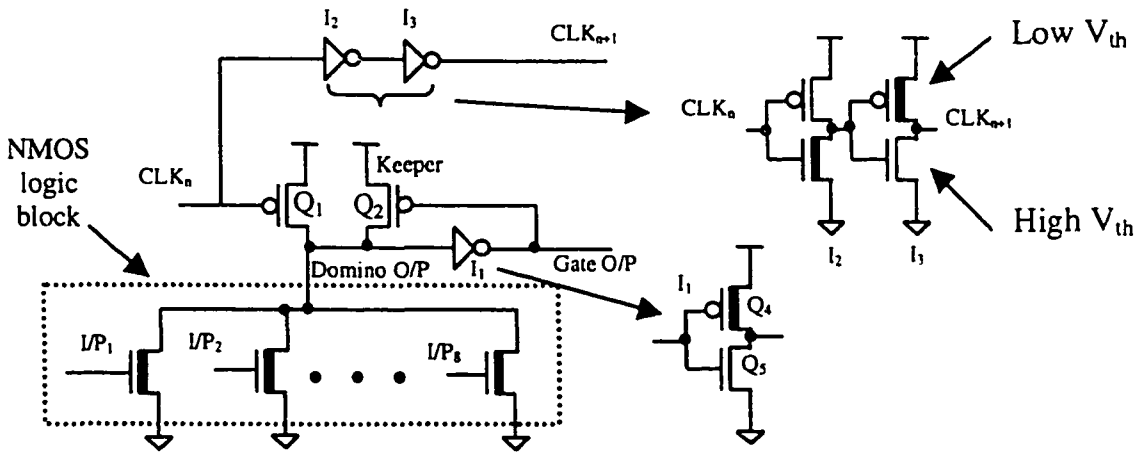


Figure 5.9: An 8-input MTCMOS CD-Domino OR gate

During evaluation, the gate may have "0" to "1" transition at the output node. All the devices involved in this transition should be LVT devices to speed up eval-

uation. These transistors are the logic block transistors, transistor Q_4, the NMOS transistor of I_2, and the PMOS transistor of I_3. The keeper transistor is important only after the evaluation.

During standby, the gate operates in evaluation mode and all the inputs are "1"s. Therefore, transistors Q_1, Q_2, Q_5, the PMOS of I_2, and the NMOS of I_3 are OFF, passing leakage current from V_{dd} to G_{ND}. All these transistors are HVT devices. Therefore, this scheme reduces the evaluation delay by using the LVT devices and the leakage currents by using HVT devices.

One problem with such a mechanism is that forcing "1"s at the input, means that these inputs should be gated. This is required only for the external inputs to the Domino block, because during standby, all the Domino gates inside the block have output "1". The gating circuitry usually consists of OR gates, which increases the dynamic & leakage power, delay, and area of the circuit. This implementation does not take noise margin into consideration. The worst case NM of an 8-input conventional Domino OR gate for different V_{th}, while keeping the ratio W_{keeper}/W_n constant, is similar to that shown in Figure 5.2.

5.5 MTCMOS High Speed Domino Logic (MHS-Domino)

To resolve the speed-NM trade-off, and thus remove the contention, as well as achieving ultra low leakage values, the MTCMOS HS-Domino (MHS-Domino) circuit shown in Figure 5.10 is devised.

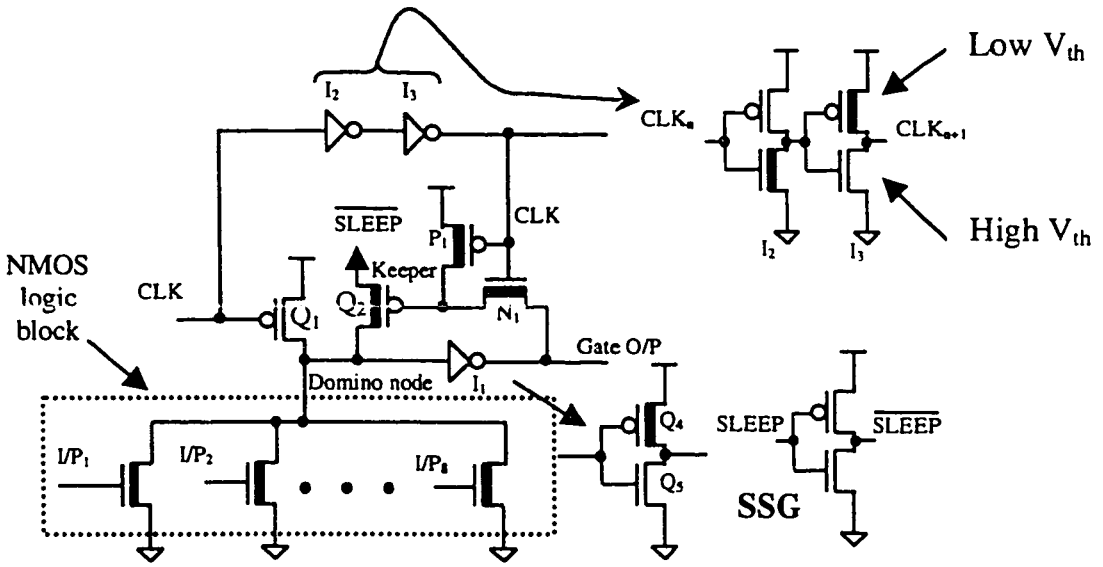


Figure 5.10: An 8-input MHS-Domino OR Gate

The MHS-Domino gate is similar to the HS-Domino gate, except that the keeper source is connected to $\overline{SLEEP}$ signal instead of V_{dd}. Transistors of the NMOS logic block, Q_2, Q_4, P_1, N_1, the NMOS transistor of I_2, and the PMOS transistor of I_3 are LVT transistors to speed up evaluation. The MHS-Domino circuit employs a Sleep Signal Generator (SSG) , which is realized by a simple inverter as shown in the schematic. The PMOS device of the SSG must be kept HVT in order to reduce the leakage during standby. The $SLEEP$ signal is "0" for normal operation and "1" for standby.

During normal operation, the operation of MHS-Domino gate is the same as HS-Domino. In this mode, $\overline{SLEEP}$ signal is "1". Therefore, the keeper source is connected to V_{dd} through the SSG block. In this scheme, all the transistors involved during evaluation are LVT devices to reduce delay.

In standby mode, the $SLEEP$ signal is "1", $\overline{SLEEP}$ becomes "0", and clock

is high. The Domino node has two possible outputs, "1" or "0". When the domino node is "0", the gate output is "1". Therefore the input to the following gate is "1". When the Domino node is "1", the output is "0" which turns the keeper transistor ON discharging the Domino node to $\overline{SLEEP}$ signal and causing the gate output to turn into "1". This transition is fast because the keeper transistor is LVT. Therefore, at the beginning of standby operation, all MHS-Domino gates change their output to become "1" without using input gating which reduces hardware and power. After the initial transitions, transistors Q_1, Q_4, the NMOS of I_2, the PMOS I_3, and the PMOS of the SSG turn OFF. All these transistors have HVT to reduce the leakage current.

5.5.1 Speed Comparison

To illustrate the speed advantage of MHS-Domino circuit, the normalized delay of a 3-stage chain of an 8-input Domino OR gates with a fan-out of 3 for the new MHS-Domino circuit versus V_{th} is shown as the 3rd curve in Figure 5.11. The delay curves of the MTCMOS Domino with constant NM and constant W_{keeper}/W_n ratio are plotted on the same graph to illustrate the speed advantage of the MHS-Domino circuit. Figure 5.11 shows the delay of the MTMCOS modified circuit at constant NM continues to decrease as V_{th} is scaled down because there is no contention current.

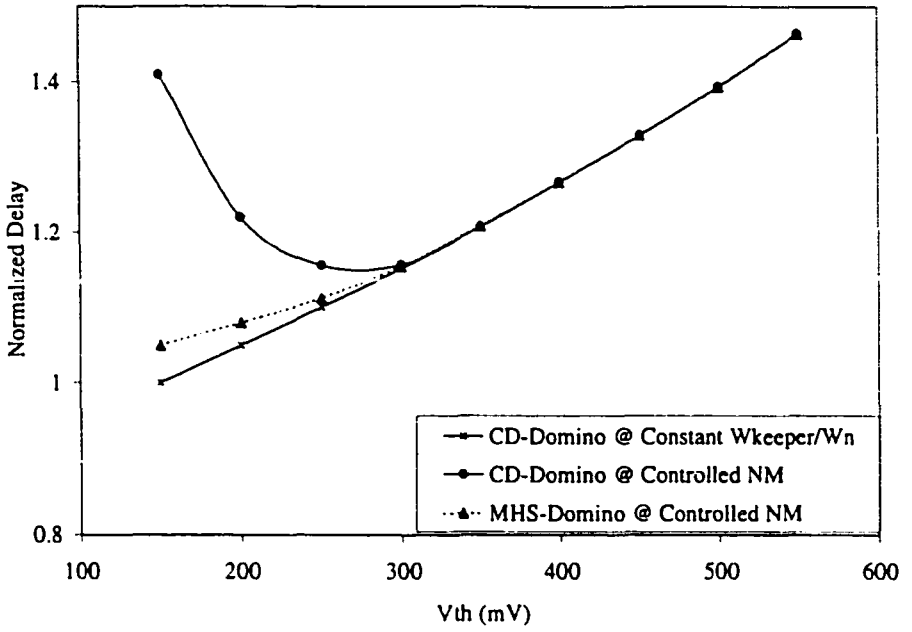


Figure 5.11: Normalized delay of DVT Domino versus V_{th}

5.5.2 Dynamic Power Comparison

Figure 5.12 compares the MHS-Domino circuit with the MTCMOS CD-Domino circuit in terms of dynamic power.

Although the MHS-Domino circuit introduces slightly higher loading, and an SSG, it actually has significantly lower power dissipation than the conventional version. This is attributed to the following:

1. Elimination of the contention in the modified domino gate, which means that there are no short-circuit currents during switching.

2. N_1 and P_1 are minimum sized devices, contributing to a very small loading effect.

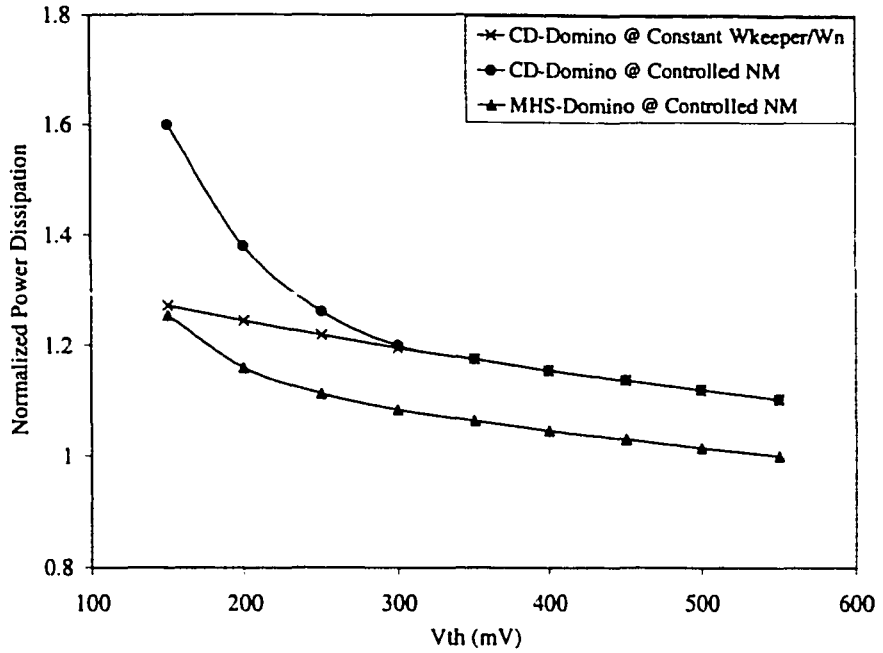


Figure 5.12: Normalized dynamic power in MTCMOS Domino versus V_{th}

3. The sleep signal generator hardly consumes any dynamic power because its output is always "1" during the active mode, and "0" during standby. Thus, no switching occurs except during transition from one mode to the other. These transition are not frequent and its power dissipation is negligible.

5.5.3 Leakage Comparison

A comparison between the normalized leakage power of the MTCMOS CD-Domino and MHS-Domino is shown in Figure 5.13. The leakage curves are plotted versus V_{th} for the MTCMOS Domino gate for constant NM and constant W_{keeper}/W_n ratio (NM is ignored).

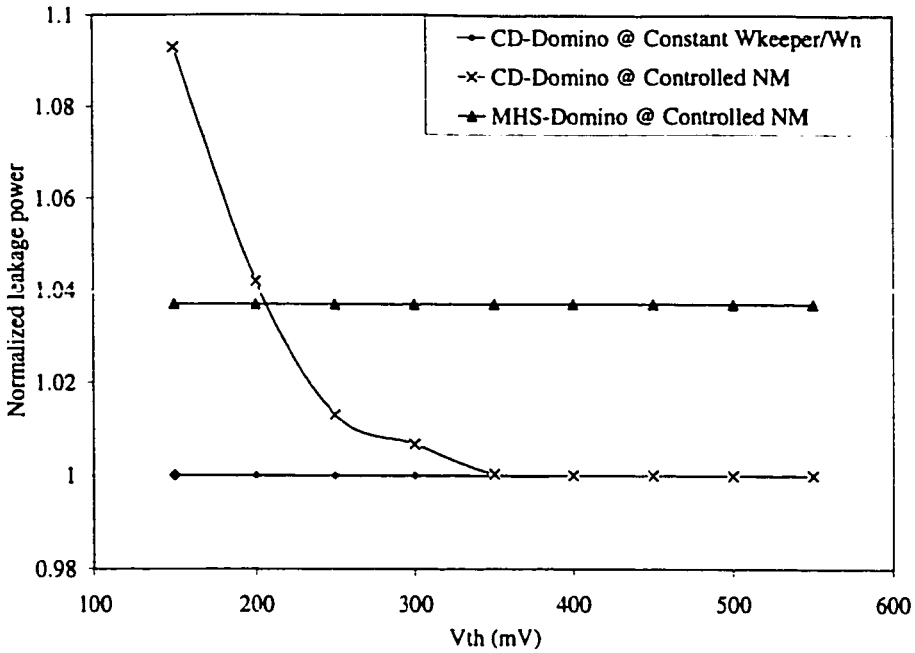


Figure 5.13: Normalized leakage power in MTCMOS Domino versus V_{th}

MHS-Domino consumes only 4% more leakage power than the MTCMOS CD-Domino (ignoring NM), while it also consumes slightly more leakage than the conventional MTCMOS case at constant NM until a V_{th} of $\approx$ 220mV. For V_{th} below 220mV, the MHS-Domino at constant NM has a leakage advantage over the conventional version at constant NM. It is necessary to offset the power penalty paid in turning devices ON and OFF, especially due to switching the devices from standby to active states, and vice versa. This power penalty becomes less significant if the system spends most of its time in the idle state (95%).

Therefore, MHS-Domino eliminates the contention, enhances the performance, reduces power dissipation and reduces leakage current during standby.

The scheme used to convert HS-Domino to an MTCMOS logic style is generic

and may be applied to any other dynamic logic style. In the following section, this scheme is extended to Domino Dual Cascode Voltage Switch logic (DDCVS). A brief introduction to the conventional DDCVS is given first.

5.6 MTCMOS Implementation for DDCVS Logic (MDDCVS)

The architecture of DDCVS circuits and theory of operation has been explained in in detail in section 3.3.4. The rest of this chapter concentrates on leakage power dissipation in DDCVS logic, and how to use the new MTCMOS scheme to reduce this leakage power.

Figure 5.14 illustrates a two input XOR gate using DDCVS logic. The cross coupled transistors Q_1 and Q_2 are OFF at the beginning of evaluation phase. Therefore, no contention current exists during evaluation and the keeper devices may be sized up to achieve a sufficient NM, without worrying about degrading the gate's speed.

Although the speed-NM problem is in DDCVS circuits, leakage becomes a challenge as CMOS technologies scale down into the submicron regime. Therefore, a new MTCMOS DDCVS (MDDCVS) is devised. The new circuit exhibits extremely low leakage during the sleep mode (standby mode), while maintaining the speed of LVT devices and dynamic power dissipation during the active mode.

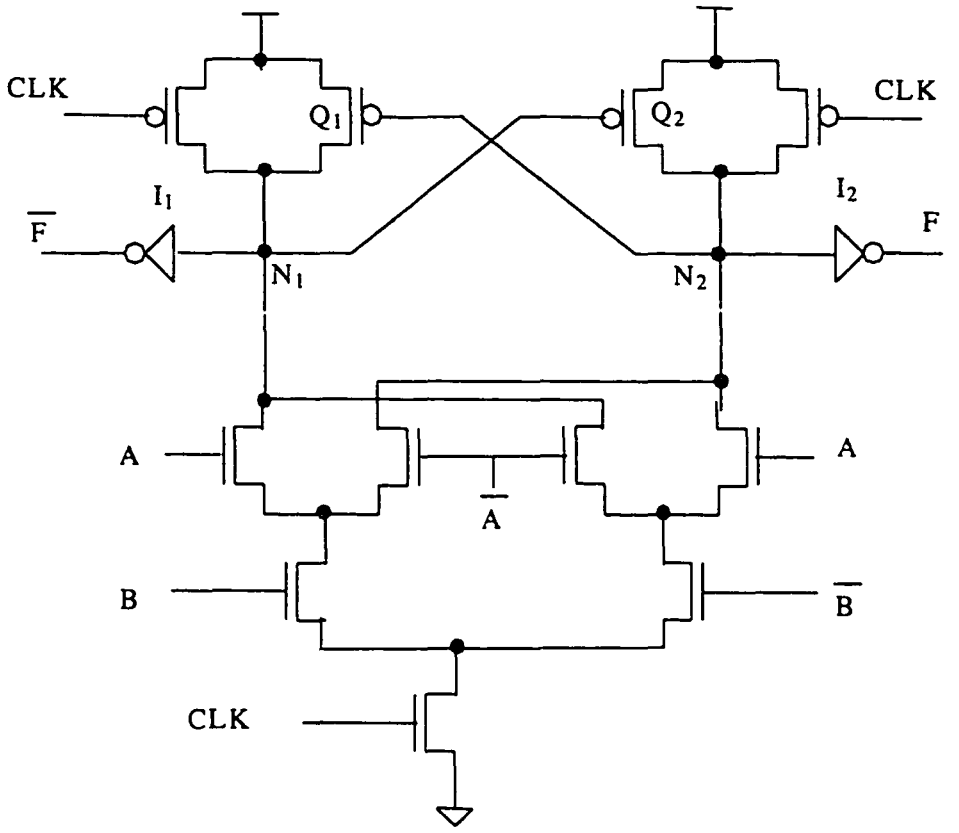


Figure 5.14: An XOR DDSVSL Gate

5.6.1 MDDCVS Architecture and Operation

MDDCVS has a similar implementation to the original DDCVS, except for a sleep signal generator (SSG), which connects the sources of the PMOS keeper devices to V_{dd} during the active mode, or to G_{ND} during the sleep mode. Figure 5.15 shows a typical clock-delayed MDDCVS stage.

During normal operation $\overline{SLEEP}$ signal is high and the MDDCVS circuit operates exactly like a regular DDCVS circuit. Because precharge time is not critical, transistors involved in the precharge process may be LVT or HVT devices. These

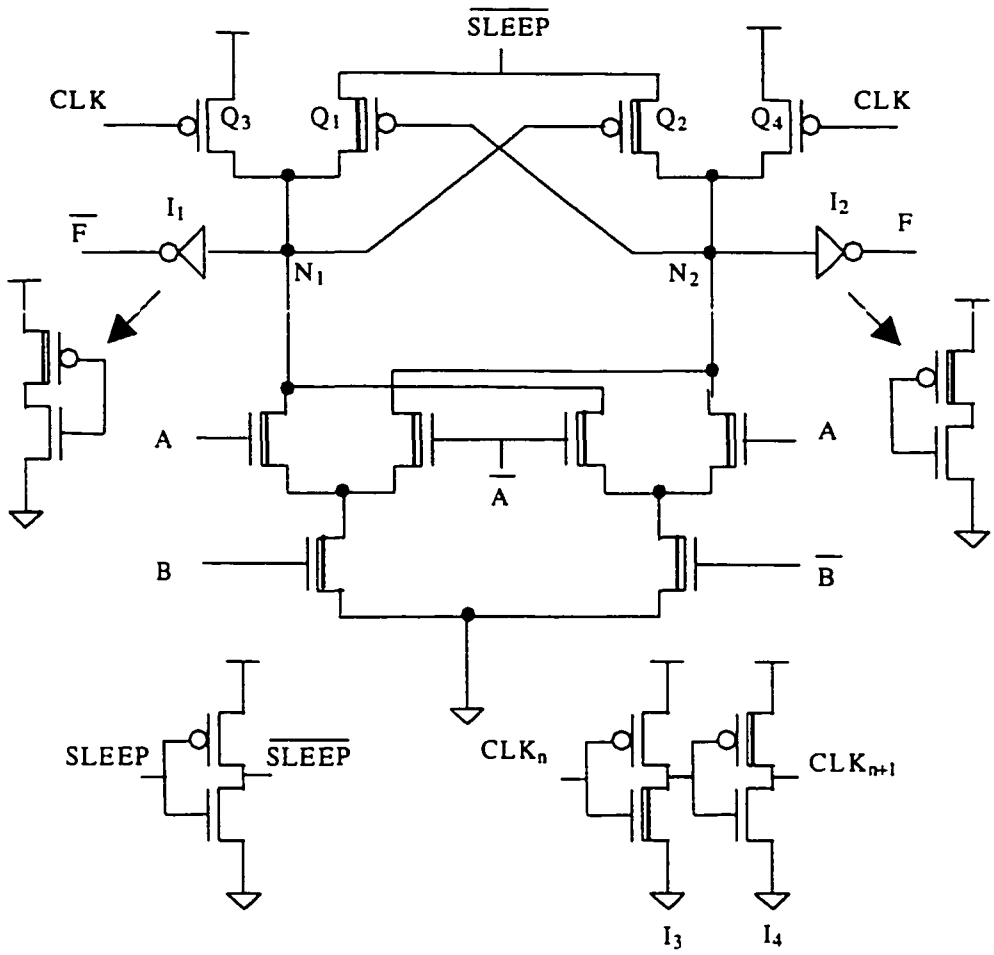


Figure 5.15: A two input MDDCVS XOR logic gate

transistors are Q_3, Q_4 and the NMOS transistors of inverters I_1, I_2. On the other side, transistors involved in the evaluation should be LVT to speed up the logic gate. The transistors responsible for evaluation are the NMOS block transistors and the PMOS transistors of I_1 and I_2.

During the standby mode, the clock is HIGH in order to turn OFF the HVT Q_3 and Q_4 devices. The SLEEP signal becomes HIGH, which supplies G_{ND} to the

sources of transistors Q_1 and Q_2.

Whether standby occurs right after precharge or evaluation, this is not an issue. In both cases, whatever the input values to the gate are, a state will always be reached where one branch is ON and the other is OFF. Assuming that the value of the inputs to the N-block cause N_1 to be LOW and thus N_2 to be HIGH. Q_2 is turned ON, allowing node N_2 to start discharging through Q_2, until it eventually reaches "0". If the inputs were to cause N_2 to go LOW and N_1 HIGH, Q_1 will turn ON, allowing node N_1 to start discharging through Q_1, until it eventually reaches "0". Therefore, independent on the inputs to the gate, both N_1 and N_2 will be "0" during the standby mode. The time taken to reach "0" for both N_1 and N_2 was calculated to be $\approx$ 200psec. Therefore, nodes F and $\overline{F}$ will both go HIGH, which cause the input NMOS devices in the successive stage to turn ON completely, and pull down the 2 internal nodes to G_{ND} very quickly. In the second stage, the discharging time takes $\approx$ 60psec. Similarly with any other cascaded gates in the pipeline.

Therefore, the first stage consumes the longest time to reach the standby state, while the other consecutive gates in the pipeline take very short times. This is not critical, especially in mobile systems, where over 95% of the time is spent as idle time. The 200psec is by far negligible compared to the long minutes a mobile system could be idle for.

An important advantage of the MDDCVS is that it does not require specific input values to the gate at the standby mode. This eliminates any increase in area, power, or delay as a result of gating the inputs to the DDCVS gate.

The PMOS of the SSG is OFF during standby, and is thus made HVT to cut-off any leakage. Similarly, during standby, nodes N_1 and N_2 will be LOW, which would turn OFF the NMOS device in inverters I_1 and I_2. Therefore those NMOS devices are made HVT. Furthermore, the I_3 PMOS and I_4 NMOS are HVT because they are also OFF during standby.

5.7 MDDCVS Simulation and Comparison

To verify the functionality and benefit of the DVT-DDCVSL, simulations were performed on a pipeline of 3 DVT-DDCVSL XOR gates operating at 500MHz and using $0.25\mu m$ CMOS technology at 2.5V supply voltage. XOR gates were used as a test vehicle because DCVS logic is normally used to implement XOR and MUX circuits due to its differential nature. Each gate in the pipeline had a fan-out of 3.

Figure 5.16 shows the normalized delay of the 3-stage chain of conventional DDCVS gates with a fan-out of 3 versus V_{th} for 3 cases: single V_{th} DDCVS at constant keeper size (ignoring NM), single V_{th} DDCVS at constant NM, and MDDCVS at constant NM.

Similar to the Domino logic case, the NM was defined as the input voltage above ground that causes a 10% drop from V_{dd} at the DDCVS internal nodes (N_1 or N_2). The NM was set to 10% of V_{dd}. Figure 5.16 shows that MDCVS with constant NM has similar performance to the Single V_{th} with constant NM case. The constant keeper curve shows the maximum possible gain in speed with lowering the V_{th}, which wrongfully ignores the NM. Again, the slight difference in speed starts to develop between the MTCMOS modified circuit and single V_{th} DDCVS with

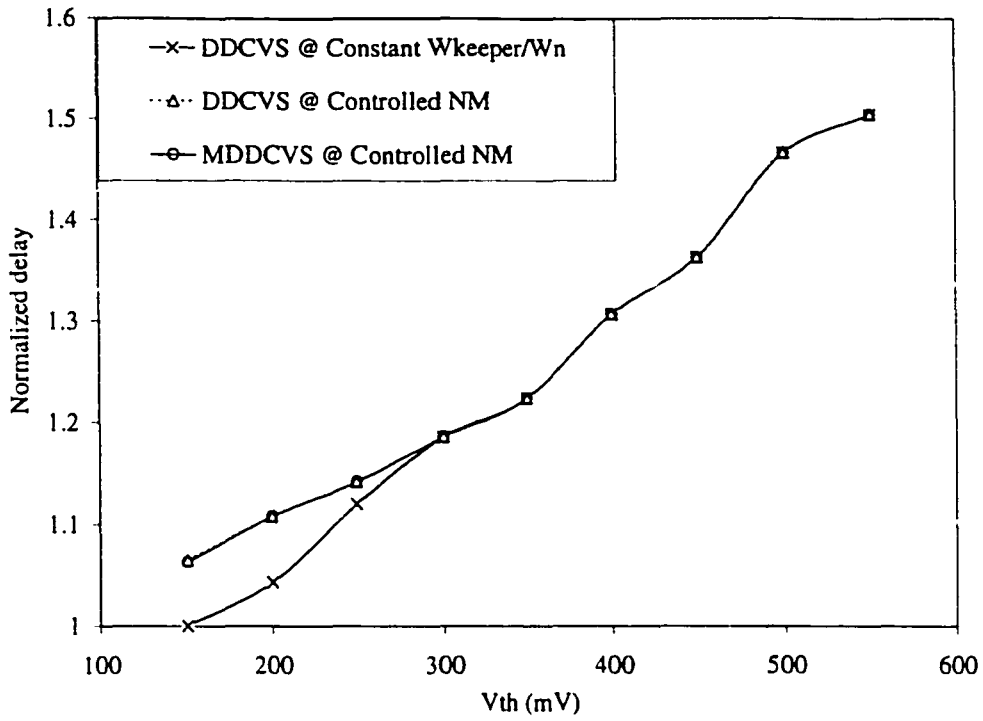


Figure 5.16: Normalized delay for DDCVS logic styles versus V_{th}

constant W_{keeper}/W_n ratio as V_{th} decreases. This is due to the increased loading at the DDCVS internal nodes (N_1 and N_2) as the keeper transistors are sized up to keep the NM constant.

5.7.1 Dynamic Power Comparison

Figure 5.17 shows a comparison between the 3 cases previously mentioned in terms of dynamic power at 500MHz.

The Figure shows that the MDCVS consumes approximately the same dynamic power as the single V_{th} DDCVS at constant NM.

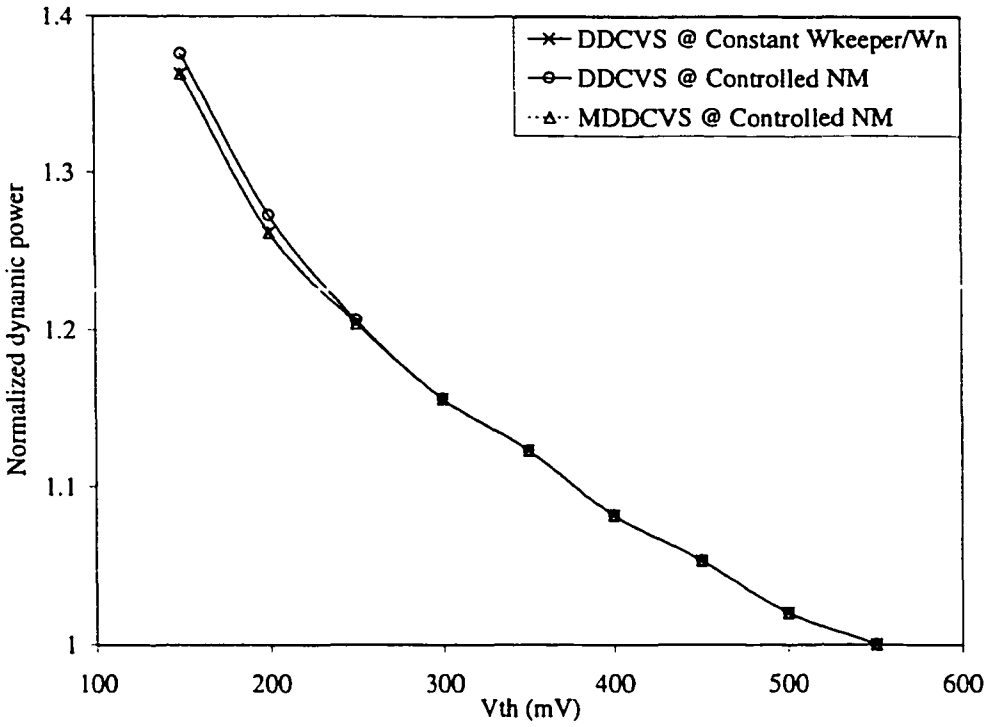


Figure 5.17: Normalized dynamic power in DDCVS logic styles

5.7.2 Leakage Power

The leakage current for the three cases is shown in Figure 5.18. The figure shows that MDDCVS has a a much smaller leakage consumption over single V_{th} designs. This advantage is important to reduce leakage currents of LVT devices and enhance the speed by using LVT devices.

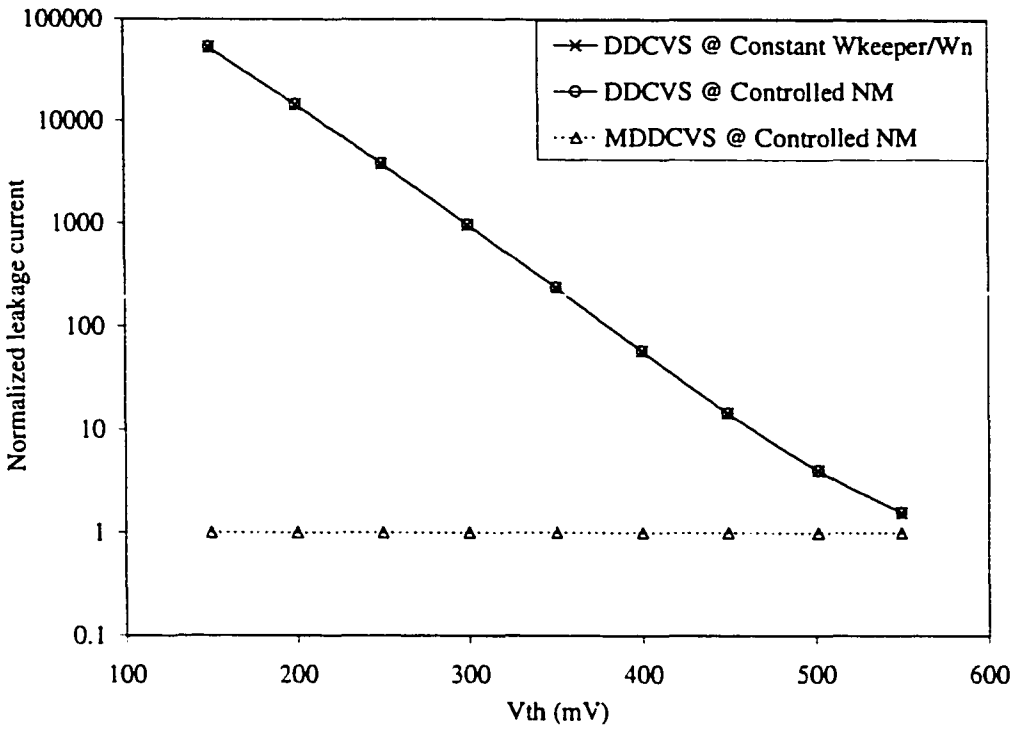


Figure 5.18: Normalized leakage power in MDDCVS versus V_{th}

5.8 Summary

A modified Domino circuit, called HS-Domino is developed. HS-Domino resolves the trade-off between performance and noise margins in DSM CMOS technologies. This logic style can now benefit from the scaling down of the technology and supply voltages since it could now tolerate the lower threshold voltages. The speed of the new Domino logic continues to improve as the threshold voltages are scaled down, while controlling the noise margin. The new circuit also dissipates less dynamic and leakage power.

An MTCMOS implementation of the new Domino logic style is also devised.

This dual threshold implementation achieves low leakage values during standby, while maintaining high speed and low dynamic power during the active mode. A dual threshold implementation of the DCVS logic gate is also presented. It achieves substantially low leakage values during standby, while attaining high performance and low dynamic power during the active mode.

Chapter 6

Low-Power High-Radix Floating Point Division Algorithm Using Quotient Approximation and Error Correction

In real time digital signal processing, high performance modules for division are essential for many powerful algorithms. Unfortunately, division process is much slower than all other arithmetic operations because division has to be executed sequentially. Division operation also consumes high power because of the repetitive multiply and subtract operations inherent in the division process.

In this chapter, a modified algorithm for high radix floating point division is presented. The algorithm uses a look-up table to estimate an approximate value for the quotient digit. The quotient digits has redundant bits to substitute for the

error in estimating the previous quotient digit.

The first section of this chapter is an introduction and classification of various division algorithms. Section 2is a survey of some fast division techniques. In the third section, the new division algorithm is introduced. In the fourth section, the details of the VLSI implementation are discussed and the simulation results for radix-8 and radix-16 dividers are presented. Section 5 is a summary.

6.1 Introduction

The division operation is defined as follows:

$$Q = \frac{X}{D} \tag{6.1}$$

where Q is the quotient, X is the dividend and D is the divisor.

Unlike multiplication and addition, which may be executed in parallel, division has to be executed sequentially because each quotient digit is dependent on the previous remainder. Thus, division operations are slow creating a bottle neck for high performance systems. Researchers have developed various division algorithms to speed up the division process. Some of these algorithms are not suitable for VLSI implementation because of their hardware requirement. Some other algorithms are complex to design. The Pentium Floating point DIVision bug (FDIV) is a clear example of what the complex division algorithms may lead to [88].

6.1.1 Types of Division Algorithms

Division algorithms are classified into two main categories: Multiplicative Division (MD) and Digit Recurrence Division (DRD) [89].

Multiplicative division multiplies both the dividend and divisor by a set of factors until the divisor ≈ 1. The final value of the dividend becomes the quotient of the division process. Each step in in the multiplicative division involves two n bit multiplication operations, where n is the total number of quotient bits. The multiplicative division execution time is proportional to $(log_2 n)$. Multiplicative algorithms are not suitable for VLSI implementation because of their large area and power dissipation.

The digit recurrence algorithm consists of n iterations of a recurrence, in which each iteration (step) produces one digit of the quotient, most-significant digit first. Usually, DRD algorithms require less area and power compared to MD algorithms. Therefore, DRD algorithms are more appropriate for VLSI implementation. The execution time of digit recurrence algorithms is proportional to n, which makes them slower than multiplicative division algorithms.

6.1.2 SRT Division Algorithm

In 1957, Sweeney, Robertson and Tocher developed division algorithm that generates a signed digit quotient digit in each iteration [90], [91], [92]. The algorithm is called SRT after the names of its inventors. SRT algorithm utilizes arithmetic redundancy to reduce the required precision of comparisons between the divisor and residual. Most VLSI implementations of the division operation are based on

the SRT algorithm because of its reduced complexity.

The radix-β RST recurrence algorithm for computing successive residuals is

$$R_i = \beta R_{i-1} - q_i D \qquad i = 1, 2, 3...., n \tag{6.2}$$

where R_i is the residual after i^{th} step, D is the divisor and $q_i \in \{-\rho..\rho\}$ is the ith quotient digit. Here, upper case variables refer to complete words, while lower case variables refer to individual digit(s) of a word. The initial residual R_0 is set to the dividend X and, to ensure convergence, the residual at the ith step must satisfy $|R_i| < \rho D.\beta^{-1})$ where ρ is the maximum value of the quotient digit.

The quotient is accumulated by appending successive quotient digits to the partial quotient Q_i, i.e., $Q_i = Q_{i-1} + q_i\beta^{-i}$. The representation of the quotient is redundant and usually employs the radix β signed digit number representation [84]. One consequence of the redundant representation is that the quotient has alternative representations e.g., in the Signed Binary Number Representation (SBNR) with the digit set $\{-1, 0, 1\}$, the number $101\bar{1}0$ is equivalent to 10010 where $\bar{1}$ denotes "-1". Therefore, at each step, there may exist a degree of choice in selecting a valid quotient digit from the given digit set.

Alternatively, quotient digits can be determined by examining a low precision estimate of the residual. For radix-2, this is typically the three most significant digits (MSDs). Any error introduced into the accumulating quotient can then be corrected on subsequent iterations. This is in contrast to conventional division where the full precision residual must be examined in determining a quotient digit. The previous algorithm requires n clock cycle for execution. The SRT algorithm

(like all recurrence division algorithms) may be implemented in an array-like manner, but it is still sequential because of the dependence of the next quotient digit on the current remainder [93].

6.2 FAST Division Algorithms

As mentioned earlier, DRD algorithms are slow because of their sequential nature and because their execution time is proportional to n. SRT algorithm reduces the complexity of the division hardware with almost no effect on speed. Therefore, scientists developed various techniques to speed up division algorithms. High radix quotient bit selection is the simplest and most effective method to speed up division process. For example, a radix-β ($\beta = 2^m$ where m is the number of bits per quotient digit) division algorithm would require n/m iterations compared to n iterations for the equivalent radix-2 algorithm. However, as the radix increases, the added complexity of the quotient digit selection function increases the iteration delay and eliminates the advantage [84]. P-D charts simplify the quotient digit selection for simple radices (2, 4). However, creating the P-D chart itself is complex even for low radices. A mistake in the P-D look up table caused the famous Pentium processor bug in 1995 (radix-4).

Three main concepts have been developed to reduce the complexity of the quotient digit selection [94].

- *Pre-scaling*: both divisor and dividend are pre-scaled to preserve the value of the quotient, while the divisor is pre-scaled to a range close to unity [95].

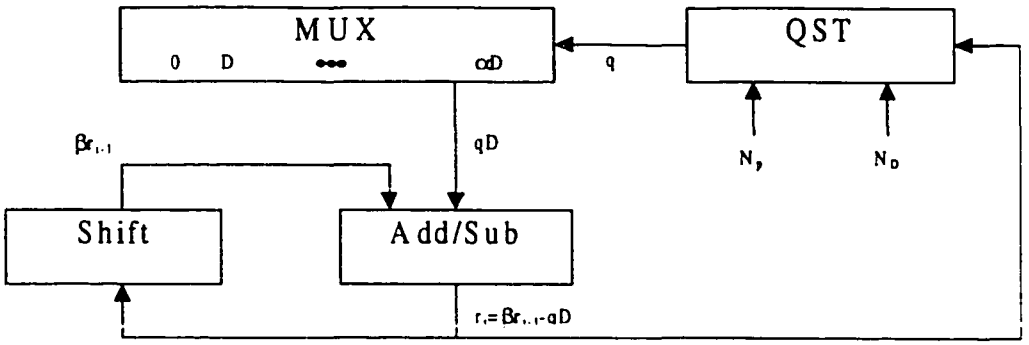


Figure 6.1: SRT algorithm with QST approach

Thus, the quotient digit is just the most significant part of the partial remainder. The new remainder is kept in the same interval as the previous one with a suitable update operation. Although pre-scaling involves two extra multiplication operations, the comparison hardware becomes smaller and faster. If the final remainder is required, an extra multiplication step should be used to restore the weight of the remainder.

- *Prediction*: the traditional division algorithms calculate the partial quotient digit in one clock cycle and the partial remainder in another clock cycle. Ercegovac *et. al.* [96] overlapped the selection of the new quotient digit with the update of the remainder to reduce the number of clock cycles required.

- *Quotient digit selection tables*: the complexity of quotient digit selection can be reduced significantly by using a look-up table, referred to as quotient-digit selection table (QST) as shown in figure 6.1. Usually, a multiplier is used instead of the MUX if D is not constant. The look up table is normally large (1 Mbits for radix-16 divider) which is impractical for VLSI implementation.

In [94], to reduce the ROM size, two smaller tables are used to determine the quotient digit at each step. This may be achieved by speculating the quotient bits, [95]. The modified algorithm with reduced quotient selection table is shown in Figure 6.2. In the algorithm, the correct quotient digit is either q or $(q+1)$. Both possible new partial remainders; $R_i^0 = \beta R_{i-1} - qD$ and $R_i^1 = \beta R_{i-1} - (q+1)D$ are updated in parallel with the quotient-digit correction process which determines the correct q.

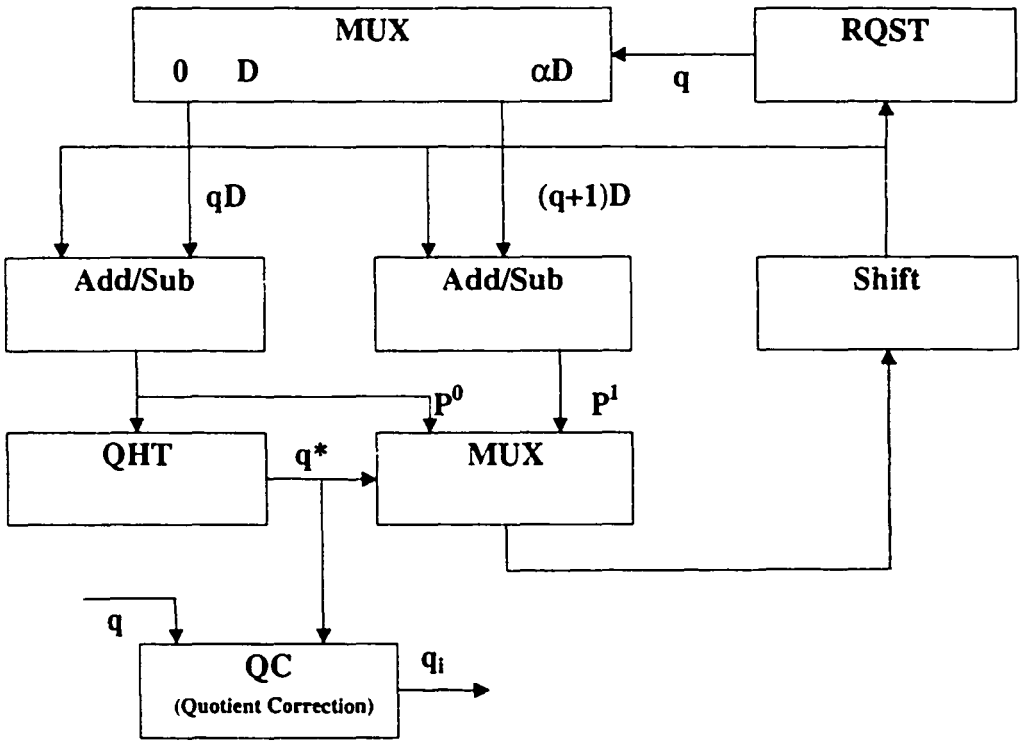


Figure 6.2: Modified SRT algorithm with QST approach

A detailed survey of division algorithms and their power dissipation can be found in [97].

6.3 Modified Division Algorithm

The purpose of this chapter is to develop a low-power high-radix floating point division algorithm. The algorithm is simple to design and can be used for any radix division. The algorithm uses a reduced size look up table to *underestimate* the new quotient digit with certain accuracy. To account for the error in speculating the previous quotient digit, the new quotient digit is allowed to overlap certain number of bits from the previous quotient digit. The look up table size is smaller in this algorithm because of the added redundancy in the quotient digit.

6.3.1 Algorithm Description

The following definitions will be used to describe the algorithm:

- The radix is 2^j.
- The quotient $Q = \frac{X}{D}$ consists of n digits.
- $$D = D^H.2^b + D^L \tag{6.3}$$

  where D^H is the l most significant bits of the partial remainder and D^L is the b least significant bits of D.
- $$R_i = R_i^H.2^a + R^L \tag{6.4}$$

where R_i^H is the k most significant bits of the partial remainder and R_i^L is the a least significant bits of R_i.

- The partial quotient of the ith iteration q_i is u bits wide to add redundancy which will be used later to substitute for errors in estimating the partial quotient, where $u > j$.

- The total quotient after the ith step is

$$Q_i = \sum_{w=0}^{i} q_w . \beta^{-w} \tag{6.5}$$

The values of l and k are function of the division radix. Some examples for these values will be given later in the next section.

The algorithm is described as follows:

1. Initialize the quotient Q to 0, the remainder R_0 to the dividend X and the step counter $i = 1$.

2. Using R_{i-1}^H and the l most significant bits of D^H, calculate the new q_i using the equation

$$q_i = \lfloor \frac{R_{i-1}^H . 2^b}{D^H . 2^a} \rfloor \tag{6.6}$$

3. Calculate the quotient Q_i using Equation 6.5.

4. Calculate the value of the new remainder R_i using the following equation:

$$R_i = R_{i-1} - q_i.D \tag{6.7}$$

5. $i = i + 1$. If $i < n$ where n is the number of digits of the quotient go to 2, else exit.

For the algorithm to converge, the following condition must be satisfied.

$$R_i \leq \frac{R_{i-1}}{2^j} \qquad i = 1, 2, ..., n \tag{6.8}$$

Complete proof of the algorithm convergence and the calculations of l, k and u are shown in appendix B.

6.4 Algorithm Implementation and Simulation Results

6.4.1 Hardware Implementation

A block diagram of the division algorithm is shown in Figure 6.3. The divider works as follows: at startup, the initialize signal is high. The divisor is stored in the divisor register while the dividend is saved in the remainder register. The l most significant bits from the divisor and the k most significant bits of the remainder register are used as an address for a look up table (ROM). The output of the ROM is multiplied by the divisor and subtracted from the previous remainder to calculate

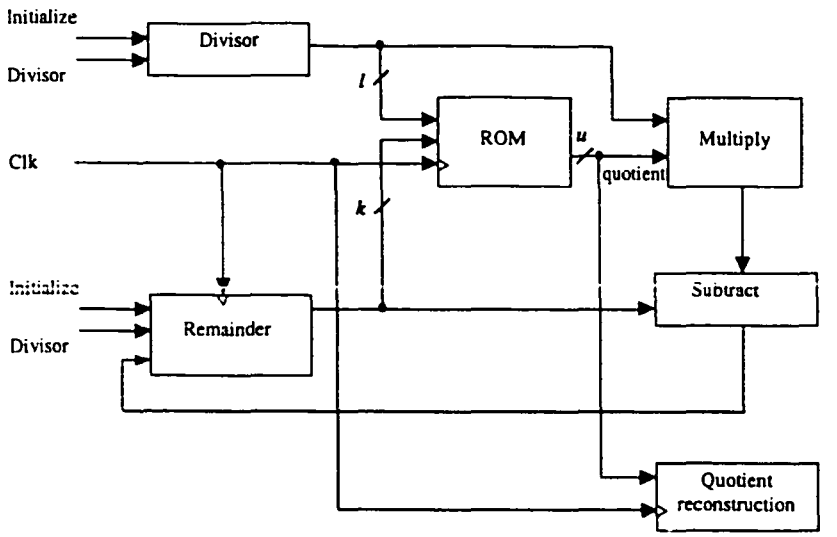


Figure 6.3: New division algorithm

the new remainder.

The quotient reconstruction register is a shift + ADD register. At the positive clock edge, the value in the register is shifted j bits to the left and the new quotient digit is added using a CRA (Carry Ripple Adder). The new quotient is then stored in the register at the negative edge of the clock cycle. Because the divisor is a floating number (≤ 0.5 divisor < 1), the quotient digit range is limited. By investigating the different radices, it is possible to prove that the carry from adding the partial quotient digit to the total quotient will not propagate more than certain number of bits depending on the radix used and the l, k combination. The details are explained in appendix B. Therefore, the CRA adder does not have to be an n digit adder. For example, for radix-8 divider with $l = 5$ and $k = 4$, the adder should be 6 bits wide.

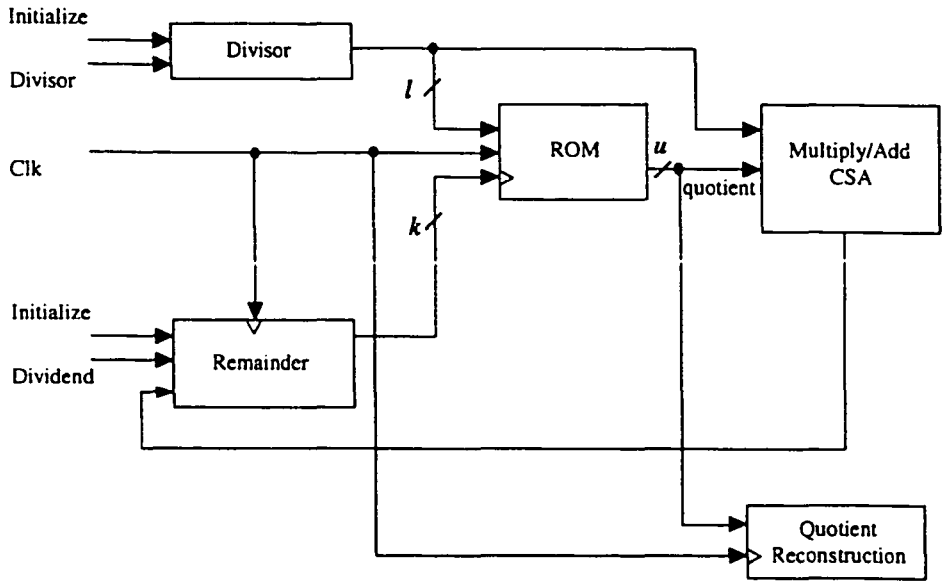


Figure 6.4: Modified algorithm with positive quotient digits only

To simplify the implementation even more, the subtractor and the multiplier may be merged together in one MAC (Multiplier Accumulator) block as shown in figure 6.4. Using CSA tree (Carry Save Adder) reduces the delay of the MAC block [84]. Because the remainder may be positive or negative, the look up table must have entries for both cases. Therefore, the table size is doubled. It was noticed that if the partial quotients in the look up table are calculated using $D^H + 1$ instead of D^H, all the remainders and quotient digits will be positive. Thus, using $D^H + 1$ to construct the look up table will reduce the hardware complexity and requires half the size of the look up table for the positive results only. The price of this change is

a one bit increase in the width of the quotient digit to account for the added error because of using $D^H + 1$. The proof of convergence for this case is also shown in appendix B.

Table 6.1: Look up table dimensions for different radices

Radix	l (divisor)	k (remainder)	u (quotient width)	table size
4	3	4	4	64 rows x 4 bits
8	4	5	8	256 rows x 8 bits
8	4	6	6	512 rows x 6 bits
8	5	4	6	256 rows x 6 bits
8	5	5	5	512 rows x 5 bits
16	5	7	7	2024 rows x 7 bits
16	6	5	7	1024 rows x 7 bits
16	7	6	6	4096 rows x 6 bits
16	7	8	5	16384 rows x 5 bits
32	6	9	8	16384 rows x 8 bits
32	6	8	10	8192 rows x 10 bits
32	7	6	9	4096 rows x 9 bits
32	7	7	8	8192 rows x 8 bits
32	7	8	7	16384 rows x 7 bits
32	8	6	8	8192 rows x 8 bits
32	9	7	6	32768 rows x 6 bits

Table 6.4.1 shows some possible combinations for k, l and u for different radices. For floating point numbers, the most significant bit of the divisor should be "1". This feature is used to reduce the number of rows in the look up table by half.

Choosing the best look up table combination is a function of the design parameters and based on the VLSI implementation of the divider. For example, if the MAC is implemented using a CSA tree with 3-2 counters, the optimum heights (number of levels in the tree) of the CSA tree would be 4, 6, 9 and 13 [84]. Be-

cause the CSA adds the partial products of the multiplication and the previous remainder, the number of partial products (quotient digit width) should be as close as possible to one of those numbers, 3, 5, 8 or 12 depending on the radix. This condition guarantees the maximum usage of the CSA tree. The look up table may be synthesized into a combinational logic block instead of the complex design of the ROM.

The MAC is used as a multiplier subtractor to calculate the term $R_{i-1} - q_i.D$. Since q_i is available, a two's complement circuit should be used to obtain the negative of q_i before the multiplication. To avoid the extra delay and hardware another scheme was used to calculate the partial remainder as follows:

Assume that it is required to calculate $A-B$ where A and B are binary numbers. The two's complement of A is $-A = \overline{A} + 1$. Therefore, if $\overline{A}$ is added to B, the result will be

$$\overline{A} + B = -A - 1 + B = -(A - (B - 1)) \tag{6.9}$$

Taking the one's complement of the result

$$\overline{-(A - (B - 1))} = -1 + (A - (B - 1)) = A - B \tag{6.10}$$

Therefore, the one's complement of the previous remainder will be added to the multiplication result and the final result will be complemented to get the new remainder. Using this scheme reduces the size of the MAC by removing the sign extension gates because all the partial products are positive. In the original case the

partial products are negative and therefore, require a sign extension to be aligned to the previous remainder.

6.4.2 Simulation Results

To evaluate the new division algorithm, radix-8 and radix-16 dividers were implemented using $0.6\mu m$ CMOS standard cells. The width of the dividend is the IEEE double precession floating point standard of 54 bits (53+(MSB=1)). The divisor is a single precession IEEE floating point number with width of 24 bits. Due to the complexity of the other division algorithms and the time required to implement each algorithm, only modified QST-SRT division algorithm [94] was chosen for comparison. For a detailed comparison and power analysis of various division algorithms, refer to [97]. A radix-8 and radix-16 dividers were also implemented using the modified QST-SRT algorithm.

The RTL (Register Transfer Language) description of the different dividers were written in Verilog and synthesized using Synopsys tools. The synthesis objectives were set to minimum power and delay consecutively. The look up tables in both divider architectures are synthesized into combinational logic blocks.

To evaluate the performance of the different dividers, a set of ten thousand test vectors were simulated using the different dividers in Verilog-XL. The switching activity extracted from the simulation was then exported to Synopsys environment, where Design Power weas used to report the power dissipation. The simulations were executed at a frequency of 10 MHz. Synopsys Design Compiler was also used to report the delay of the different dividers.

Table 6.2: Simulation results

Radix	SRT-QST			The new algorithm		
	area (gates)	delay per cycle	power total	area (gates)	delay per cycle	power total
8	785	8.2ns	13 m Watt	890	7.1ns	10.2 m Watt
16	1132	10.3ns	16.7 m Watt	1397	9.3ns	14.7 m Watt

Simulation results show that the modified algorithm reduces the over all power by 22% and 12% for radix-8 and radix-16 respectively. The power dissipation is reduced because of the simpler hardware and small logic depth of the combinational logic. The power dissipated in the look up table is also small because the divisor is fixed during the division which reduces the switching activity.

The delay is reduced by 13% and 10% for radix-8 and radix-16 respectively because of the smaller logic depth of the new implementation. However the area increased by 13% and 24% respectively because of relatively large look up table size compared to the modified SRT-QST algorithm.

Therefore the modified algorithm produces a power and delay gain compared to the SRT-QST algorithm. The price for the gain in power and speed is an increase in the area of the look up table.

6.5 Summary

A new algorithm for high radix floating point division has been developed. The algorithm uses a look up table to simplify the process of quotient digit selection. The VLSI implementation of the algorithm has been discussed in details.

The new divider shows power and delay gain for radix-8 and radix-16 dividers.

The area of the divider is larger than the other division algorithms because of the look up table size. More importantly, the new adder architecture is simple and may be generalized for any radix. The algorithm avoids the complex P-D tables and the extra pre-scaling hardware that exists in many division algorithms.

Chapter 7

Conclusion

Design for low-power has become an important concern in digital VLSI design, specially for portable systems where the energy source, the battery, is limited. The reduction of the transistor size allows higher integration density and increases the operating frequency. The rapid switching of millions of transistors dissipates lots of power and overheats the chip. This excessive temperature reduces the reliability of the chip and raises the need for expensive and large cooling systems. However, the performance is still an important issue which can not be sacrificed for low-power because any portable system has to achieve a minimum processing power.

This work introduced new methodologies for low-power digital design, on process, circuit and algorithm levels. Energy delay product is used as a comparison figure in most of this dissertation, to emphasise the importance of speed in digital design.

7.1 Thesis Contributions

The contributions of this thesis are as follows:

Impact of technology scaling on CMOS logic styles

This work discusses CMOS technology scaling trends and limitations, and how it impacts the functionality of CMOS logic styles. The study covers five CMOS logic styles, namely, conventional CMOS, Complementary Pass Logic (CPL), Domino logic, MOS Current Mode Logic (MCML) and Differential Cascode Voltage Switch Logic (DCVS). The analysis shows also that future CMOS technologies will give an advantage for conventional CMOS circuits in terms of power and speed. This study helps VLSI designers choose the appropriate logic style for each design based on objectives of each design.

Dynamic Current Mode Logic (DyCML)

A new logic style, called DyCML, for high speed low-power design is presented. DyCML utilizes reduced voltage swing circuits to reduce delay and dynamic power dissipation. The dynamic architecture of DyCML gate avoids the static power dissipation of the equivalent MCML circuits. Simulation results showed that DyCML circuits had better delay, power delay and energy delay products compared to other logic styles. Experimental results proved the superiority of DyCML circuits at various operating conditions.

High Speed Domino logic (HS-Domino)

HS-Domino is a new contention free Domino logic style. HS-Domino resolves the trade-off between noise margins and speed that exists in the conventional Domino circuits. Though the performance of conventional Domino degrades in DSM technologies, because of the high leakage currents, HS-Domino scales down smoothly with minimum effect on power and speed.

New MTCMOS implementations for Domino and DDCVS logic gates

New Multiple Threshold CMOS (MTCMOS) implementations for domino and DCVS dynamic circuits, are also presented. The new implementations reduce the leakage power by orders of magnitude keeping the noise margin intact. They also maintain the high performance and low dynamic power of low V_{th} circuits. Unlike other MTCMOS Domino logic implementations, the new architecture does not add extra hardware.

New high radix division algorithm

A high radix division algorithm for floating point numbers is introduced. The algorithm uses a look up table to estimate the quotient digit at each iteration. One of the major advantages of the new algorithm is that it can be generalized for any radix with minor changes in architecture. The algorithm reduces power and delay compared to other existing division algorithms.

7.2 Future Work

As much as this work introduced solutions for existing problems, it opened the door for new questions that could not be answered because of the limited time allocated to this work.

For example, the work on impact of technology scaling on logic styles discussed CMOS technologies down to 0.25 μm feature length. The question is now how the copper interconnects or low temperature devices will affect the performance of the various logic styles. Furthermore, how do these styles fit into the new technologies like SOI (Silicon On Insulator) and SIMOX-SOI (Separation by IMplantation of OXygen).

The DyCML logic style re-introduced CML logic as a general purpose logic style instead of being limited to mixed signal applications only. Future work in this area involves applying the same circuit scheme for other circuits that require high evaluation currents like memory sense amplifiers. Another extension to this work is to transfer this circuit scheme to SOI technology and validate the circuit functionality.

Engineers expected that the dynamic logic styles will not be attractive in DSM technologies. This thesis introduced new circuit techniques to prove that this is not true and to extend their operation into the DSM era. Further enhancements to this work should include developing more accurate models for leakage currents and implement those circuits on real silicon to validate their functionality when such technologies become available.

The last part of this thesis introduced a simple low-power division algorithm.

The new lookup table scheme may be extended to many other division algorithms. Detailed analysis of the lookup table may also lead to simple logic replacement which may reduce power dissipation even more.

7.3 Publications

The following is a list of the publications contributed during the course of this work.

1. M.W. Allam, M.H. Anis and M.I. Elmasry, "New Dynamic Logic Circuits for CMOS and MTCMOS Technologies", to be presented as a full paper in IEEE International Symposium on Low Power Electronics Design (ISLPED), July 2000, Italy

2. M.W. Allam and M.I. Elmasry, "Dynamic Current Mode Logic, A New Low-Power High-Performance Logic Family", submitted to IEEE Journal of Solid State Circuits, July 2000.

3. M.H. Anis, M.W. Allam and M.I. Elmasry, "Effect of Technology Scaling on CMOS Logic Styles", submitted to IEEE Journal of Solid State Circuits, June 2000.

4. M.W. Allam, M.H. Anis and M.I. Elmasry, "Effect of Technology Scaling on CMOS Logic Styles", IEEE Custom Integrated Circuits Conference (CICC), Orlando, Florida, May 2000.

5. M.W. Allam and M.I. Elmasry, "Dynamic Current Mode Logic, A New Low-Power High-Performance Logic Family", Custom Integrated Circuits Conference (CICC), Orlando, Florida, May 2000.

6. M.W. Allam and M. I. Elmasry, "Low-Power CMOS Logic Families", presented in The Midwest Symposium on Circuits and Systems (MWSCAS), NotreDame, Indiana, August 1998.

7. M.W. Allam and M. I. Elmasry, "A Dynamic Current Mode Logic Family", US Patent # US06028454, February 2000.

8. M.W. Allam and M. I. Elmasry, "Low-Power VLSI Implementation of Fast Addition Algorithms", Canadian Conference on Electrical and Computer Engineering (CCECE'98), Waterloo, Ontario, May 1998.

9. M.W. Allam, A. Vannelli and M.I. Elmasry, "A Clustering Algorithm for Network Partitioning", Canadian Conference on Electrical and Computer Engineering (CCECE'97), NewfoundLand, May 1997.

Appendix A

CMOS Logic Gates

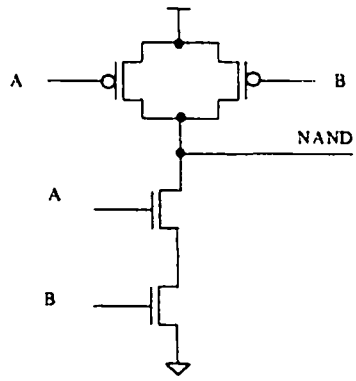


Figure A.1: NAND implemented in CMOS

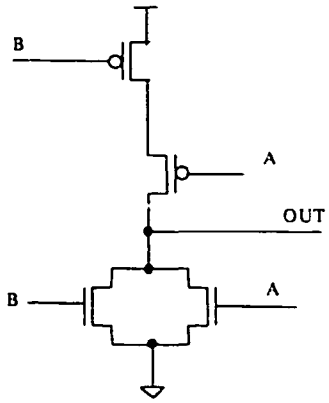


Figure A.2: NOR implemented in CMOS

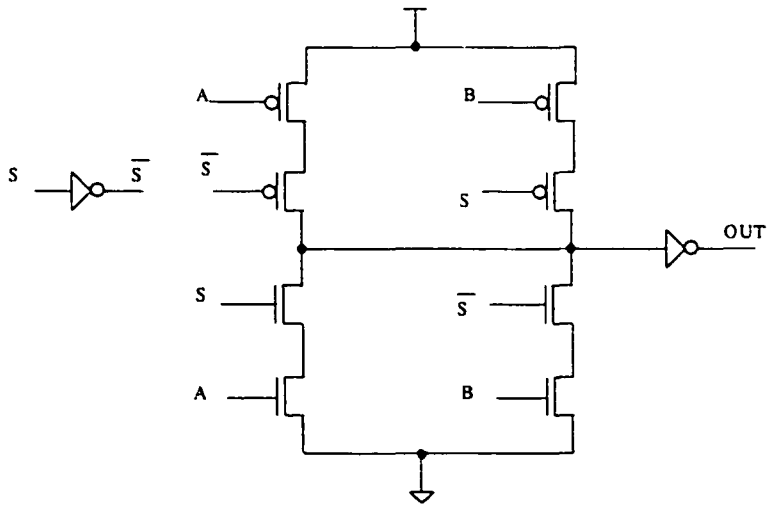


Figure A.3: MUX implemented in CMOS

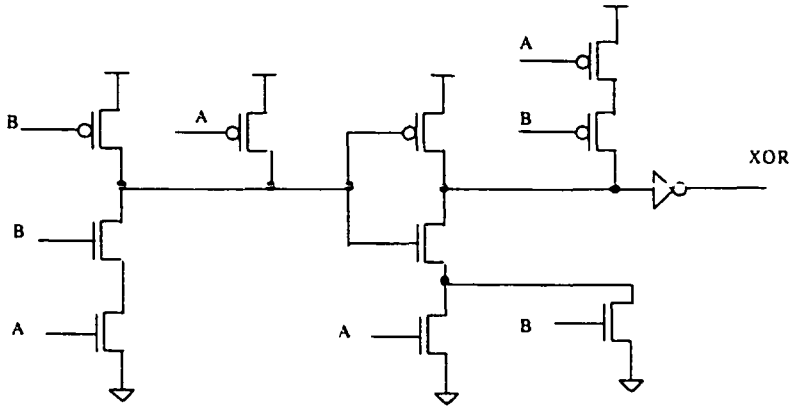


Figure A.4: XOR implemented in Conventional CMOS

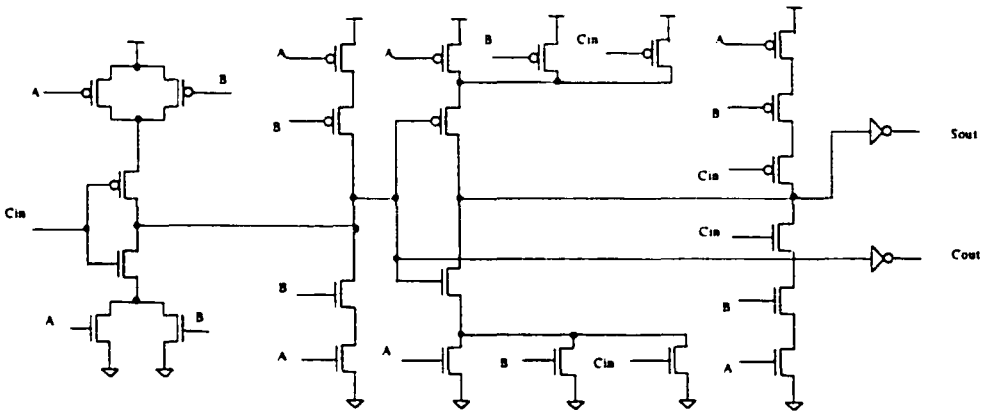


Figure A.5: Full Adder implemented in Conventional CMOS

CPL Logic Gates

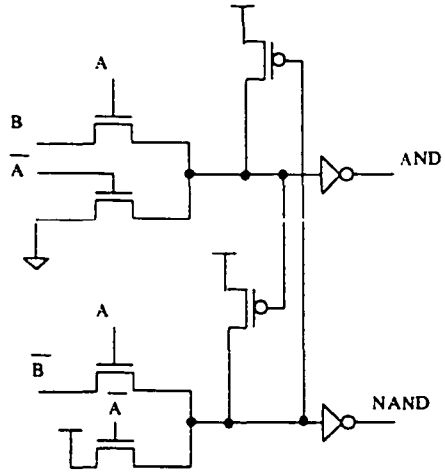


Figure A.6: NAND implemented in CPL

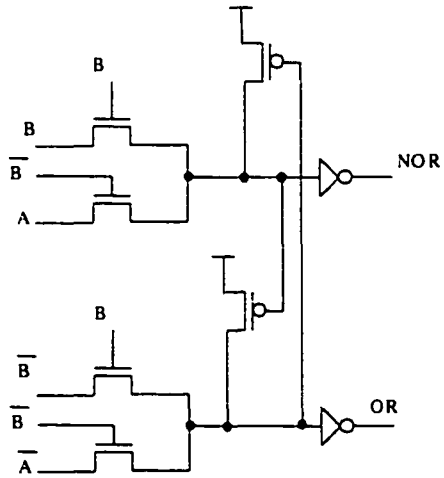


Figure A.7: NOR implemented in CPL

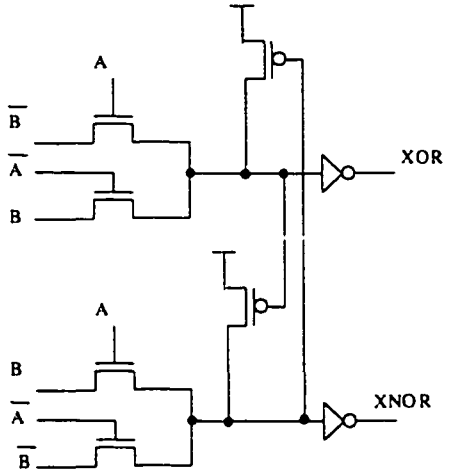


Figure A.8: XOR implemented in CPL

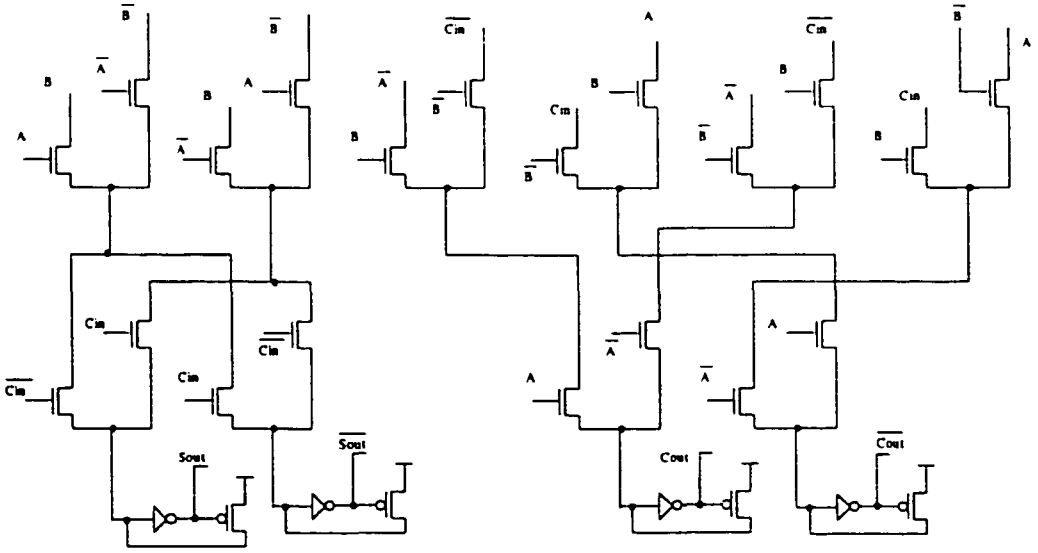


Figure A.9: Full Adder implemented in CPL

Domino Logic Gates

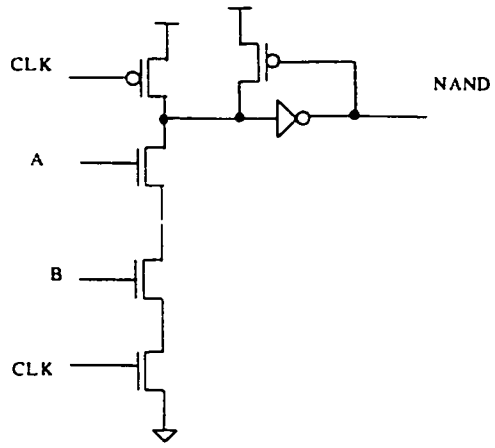


Figure A.10: AND implemented in Domino

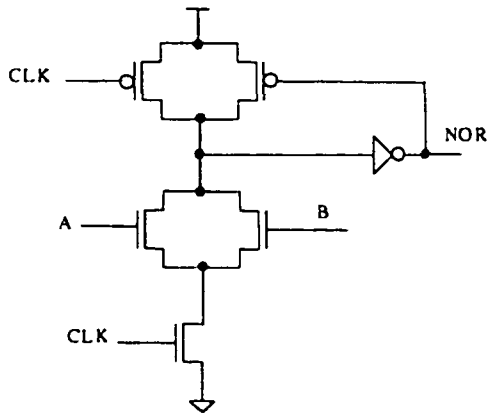


Figure A.11: OR implemented in Domino

MCML Logic Gates

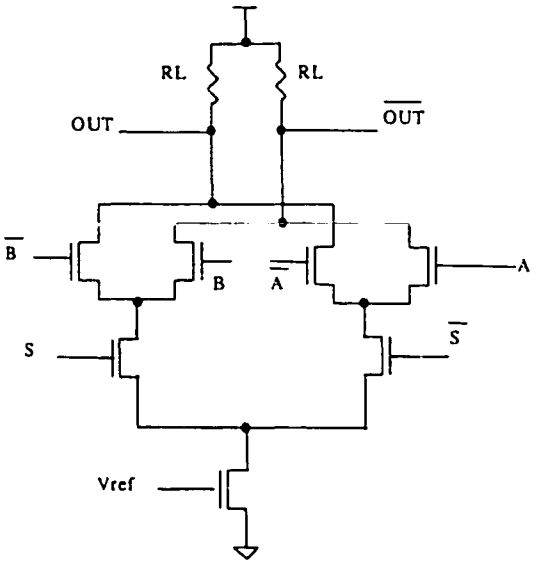


Figure A.12: MUX implemented in MCML

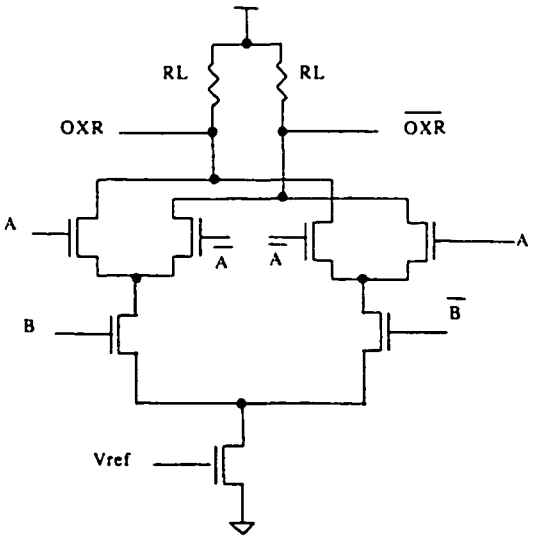


Figure A.13: XOR implemented in MCML

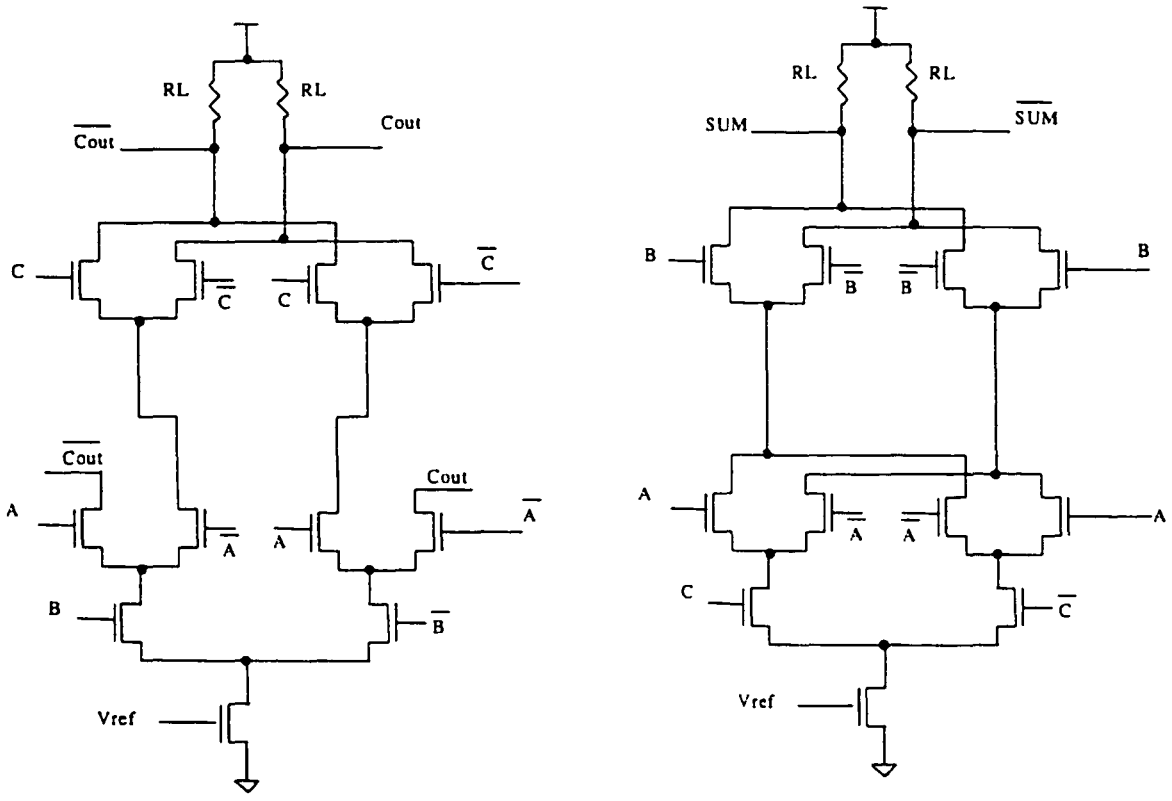


Figure A.14: Full Adder implemented in MCML

DDCVS Logic Gates

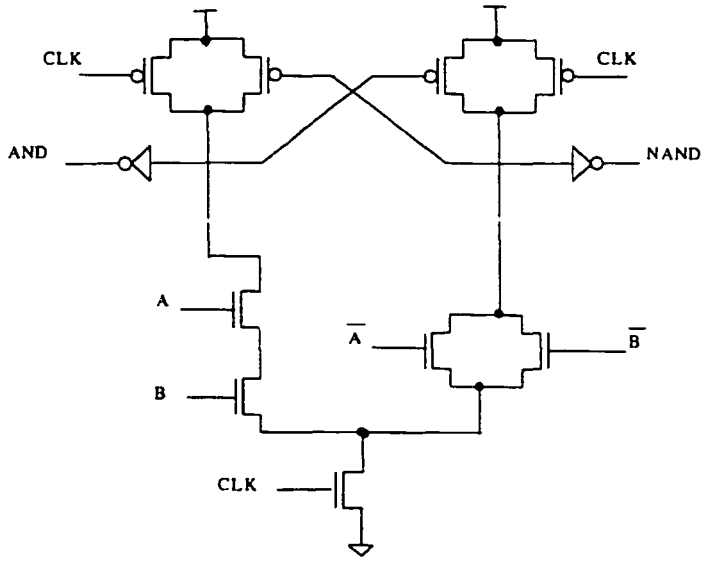


Figure A.15: AND/NAND/OR/NOR implemented in DCVS

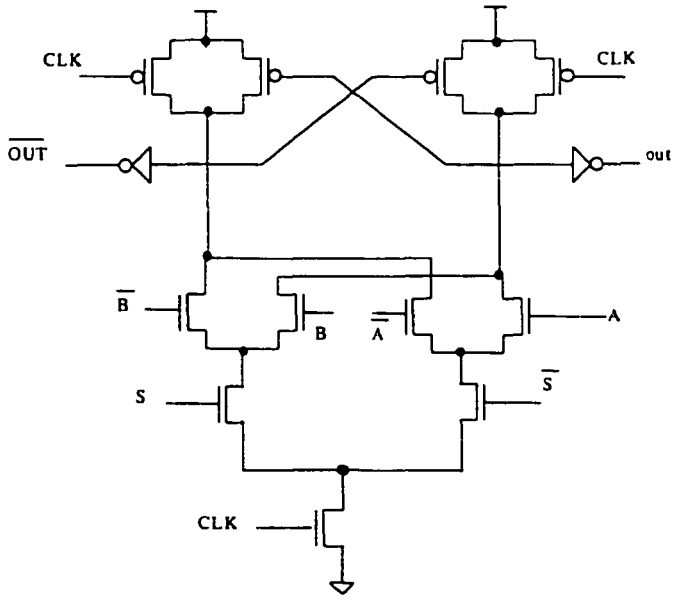


Figure A.16: MUX implemented in DCVS

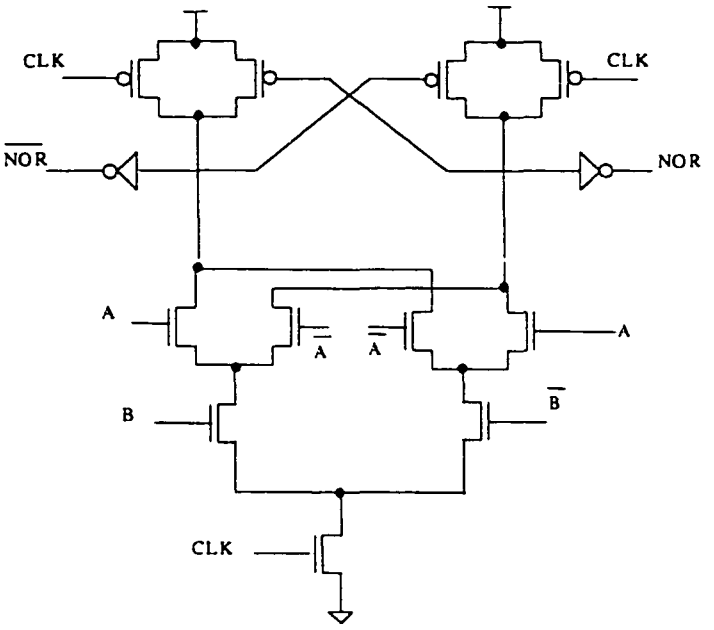


Figure A.17: XOR implemented in DCVS

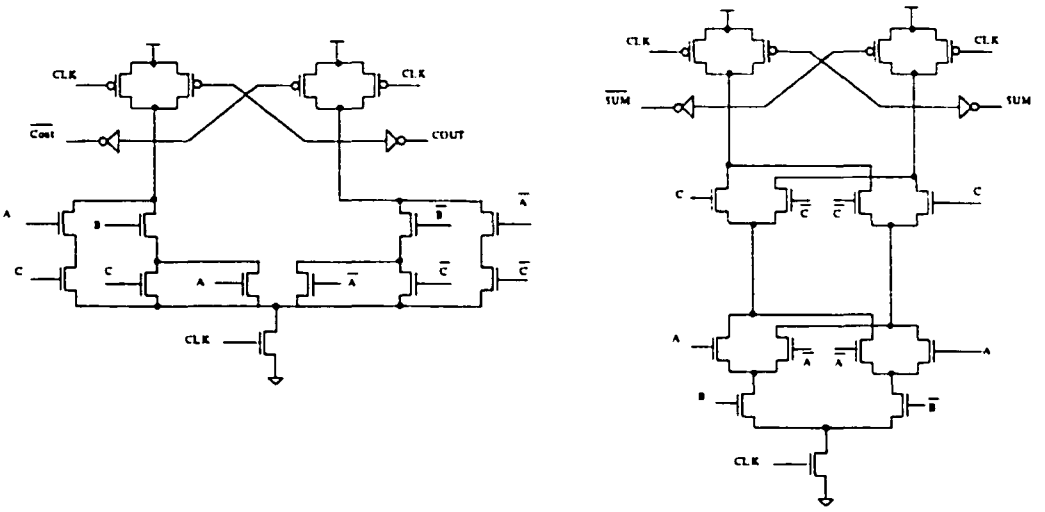


Figure A.18: Full Adder implemented in DCVS

Appendix B

Convergence of The Division Algorithm

The division process is defined as

$$Q = \frac{A}{D} \tag{B.1}$$

where Q is the quotient, A is the dividend and Q is the divisor.

In recurrence division, the quotient is calculated by shifting the previous quotient k bits (k is a function of the radix) and adding it to the partial quotient of the new recurrence. The following equation may be used to calculate the quotient after j recurrences:

$$Q_j = \sum_{i=j}^{i=1} Q_{i-1}.2^k + q_i \tag{B.2}$$

where Q_j is the quotient after j recurrences, q_i is the ith partial quotient, 2^k is the division radix and $Q_0 = 0$.

The ith partial remainder is calculated as follows:

$$R_i = R_{i-1} - q_i.D \tag{B.3}$$

where R_i is the remainder of the ith recurrence and R_0 equals A.

Therefore, R_i may be defined as

$$R_i = 2^u.R_i^H + R_i^L \tag{B.4}$$

where R_i^L is the least significant u bits of R_i and R_i^H is the most significant m bits of R_i starting from the uth bit.

Similarly,

$$D = 2^v.D^H + D^L \tag{B.5}$$

where D^L is the least significant v bits of D and D^H is the most significant n bits of D starting from the vth bit.

The new division algorithm calculates the approximate partial quotient $\hat{q}$ using the following equation

$$\hat{q} = \lfloor \frac{2^b.R_{i-1}^H}{D^H + 1} \rfloor \tag{B.6}$$

where b is an arbitrary value used to shift the the result of the partial quotient

multiplication before subtracting it from the previous remainder. The value of b depends on the division radix, u and v. The details of calculating b is shown later on this proof. The value of the quotient in this case is underestimated. Therefore

$$\hat{q} = \frac{2^b.R_{i-1}^H}{D^H + 1} - e_f \tag{B.7}$$

where e_f is a positive value which represents the approximation error. e_f is limited by the following inequality

$$0 \leqslant e_f \leqslant \frac{D^H}{D^H + 1} \tag{B.8}$$

Substituting B.4, B.5, B.6, B.7 into B.3 leads to:

$$R_i = 2^u.R_{i-1}^H + R_{i-1}^L - 2^{u-v-b}.\hat{q}_i.(2^v.D^H + D^L)$$

$$= R_{i-1}^L + \frac{2^u.R_{i-1}^H}{D^H + 1} + 2^{u-b}.e_f.D^H + 2^{u-v-b}.e_f.D^L - \frac{2^{u-v}D^L.R_{i-1}^H}{D^H + 1}$$

$$= R_{i-1}^L + \frac{2^u.R_{i-1}^H}{D^H + 1} - \frac{2^{u-v}.D^L.R_{i-1}^H}{D^H + 1} + e_f.(2^{u-b}.D^H + 2^{u-v-b}.D^L) \tag{B.9}$$

The largest value for R_i occurs when $e_f = \frac{D^H}{D^H+1}$.

For the algorithm to converge, R_i must be less than $\frac{R_{i-1}}{2^k}$. To prove this, the four components of equation B.9 must be studied.

First is R_{i-1} which consists of u bits

$$0 \leqslant R^L_{i-1} \leqslant 2^u - 1 \tag{B.10}$$

Second is $\frac{2^u.R^H_{i-1}}{D^H+1}$. Since D is a floating point number, the most significant bit should be 1. Therefore,

$$\frac{2^u.R^H_{i-1}}{D^H+1} \leqslant \frac{2^u.(2^m-1)}{2^{n-1}+1} \tag{B.11}$$

i.e., if $n \geqslant m+1$, equation B.11 will be in the order of 2^u.

Third is $\frac{2^{u-v}.D^L.R^H_{i-1}}{D^H+1}$. Since $0 \leqslant D^L \leqslant 2^v - 1$ and $0 \leqslant R^H_{i-1} \leqslant 2^m - 1$, the largest value for this term occurs when D^L, R^H_{i-1} are maximized and D^H is minimized.

$$0 \leqslant \frac{2^{u-v}.D^L.R^H_{i-1}}{D^H+1} \leqslant \frac{2^{u-v}.(2^m-1).(2^v-1)}{2^{n-1}+1}$$

$$0 \leqslant \frac{2^{u-v}.D^L.R^H_{i-1}}{D^H+1} \leqslant \frac{2^{m+u} - 2^u - 2^m + 2^{u-v} + 1}{2^{n-1}+1} \approx 2^{u+m-n+1} \tag{B.12}$$

Therefore, if $n \geqslant m+1$, equation B.12 becomes $\leqslant 2^u$

Fourth component is $e_f.(2^{u-b}.D^H + 2^{u-v-b}.D^L)$. The maximum value for this term occurs when $e_f = \frac{D^H}{D^H+1}$ and D^H is maximized. Therefore

$$0 \leqslant e_f.(2^{u-b}.D^H + 2^{u-v-b}.D^L) \leqslant \frac{2^{n-1}.2^{u-v-b}.(2^v-1)}{2^{n-1}+1} \tag{B.13}$$

Therefore, this term is $\leqslant 2^{u-b}$ which is less than 2^u because b is a positive number.

Therefore, the largest value for Equation B.9 occurs when the third term is zero while all the other terms $\approx 2^u$. i.e.,

$$R_i \leqslant 3.(2^u) \leqslant 2^{u+2} \tag{B.14}$$

Since the original order of R_{i-1} was $2^m + u$, the order of R_i has dropped by $2^m - 2$. i.e., for the algorithm to converge, the number of bits taken from the remainder in the look up table (m) must be more than $k + 2$, where 2^k is the division radix. From Equation B.10, $m \leqslant n - 1$.

Condition of convergence of the algorithm is

$$k + 2 \leqslant m \leqslant n - 1 \tag{B.15}$$

Bibliography

[1] Anantha P. Chandrakasan et al., "Low-Power CMOS Digital Design", *IEEE Journal of Slid-State Circuits*, vol. SC-27, n. 4, pp. 473–484, 1992.

[2] Technical Support, *Pentium III Processor for the SC242 at 450 MHz to 800 MHz, Datasheet*, Intel Corporation, Santa Clara, CA, 1999.

[3] H.Sasaki, "Multimedia Complex on a Chip", *International Solid-State Circuits Conference*, pp. 16–19, 1996.

[4] G. K. Yeap, *Practical low power digital VLSI design*, Kluwer Academic Publishers, Boston, Mass. ; London, 1998.

[5] P. Landman and J. Rabaey, "Power Estimation for High Level Synthesis", *Proceedings of The European Conference on Design Automation with the European Event in ASIC Design (EDAC)*, 1993.

[6] A. Bellaouar and M. I. Elmasry, *Low-Power Digital VLSI Design Circuits and Systems*, Kluwer Academics Publications, 1995.

[7] J.M.Rabaey, *Digital Integrated Circuits*, Prentice Hall, 1996.

[8] H. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits", *IEEE Journal of Solid State Circuits,*, vol. 19, n. 4, August 1984.

[9] N. Jha A. Raghunathan, S. Dey, "Glitch Analysis and Reduction in Register Transfer Level Power Optimization", *Proceedings of the 1996 33rd Annual Design Automation Conference*, 1996.

[10] C. Huizer, "Optimized Application of Submicron CMOS for Vlsi Logic - a Systems Oriented View on Design and Technology", *Proceedings of the IEEE Custom Integrated Circuits Conference*, 1987.

[11] S. Sze, *Physics of Semiconductor Devices*, Wiley, New York, 1981.

[12] M.I. Elmasry, "Field Effect Current Mode Logic Gate", *U.S. Patent # 4445051*, 1984.

[13] H. Samueli F. Lu, "A High-Speed CMOS Full-Adder Cell Using a New Circuit Design Technique-Adaptively-Biased Pseudo-NMOS Logic", *Proceedings of IEEE International Symposium on Circuits and Systems*, 1990.

[14] C. Svensson D. Liu, "Trading Speed for Low Power by Choice of Supply and Threshold Voltages", *IEEE Journal of Slid-State Circuits*, vol. 28, n. 1, January 1993.

[15] M. Kakumu H. Oyamatsu, K. Kinugawa, "Design Methodology of Deep Submicron CMOS Devices for 1 V Operation", *Symposium on VLSI Technology Tech. Dig.*, pp. 89–90, 1993.

[16] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, Menlo Park, CA, 1990.

[17] A. K. Stamper, "Interconnection Scaling to 1 GHz and Beyond", *Micronews, IBM Technical Journal*, Q2, 1998.

[18] H.Iwai, "CMOS Technology-YEAR 2010 and Beyond", *IEEE Journal of Solid State Circuits*, pp. 357–366, 1999.

[19] Jean-Pierre Colinge, *Silicon-On-Insulator Technology : Materials to VLSI*, Kluwer Academic Publishers, Boston, 1997.

[20] L.Wei et al., "Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 16–24, 1999.

[21] M.Kakumu et al., "Design Optimization Methodology for Deep-Submicron CMOS Device at Low-Temperature Operation", *IEEE Transactions on Electron Devices*, pp. 370–377, 1992.

[22] Z.Chen et al., "Low Threshold Voltage Quarter Micron MOSFETs for Low Power Applications", *Lecture*, Nov. 1995.

[23] John H. Lau and Shi-Wei R. Lee, *Chip Scale Package (CSP) : Design, Materials, Processes, Reliability, and Applications*, Electronic Packaging and Interconnection Series, McGraw-Hill, New York, London, 1999.

[24] C. Chao, *Multiport Memory Design for an MCM Coprocessor*, Ph.D. thesis, Stanford University, 1994.

[25] John H Lau, *Flip Chip Technologies*, Electronic Packaging and Interconection Series, McGraw-Hill, New York, 1996.

[26] John H. Lau, *Ball grid array technology*, Electronic Packaging and Interconnection Series, McGraw-Hill, New York, Toronto, 1995.

[27] K. Chao and F. Wong, "Low-Power Consideration in Floorplan Design", *Proc. of the 1994 International Workshop on Low-Power Design*, pp. 45–50, 1994.

[28] H. Vaishnav and M. Pedram, "Delay Optimal Partitioning Targeting Low-Power VLSI Circuits", *Proc. of the IEEE International Conference on Computer Aided Design*, pp. 638–643, 1995.

[29] J. Xi and W-M Dai, "Buffer Insersion and Sizing Under Process Variation for Low-Power", *Proc. of the 32nd Design Automation Conference*, pp. 491–496, 1995.

[30] H. Vaishnav, *Optimization of Post-Layout Area, Delay, and Power Dissipation*, Ph.D. thesis, Computer Engineering Dept., University of Southern California, 1995.

[31] M. Kakumu H. Yoshimura, F. Matsuoka, "New CMOS Shallow Junction Well FET Structure (CMOS-SJET) for Low Power-Supply Voltage", *International Electron Devices Meeting Tech. Dig.*, pp. 909–912, December 1992.

[32] A. Chandrakasan, *Low-power CMOS Design*, PhD thesis, University of California, Berkeley, Aug 1994.

[33] T. Burd, "Low Power CMOS Library Design Methodology", Master's thesis, University of California, Berkeley, 1995.

[34] K. Jeppson N. Hedenstierna, "CMOS Circuit Speed and Buffer Optimization", *IEEE Transactions on Computer-Aided Design*, vol. CAD-6, pp. 270–281, 1987.

[35] D.W. Dobberpuhl et al., "A 200-MHz 64-bit Dual-issue CMOS Microprocessor", *Digital Technical Journal*, vol. 4, n. 4, 1992.

[36] C. Svensson J. Yuan, "High-Speed CMOS Circuit Techniques", *IEEE Journal of Slid-State Circuits*, vol. 24, n. 1, February 1989.

[37] I. Karlsson, "True Single Phase Clock Dynamic CMOS circuit technique", *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 475–478, 1988.

[38] S. Iman and M. Pedram, "Multi-Level Network Optimization for Low-Power", *Proc. of the International Conference on Computer Aided Design*, pp. 371–377, November 1994.

[39] K. Roy and S. Parasad, "Circuit Activity Based Logic Synthesis for Low Power Reliable Operations", *IEEE Transactions on VLSI Systems*, vol. 1, n. 4, December 1993.

[40] C. Tsui and M. Pedram, "Low Power State Assignment Targeting Two- and Multi-level Logic Implementations", *Proc. of the International Conference on Computer Aided Design*, pp. 82–87, November 1994.

[41] O. Coudert, "Gate Sizing for Constrained Delay/Power/Area Optimization", *IEEE Transactions on VLSI Systems*, vol. 5, n. 4, December 1997.

[42] J. Monterio, S. Devadas and A. Ghosh, "Retiming Sequential Circuits for Low Power", *Proc. of the International Conference on Computer Aided Design*, pp. 398–402, 1993.

[43] A. Chandrakasan et al., "HYPER-LP: A system for Power Minimization Using Architectural Transformations", *Proc. of the International Conference on Computer Aided Design*, pp. 300–303, 1992.

[44] A. Raghunathan, S. Dey and N. K. Jha, "Glitch Analysis and Reduction in Register Transfer Level Power Optimization", *33rd Design Automation Conference*, pp. 331–336, 1996.

[45] J-M. Cheng and M. Pedram, "Energy Minimization Using Multiple Supply Voltages", *Proc. of the International Conference on Low Power Electronics and Design*, pp. 157–162, 1996.

[46] C. Papachristou and M. Nourani M. Spining, "A Multiple Clocking Scheme for Low Power RTL Design", *Proc. of the International Symposium on Low Power Design*, pp. 27–32, 1995.

[47] M. Lee and V. Tiwari, "A Memory Allocation Technique for Low-Energy Embeded DSP Software", *Proc. of the IEEE Symposium on Low Power Electronics*, pp. 24–25, 1995.

[48] G. Gerosa et al., "A 2.2 Watt 80 MHz Superscalar RISC Microprocessor", *IEEE Journal of Slid-State Circuits*, vol. 29, n. 12, pp. 1440–1454, December 1996.

[49] A. Chandrakasan J. Ludwig, S. Nawab, "Low-Power Digital Filtering Using Approximate Precessing", *IEEE Journal of Slid-State Circuits*, vol. 31, n. 3, pp. 395–400, March 1996.

[50] A. Chandtakasan et al., "Design Considerations and Tools for Low-Voltage Digital System Design", *Proc. of the 33rd Design Automation Conference*, 1996.

[51] G.McFarland, *CMOS Technology Scaling and its Impact on Cache Delay*, Ph.D. thesis, Standford University, 1997.

[52] G.E. Moore, "Progress in digital integrated circuits", *International Electronic Devices Meeting*, pp. 11–13, 1975.

[53] Semiconductor Industry Association, "The national technology roadmap for semiconductors", 1994.

[54] R.Dennard et al., "Design of ion-implanted MOSFETs with very small dimensions", *IEEE Journal of Slid-State Circuits*, pp. 256–268, 1974.

[55] P.Chatterjee et al., "The Impact of Scaling Laws on the Choice of N-channel or P-channel for MOS VLSI", *IEEE Electron Device Letters*, pp. 220–223, 1980.

[56] R.Gonzalez et al., "Supply and threshold voltage scaling for low power CMOS", *IEEE Journal of Slid-State Circuits*, pp. 1210–1216, 1997.

[57] M.Bohr et al., "A high-performance 0.25-μm logic technology optimized for 1.8V operation", *International Electronic Devices Meeting*, pp. 847–850, 1996.

[58] Y.Taur, "The 100-Million Transistor IC", *IEEE SPECTRUM Journal*, pp. 23–29, August 1999.

[59] K.Shimohigashi et al., "Low-Voltage ULSI Design", *IEEE Journal of Slid-State Circuits*, pp. 408–413, 1993.

[60] T.Sakurai et al., "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas", *IEEE Journal of Slid-State Circuits*, pp. 584–594, 1990.

[61] J.Pfiester, "Performance Limits of CMOS ULSI", *IEEE Transactions on Electron Devices*, page 333, 1985.

[62] S.Thompson et al., "Dual Threshold Voltage and Substrate Bias: Keys to High Performance, Low Power, 0.1μm Logic Designs", *IEEE Symposium on VLSI Technology Digest of Technical Papers*, pp. 69–70, 1997.

[63] T.Chan et al., "The Impact of Gate-Induced Drain Leakage Current on MOSFET Scaling", *International Electronic Devices Meeting*, pp. 718–721, 1987.

[64] K.Yamabe et al., "Time-dependant Dielectric Breakdown of Thin Thermally Grown SiO2 Films", *IEEE Transactions on Electron Devices*, pp. 423–428, 1985.

[65] T.Hayashi et al., "Hot Carrier Injection in PMOSFETs", *OKI Technical Review*, pp. 59–62, 1991.

[66] P.Srivastava et al., "Issues in the Design of Domino Logic Circuits", *Proc. of IEEE Great Lakes Symposium on VLSI*, pp. 108–112, 1998.

[67] K.Saraswat et al., "Effect of Scaling Interconnections on the Time Delay of VLSI Circuits", *IEEE Transactions on Electron Devices*, pp. 645–650, 1982.

[68] M.Bohr and Y.Elmansy, "Technology for Advanced High-Performance Microprocessors", *IEEE Transactions on Electron Devices*, pp. 620–625, Vol.45 1998.

[69] R.Zimmermann and W.Fichtner, "Low-Power Logic Styles: CMOS Versus Pass-Transistor Logic", *IEEE Journal of Solid State Circuits*, pp. 1079–1090, July 1997.

[70] K.Yano et al., "Top-Down Pass-Transistor Logic Design", *IEEE Journal of Solid State Circuits*, pp. 792–803, June 1996.

[71] Ruchir Puri, "Design Issues in Mixed Static-Domino Circuit Implementations", *Proc. IEEE International Conf. on Computer Design*, pp. 270–275, Oct. 1998.

[72] N.Weste and K.Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley Publishing Company, 1994.

[73] L. G. Heller et al., "Cascode Voltage Switch Logic: A Differential CMOS Logic Family", *International Solid State Circuits Conference*, pp. 16–17, 1984.

[74] DM Wu, JW Davis and NG Thoma, "Design and Test Strategy for Differential Cascode Voltage Switch Circuits", *Proceedings of the Third Annual IEEE ASIC Seminar and Exhibit*, pp. 1–4, 1990.

[75] F. Lai et al., "Design and Implementation of Differential Cascode Switch with

Pass-Gate (DCVSPG) Logic for High-Performance Digital Systems", *Journal of Solid-State Circuits*, pp. 563–573, April 1997.

[76] A.J. Acosta et al., "SODS: A New CMOS Differential Type Structure", *Journal of Solid-State Circuits*, pp. 835–838, July 1995.

[77] B. Kong et al., "Charge Recycling Differential Logic CRDL for Low Power Applications", *IEEE Journal of Solid-State Circuits*, pp. 1267–1276, September 1996.

[78] M. Mizuno et al., "A GHz MOS Adaptive Pipeline Technique Using MOS Current-Mode Logic", *IEEE Journal of Solid-State Circuits*, pp. 784–791, June 1996.

[79] M. Yamashina and H. Yamada, "MOS current mode logic MCML circuit for low-power GHz processors", *NEC Research & Development*, vol. 36, n. 1, pp. 54–63, Jan 1995.

[80] P. Ng, P. T. Balsara and D. Steiss, "Performance of CMOS Differential Circuits", *IEEE Journal of Solid-State Circuits*, pp. 841–846, June 1996.

[81] S.Thompson et al., "MOS Scaling: Transistor Challenges for the 21st Century", *Intel Technology Journal*, Q3, 1998.

[82] et al. S. Venkatesan, "A High Performance 1.8V, 0.20μm CMOS Technology with Copper Metallization", *IEEE Transactions on Electron Devices*, pp. 769–772, 1997.

[83] M. Bohr, "Technology development strategies for the 21st century", *Applied Surface Science*, pp. 534–540, 100/101 1996.

[84] I. Koren, *Computer Arithmetic Algorithms*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[85] Z.Chen et al., "0.18μm Dual V_t MOSFET Process and Energy-Delay Measurement", *IEDM Technical Digest*, pp. 851–853, 1996.

[86] S.Mutah et al., "1-V Power Supply High-Speed Digital Circuit Technology with Multi-Threshold Voltage CMOS", *IEEE Journal of Slid-State Circuits*, pp. 847–853, 1995.

[87] J.Kao, "Dual Theshold Voltage Domino Logic", *IEEE 25th European Solid-State Circuits Conference*, To appear, 1999.

[88] T. Coe et al., "Computational Aspect of the Pentium Affair", *IEEE computational Science & Engineering*, pp. 18–30, 1995.

[89] P. Soderquist and M. Leeser, "Area/performance comparison of subtractive and multiplicative divide/square root implementations", *IEEE 12th Symposium on Computer Arithmetic*, pp. 132–139, 1995.

[90] J. Cocke and D.W. Sweeney, "High Speed Arithmetic in a Parallel Device", *Technical Report, IBM*, feb 1957.

[91] J.E. Robertson, "On the Design of High-Speed Computer", Technical Report 80, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, 1957.

[92] K.D. Tocher, "Techniques of Multiplication and Division for Automatic Binary Computers", *Quart. J. Mech. Applied Mathematics*, pp. 364–384, 1958.

[93] S. McQuillan, J. McCanny and R. Hamill, "New algorithms and VLSI architectures for SRT division and square root", *IEEE 11th Symposium on Computer Arithmetic*, 1993, p 80-86.

[94] T. Lang and P. Montuschi, "Very-High Radix Combined Division and Square Root with Prescaling and Selection By Rounding", *Proceedings of the 1995 IEEE 12th Symposium on Computer Arithmetic*, pp. 124–131, 1995.

[95] T. Pan, H. Kay and C. Wey, "High-Radix SRT Division with Speculation of Quotient Digits", *Proceedings of IEEE International Conference on Computer Design*, 1995, p 479-484.

[96] M. Ercegovac and T. Lang, *Division and Square root*, Kluwer Academic Publishers, Boston / Detroit / London, 1994.

[97] Alberto Nannarelli, *Low Power Division and Square Root*, Ph.D. thesis, University of California, Irvine, 1999.